Duke Attendance Stats 2022-23

Packages

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(tidymodels)
```

```
-- Attaching packages -------------------------------------- tidymodels 1.1.1 --
v broom        1.0.5     v rsample      1.2.0
v dials        1.2.0     v tune         1.1.2
v infer        1.0.4     v workflows    1.1.3
v modeldata    1.2.0     v workflowsets 1.0.1
v parsnip      1.1.1     v yardstick    1.2.0
v recipes      1.0.8
-- Conflicts ----------------------------------------- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
```

```
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
* Use tidymodels_prefer() to resolve common conflicts.
```

Home Game Attendance

Import Data

```r
attendance_data <- read_csv("data/Duke Stats - DukeAttendanceV2.csv")
```

```
Rows: 26 Columns: 29
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (8): OppName, Surface, Day, Site, Result, TV_Coverage, City, State
dbl (12): FPI, FPI_diff, Month, Date, Year, Start_Time, DukePts, OppPts, Poi...
lgl  (9): Rain, 1stSeedQB, SchoolBreak, NatlHoliday, Bowl, UNC_Game, Undefea...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

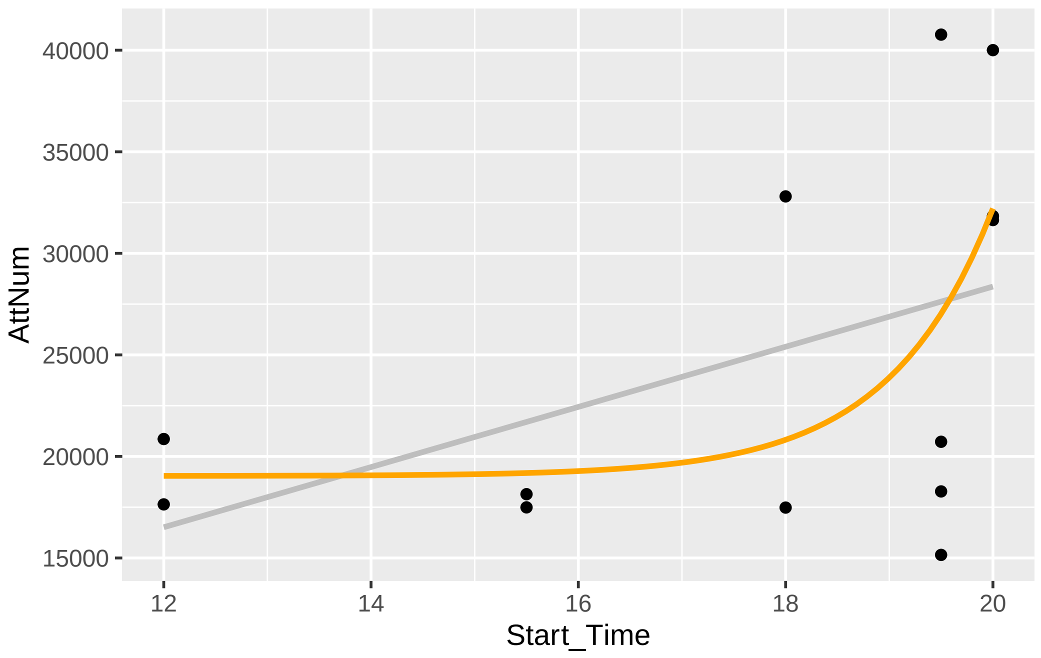
```r
attendance_data <- attendance_data |>
  mutate(isHome = if_else(Site == "Home", TRUE, FALSE)) |>
  mutate(Day = as.factor(Day))

home_attendance_data <- attendance_data |>
  filter(isHome == TRUE)

home_attendance_data
```

```
# A tibble: 13 x 30
   OppName         FPI FPI_diff Surface Month  Date  Year Day   Start_Time Site
   <chr>         <dbl>    <dbl> <chr>   <dbl> <dbl> <dbl> <fct>      <dbl> <chr>
 1 Clemson        13.8      4.8 Grass       9     4  2023 Mon         20    Home
 2 Lafayette      NA       NA   Grass       9     9  2023 Sat         18    Home
 3 Northwestern    0.8     -8.2 Grass       9    16  2023 Sat         15.5 Home
 4 Notre Dame     20.7     11.7 Grass       9    30  2023 Sat         19.5 Home
 5 North Caroli~   6.9     -2.1 Grass      10    14  2023 Sat         20    Home
 6 Wake Forest    -1.7    -10.7 Grass      11     2  2023 Thu         19.5 Home
 7 Pittsburgh     -0.5     -9.5 Grass      11    25  2023 Sat          12   Home
 8 Temple        -11.8    -17.1 Grass       9     2  2022 Fri          19.5 Home
 9 N.C. A&T       NA       -5.3 Grass       9    17  2022 Sat          18   Home
10 Virginia       -4       -9.3 Grass      10     1  2022 Sat          19.5 Home
11 North Caroli~   6.2      0.9 Grass      10    15  2022 Sat          20   Home
12 Virginia Tech  -6.2    -11.5 Grass      11    12  2022 Sat          12   Home
13 Wake Forest     7.6      2.3 Grass      11    26  2022 Sat          15.5 Home
# i 20 more variables: Result <chr>, DukePts <dbl>, OppPts <dbl>,
#   PointDiff <dbl>, AttNum <dbl>, AttPct <dbl>, ESPN_WinPred <dbl>,
#   Rain <lgl>, `1stSeedQB` <lgl>, SchoolBreak <lgl>, NatlHoliday <lgl>,
#   TV_Coverage <chr>, City <chr>, State <chr>, Bowl <lgl>, UNC_Game <lgl>,
#   Undefeated_All <lgl>, Undefeated_Home <lgl>, Game_After_Loss <lgl>,
#   isHome <lgl>
```

Time of Day

```r
home_attendance_data |>
  ggplot(
    aes(x = Start_Time, y = AttNum)
  ) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "gray") +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE, color = "orange") #+
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
  #scale_colour_viridis_c()

time_lm <- linear_reg() |>
  set_engine("lm") |>
  fit(AttNum ~ Start_Time, data = home_attendance_data)

time_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time), data = home_attendance_data)

tidy(time_lm)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic p.value
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)   -1262.    14851.   -0.0850   0.934
2 Start_Time     1481.      832.    1.78     0.103
```

```
tidy(time_glm)
```

```
# A tibble: 2 x 5
  term                 estimate    std.error statistic  p.value
  <chr>                   <dbl>        <dbl>     <dbl>    <dbl>
1 (Intercept)     19037.        3260.             5.84 0.000112
2 exp(Start_Time)     0.0000271    0.0000114      2.38 0.0365
```

```
glance(time_lm)$AIC
```

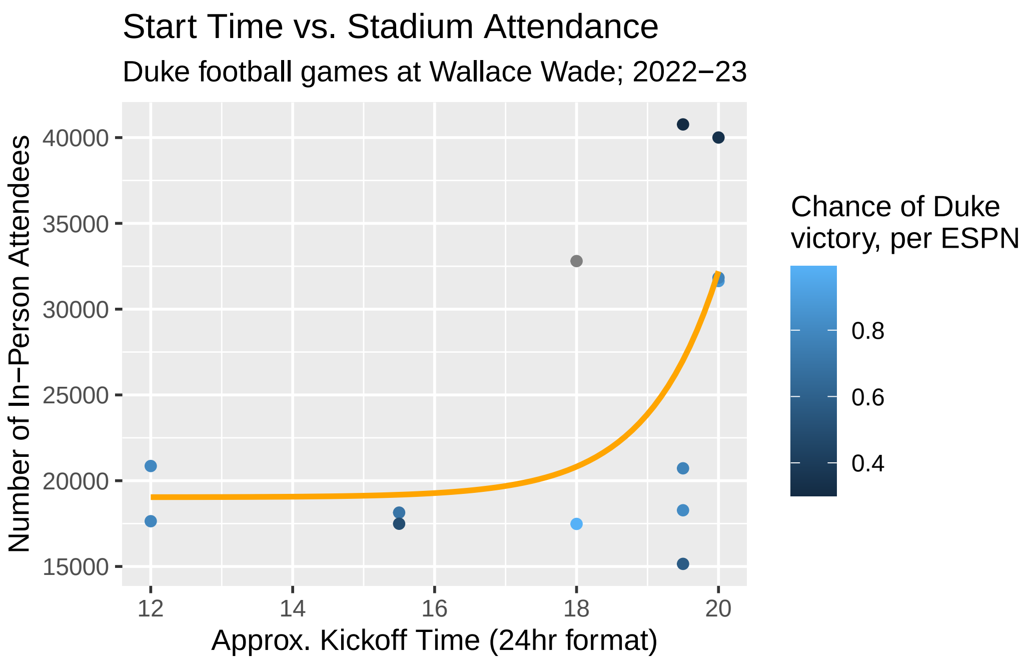
```
[1] 275.8782
```

```
glance(time_glm)$AIC
```

```
[1] 273.7693
```

+ Win Chance

```
home_attendance_data |>
  ggplot(
    aes(x = Start_Time, y = AttNum, color = ESPN_WinPred)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE, color = "orange") +
  labs(title = "Start Time vs. Stadium Attendance",
       subtitle = "Duke football games at Wallace Wade; 2022-23",
       x = "Approx. Kickoff Time (24hr format)",
       y = "Number of In-Person Attendees",
       color = "Chance of Duke\nvictory, per ESPN") #+
```



```
  #scale_colour_viridis_c()

time_winpred_add_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) + ESPN_WinPred, data = home_attendance_data)

time_winpred_int_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) * ESPN_WinPred, data = home_attendance_data)

tidy(time_winpred_add_glm)
```

```
# A tibble: 3 x 5
  term            estimate     std.error statistic p.value
  <chr>              <dbl>         <dbl>     <dbl>   <dbl>
1 (Intercept)      3.03e+4 7130.              4.25 0.00215
2 exp(Start_Time)  2.76e-5    0.00000954      2.89 0.0180
3 ESPN_WinPred    -1.81e+4 8969.             -2.01 0.0750
```

```
tidy(time_winpred_int_glm)
```

```
# A tibble: 4 x 5
  term                              estimate     std.error statistic p.value
  <chr>                                <dbl>         <dbl>     <dbl>   <dbl>
1 (Intercept)                  21221.        12894.            1.65    0.138
2 exp(Start_Time)                  0.0000586     0.0000378     1.55    0.160
3 ESPN_WinPred                 -5628.        17228.           -0.327   0.752
4 exp(Start_Time):ESPN_WinPred    -0.0000440     0.0000517    -0.850   0.420
```

```r
glance(time_winpred_add_glm)$AIC
```

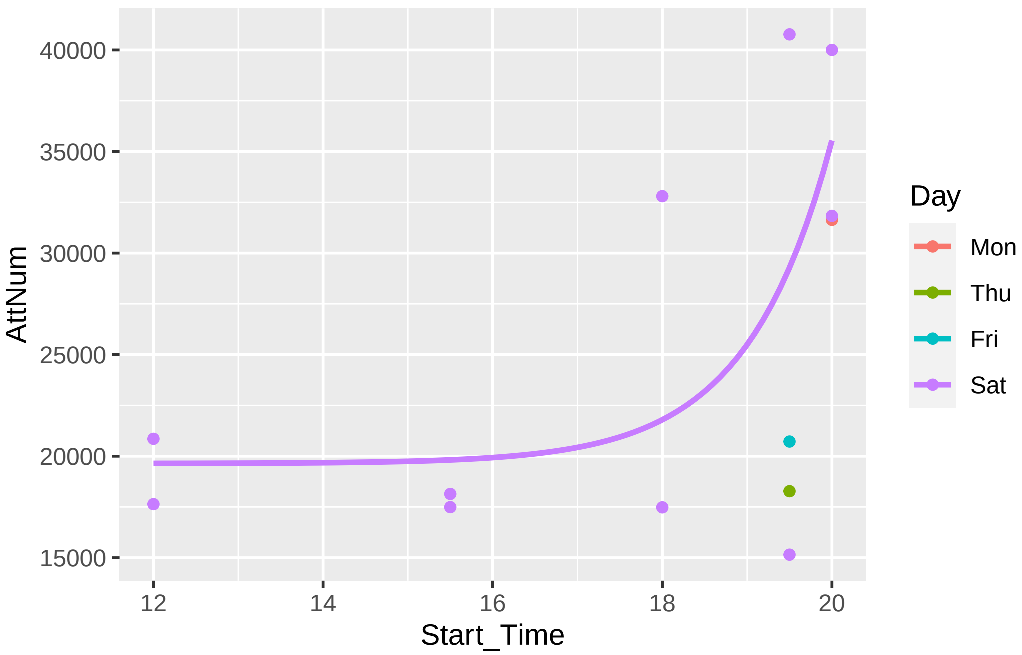
```
[1] 248.3154
```

```r
glance(time_winpred_int_glm)$AIC
```

```
[1] 249.2786
```

+ Win Chance + Day of Week

```r
home_attendance_data |>
  mutate(Day = fct_relevel(Day, "Mon", "Thu", "Fri", "Sat")) |>
  ggplot(
    aes(x = Start_Time, y = AttNum, color = Day)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) #+
```



```
  #scale_colour_viridis_c()

time_winpred_day_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) + Day + ESPN_WinPred, data = home_attendance_data)

tidy(time_winpred_day_glm)
```

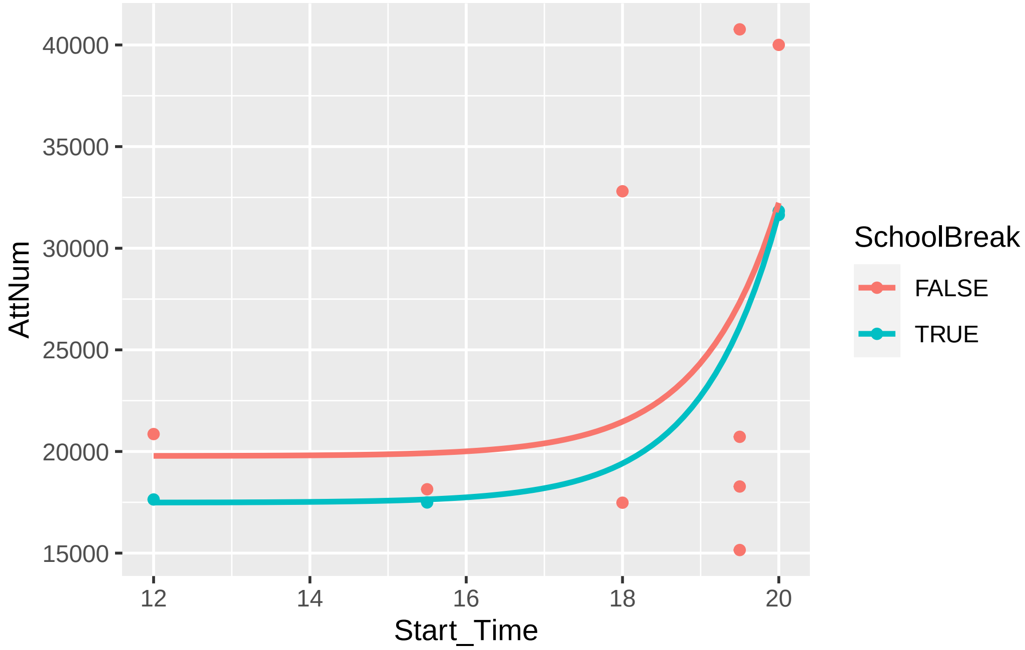
```
# A tibble: 6 x 5
  term            estimate     std.error statistic p.value
  <chr>              <dbl>         <dbl>     <dbl>   <dbl>
1 (Intercept)      2.52e+4 13436.             1.87    0.110
2 exp(Start_Time)  2.81e-5     0.0000132      2.13    0.0768
3 DayMon           7.22e+3 10322.             0.699   0.510
4 DaySat           4.63e+3  7757.             0.597   0.572
5 DayThu          -1.77e+3  9810.            -0.180   0.863
6 ESPN_WinPred    -1.65e+4 11943.            -1.38    0.217
```

```
glance(time_winpred_day_glm)$AIC
```

```
[1] 251.9767
```

+ Win Chance + School Break

```
home_attendance_data |>
  ggplot(
    aes(x = Start_Time, y = AttNum, color = SchoolBreak)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) #+
```



```
  #scale_colour_viridis_c()

time_winpred_break_int_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) * SchoolBreak * ESPN_WinPred, data = home_attendance_data)

time_winpred_break_add_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) + SchoolBreak * ESPN_WinPred, data = home_attendance_data)

tidy(time_winpred_break_int_glm)
```

```
# A tibble: 8 x 5
  term                                     estimate std.error statistic p.value
  <chr>                                       <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)                                2.35e+4   1.91e+4     1.23    0.287
2 exp(Start_Time)                            7.60e-5   5.35e-5     1.42    0.228
3 SchoolBreakTRUE                           -6.67e+3   2.73e+4    -0.244   0.819
4 ESPN_WinPred                              -4.53e+3   2.40e+4    -0.188   0.860
5 exp(Start_Time):SchoolBreakTRUE           -4.19e-5   1.67e-4    -0.251   0.814
6 exp(Start_Time):ESPN_WinPred              -1.09e-4   7.60e-5    -1.44    0.224
7 SchoolBreakTRUE:ESPN_WinPred               5.57e+3   3.81e+4     0.146   0.891
8 exp(Start_Time):SchoolBreakTRUE:ESPN_Win~  1.03e-4   2.09e-4     0.492   0.648
```

```
tidy(time_winpred_break_add_glm)
```

```
# A tibble: 5 x 5
  term                         estimate     std.error statistic p.value
  <chr>                           <dbl>         <dbl>     <dbl>   <dbl>
1 (Intercept)                   3.85e+4  9326.             4.12 0.00444
2 exp(Start_Time)               1.70e-5     0.0000122      1.40 0.205
3 SchoolBreakTRUE              -2.79e+4 21839.            -1.28 0.242
4 ESPN_WinPred                 -2.76e+4 11288.            -2.45 0.0443
5 SchoolBreakTRUE:ESPN_WinPred  4.12e+4 30306.             1.36 0.216
```

```
glance(time_winpred_break_int_glm)$AIC
```

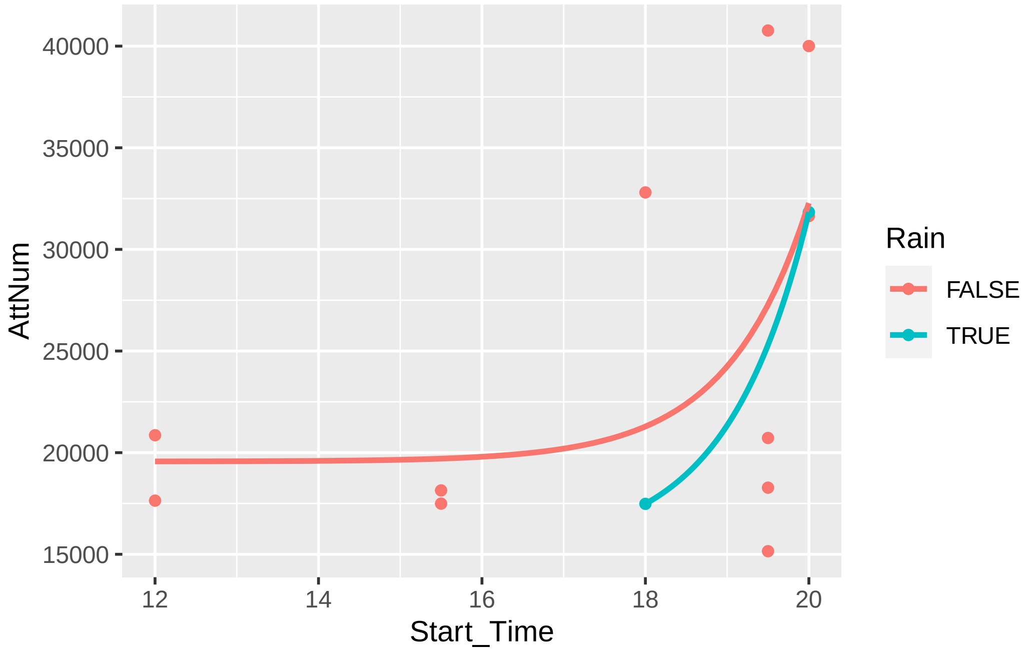
```
[1] 248.2658
```

```
glance(time_winpred_break_add_glm)$AIC
```

```
[1] 249.3467
```

+ Win Chance + Gameday Rain

```
home_attendance_data |>
  ggplot(
    aes(x = Start_Time, y = AttNum, color = Rain)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) #+
```



```
  #scale_colour_viridis_c()

time_winpred_rain_int_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) * Rain + ESPN_WinPred, data = home_attendance_data)

time_winpred_rain_add_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) + Rain + ESPN_WinPred, data = home_attendance_data)

tidy(time_winpred_rain_int_glm)
```

```
# A tibble: 5 x 5
  term                      estimate    std.error statistic p.value
  <chr>                        <dbl>        <dbl>     <dbl>   <dbl>
1 (Intercept)                3.25e+4  8410.            3.87  0.00617
2 exp(Start_Time)            2.65e-5     0.0000117     2.25  0.0588
3 RainTRUE                   5.32e+3  9413.            0.566 0.589
```

```
4 ESPN_WinPred             -2.20e+4 11233.           -1.96  0.0911
5 exp(Start_Time):RainTRUE -3.76e-6     0.0000259    -0.145 0.889
```

```
tidy(time_winpred_rain_add_glm)
```

```
# A tibble: 4 x 5
  term            estimate     std.error statistic p.value
  <chr>              <dbl>         <dbl>     <dbl>   <dbl>
1 (Intercept)      3.25e+4  7879.             4.13  0.00331
2 exp(Start_Time)  2.58e-5     0.0000101      2.57  0.0333
3 RainTRUE         4.28e+3  5653.             0.756 0.471
4 ESPN_WinPred    -2.18e+4 10427.            -2.09  0.0702
```

```
glance(time_winpred_rain_int_glm)$AIC
```

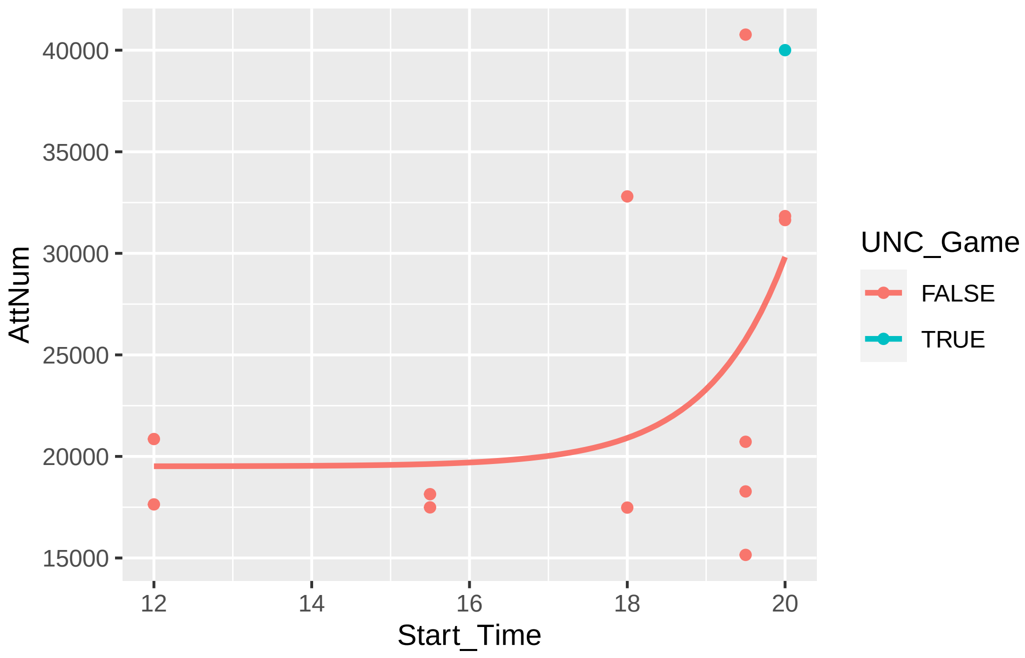
```
[1] 251.4506
```

```
glance(time_winpred_rain_add_glm)$AIC
```

```
[1] 249.4866
```

+ Win Chance + is UNC game

```
home_attendance_data |>
  ggplot(
    aes(x = Start_Time, y = AttNum, color = UNC_Game)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) #+
```



```
  #scale_colour_viridis_c()

time_winpred_UNC_int_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) * UNC_Game + ESPN_WinPred, data = home_attendance_data)

time_winpred_UNC_add_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) + UNC_Game + ESPN_WinPred, data = home_attendance_data)

tidy(time_winpred_UNC_int_glm)
```

```
# A tibble: 5 x 5
  term                         estimate     std.error statistic  p.value
  <chr>                           <dbl>         <dbl>     <dbl>    <dbl>
1 (Intercept)                   2.87e+4  8169.            3.51   0.00796
2 exp(Start_Time)               2.57e-5     0.0000107     2.40   0.0429
3 UNC_GameTRUE                  4.10e+3  8490.            0.482  0.642
4 ESPN_WinPred                 -1.56e+4 10684.           -1.46   0.183
5 exp(Start_Time):UNC_GameTRUE NA          NA            NA     NA
```

```
tidy(time_winpred_UNC_add_glm)
```

```
# A tibble: 4 x 5
  term             estimate    std.error statistic p.value
  <chr>               <dbl>        <dbl>     <dbl>   <dbl>
1 (Intercept)       2.87e+4  8169.            3.51  0.00796
2 exp(Start_Time)   2.57e-5     0.0000107     2.40  0.0429
3 UNC_GameTRUE      4.10e+3  8490.            0.482 0.642
4 ESPN_WinPred     -1.56e+4 10684.           -1.46  0.183
```

```
glance(time_winpred_UNC_int_glm)$AIC
```

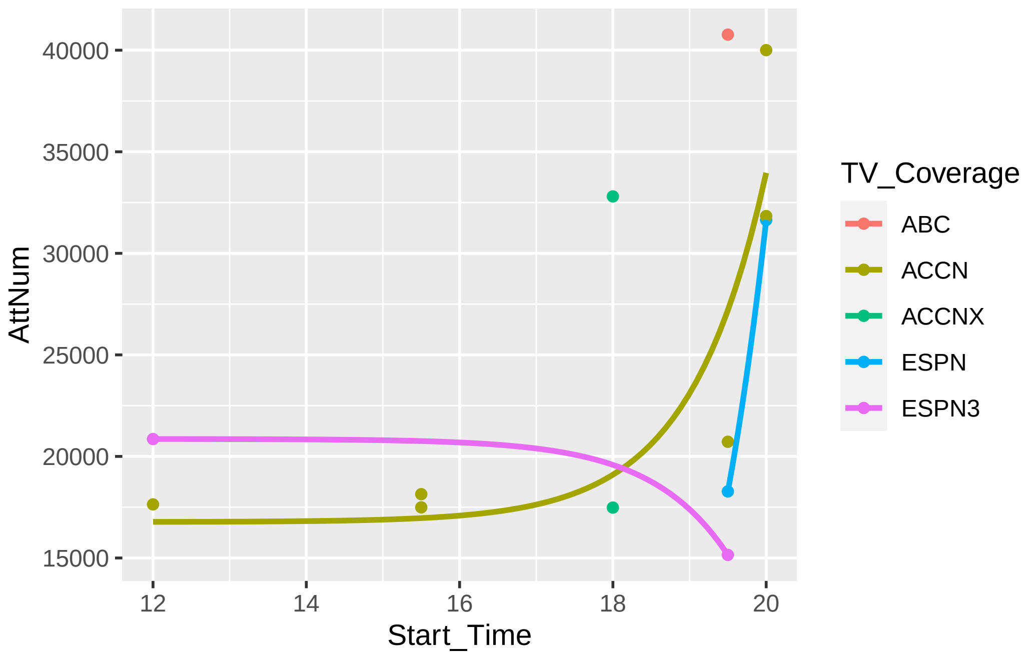
```
[1] 249.9714
```

```
glance(time_winpred_UNC_add_glm)$AIC
```

```
[1] 249.9714
```

+ Win Chance + Coverage

```
home_attendance_data |>
  ggplot(
    aes(x = Start_Time, y = AttNum, color = TV_Coverage)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) #+
```



```
  #scale_colour_viridis_c()

time_winpred_TV_int_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) * TV_Coverage * ESPN_WinPred, data = home_attendance_data)

time_winpred_TV_add_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) + TV_Coverage * ESPN_WinPred, data = home_attendance_data)

tidy(time_winpred_TV_int_glm)
```

```
# A tibble: 20 x 5
   term                                       estimate std.error statistic p.value
   <chr>                                         <dbl>     <dbl>     <dbl>   <dbl>
 1 (Intercept)                                 4.40e+4   1.22e+4     3.61   0.0689
 2 exp(Start_Time)                             4.65e-6   3.63e-5     0.128  0.910
 3 TV_CoverageACCN                            -2.55e+4   6.31e+3    -4.04   0.0562
 4 TV_CoverageACCNX                           -2.18e+4   7.71e+3    -2.83   0.106
 5 TV_CoverageESPN                            -4.68e+4   1.15e+4    -4.05   0.0558
 6 TV_CoverageESPN3                           -2.14e+4   5.19e+3    -4.13   0.0538
 7 ESPN_WinPred                               -2.14e+3   1.53e+4    -0.140  0.902
 8 exp(Start_Time):TV_CoverageACCN           5.61e-5   1.93e-5     2.91   0.100
 9 exp(Start_Time):TV_CoverageACCNX         NA        NA          NA     NA
10 exp(Start_Time):TV_CoverageESPN           1.09e-4   3.05e-5     3.58   0.0698
11 exp(Start_Time):TV_CoverageESPN3         NA        NA          NA     NA
12 exp(Start_Time):ESPN_WinPred             -4.49e-5   3.88e-5    -1.16   0.367
13 TV_CoverageACCN:ESPN_WinPred             NA        NA          NA     NA
14 TV_CoverageACCNX:ESPN_WinPred            NA        NA          NA     NA
15 TV_CoverageESPN:ESPN_WinPred             NA        NA          NA     NA
16 TV_CoverageESPN3:ESPN_WinPred            NA        NA          NA     NA
17 exp(Start_Time):TV_CoverageACCN:ESPN_Wi~ NA        NA          NA     NA
18 exp(Start_Time):TV_CoverageACCNX:ESPN_W~ NA        NA          NA     NA
19 exp(Start_Time):TV_CoverageESPN:ESPN_Wi~ NA        NA          NA     NA
20 exp(Start_Time):TV_CoverageESPN3:ESPN_W~ NA        NA          NA     NA
```

```r
tidy(time_winpred_TV_add_glm)
```

```
# A tibble: 11 x 5
   term                          estimate std.error statistic p.value
   <chr>                            <dbl>     <dbl>     <dbl>   <dbl>
 1 (Intercept)                    1.06e+4   8.80e+3     1.21   0.313
 2 exp(Start_Time)                3.25e-5   6.68e-6     4.87   0.0165
 3 TV_CoverageACCN                1.78e+4   9.54e+3     1.86   0.160
 4 TV_CoverageACCNX              -6.33e+4   1.67e+4    -3.78   0.0324
 5 TV_CoverageESPN               -9.87e+4   7.25e+4    -1.36   0.266
 6 TV_CoverageESPN3              -4.42e+4   9.80e+3    -4.52   0.0203
 7 ESPN_WinPred                   6.85e+4   2.38e+4     2.88   0.0637
 8 TV_CoverageACCN:ESPN_WinPred  -8.56e+4   2.46e+4    -3.47   0.0402
 9 TV_CoverageACCNX:ESPN_WinPred NA        NA          NA     NA
10 TV_CoverageESPN:ESPN_WinPred   5.07e+4   9.03e+4     0.561  0.614
11 TV_CoverageESPN3:ESPN_WinPred NA        NA          NA     NA
```

```r
glance(time_winpred_TV_int_glm)$AIC
```

```
[1] 229.0432
```

```r
glance(time_winpred_TV_add_glm)$AIC
```

```
[1] 233.1881
```

Best Models So Far

Code

```r
tidy(time_winpred_add_glm)
```

```
# A tibble: 3 x 5
  term            estimate     std.error statistic p.value
  <chr>              <dbl>         <dbl>     <dbl>   <dbl>
1 (Intercept)      3.03e+4 7130.              4.25 0.00215
2 exp(Start_Time)  2.76e-5    0.00000954      2.89 0.0180
3 ESPN_WinPred    -1.81e+4 8969.             -2.01 0.0750
```

```r
tidy(time_winpred_TV_int_glm)
```

```
# A tibble: 20 x 5
   term                                     estimate std.error statistic p.value
   <chr>                                       <dbl>     <dbl>     <dbl>   <dbl>
 1 (Intercept)                               4.40e+4   1.22e+4     3.61   0.0689
 2 exp(Start_Time)                           4.65e-6   3.63e-5     0.128  0.910
 3 TV_CoverageACCN                          -2.55e+4   6.31e+3    -4.04   0.0562
 4 TV_CoverageACCNX                         -2.18e+4   7.71e+3    -2.83   0.106
 5 TV_CoverageESPN                          -4.68e+4   1.15e+4    -4.05   0.0558
 6 TV_CoverageESPN3                         -2.14e+4   5.19e+3    -4.13   0.0538
 7 ESPN_WinPred                             -2.14e+3   1.53e+4    -0.140  0.902
 8 exp(Start_Time):TV_CoverageACCN           5.61e-5   1.93e-5     2.91   0.100
 9 exp(Start_Time):TV_CoverageACCNX         NA        NA          NA     NA
10 exp(Start_Time):TV_CoverageESPN           1.09e-4   3.05e-5     3.58   0.0698
11 exp(Start_Time):TV_CoverageESPN3         NA        NA          NA     NA
12 exp(Start_Time):ESPN_WinPred             -4.49e-5   3.88e-5    -1.16   0.367
13 TV_CoverageACCN:ESPN_WinPred             NA        NA          NA     NA
14 TV_CoverageACCNX:ESPN_WinPred            NA        NA          NA     NA
15 TV_CoverageESPN:ESPN_WinPred             NA        NA          NA     NA
16 TV_CoverageESPN3:ESPN_WinPred            NA        NA          NA     NA
17 exp(Start_Time):TV_CoverageACCN:ESPN_Wi~ NA        NA          NA     NA
18 exp(Start_Time):TV_CoverageACCNX:ESPN_W~ NA        NA          NA     NA
19 exp(Start_Time):TV_CoverageESPN:ESPN_Wi~ NA        NA          NA     NA
20 exp(Start_Time):TV_CoverageESPN3:ESPN_W~ NA        NA          NA     NA
```

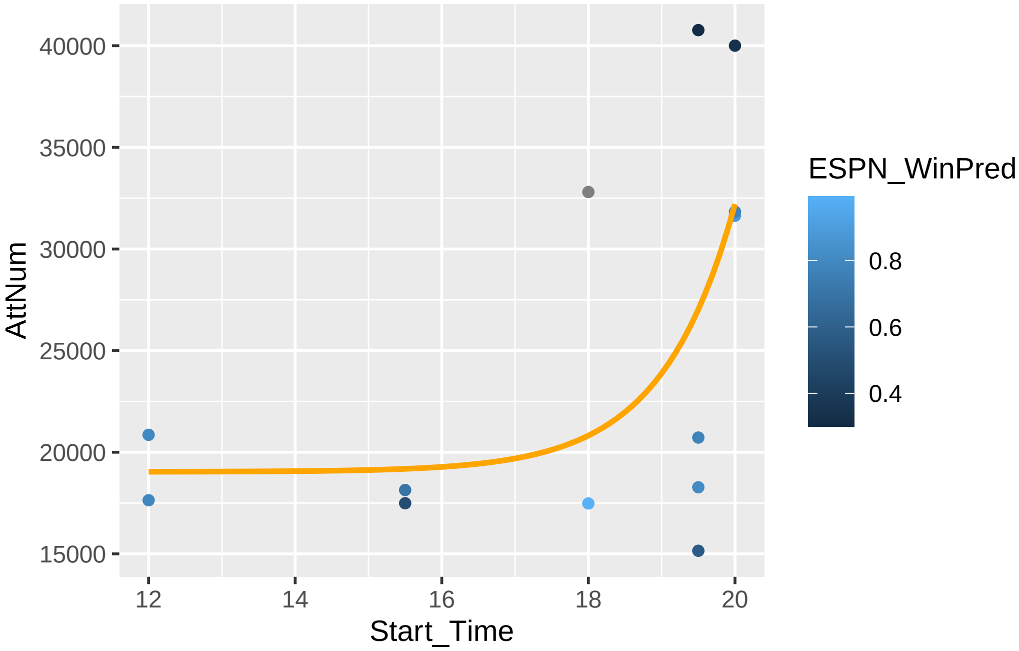
```
glance(time_winpred_add_glm)$AIC
```

```
[1] 248.3154
```

```
glance(time_winpred_TV_int_glm)$AIC
```

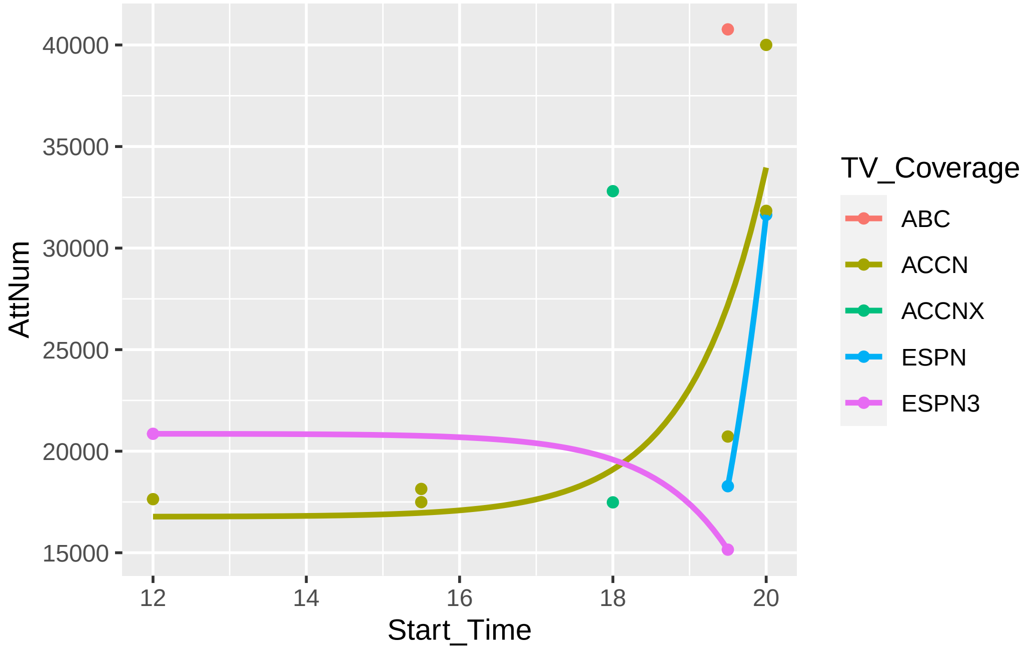
```
[1] 229.0432
```

```
home_attendance_data |>
  ggplot(
    aes(x = Start_Time, y = AttNum, color = ESPN_WinPred)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE, color = "orange") #+
```



```
  #scale_colour_viridis_c()

home_attendance_data |>
```

```r
ggplot(
  aes(x = Start_Time, y = AttNum, color = TV_Coverage)
) +
geom_point() +
geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) #+
```



```r
#scale_colour_viridis_c()
```

Model 1 (simpler):

$$\widehat{AttNum} = 30285 + 0.0000276 * e^{(Start\_Time)} - 18051 * (ESPN\_WinPred)$$

The further past 12 PM (earliest) that a game starts, the *more* people are predicted to attend. The evidence for this claim is strongly **statistically significant.**

The more likely it is that Duke will win, the *less* people are predicted to attend. The evidence for this claim is only marginally significant.

Model 2 (better matches observed attendance):

$$\widehat{AttNum} = 44002 + 0.0000047 * e^{(Start\_Time)} - 25470 * ACCN - 21778 * ACCNX - 46798 * ESPN - 21442 * ESPN3 -$$

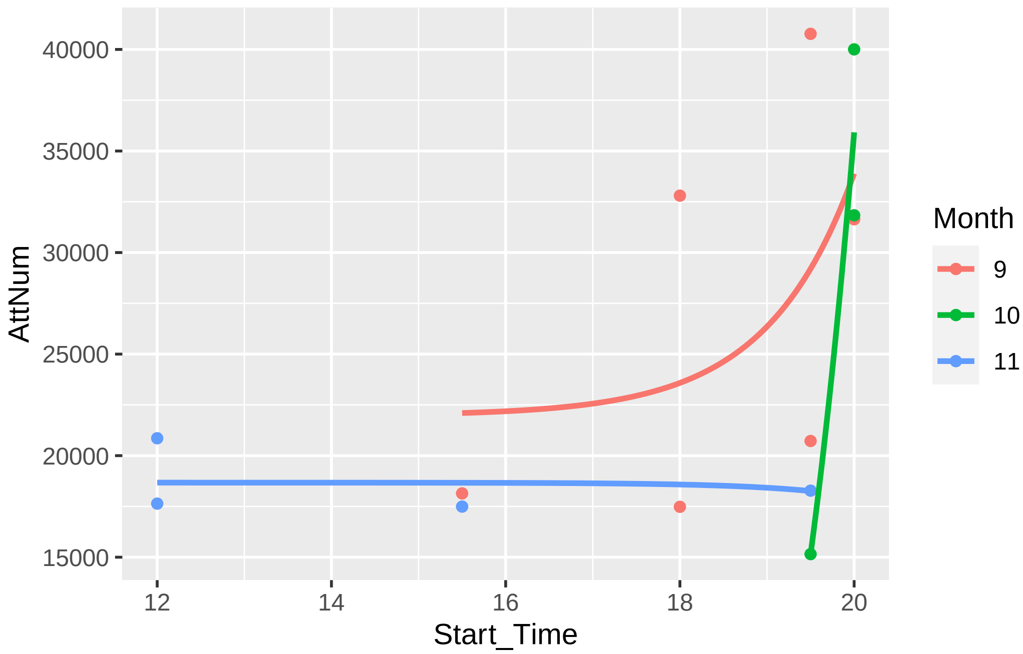
$$ACCN = \begin{cases} 1 & \text{if broadcast on ACCN} \\ 0 & \text{else} \end{cases} \; ACCNX = \begin{cases} 1 & \text{if broadcast on ACCNX} \\ 0 & \text{else} \end{cases} \; ESPN = \begin{cases} 1 & \text{if broadcast c} \\ 0 & \text{else} \end{cases}$$

Description of model 2 TBD.

Time of Day (cont.)

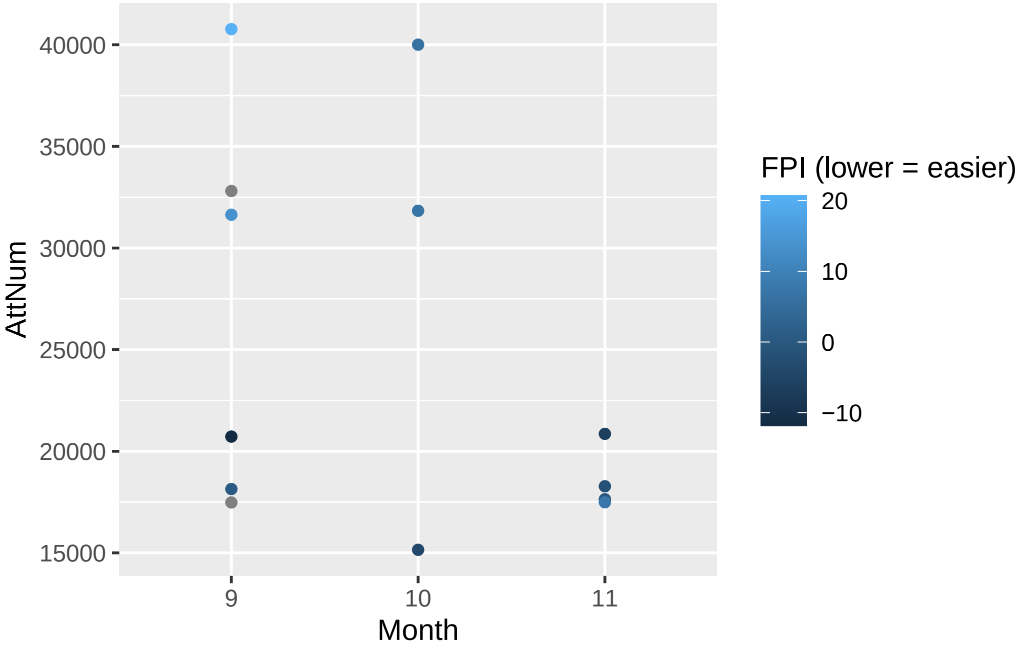
+ Win Chance + Month

```
home_attendance_data |>
  mutate(Month = as.factor(Month)) |>
  ggplot(
    aes(x = Start_Time, y = AttNum, color = Month)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) #+
```



```
    #scale_colour_viridis_c()

home_attendance_data |>
  mutate(Month = as.factor(Month)) |>
  ggplot(
    aes(x = Month, y = AttNum, color = FPI)
  ) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(color = "FPI (lower = easier)") #+
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
    #scale_colour_viridis_c()

time_winpred_month_int_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) * Month + ESPN_WinPred, data = home_attendance_data)

time_winpred_month_add_glm <- linear_reg() |>
```

```
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) + Month + ESPN_WinPred, data = home_attendance_data)

tidy(time_winpred_month_int_glm)
```

```
# A tibble: 5 x 5
  term                  estimate     std.error statistic p.value
  <chr>                    <dbl>         <dbl>     <dbl>   <dbl>
1 (Intercept)            3.08e+4 32684.          0.941    0.378
2 exp(Start_Time)        1.45e-4     0.000135    1.07     0.320
3 Month                 -1.65e+1  3045.         -0.00540  0.996
4 ESPN_WinPred          -1.85e+4  9328.         -1.98     0.0879
5 exp(Start_Time):Month -1.21e-5     0.0000136  -0.889    0.404
```

```
tidy(time_winpred_month_add_glm)
```

```
# A tibble: 4 x 5
  term            estimate     std.error statistic p.value
  <chr>              <dbl>         <dbl>     <dbl>   <dbl>
1 (Intercept)      4.93e+4 24815.            1.99   0.0822
2 exp(Start_Time)  2.49e-5     0.0000103     2.42   0.0420
3 Month           -1.81e+3  2253.           -0.802  0.446
4 ESPN_WinPred    -1.88e+4  9198.           -2.04   0.0754
```

```
glance(time_winpred_month_int_glm)$AIC
```

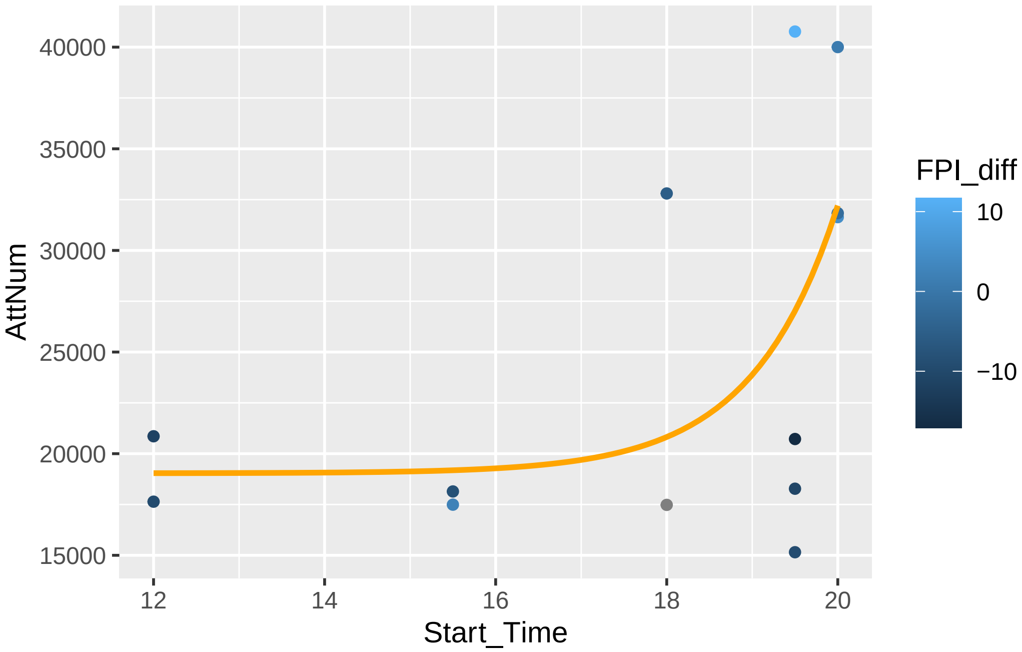
```
[1] 250.1059
```

```
glance(time_winpred_month_add_glm)$AIC
```

```
[1] 249.3883
```

+ Win Chance + Opponent Difficulty (FPI comparison)

```r
home_attendance_data |>
  ggplot(
    aes(x = Start_Time, y = AttNum, color = FPI_diff)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE, color = "orange") #+
```



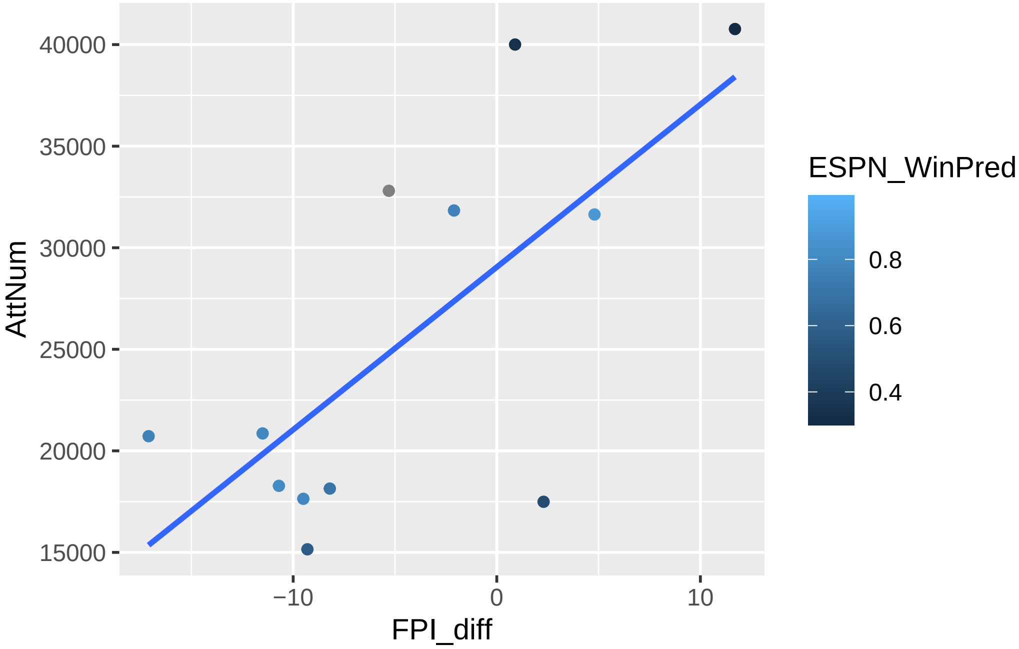
```r
  #scale_colour_viridis_c()

home_attendance_data |>
  ggplot(
    aes(x = FPI_diff, y = AttNum, color = ESPN_WinPred)
  ) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: Removed 1 rows containing non-finite values (`stat_smooth()`).
```

```
Warning: The following aesthetics were dropped during statistical transformation: colour
i This can happen when ggplot fails to infer the correct grouping structure in
  the data.
i Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?
```

```
Warning: Removed 1 rows containing missing values (`geom_point()`).
```



```
#FPI only
fpi_lm <- linear_reg() |>
  set_engine("lm") |>
  fit(AttNum ~ FPI_diff, data = home_attendance_data)

#tidy(fpi_lm)
glance(fpi_lm)$AIC #[1]
```

```
[1] 250.228
```

```
#FPI and ESPN_WinPred
fpi_winpred_int_lm <- linear_reg() |>
  set_engine("lm") |>
  fit(AttNum ~ FPI_diff * ESPN_WinPred, data = home_attendance_data)

fpi_winpred_add_lm <- linear_reg() |>
  set_engine("lm") |>
  fit(AttNum ~ FPI_diff + ESPN_WinPred, data = home_attendance_data)

#tidy(fpi_winpred_int_lm)
#tidy(fpi_winpred_add_lm)

glance(fpi_winpred_int_lm)$AIC #[2]
```

```
[1] 232.2951
```

```
glance(fpi_winpred_add_lm)$AIC #[3]
```

```
[1] 230.9492
```

```
#FPI and Time
fpi_time_int_lm <- linear_reg() |>
  set_engine("lm") |>
  fit(AttNum ~ FPI_diff * Start_Time, data = home_attendance_data)

fpi_time_add_lm <- linear_reg() |>
  set_engine("lm") |>
  fit(AttNum ~ FPI_diff + Start_Time, data = home_attendance_data)

#tidy(fpi_time_int_lm)
#tidy(fpi_time_add_lm)

glance(fpi_time_int_lm)$AIC #[4]
```

```
[1] 246.4937
```

```
glance(fpi_time_add_lm)$AIC #[5]
```

```
[1] 250.0496
```

```
#Time, ESPN_WinPred, and FPI
time_winpred_fpi_int_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) * FPI_diff + ESPN_WinPred, data = home_attendance_data)

time_winpred_fpi_add_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) + FPI_diff + ESPN_WinPred, data = home_attendance_data)

glance(time_winpred_fpi_int_glm)$AIC #[6]
```

```
[1] 221.3307
```

```
glance(time_winpred_fpi_add_glm)$AIC #[7]
```

```
[1] 226.274
```

```
#Time, ESPN_WinPred, TV, and FPI
time_winpred_TV_fpi_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(Start_Time) * TV_Coverage + FPI_diff * ESPN_WinPred, data = home_attend

glance(time_winpred_TV_fpi_glm)$AIC #[8]
```

```
[1] 214.2946
```

Best Model Using FPI

```
tidy(time_winpred_fpi_int_glm)
```

```
# A tibble: 5 x 5
  term                      estimate     std.error statistic p.value
  <chr>                        <dbl>         <dbl>     <dbl>   <dbl>
1 (Intercept)                2.71e+4 5565.              4.87  0.00278
2 exp(Start_Time)            3.54e-5    0.00000924      3.83  0.00863
3 FPI_diff                  -4.67e+2  478.             -0.977 0.367
4 ESPN_WinPred              -1.72e+4 9564.             -1.79  0.123
5 exp(Start_Time):FPI_diff   3.21e-6    0.00000140      2.30  0.0613
```

```r
glance(time_winpred_fpi_int_glm)$AIC
```

```
[1] 221.3307
```

$$\widehat{AttNum} = 27119 + 0.0000354 * e^{(Start\_Time)} - 466.638(FPI\_diff) - 17156 * (ESPN\_WinPred) + 0.00000321 * (F$$

The further past 12 PM (earliest) that a game starts, the *more* people are predicted to attend. *(very likely)*

When the opponent has a greater Power Index rating, somewhat *less* people are predicted to attend. *(uncertain claim)*

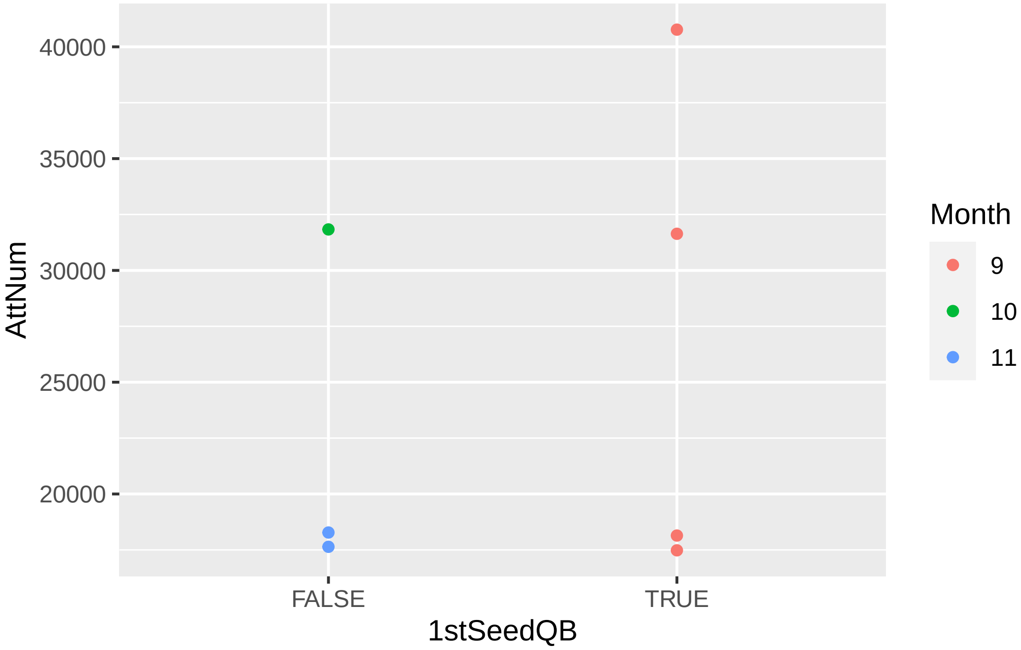
The more greatly Duke is predicted to win by ESPN, the *less* people are predicted to attend. *(somewhat uncertain claim)*

Start time and opponent Football Power Index (FPI) rating are likely *not* independent variables.

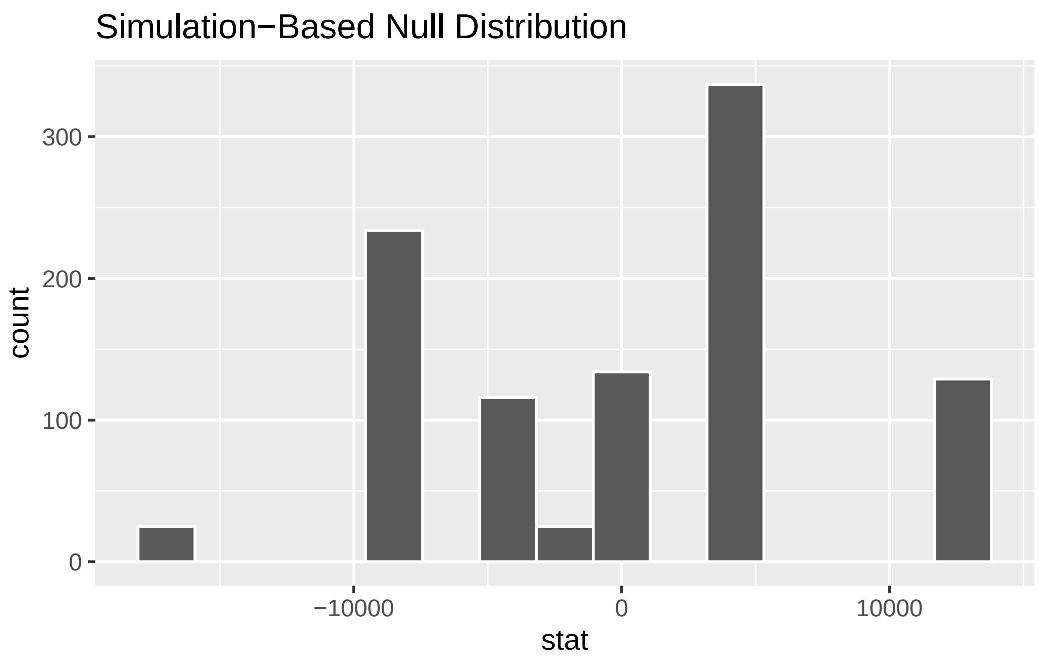
1st Seed Quarterback

With and Without

```r
# Data
home_attendance_data_2023 <- home_attendance_data |>
  filter(Year == 2023) |>
  mutate(DateDecimal = Month + Date/31) |>
  mutate(Month = as.factor(Month))

# Visualization
home_attendance_data_2023 |>
  ggplot(
    aes(x = `1stSeedQB`, y = AttNum, color = Month)
  ) +
  geom_point()
```



```r
# Hypothesis Test
set.seed(12345)

qb_null_dist <- home_attendance_data_2023 |>
  specify(response = AttNum,
          explanatory = `1stSeedQB`) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means",
            order = c("TRUE", "FALSE"))

visualize(qb_null_dist)
```



```
qb_null_dist |>
  get_confidence_interval(level = 0.90, type = "percentile")
```

```
# A tibble: 1 x 2
  lower_ci upper_ci
     <dbl>    <dbl>
1   -8993.    12796
```

Whether or not Riley Leonard started as quarterback in a game was *not* a statistically significant predictor of Duke home-game attendance in 2022-23.

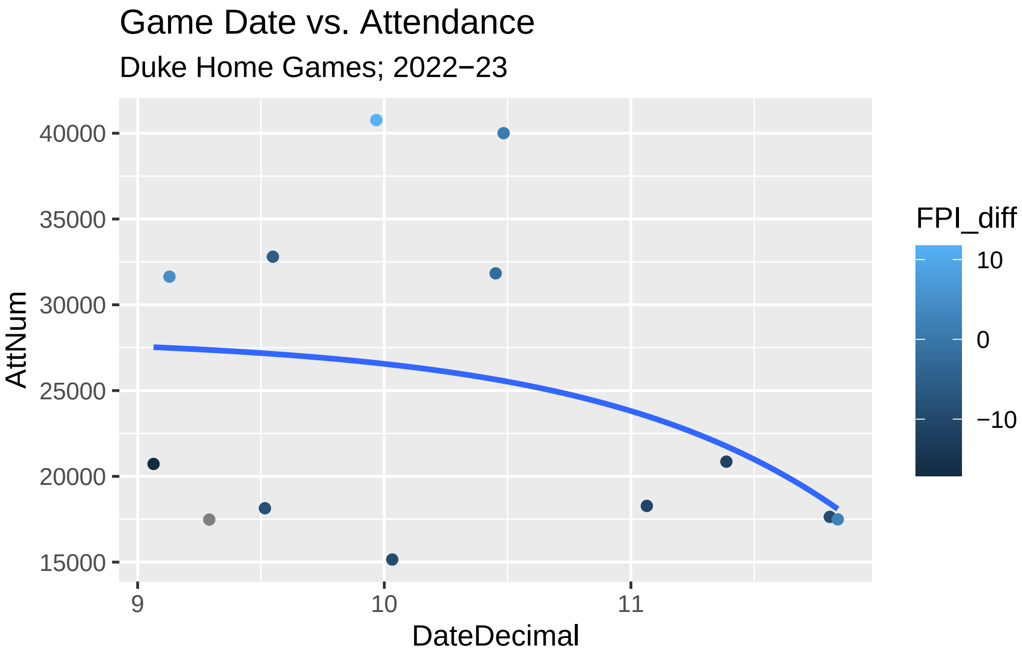
Season-Relative Date

2022-2023

```
# Calculation of time variable
home_attendance_data_chron <- home_attendance_data |>
  mutate(DateDecimal = Month + Date/31)

# Visualization
```

```
home_attendance_data_chron |>
  ggplot(
    aes(x = DateDecimal, y = AttNum, color = FPI_diff)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) +
  labs(title = "Game Date vs. Attendance",
       subtitle = "Duke Home Games; 2022-23")
```

```
Warning: The following aesthetics were dropped during statistical transformation: colour
i This can happen when ggplot fails to infer the correct grouping structure in
  the data.
i Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?
```



```
# Model
date_att_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(DateDecimal), data = home_attendance_data_chron)

tidy(date_att_glm)
```

```
# A tibble: 2 x 5
  term               estimate std.error statistic    p.value
  <chr>                 <dbl>     <dbl>     <dbl>      <dbl>
1 (Intercept)      28156.     3556.          7.92 0.00000721
2 exp(DateDecimal)    -0.0726    0.0556     -1.30 0.219
```

```
glance(date_att_glm)$AIC
```

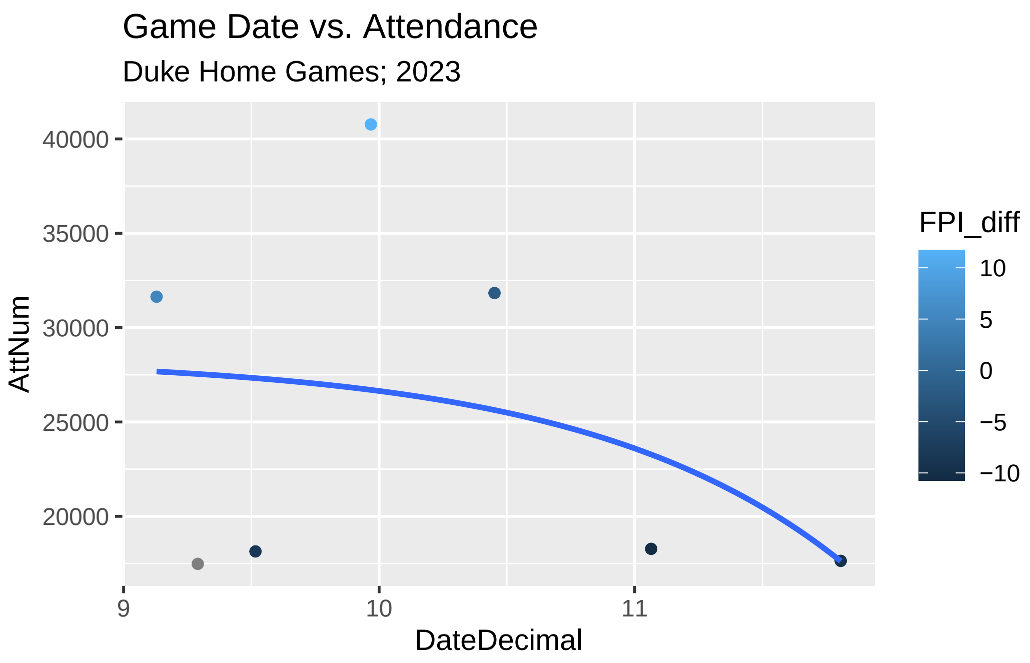
```
[1] 277.2961
```

The date on which a game took place relative to the season timeline was *not* a statistically significant predictor of Duke home-game attendance in 2022-23.

2023 only

```
# Calculation of time variable
home_attendance_data_chron_2023 <- home_attendance_data_chron |>
  filter(Year == 2023)

# Visualization
home_attendance_data_chron_2023 |>
  ggplot(
    aes(x = DateDecimal, y = AttNum, color = FPI_diff)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) +
  labs(title = "Game Date vs. Attendance",
       subtitle = "Duke Home Games; 2023")
```

```
Warning: The following aesthetics were dropped during statistical transformation: colour
i This can happen when ggplot fails to infer the correct grouping structure in
  the data.
i Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?
```



```
# Model
date_att_glm <- linear_reg() |>
  set_engine("glm") |>
  fit(AttNum ~ exp(DateDecimal), data = home_attendance_data_chron_2023)

tidy(date_att_glm)
```

```
# A tibble: 2 x 5
  term             estimate std.error statistic p.value
  <chr>               <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)      28416.     5091.       5.58  0.00255
2 exp(DateDecimal)    -0.0805    0.0868    -0.927 0.396
```

```
glance(date_att_glm)$AIC
```

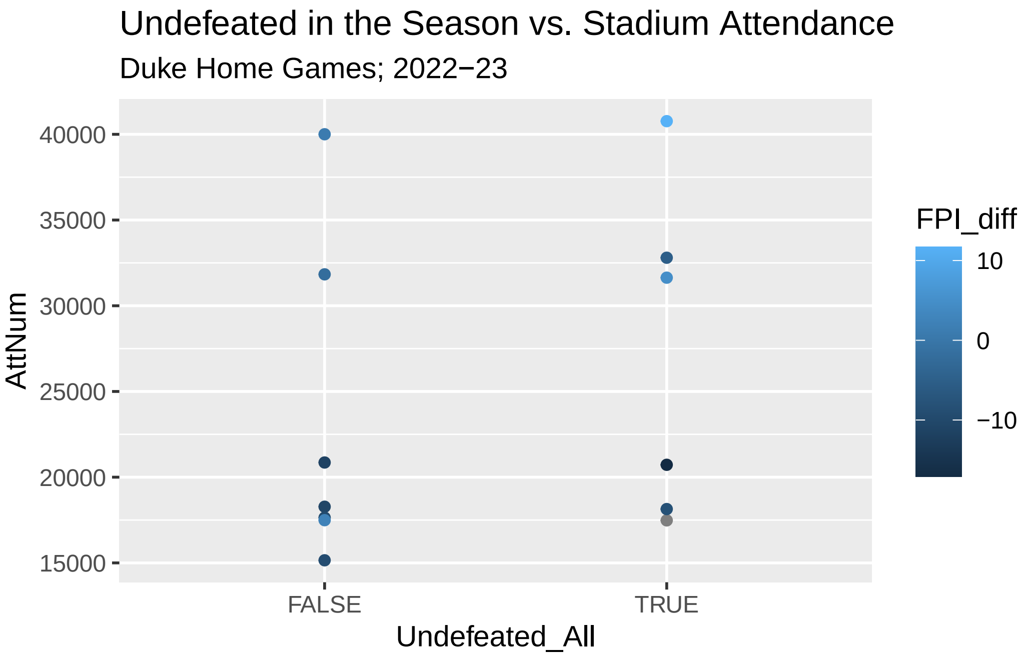
```
[1] 151.9114
```

The same is true when examining the 2023 season alone; the date on which a game took place relative to the season timeline was *not* a statistically significant predictor of Duke home-game attendance in 2023.

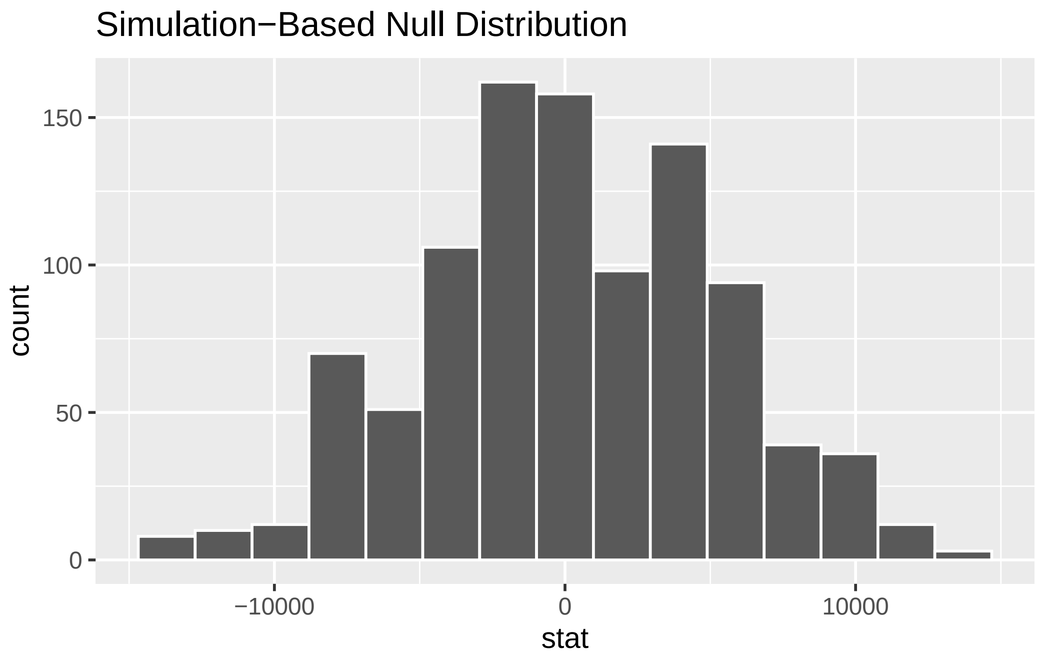
Following a Duke Loss

Undefeated Overall (in the Season)

```
# Visualization
home_attendance_data |>
  ggplot(
    aes(x = Undefeated_All, y = AttNum, color = FPI_diff)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) +
  labs(title = "Undefeated in the Season vs. Stadium Attendance",
       subtitle = "Duke Home Games; 2022-23")
```



```
# Hypothesis Test
set.seed(12345)

undefeated_all_null_dist <- home_attendance_data |>
  specify(response = AttNum,
          explanatory = Undefeated_All) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means",
            order = c("TRUE", "FALSE"))

visualize(undefeated_all_null_dist)
```



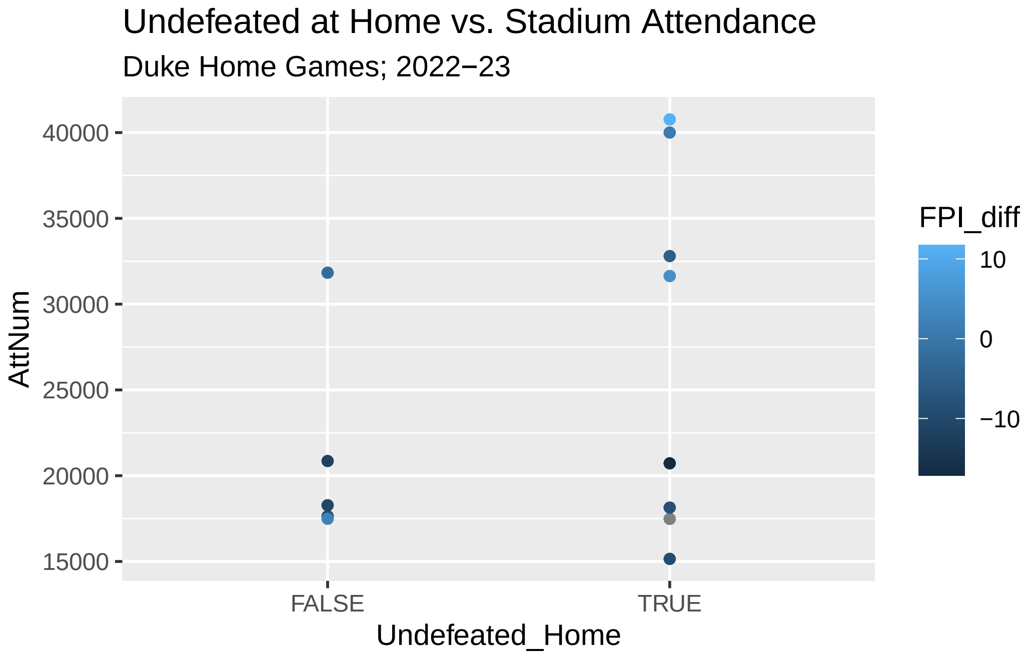
```
undefeated_all_null_dist |>
  get_confidence_interval(level = 0.90, type = "percentile")
```

```
# A tibble: 1 x 2
  lower_ci upper_ci
     <dbl>    <dbl>
1   -8445.    8932.
```

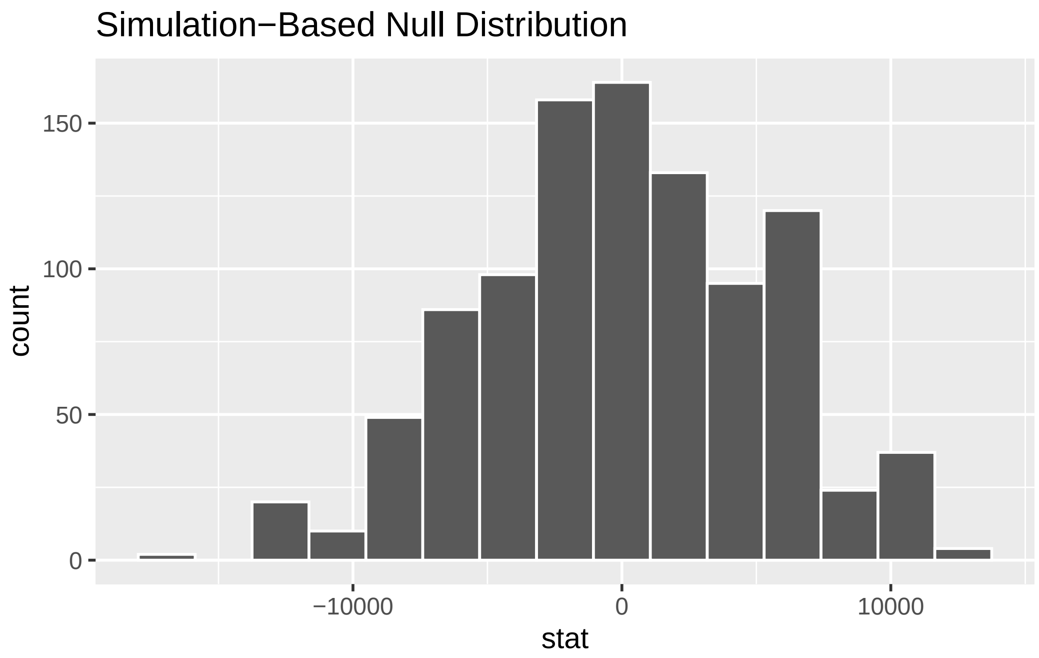
Whether or not Duke was undefeated in a season before a game was *not* a statistically significant predictor of Duke home-game attendance in 2022-23.

Undefeated at Home (in the Season)

```
# Visualization
home_attendance_data |>
  ggplot(
    aes(x = Undefeated_Home, y = AttNum, color = FPI_diff)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) +
  labs(title = "Undefeated at Home vs. Stadium Attendance",
       subtitle = "Duke Home Games; 2022-23")
```



```
# Hypothesis Test
set.seed(12345)

undefeated_home_null_dist <- home_attendance_data |>
  specify(response = AttNum,
          explanatory = Undefeated_Home) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means",
              order = c("TRUE", "FALSE"))

visualize(undefeated_home_null_dist)
```



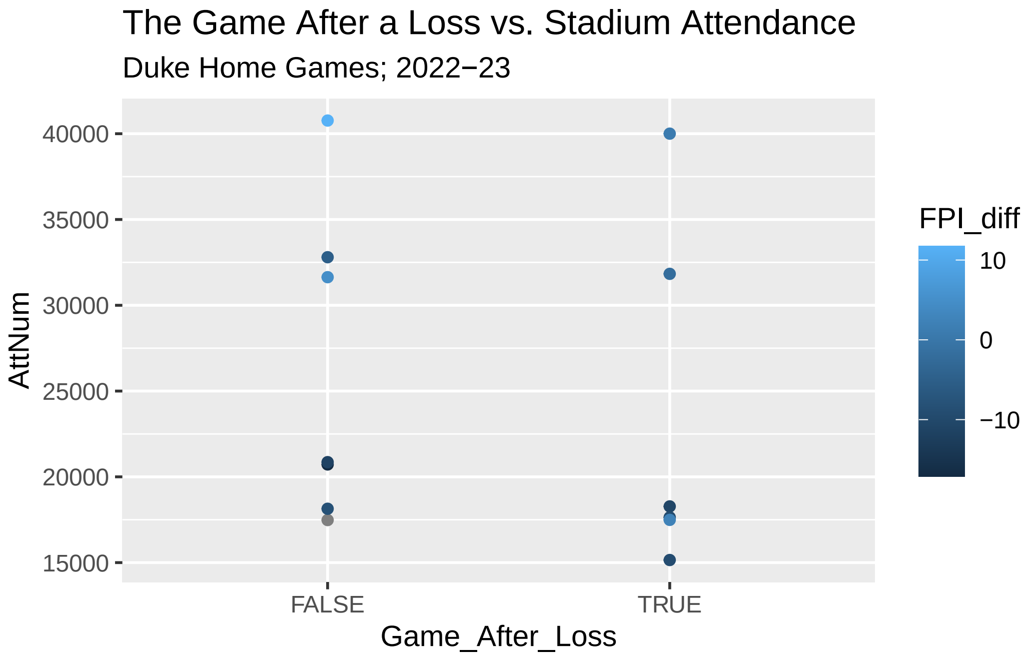
```
undefeated_home_null_dist |>
  get_confidence_interval(level = 0.90, type = "percentile")
```

```
# A tibble: 1 x 2
  lower_ci upper_ci
     <dbl>    <dbl>
1   -8605.    7786.
```

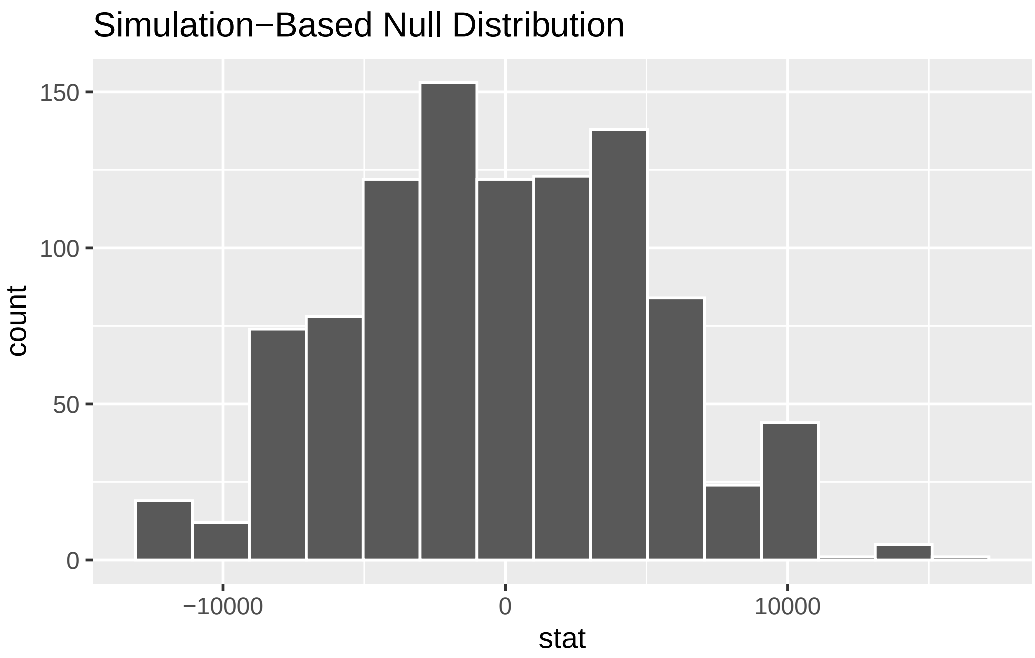
Whether or not Duke was undefeated (during the season) on their home field was *not* a statistically significant predictor of Duke home-game attendance in 2022-23.

Games Directly After a Duke Loss

```
# Visualization
home_attendance_data |>
  ggplot(
    aes(x = Game_After_Loss, y = AttNum, color = FPI_diff)
  ) +
  geom_point() +
  geom_smooth(method = "glm", formula = y ~ exp(x), se = FALSE) +
  labs(title = "The Game After a Loss vs. Stadium Attendance",
       subtitle = "Duke Home Games; 2022-23")
```



```
# Hypothesis Test
set.seed(12345)

recent_defeat_null_dist <- home_attendance_data |>
  specify(response = AttNum,
          explanatory = Game_After_Loss) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "diff in means",
            order = c("TRUE", "FALSE"))

visualize(recent_defeat_null_dist)
```



```
recent_defeat_null_dist |>
  get_confidence_interval(level = 0.90, type = "percentile")
```

```
# A tibble: 1 x 2
  lower_ci upper_ci
     <dbl>    <dbl>
1   -8537.    9139.
```

Whether or not a game directly followed a Duke loss was *not* a statistically significant predictor of Duke home-game attendance in 2022-23.