Modeling

Linus Jen

8/6/2020

Packages

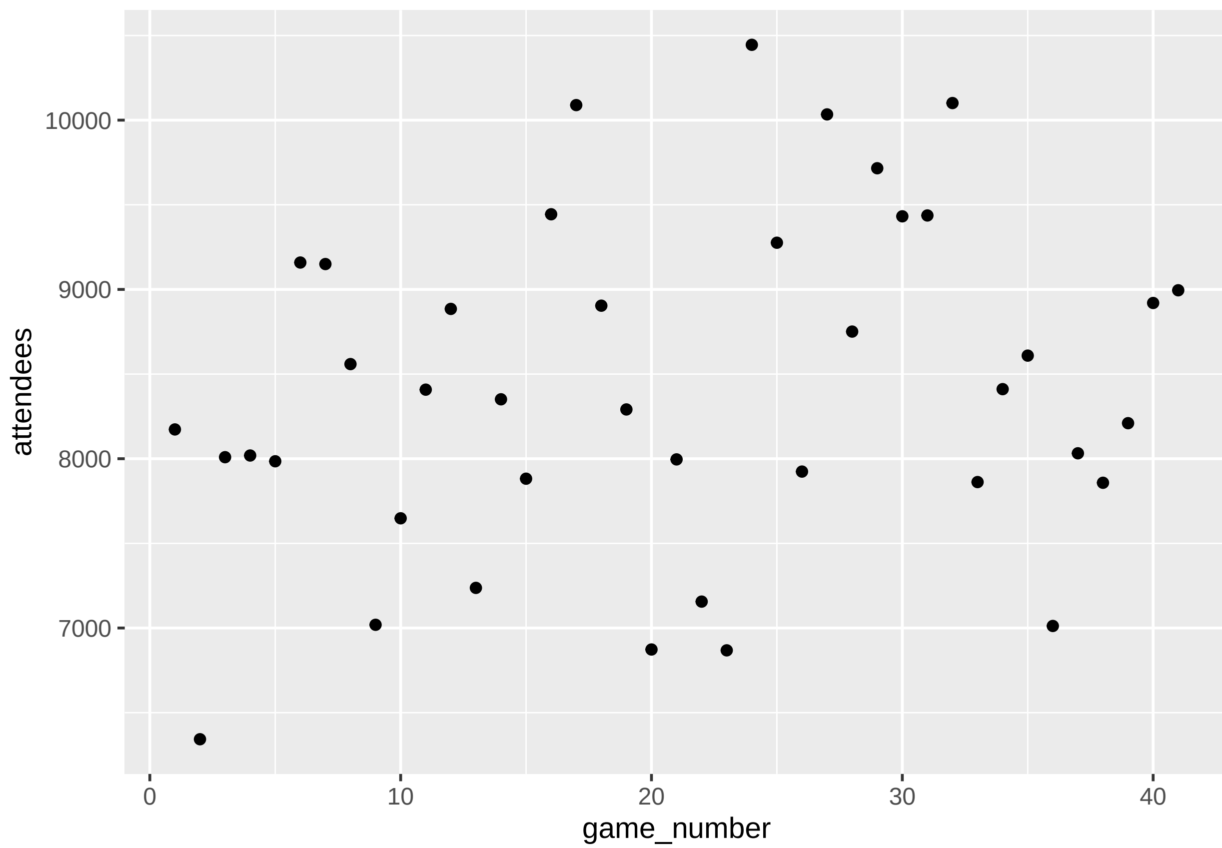
```
library(dplyr)
library(ggplot2)
library(ggthemes)
library(readxl)
library(car)
library(lubridate)
```

```
# Pull our data in
data = read_xlsx("final_dataset.xlsx")

# Add in a total game column
data = data %>% mutate(games_played = sum(win_col, lose_col), day_of_week = wday(date), weekday_status =

names(data)[19] = "Follower_Count_mil"
```

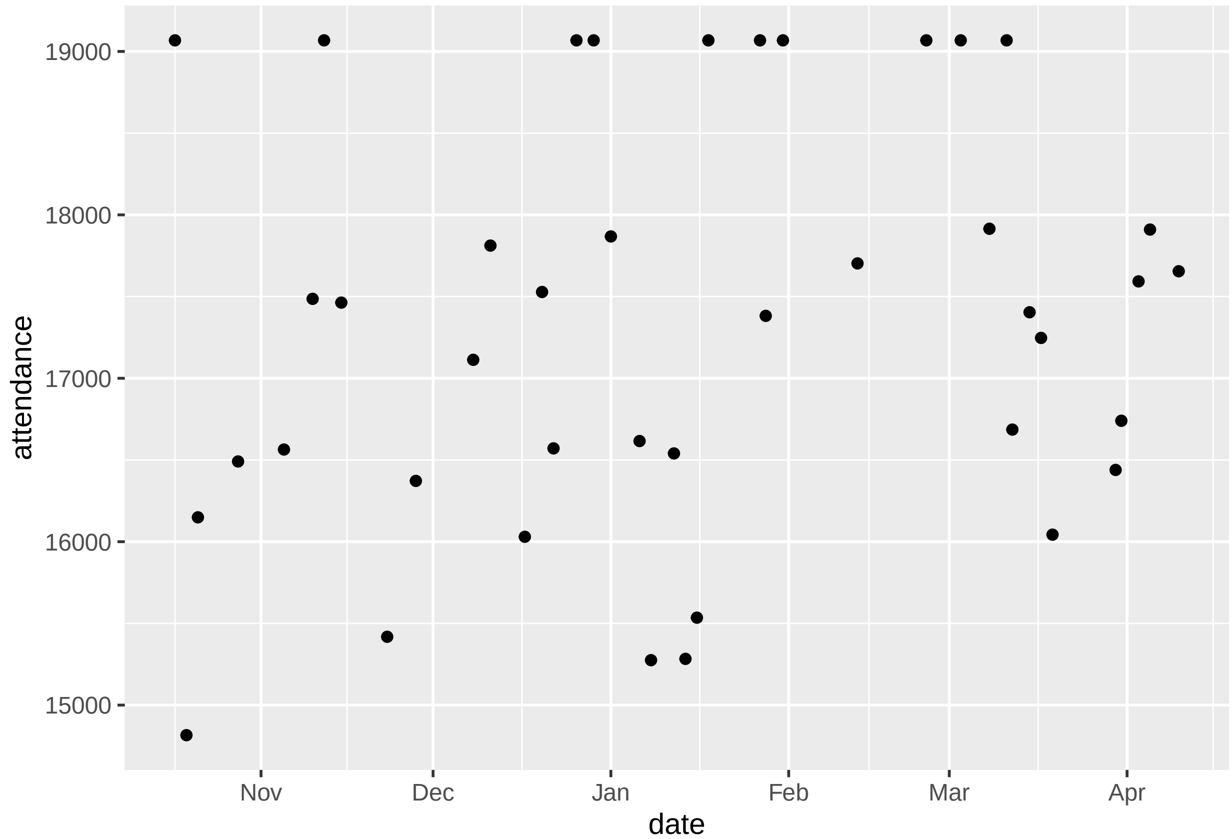
"Fixing" the Attendances

To train my model on previous seasons, I scrapped the attendances for past home games. However, when comparing the scraped attendance data for the 2018-19 season with the attendance data given by the Clippers, I couldn't help but notice a large discrepancy between the two. The following code was used to check how similar the data is, and see if there is a common value I can apply to the "real" attendances so that the models can better predict

```
real_attendees = read_xlsx("attendees_by_game_2018.xlsx")
test_prelim = data[data$season == "2018",]

ggplot(real_attendees, aes(x = game_number, y = attendees)) + geom_point()
```



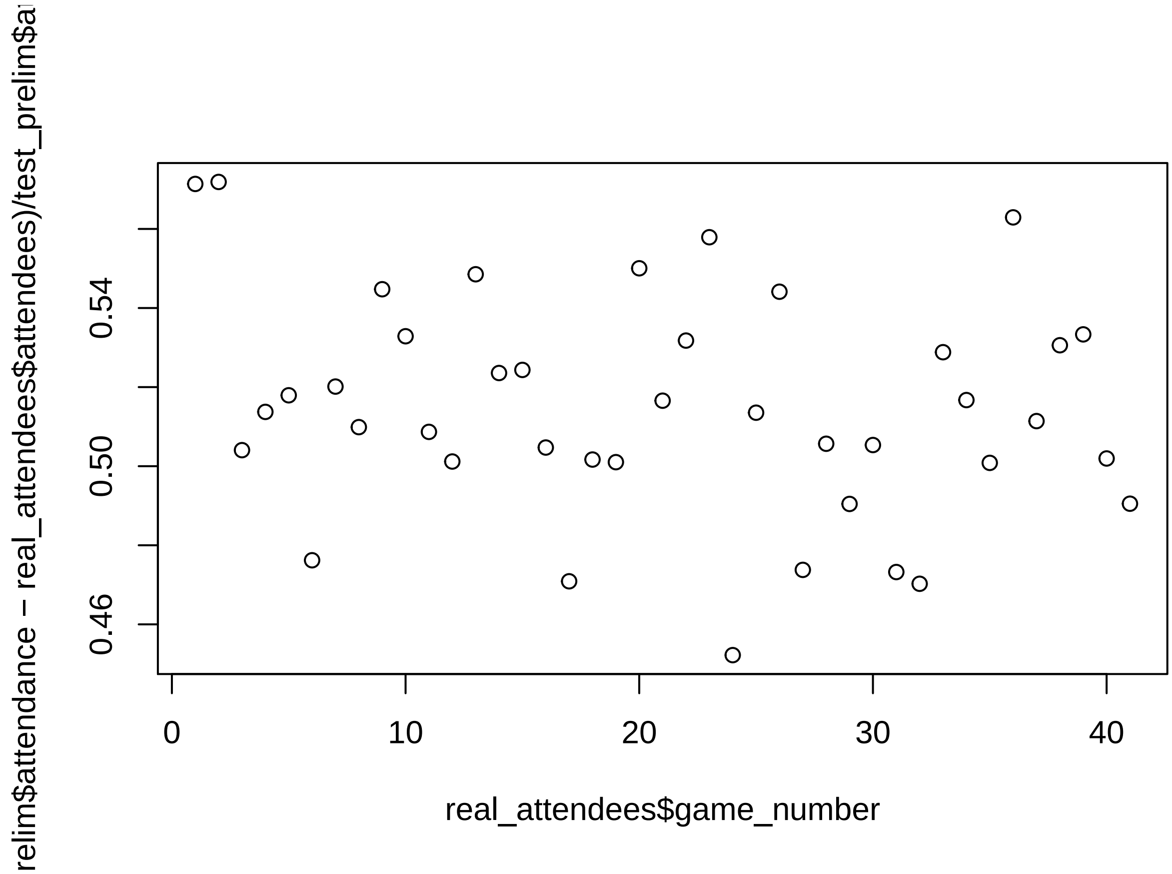

```r
ggplot(test_prelim, aes(x = date, y = attendance)) + geom_point()
```



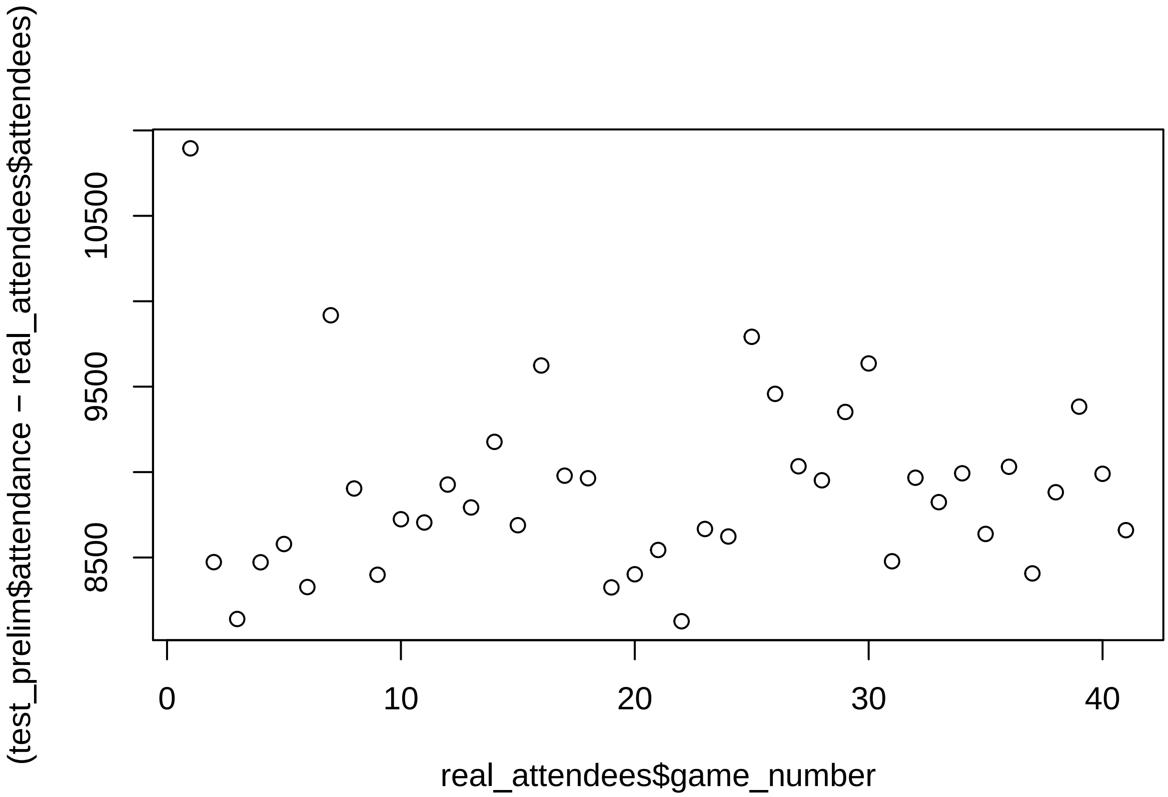

```r
cor(real_attendees$attendees, test_prelim$attendance)
```

```
## [1] 0.9149014
```

```r
plot(x = real_attendees$game_number, y = (test_prelim$attendance - real_attendees$attendees)/test_preli
```




```
plot(x = real_attendees$game_number, y = (test_prelim$attendance - real_attendees$attendees))
```




```
data$attendance = data$attendance * 0.5
```

We see from the graphs that there is roughly a 50% difference in attendees. Thus, we will multiply all the attendances from our scraped data with 0.5

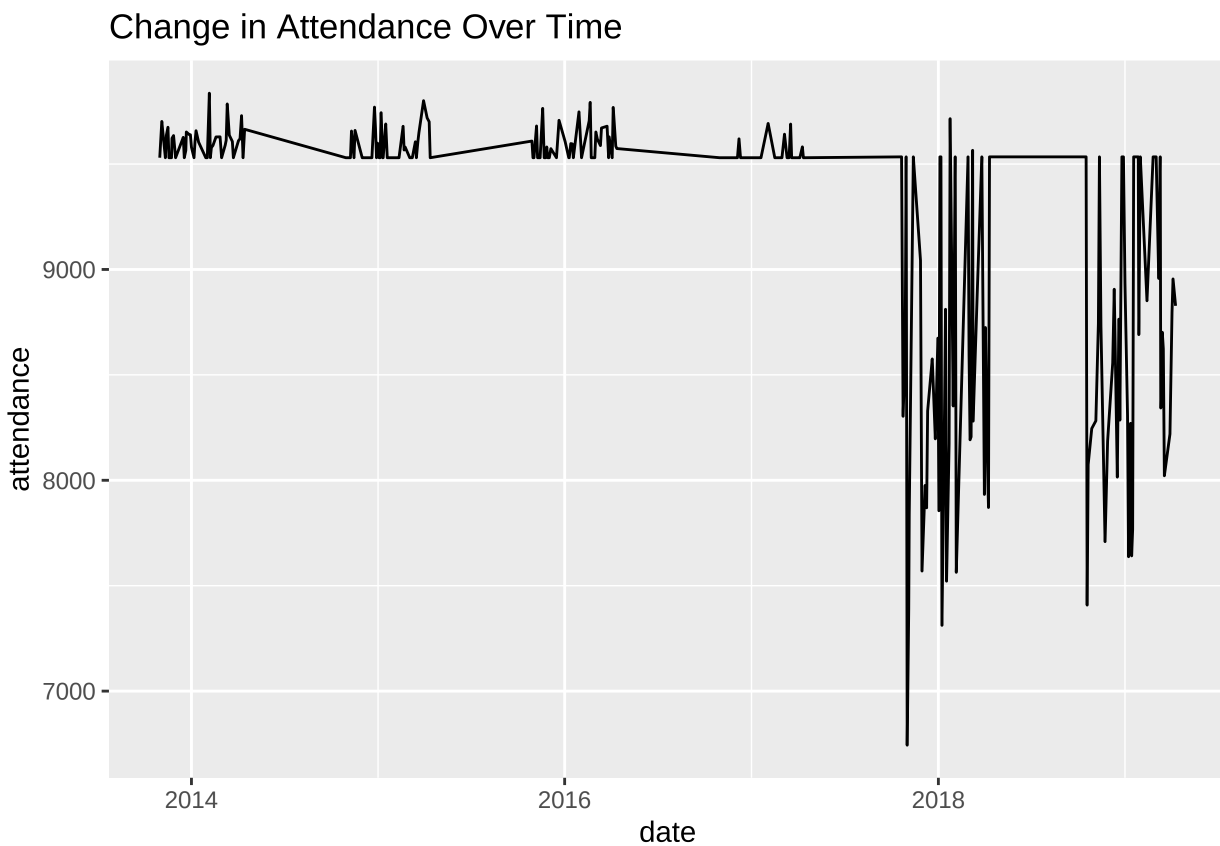
Here, we can break the data into training and testing

```
# Take a look at the data
# glimpse(data)

# Create our training and testing data, splitting by the seasons initially
train_prelim = data[data$season != "2018",]
test_prelim = data[data$season == "2018",]

# We can further break down the training and testing data at a later time

# Lastly, we have the "attendances" of the 2018-19 season given to us.
# Now, we need to standardize our data so that the averages align, as the data # provided by the Clippe
```

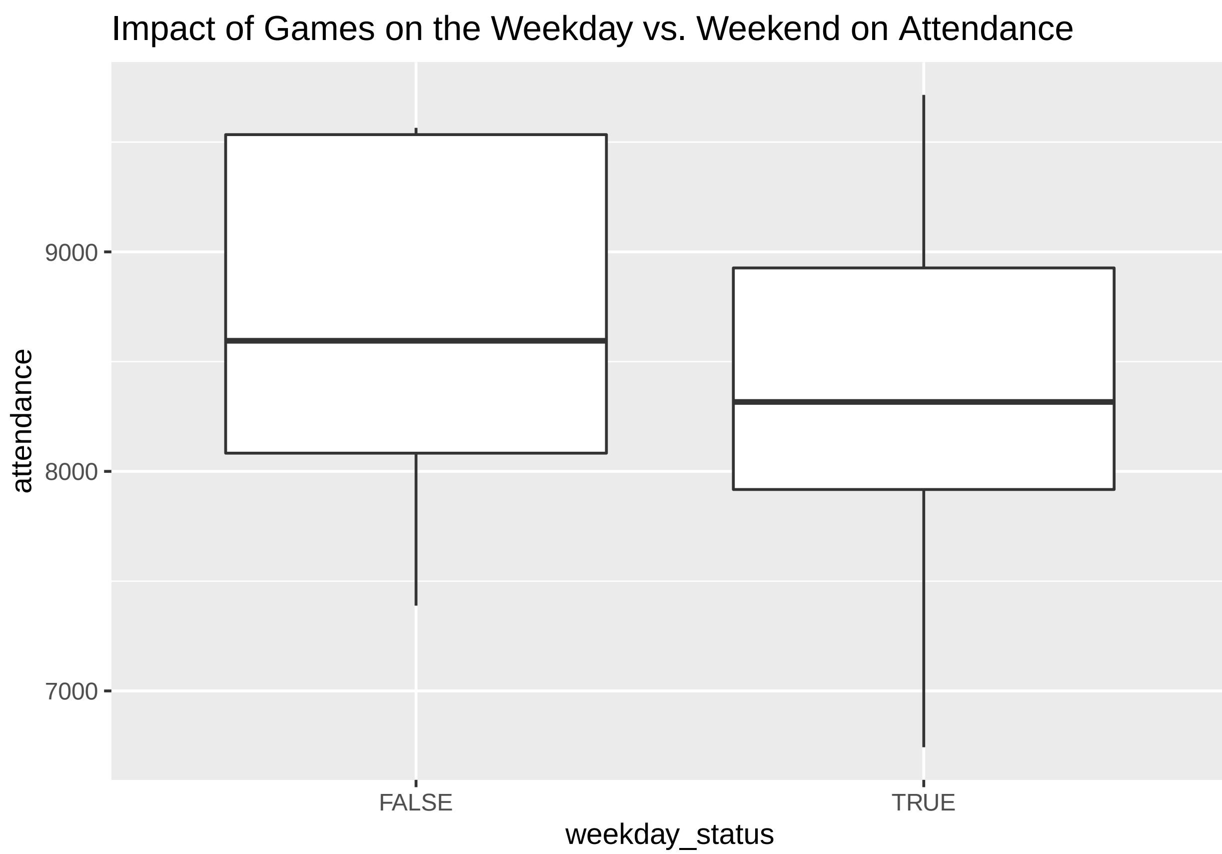
Visualizing the data

Note that the columns of importance are: win_col, lose_col, own_streak, attendance, op_W, op_L, win_perc, op_win_perc, all_star_num, pop_team, and odds.

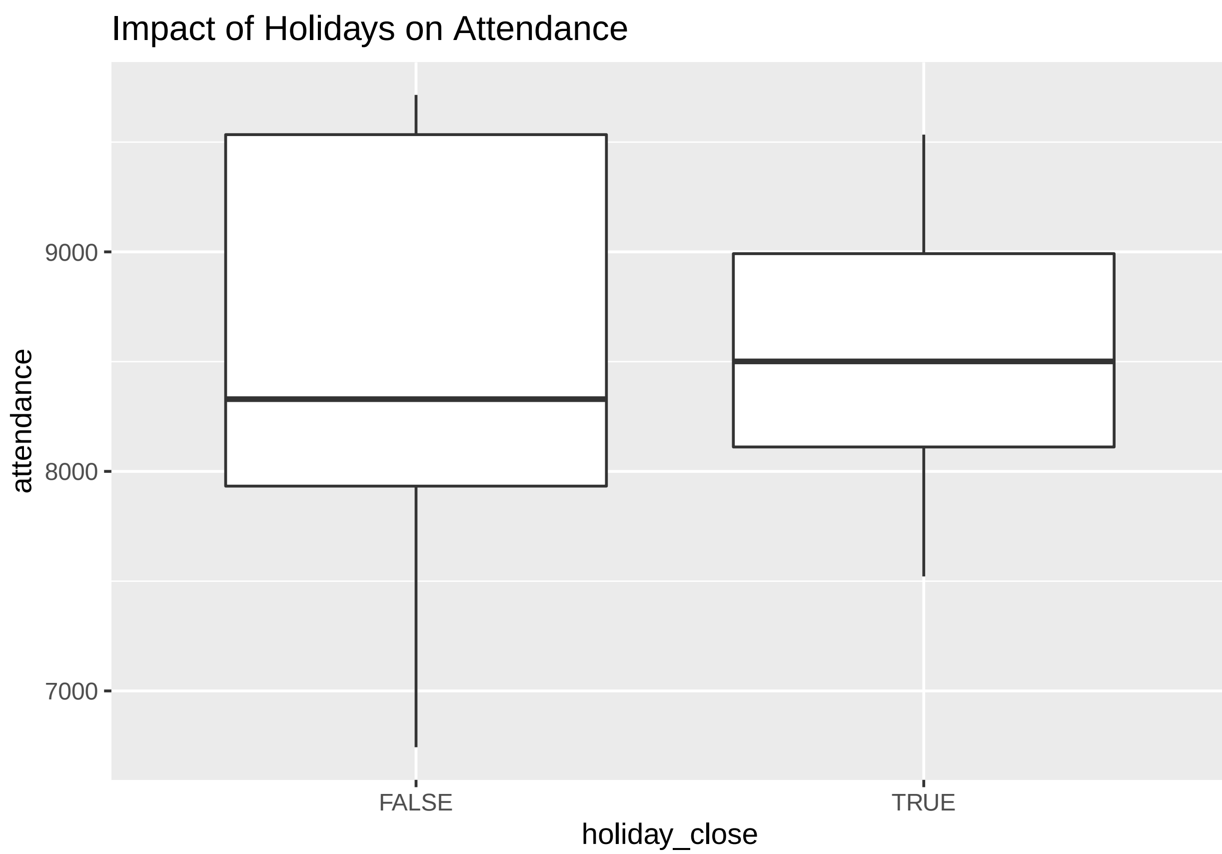
```
# First, graph the change in attendance over the past 6 years
ggplot(data, aes(x = date, y = attendance)) +
  geom_line() +
  labs(title = "Change in Attendance Over Time")
```



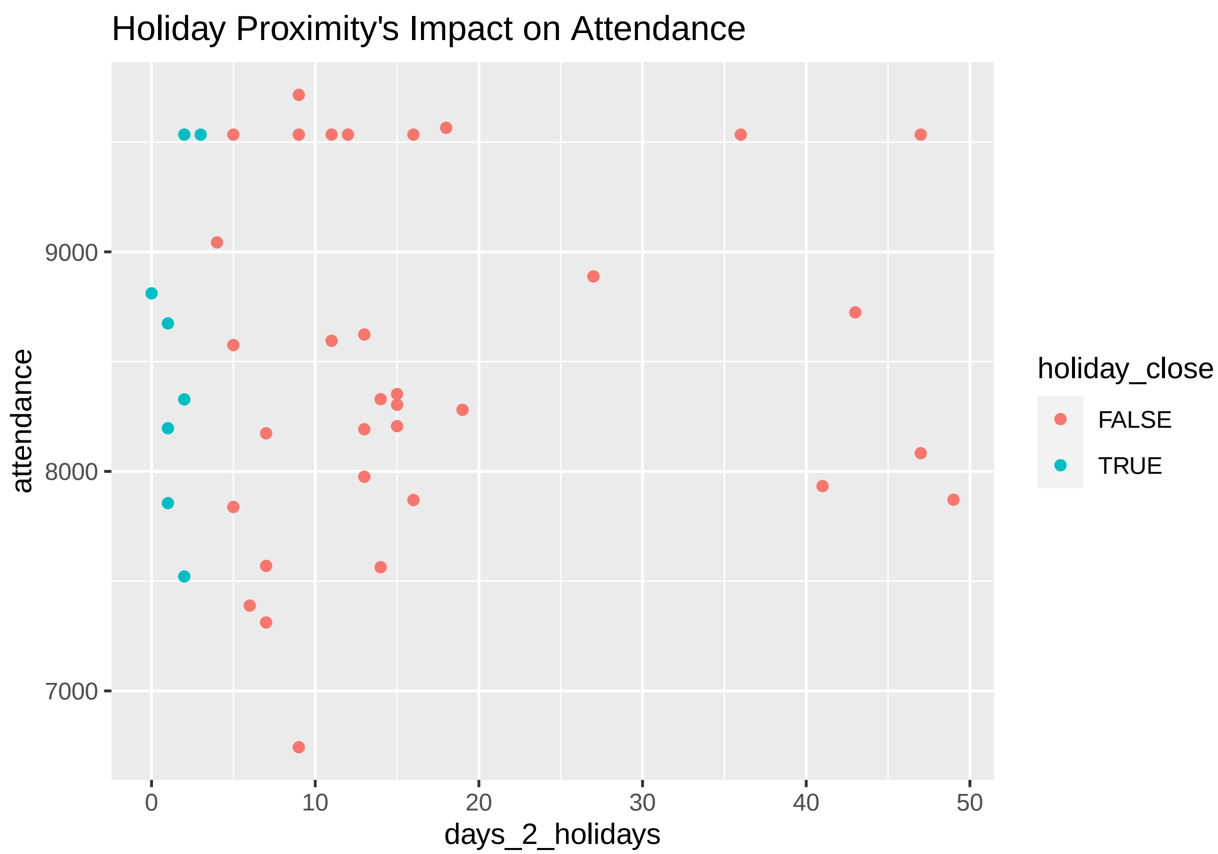

```
# What is the most striking is that the Clippers consistenly had over 19000
# attendees per game, but with the 2017/18 season, viewership dramatically
# dropped. I would attribute this to the loss of Chris Paul that year,
# and we see that the 2018/19 season has similar views.
# Thus, I will only use the 2017/18 season to predict the 2018/19 attendance

train_17 = train_prelim[train_prelim$season == "2017",]

# Let's look at the boxplots for weekday_status and attendance
ggplot(train_17, aes(x = weekday_status, y = attendance)) +
  geom_boxplot() +
  labs(title = "Impact of Games on the Weekday vs. Weekend on Attendance")
```



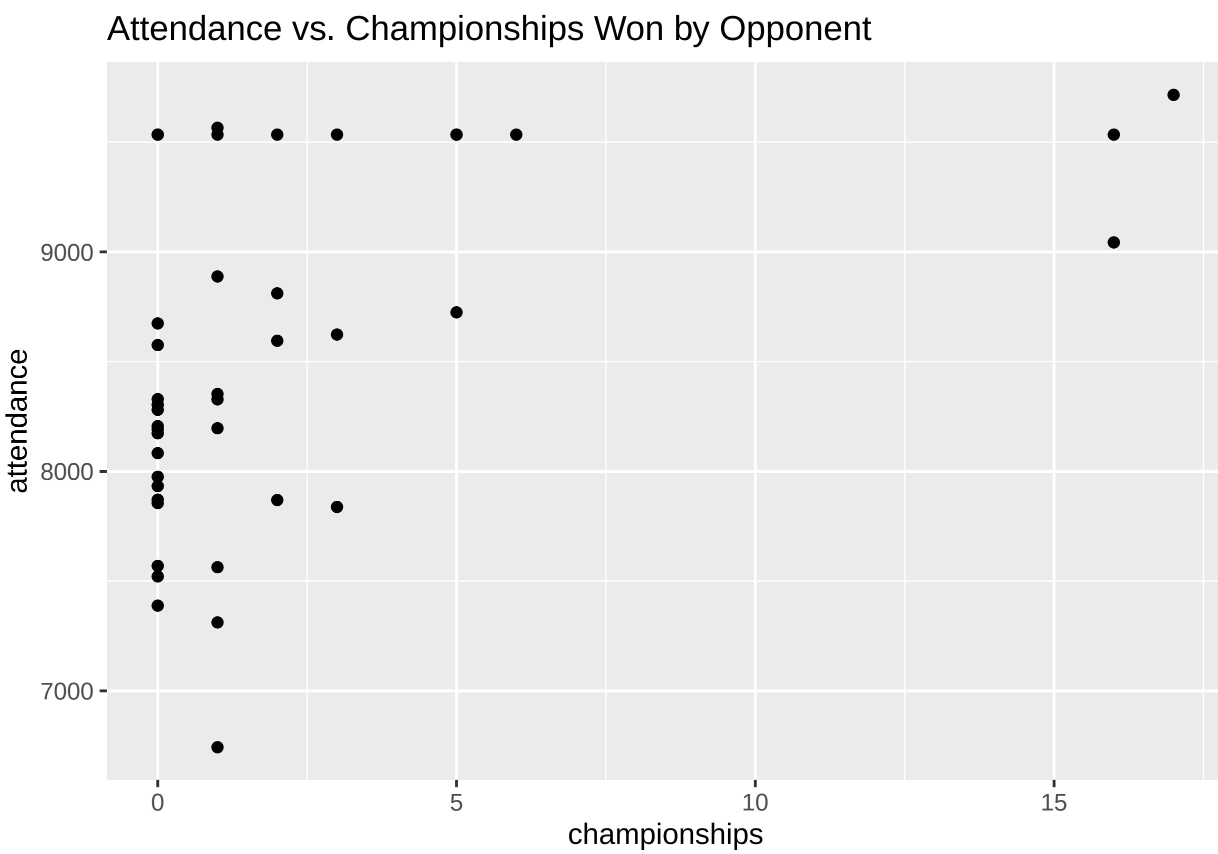

```
# There doesn't seem to be much difference - do note that there is an increase in attendance during the

# Here, we'll look at the distribution of games being close to holidays impacting attendance
ggplot(train_17, aes(x = holiday_close, y = attendance)) +
  geom_boxplot() +
  labs(title = "Impact of Holidays on Attendance")
```



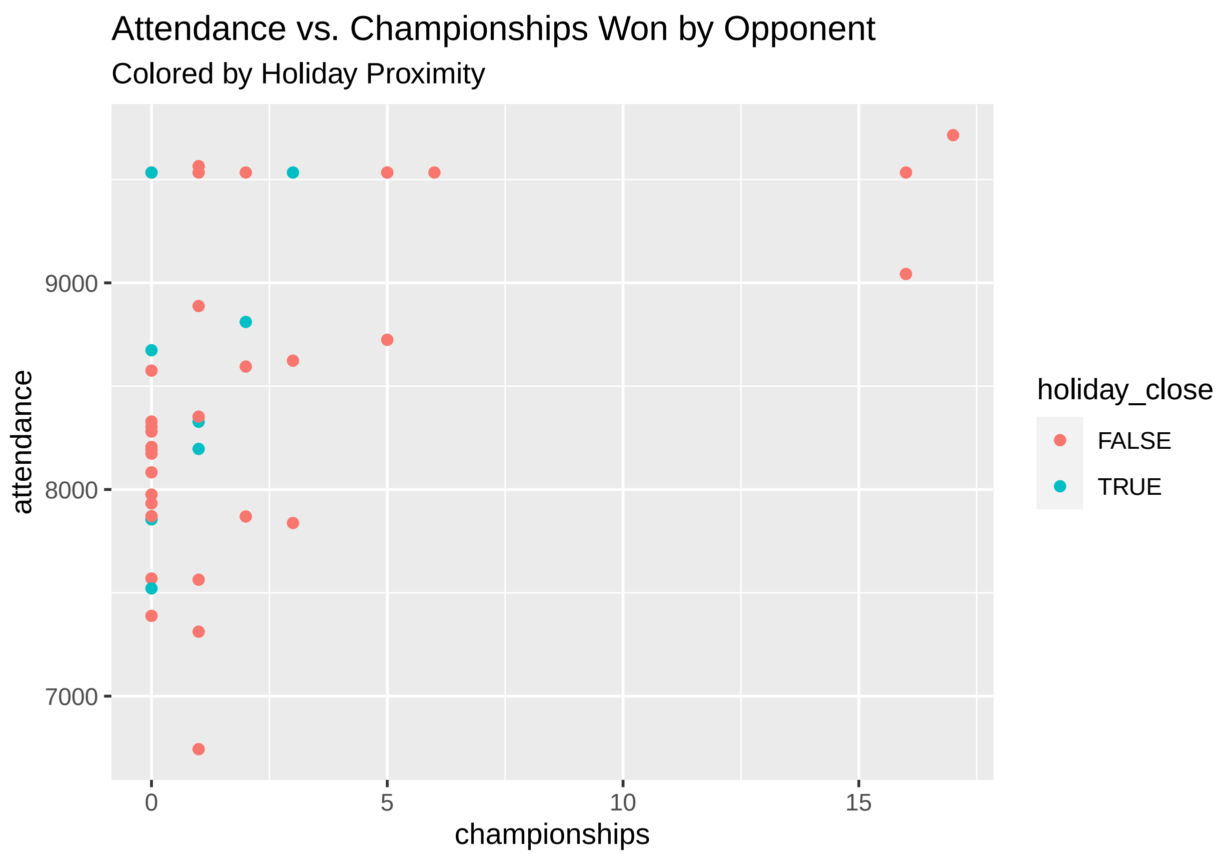

```r
# We see that while there does seem to be an impact, it does not seem like much

# Let's try plotting the number of days to a holiday on attendance level
ggplot(train_17, aes(x = days_2_holidays, y = attendance)) +
  geom_point(aes(color = holiday_close)) +
  labs(title = "Holiday Proximity's Impact on Attendance")
```



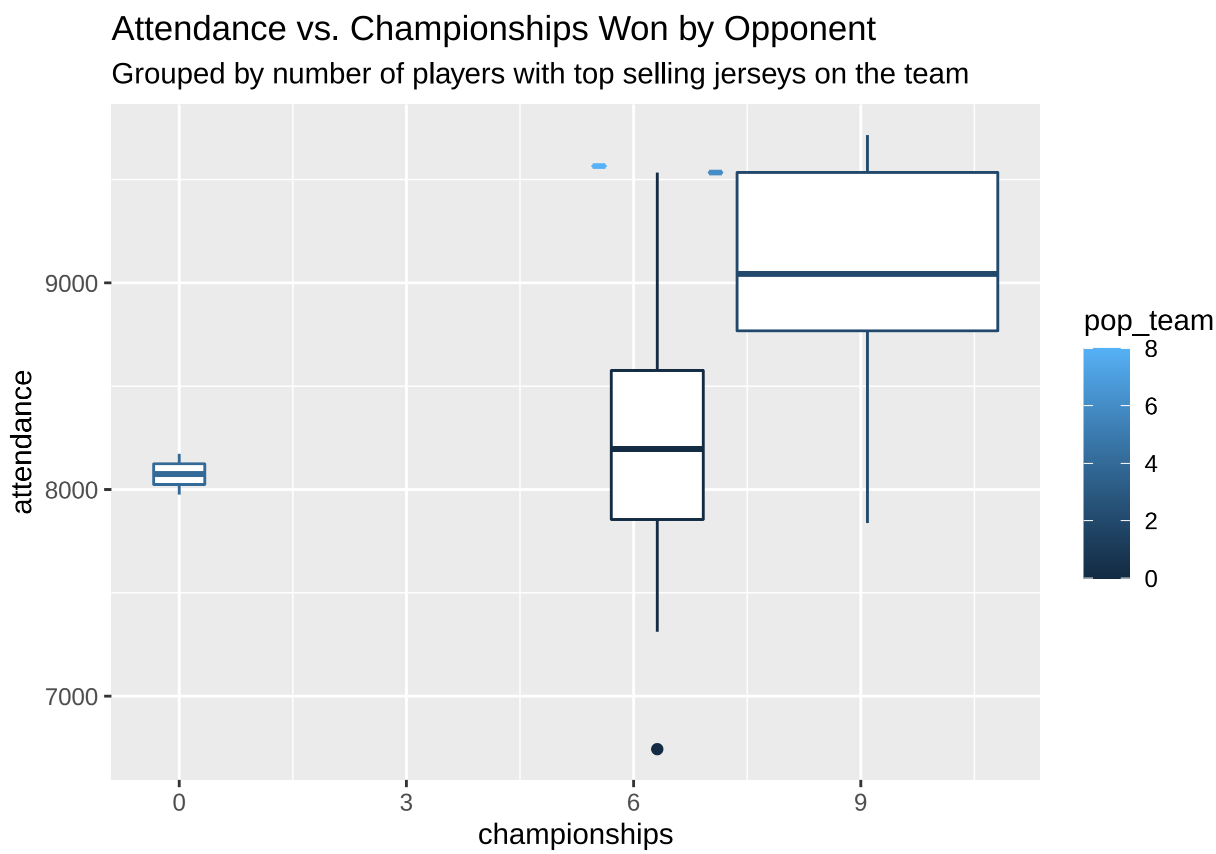

```r
# Let's graph the number of championships per team vs. attendance
ggplot(train_17, aes(x = championships, y = attendance)) +
  geom_point() +
  labs(title = "Attendance vs. Championships Won by Opponent")
```



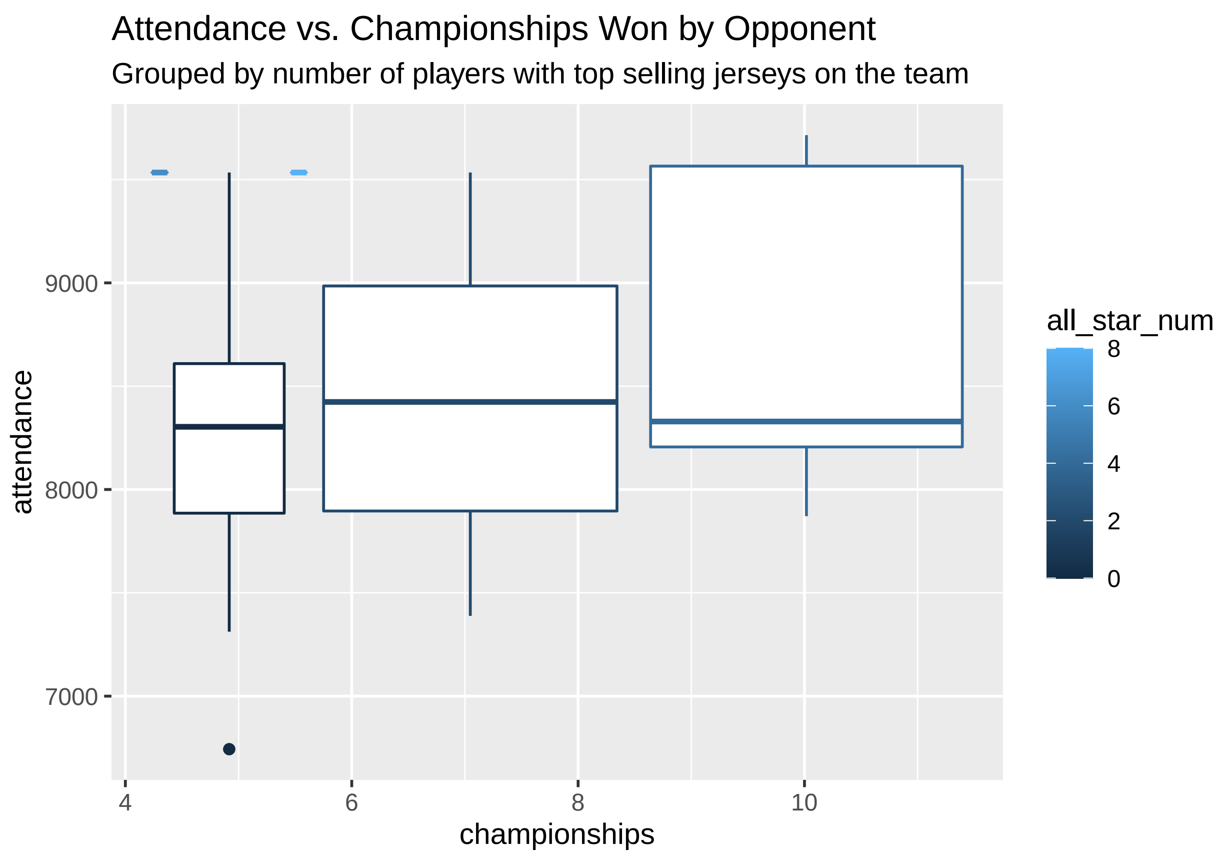

```r
# Notice that there does seem to be a correlation between number of championships and attendance
# Let us see if there is an interaction between these based off holidays
ggplot(train_17, aes(x = championships, y = attendance)) +
  geom_point(aes(color = holiday_close)) +
  labs(title = "Attendance vs. Championships Won by Opponent",
       subtitle = "Colored by Holiday Proximity")
```



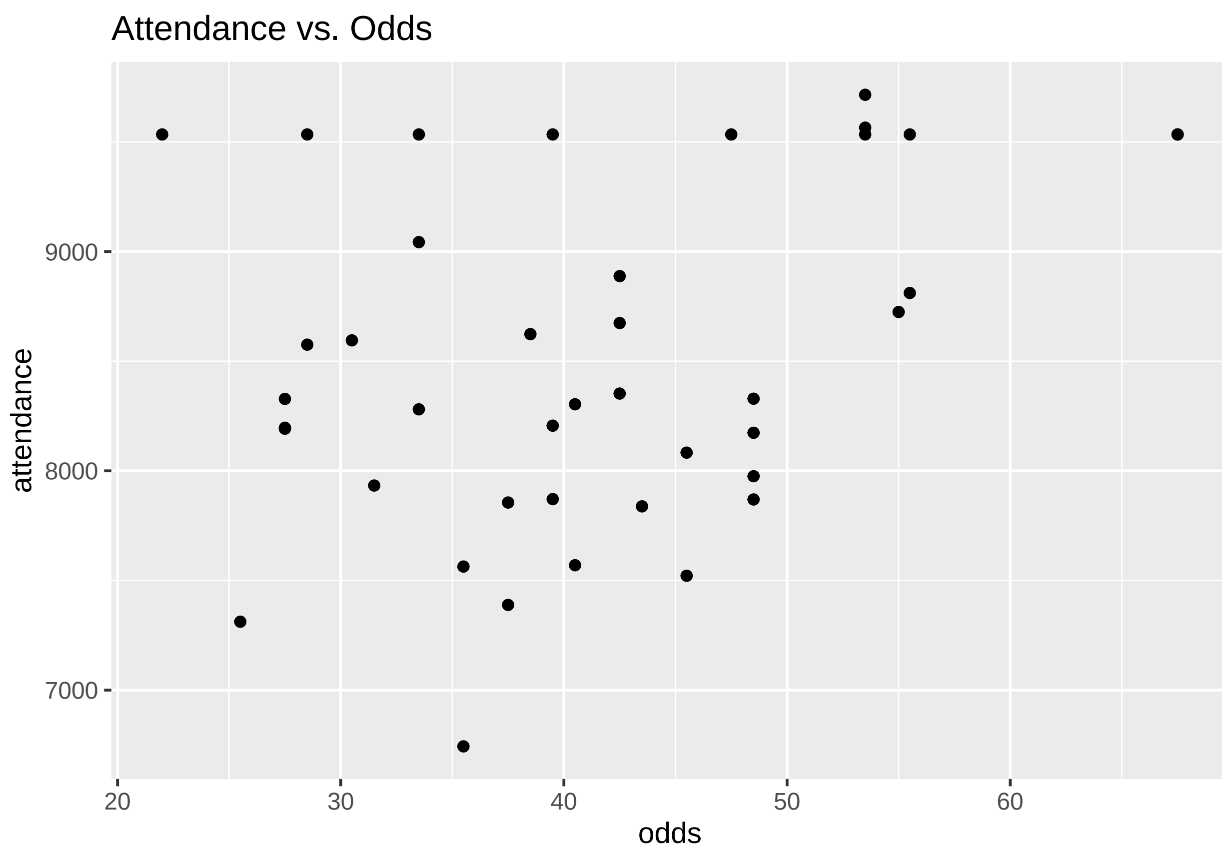

```
# We do see some interaction here, as games over the holidays tend to have a
# greater number of attendees

# Now, we check that with the popularity of a team
ggplot(train_17, aes(x = championships, y = attendance, group = pop_team)) +
  geom_boxplot(aes(color = pop_team)) +
  labs(title = "Attendance vs. Championships Won by Opponent",
       subtitle = "Grouped by number of players with top selling jerseys on the team")
```



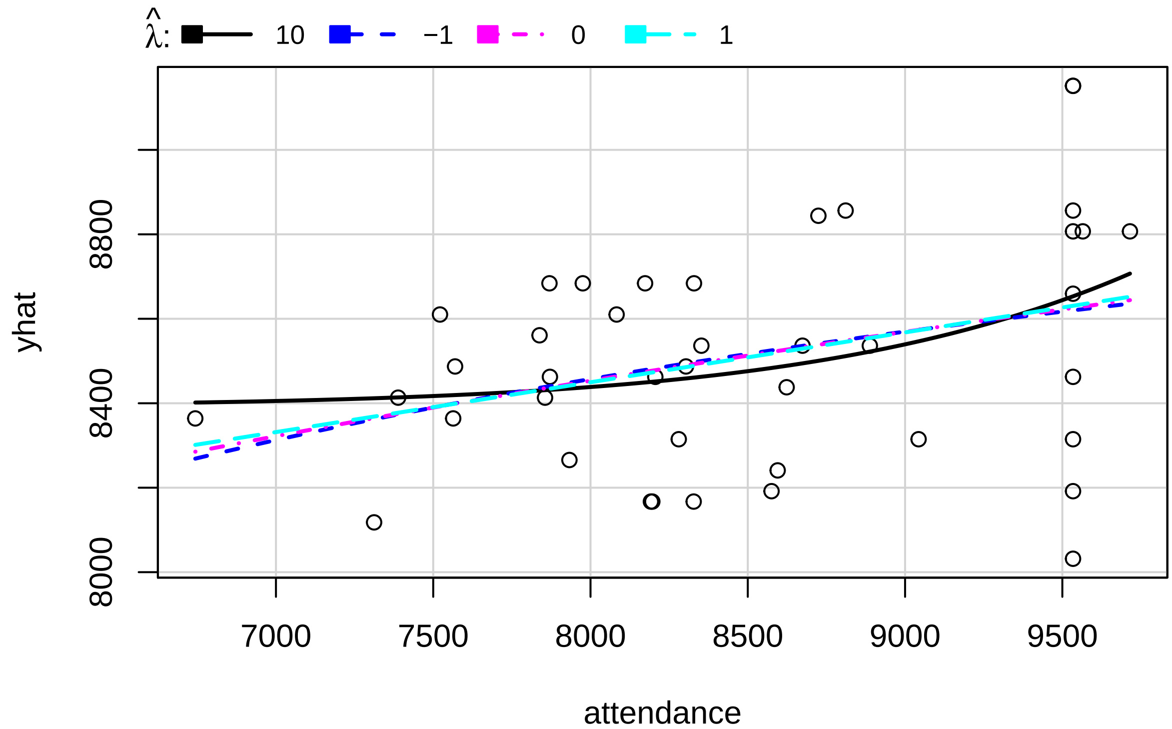

```r
# Now, see if a greater number of all stars from the prior year impacts attendance
ggplot(train_17, aes(x = championships, y = attendance, group = all_star_num)) +
  geom_boxplot(aes(color = all_star_num)) +
  labs(title = "Attendance vs. Championships Won by Opponent",
       subtitle = "Grouped by number of players with top selling jerseys on the team")
```




```r
# Let's look at the individual variables with attendance first

# How do the odds of the opposing team influence attendance?
ggplot(train_17, aes(x = odds, y = attendance)) +
  geom_point() +
  labs(title = "Attendance vs. Odds")
```




```r
# There seems to be a weak positive relationship between odds and attendance

# Let's create a new column of the transformed data
invResPlot(lm(attendance ~ odds, data = train_17))
```



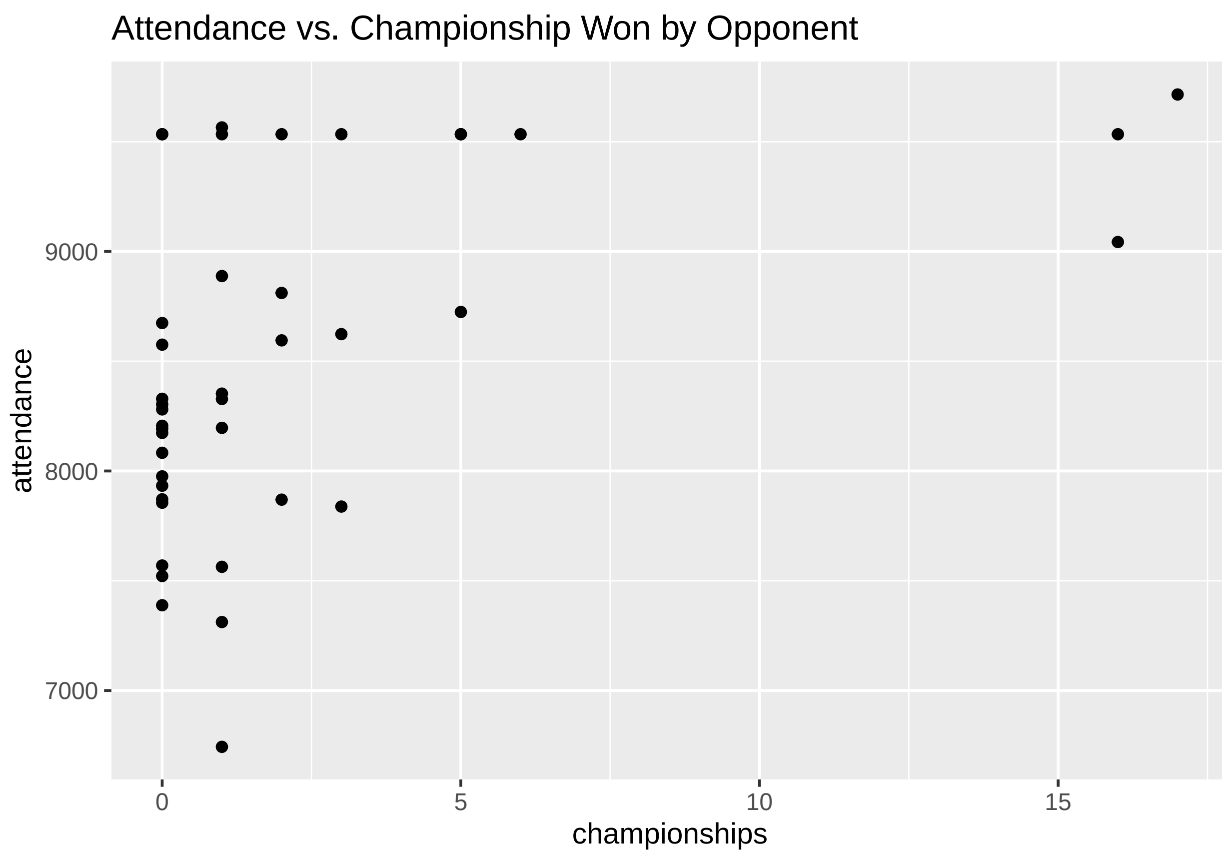

```
##       lambda     RSS
## 1  9.999926 2493171
## 2 -1.000000 2547774
## 3  0.000000 2536879
## 4  1.000000 2527555
```

```
# It recommends us to use a power of 10 here
train_17$odds_trans = train_17$odds^.1
summary(lm(attendance ~ odds_trans, data = train_17))
```

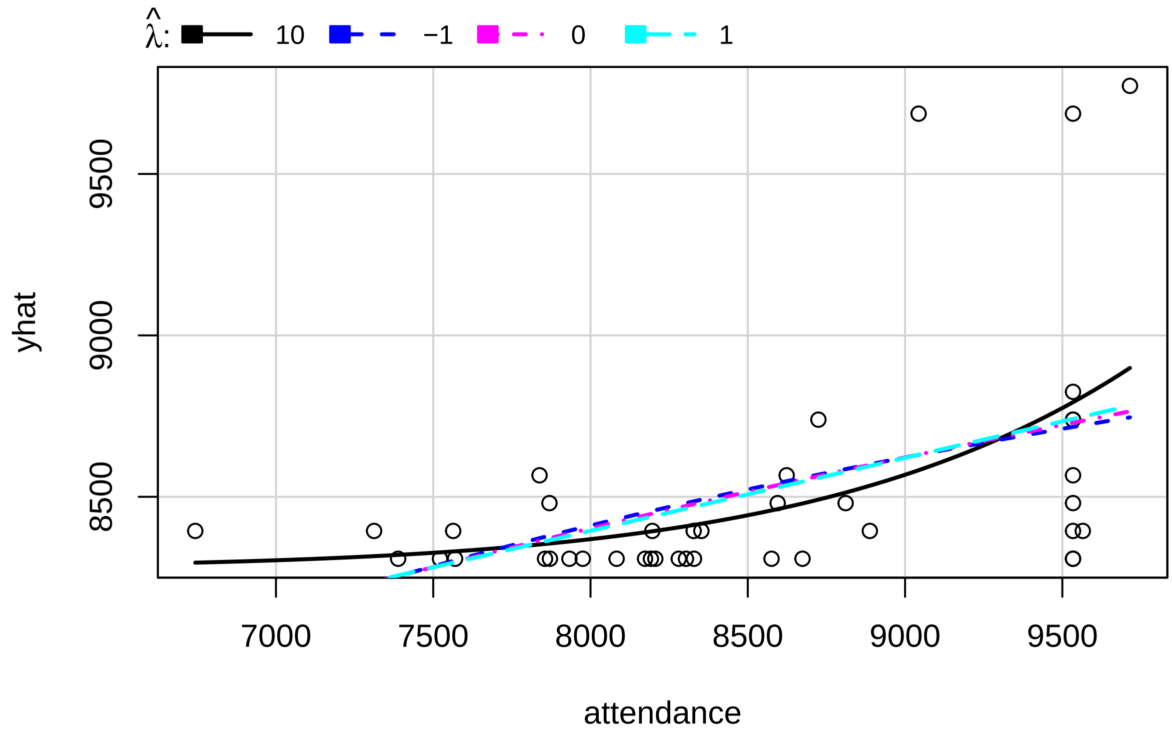
```
##
## Call:
## lm(formula = attendance ~ odds_trans, data = train_17)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1662.3  -596.2    -3.4   574.3  1518.1
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    59.67    4508.24   0.013   0.9895
## odds_trans   5840.67    3114.96   1.875   0.0683 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 755.7 on 39 degrees of freedom
## Multiple R-squared:  0.08269,    Adjusted R-squared:  0.05917
## F-statistic: 3.516 on 1 and 39 DF,  p-value: 0.06829
```

```
# Now, let's check the number of championships and its impact on attendance
ggplot(train_17, aes(x = championships, y = attendance)) +
  geom_point() +
  labs(title = "Attendance vs. Championship Won by Opponent")
```




```
# There definitely seems to be a positive, and possibly exponential, relationship here
# Let's tranform it
invResPlot(lm(attendance ~ championships, data = train_17))
```



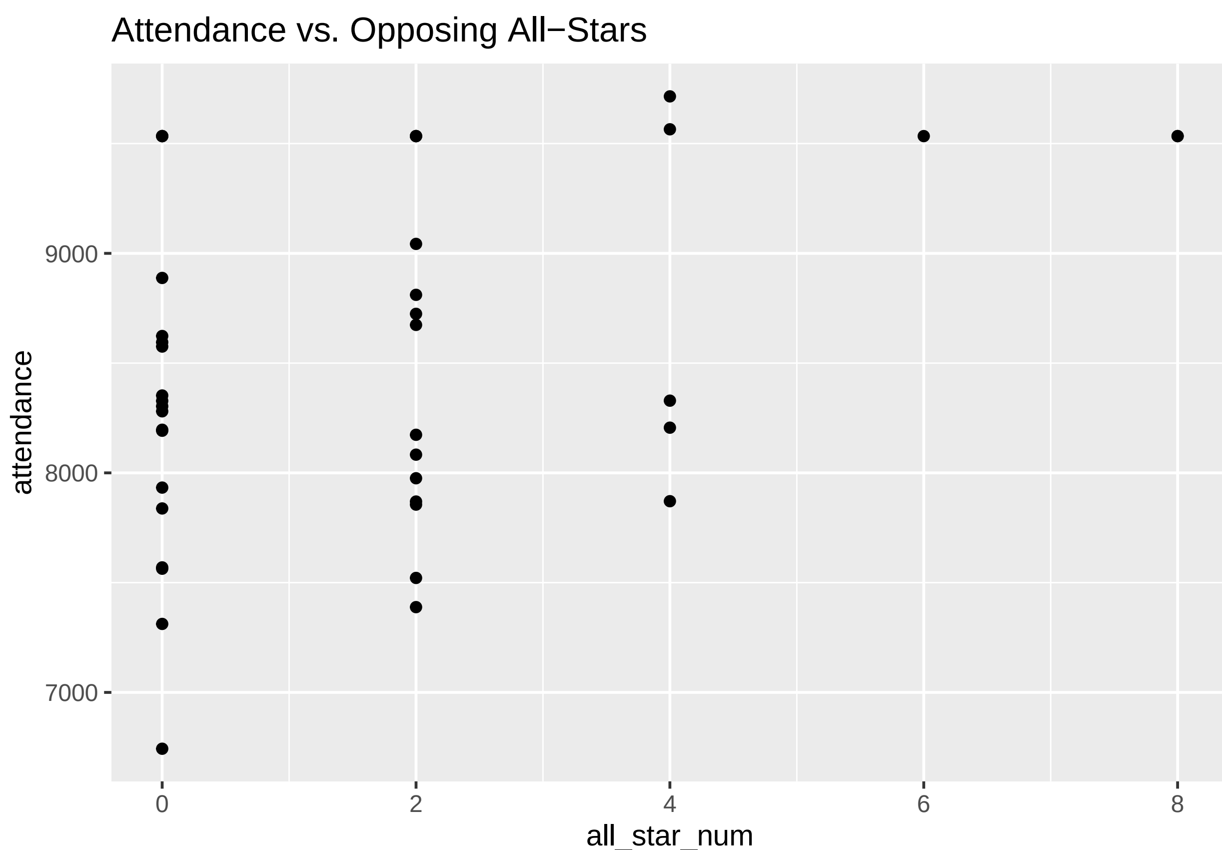

```
##       lambda     RSS
## 1  9.999926 4033807
## 2 -1.000000 4367438
## 3  0.000000 4303452
## 4  1.000000 4247298
```

```
# Again, we get a lambda value of 10, so we apply here
train_17$champ_trans = train_17$championships^10
summary(lm(attendance ~ I(championships^(1/10)), data = train_17))
```

```
##
## Call:
## lm(formula = attendance ~ I(championships^(1/10)), data = train_17)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1973.58  -364.58    16.71   643.82  1358.71
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               8175.3      170.3  48.005   <2e-16 ***
## I(championships^(1/10))    541.8      205.5   2.636    0.012 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 726.9 on 39 degrees of freedom
## Multiple R-squared:  0.1512, Adjusted R-squared:  0.1295
## F-statistic: 6.948 on 1 and 39 DF,  p-value: 0.01198
```

```r
# Now, let's see how the number of all-stars on the team influences attendance
ggplot(train_17, aes(x = all_star_num, y = attendance)) +
  geom_point() +
  labs(title = "Attendance vs. Opposing All-Stars")
```

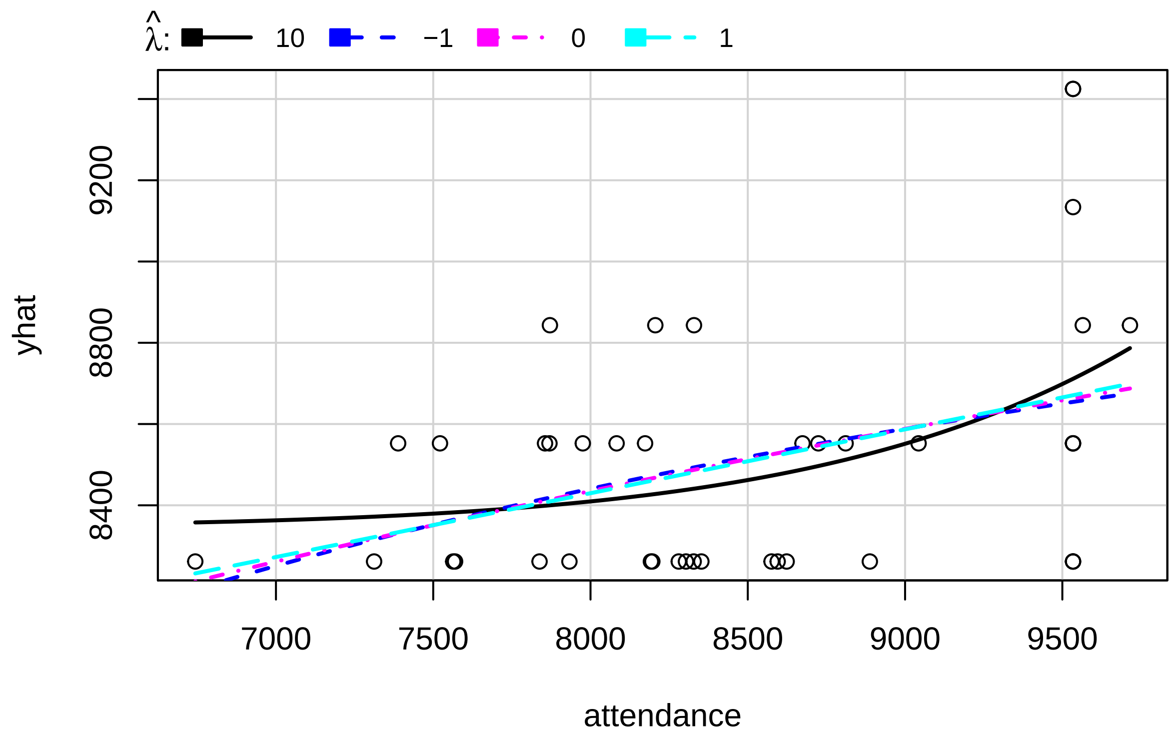



```r
# We see a similar trend like the last, as there definitely is a positive trend with the number of all
summary(lm(attendance ~ all_star_num, data = train_17))
```

```
##
## Call:
## lm(formula = attendance ~ all_star_num, data = train_17)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1518.17  -576.96    66.33   399.97  1272.33
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8261.67     145.85  56.646   <2e-16 ***
## all_star_num   145.39      53.92   2.697   0.0103 *
## ---
```

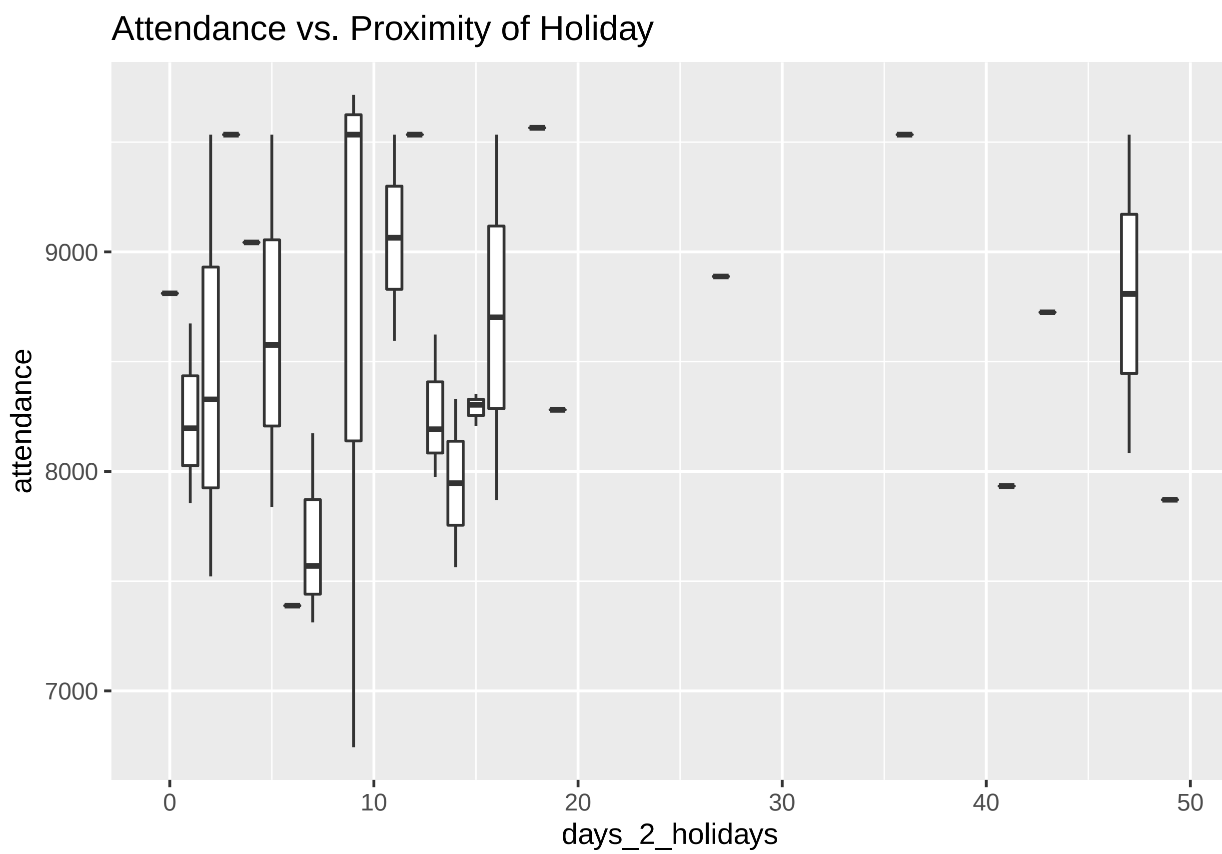
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 724.4 on 39 degrees of freedom
## Multiple R-squared:  0.1572, Adjusted R-squared:  0.1355
## F-statistic: 7.272 on 1 and 39 DF,  p-value: 0.01029
```

```r
# Let's see if transformations improve this
invResPlot(lm(attendance ~ all_star_num, data = train_17))
```

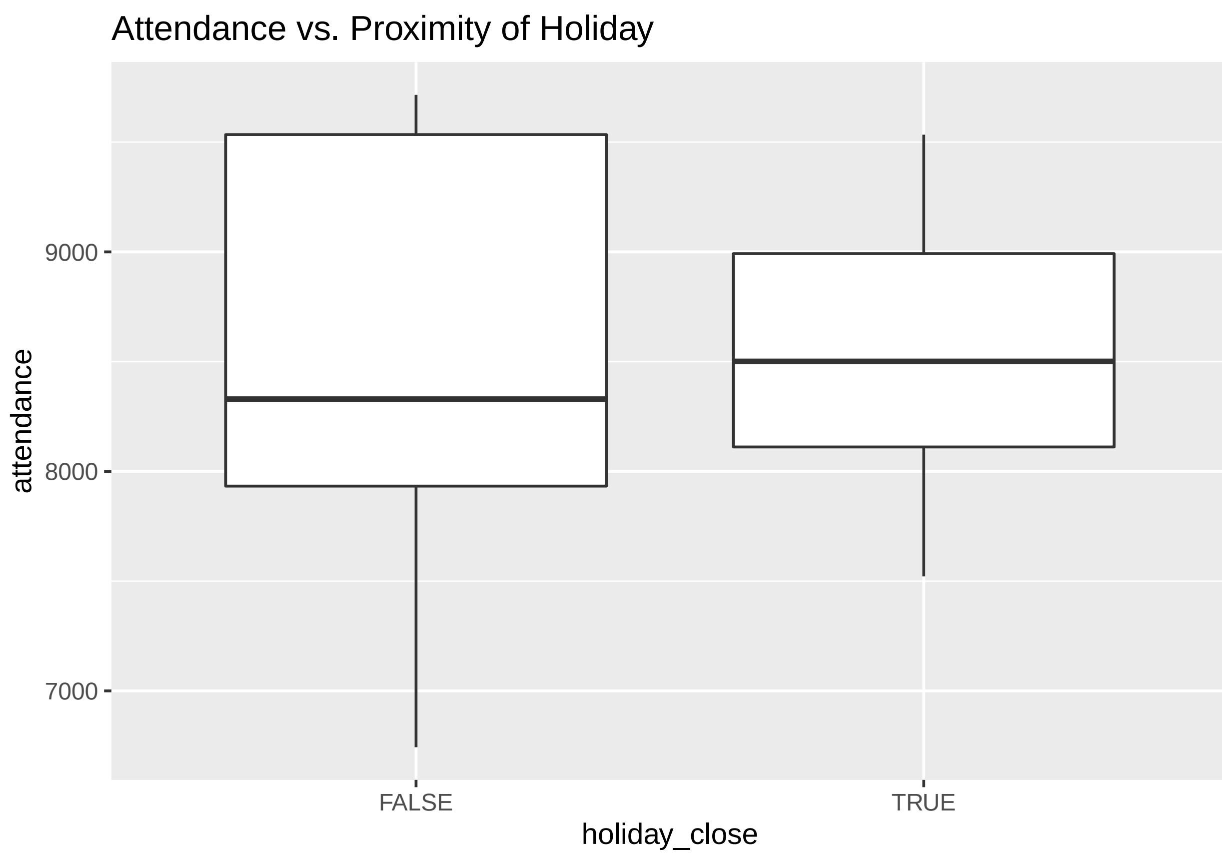



```
##      lambda     RSS
## 1  9.999926 3079523
## 2 -1.000000 3266041
## 3  0.000000 3239931
## 4  1.000000 3215810
```

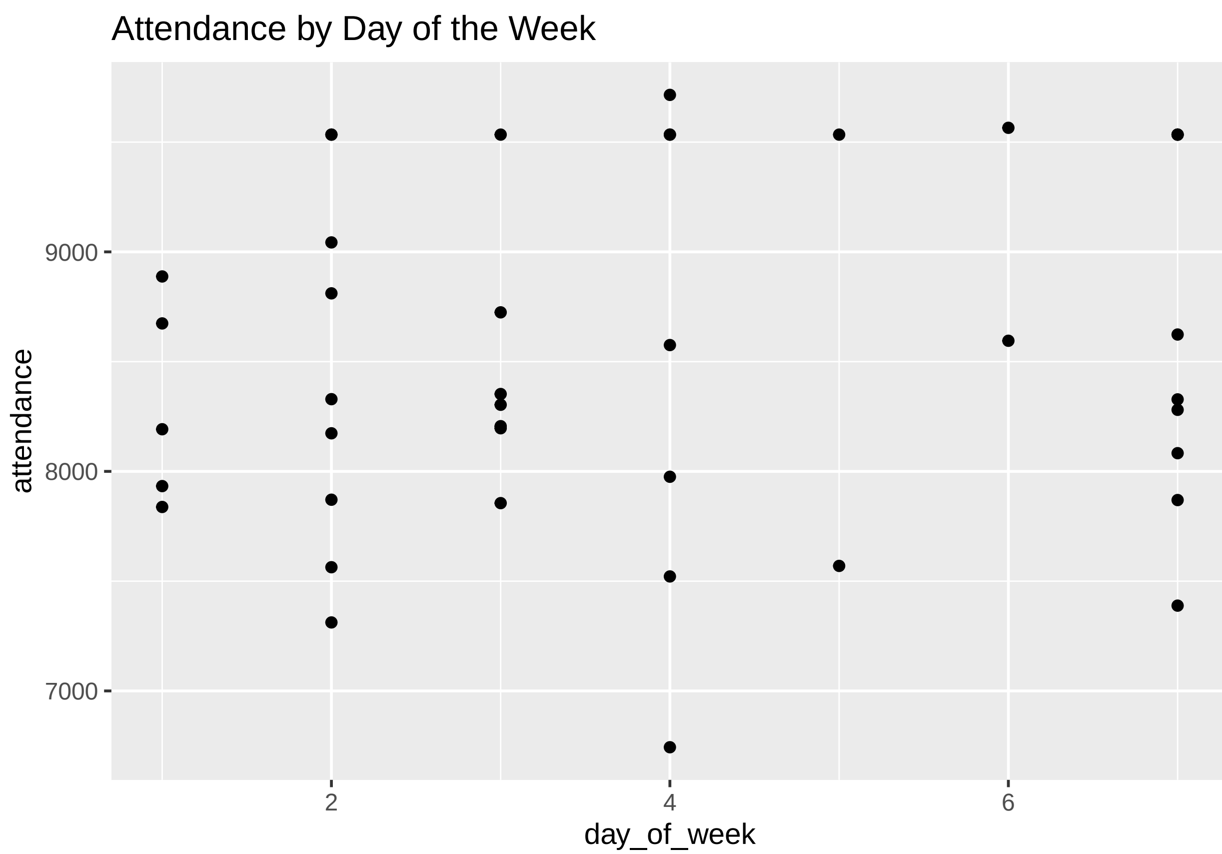
```r
# Let's plot the days away from a holiday with attendance
ggplot(train_17, aes(x = days_2_holidays, y = attendance, group = days_2_holidays)) +
  geom_boxplot() +
  labs(title = "Attendance vs. Proximity of Holiday")
```



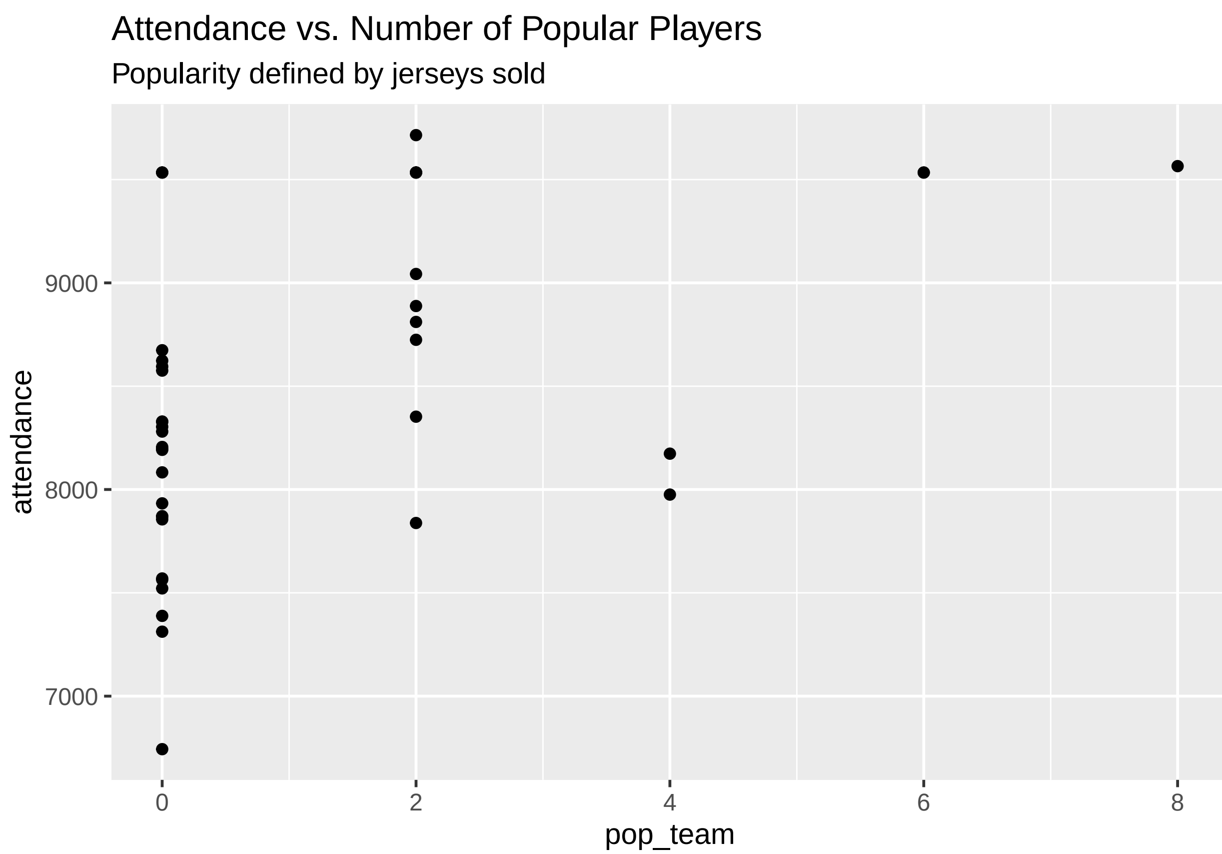

```r
# This doesn't seem to be a good predictor at all

# The distributions are similar as well. Thus, it does not seem like the holidays influence
ggplot(train_17, aes(x = holiday_close, y = attendance)) +
  geom_boxplot() +
  labs(title = "Attendance vs. Proximity of Holiday")
```



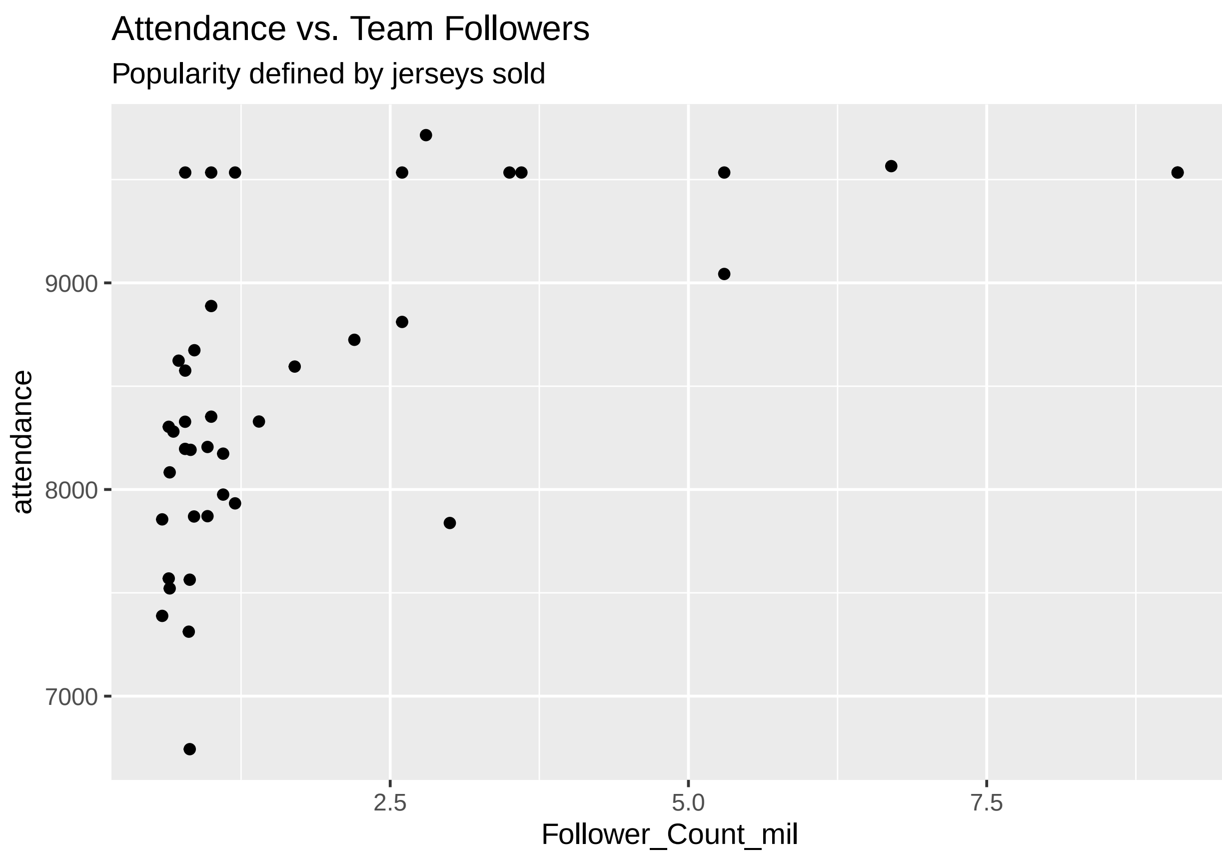

```r
# Check weekday
ggplot(train_17, aes(x = day_of_week, y = attendance)) +
  geom_point() +
  labs(title = "Attendance by Day of the Week")
```



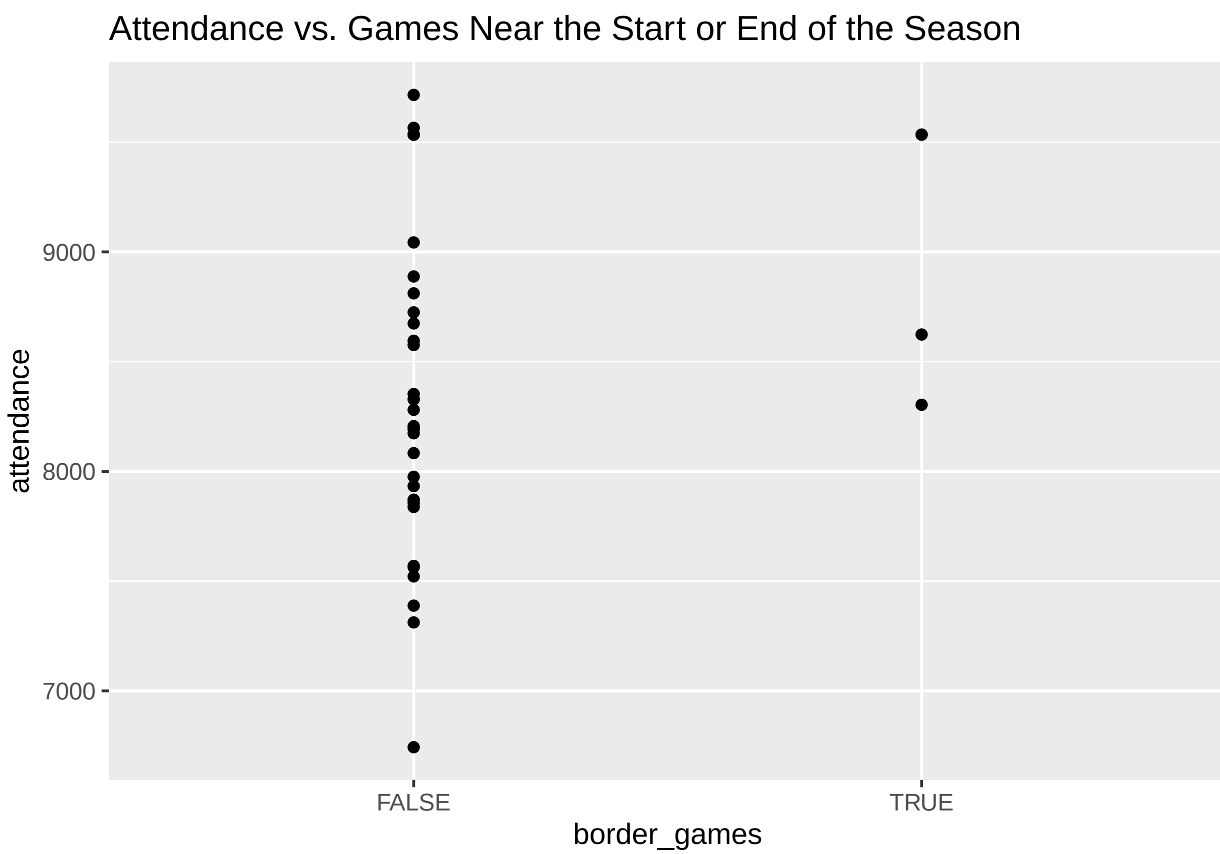

```r
# Let's see how popular teams affect attendance
ggplot(train_17, aes(x = pop_team, y = attendance)) +
  geom_point() +
  labs(title = "Attendance vs. Number of Popular Players",
       subtitle = "Popularity defined by jerseys sold")
```



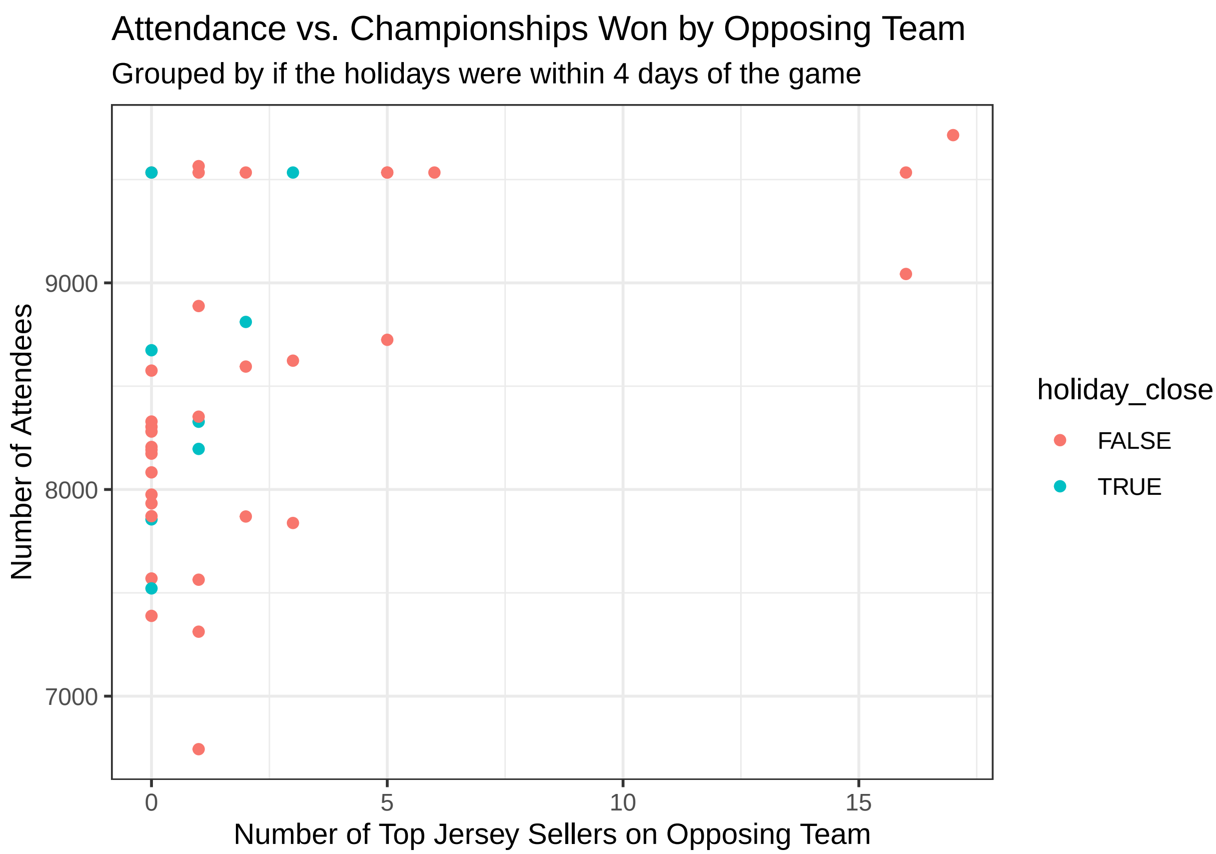

```r
# Similar distribution as before, where the more popular teams garner more fans

# Let's see how follower count for each team affects attendance
ggplot(train_17, aes(x = Follower_Count_mil, y = attendance)) +
  geom_point() +
  labs(title = "Attendance vs. Team Followers",
       subtitle = "Popularity defined by jerseys sold")
```




```r
ggplot(train_17, aes(x = border_games, y = attendance)) +
  geom_point() +
  labs(title = "Attendance vs. Games Near the Start or End of the Season")
```




```
ggplot(train_17, aes(x = championships, y = attendance)) +
  geom_point(aes(color = holiday_close, group = holiday_close)) +
  labs(title = "Attendance vs. Championships Won by Opposing Team",
       subtitle = "Grouped by if the holidays were within 4 days of the game",
       x = "Number of Top Jersey Sellers on Opposing Team",
       y = "Number of Attendees") +
  theme_bw()
```



Models!

After many hours of data scraping, it is now officially time to start creating my own models. I'm hoping to achieve an R^2 value of 0.8, and potentially more.

I plan on making a linear regression, and if time permits, possibly a random forest.

```r
# First up, a basic MLR using most of the variables
mod1 = lm(attendance ~ all_star_num + pop_team + odds + weekday_status + holiday_close + championships +

# Check the summary statistics
summary(mod1)
```

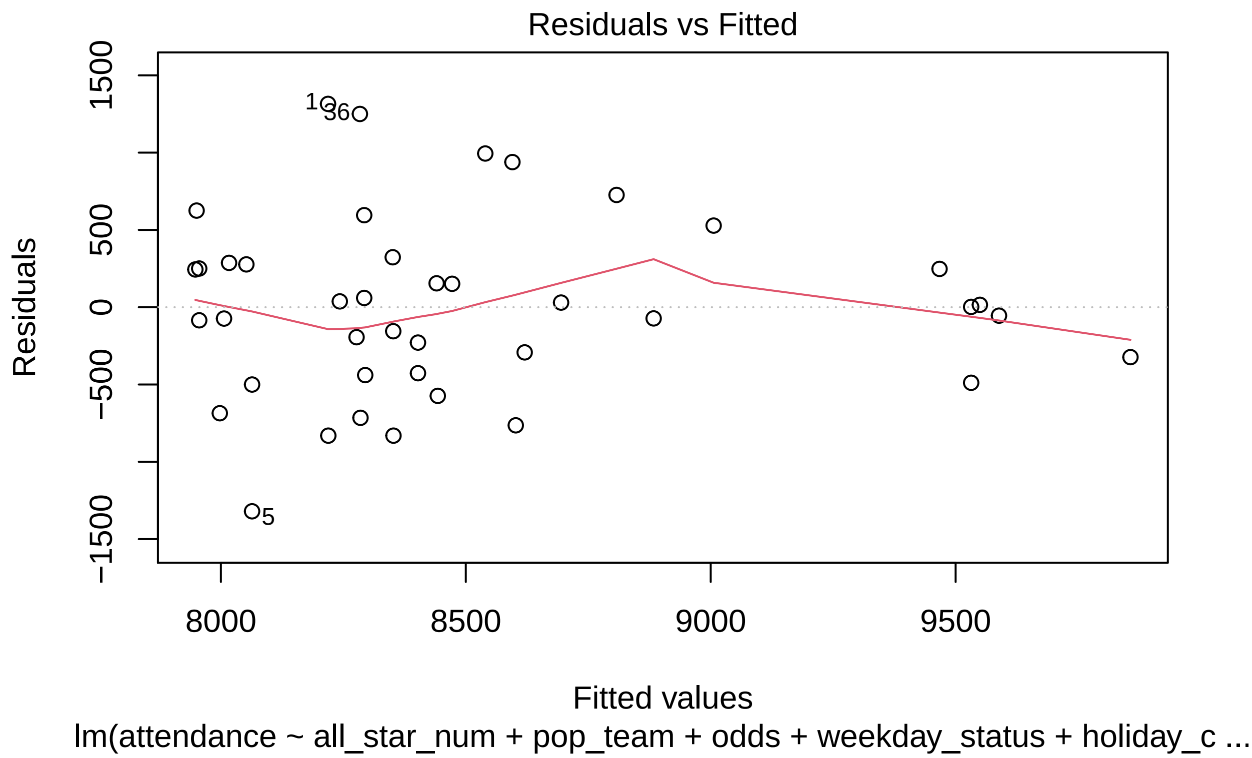
```
##
## Call:
## lm(formula = attendance ~ all_star_num + pop_team + odds + weekday_status +
##     holiday_close + championships + Follower_Count_mil, data = train_17)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1320.16  -426.86     2.24   277.00  1315.23
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          7964.78     640.46  12.436 5.29e-14 ***
```

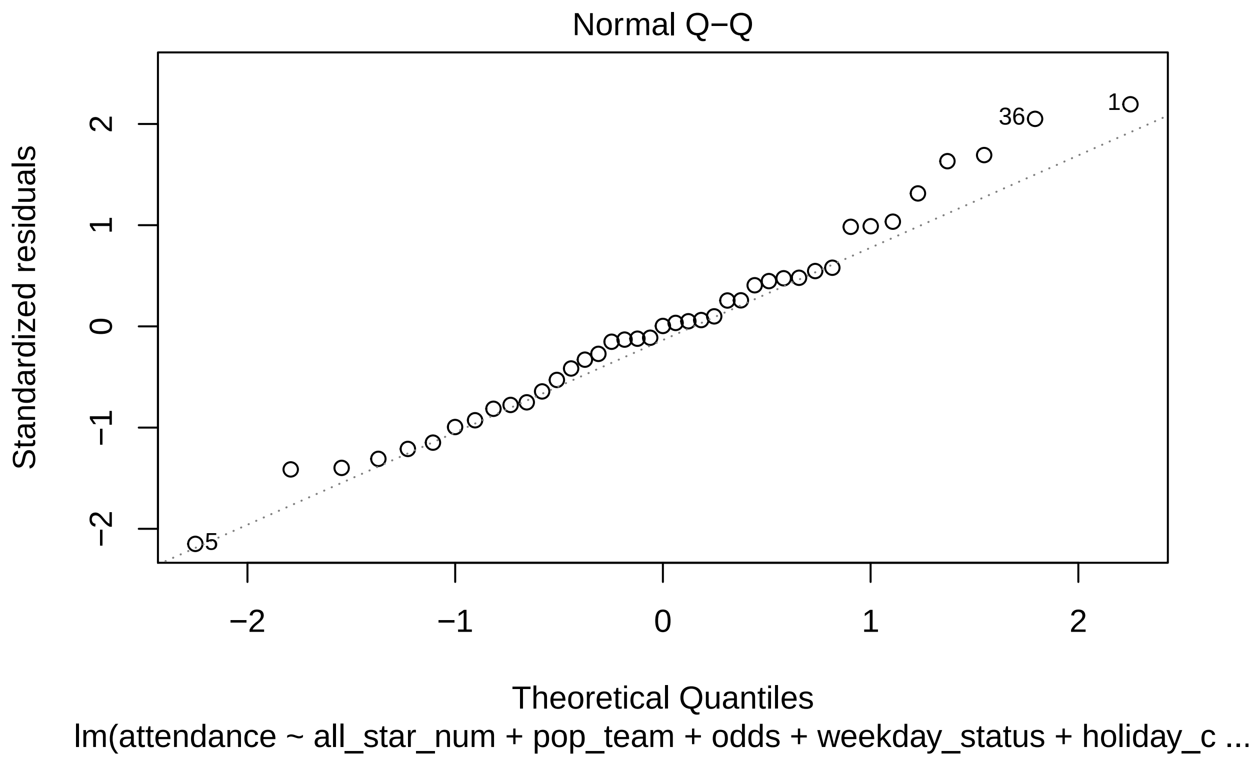
```
## all_star_num        -20.67      92.39  -0.224   0.8243
## pop_team             83.79     108.64   0.771   0.4460
## odds                  6.52      17.43   0.374   0.7107
## weekday_statusTRUE  -268.35     228.69  -1.173   0.2490
## holiday_closeTRUE    343.74     264.10   1.302   0.2021
## championships        64.29      31.26   2.057   0.0477 *
## Follower_Count_mil    87.17     111.75   0.780   0.4409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 637.7 on 33 degrees of freedom
## Multiple R-squared:  0.4472, Adjusted R-squared:   0.33
## F-statistic: 3.814 on 7 and 33 DF,  p-value: 0.003861
```

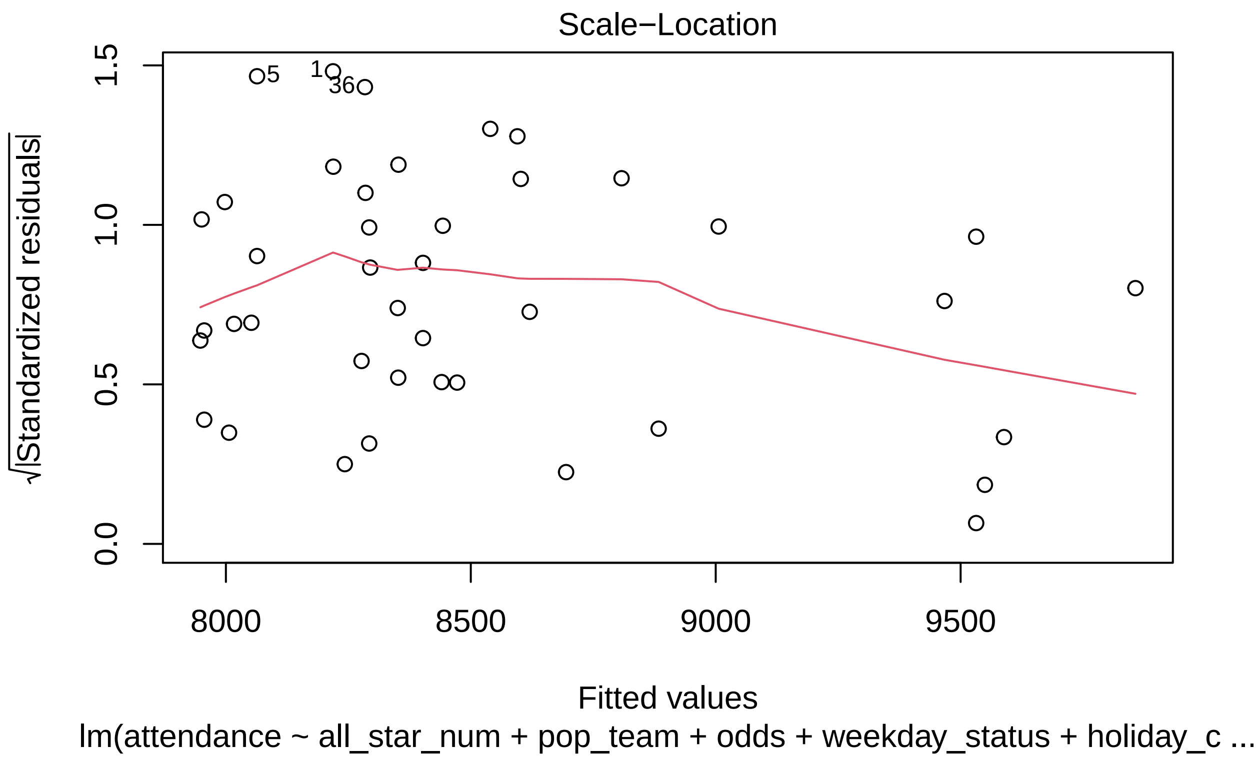
```r
# We get a respectable .3377 adjusted R^2 value
anova(mod1)
```

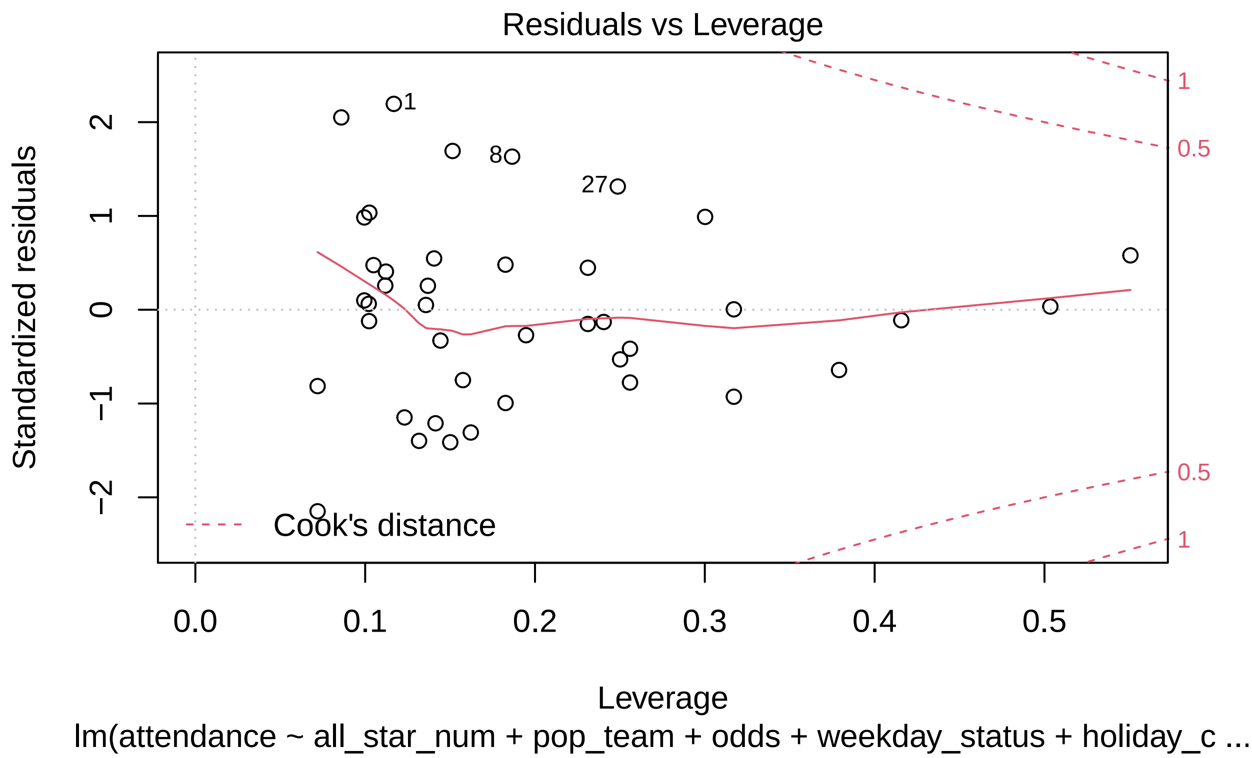
```
## Analysis of Variance Table
##
## Response: attendance
##                    Df   Sum Sq Mean Sq F value   Pr(>F)
## all_star_num        1  3815402 3815402  9.3815 0.004341 **
## pop_team            1  2152401 2152401  5.2924 0.027876 *
## odds                1   133416  133416  0.3281 0.570692
## weekday_status      1   268774  268774  0.6609 0.422077
## holiday_close       1   325753  325753  0.8010 0.377283
## championships       1  3914574 3914574  9.6254 0.003917 **
## Follower_Count_mil  1   247443  247443  0.6084 0.440940
## Residuals          33 13420893  406694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Let's check to see if this is a valid model
plot(mod1)
```


Residuals vs Fitted
Residuals
1500
500
0
−500
−1500
8000
8500
9000
9500
Fitted values
lm(attendance ~ all_star_num + pop_team + odds + weekday_status + holiday_c ...
16
36
5


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(attendance ~ all_star_num + pop_team + odds + weekday_status + holiday_c ...
36
1
5
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
8000
8500
9000
9500
Fitted values
lm(attendance ~ all_star_num + pop_team + odds + weekday_status + holiday_c ...
5
1
36



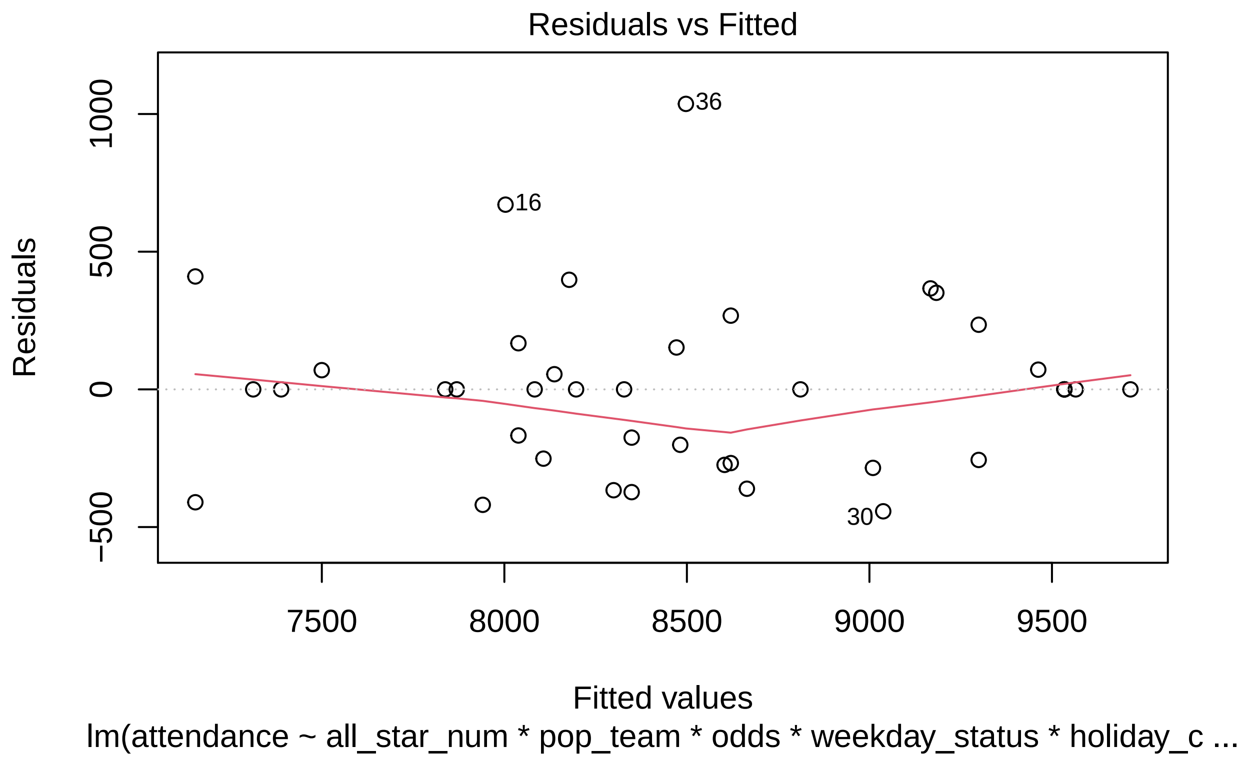

```r
# Note how we do see a mostly normal distribution for the errors, but we cannot assume constant varianc
# There also does not seem to be any bad leverage points, which is a positive

# The model below shows every interaction possible
mod2 = lm(attendance ~ all_star_num * pop_team * odds * weekday_status*holiday_close * championships, da

# Check the statistics
# summary(mod2)
# I'll save you the trouble of reading this
# But I pulled out all the important interactions, and will include them in my model below
anova(mod2)
```

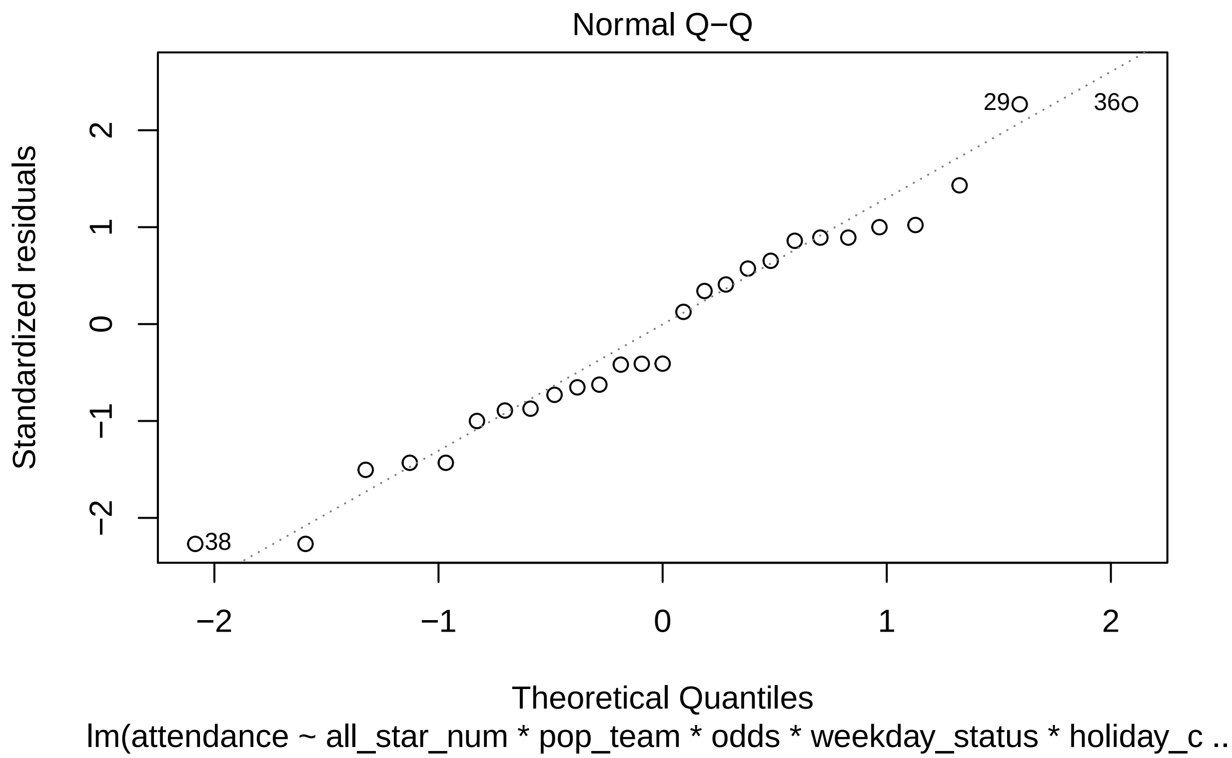
```
## Analysis of Variance Table
##
## Response: attendance
##                                     Df  Sum Sq Mean Sq F value   Pr(>F)
## all_star_num                         1 3815402 3815402 11.3439 0.006274 **
## pop_team                             1 2152401 2152401  6.3995 0.027987 *
## odds                                 1  133416  133416  0.3967 0.541674
## weekday_status                       1  268774  268774  0.7991 0.390497
## holiday_close                        1  325753  325753  0.9685 0.346200
## championships                        1 3914574 3914574 11.6388 0.005809 **
## all_star_num:pop_team                1    3327    3327  0.0099 0.922559
## all_star_num:odds                    1  744798  744798  2.2144 0.164823
## pop_team:odds                        1   57705   57705  0.1716 0.686684
## all_star_num:weekday_status          1  117263  117263  0.3486 0.566813
```

```
## pop_team:weekday_status               1  875059  875059  2.6017 0.135044
## odds:weekday_status                   1 1282769 1282769  3.8139 0.076734 .
## all_star_num:holiday_close            1     885     885  0.0026 0.960009
## pop_team:holiday_close                1  109821  109821  0.3265 0.579205
## odds:holiday_close                    1   17236   17236  0.0512 0.825060
## weekday_status:holiday_close          1  217718  217718  0.6473 0.438117
## all_star_num:championships            1   78759   78759  0.2342 0.637942
## pop_team:championships                1  128676  128676  0.3826 0.548820
## odds:championships                    1  124862  124862  0.3712 0.554706
## weekday_status:championships          1   26961   26961  0.0802 0.782338
## holiday_close:championships           1 2425554 2425554  7.2116 0.021201 *
## all_star_num:pop_team:odds            1 1015160 1015160  3.0183 0.110210
## all_star_num:pop_team:weekday_status  1  563729  563729  1.6761 0.221963
## all_star_num:odds:weekday_status      1  515564  515564  1.5329 0.241455
## pop_team:odds:weekday_status          1  365974  365974  1.0881 0.319273
## all_star_num:pop_team:championships   1   98549   98549  0.2930 0.599092
## all_star_num:odds:championships       1  939842  939842  2.7943 0.122771
## pop_team:odds:championships           1   17699   17699  0.0526 0.822770
## odds:weekday_status:championships     1  240700  240700  0.7156 0.415599
## Residuals                            11 3699726  336339
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

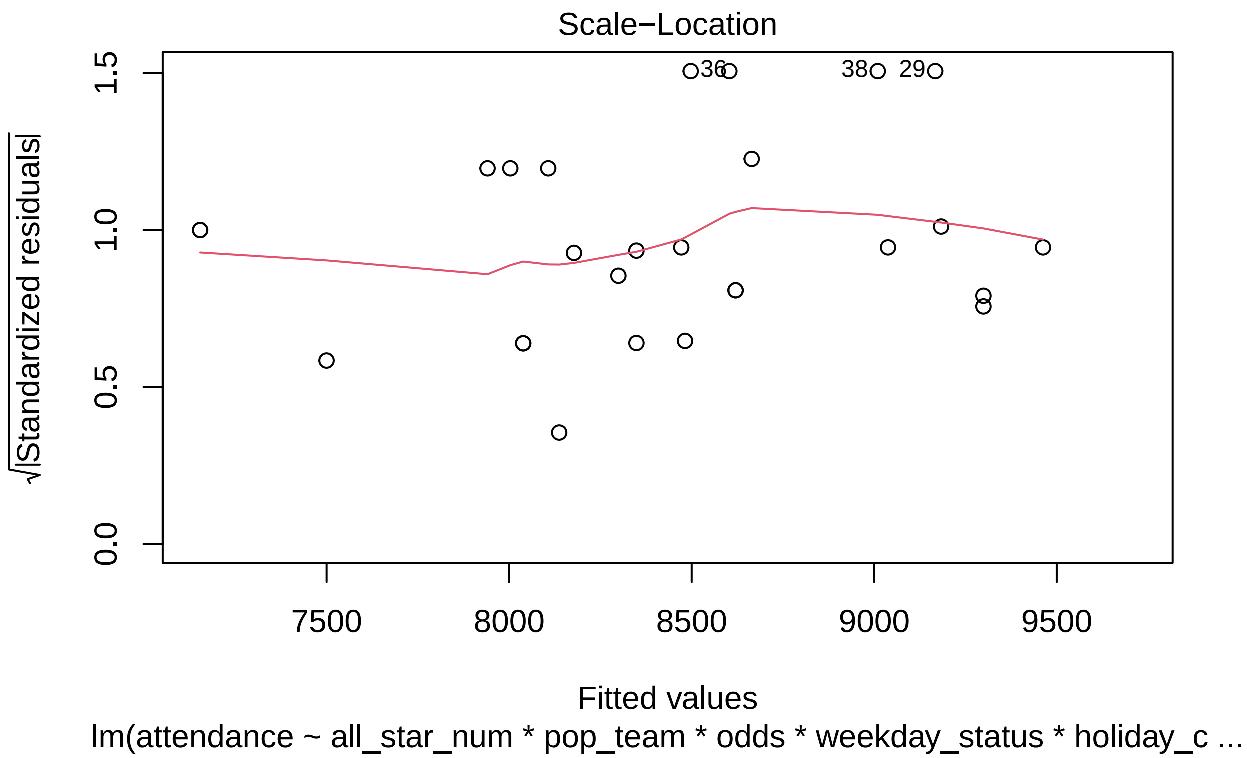
```r
# Let's check the diagnostic plots
plot(mod2)
```

```
## Warning: not plotting observations with leverage one:
##   4, 6, 7, 8, 12, 15, 18, 19, 20, 21, 22, 25, 33, 39
```

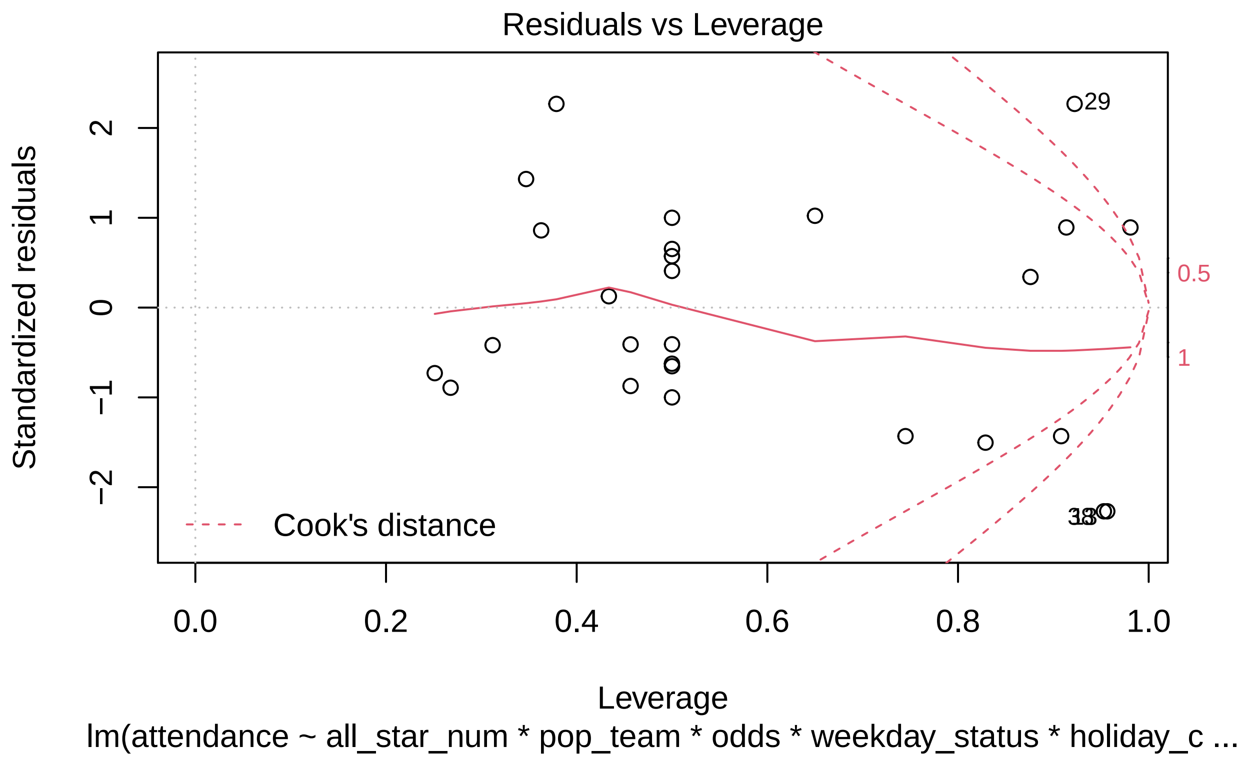

Residuals vs Fitted
Residuals
1000
500
0
−500
36
16
30
7500
8000
8500
9000
9500
Fitted values
lm(attendance ~ all_star_num * pop_team * odds * weekday_status * holiday_c ...



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(attendance ~ all_star_num * pop_team * odds * weekday_status * holiday_c ...
−2
−1
0
1
2
38
29
36




```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```




```r
# It looks ok. We can start assuming constant variance here

# Here is a new model with interactions
mod3 = lm(attendance ~ all_star_num + odds + weekday_status + championships + pop_team + odds:weekday_s

summary(mod3)
```

```
##
## Call:
## lm(formula = attendance ~ all_star_num + odds + weekday_status +
##     championships + pop_team + odds:weekday_status + weekday_status:holiday_close +
##     holiday_close:championships + Follower_Count_mil, data = train_17)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1217.26  -397.26    25.49   359.69  1167.86
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            8941.01     883.32  10.122 3.45e-11 ***
## all_star_num                             41.75     101.28   0.412   0.6831
## odds                                    -20.04      23.50  -0.853   0.4005
## weekday_statusTRUE                    -1425.71     827.59  -1.723   0.0952 .
## championships                            55.09      31.11   1.771   0.0868 .
## pop_team                                 98.67     108.54   0.909   0.3706
## Follower_Count_mil                       76.18     112.77   0.676   0.5045
```

```
## odds:weekday_statusTRUE                   29.28      19.46   1.505   0.1429
## weekday_statusFALSE:holiday_closeTRUE   240.61     526.79   0.457   0.6511
## weekday_statusTRUE:holiday_closeTRUE     13.67     383.17   0.036   0.9718
## championships:holiday_closeTRUE         292.86     231.06   1.267   0.2147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 626.4 on 30 degrees of freedom
## Multiple R-squared:  0.5151, Adjusted R-squared:  0.3535
## F-statistic: 3.187 on 10 and 30 DF,  p-value: 0.006717
```

```
anova(mod3)
```

```
## Analysis of Variance Table
##
## Response: attendance
##                              Df   Sum Sq Mean Sq F value   Pr(>F)
## all_star_num                  1  3815402 3815402  9.7233 0.003994 **
## odds                          1    93134   93134  0.2373 0.629671
## weekday_status                1   323529  323529  0.8245 0.371105
## championships                 1  4595285 4595285 11.7108 0.001815 **
## pop_team                      1  1042225 1042225  2.6561 0.113613
## Follower_Count_mil            1   299223  299223  0.7626 0.389467
## odds:weekday_status           1   979483  979483  2.4962 0.124612
## weekday_status:holiday_close  2   728106  364053  0.9278 0.406485
## championships:holiday_close   1   630386  630386  1.6065 0.214734
## Residuals                    30 11771884  392396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vif(mod3)
```

```
##                                   GVIF Df GVIF^(1/(2*Df))
## all_star_num                  4.718489  1        2.172208
## odds                          6.655932  1        2.579909
## weekday_status               15.496001  1        3.936496
## championships                 1.823658  1        1.350429
## pop_team                      4.533962  1        2.129310
## Follower_Count_mil            6.129824  1        2.475848
## odds:weekday_status          17.640649  1        4.200077
## weekday_status:holiday_close  2.462329  2        1.252670
## championships:holiday_close   1.878259  1        1.370496
```

```
# Use our transformed data
mod4 = lm(attendance ~ all_star_num + odds_trans + champ_trans + weekday_status + pop_team + Follower_C

summary(mod4)
```

```
##
## Call:
## lm(formula = attendance ~ all_star_num + odds_trans + champ_trans +
##     weekday_status + pop_team + Follower_Count_mil + holiday_close +
```

```
##     weekday_status, data = train_17)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1356.53  -364.66   -44.39   272.41  1323.98
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.904e+03  6.671e+03   0.885   0.3825
## all_star_num        -4.839e+01  9.081e+01  -0.533   0.5977
## odds_trans           1.604e+03  4.681e+03   0.343   0.7339
## champ_trans          5.342e-10  2.873e-10   1.860   0.0719 .
## weekday_statusTRUE -2.388e+02  2.292e+02  -1.042   0.3050
## pop_team             5.398e+01  1.062e+02   0.508   0.6146
## Follower_Count_mil  1.727e+02  9.862e+01   1.751   0.0893 .
## holiday_closeTRUE   3.321e+02  2.664e+02   1.246   0.2214
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 645.1 on 33 degrees of freedom
## Multiple R-squared:  0.4344, Adjusted R-squared:  0.3144
## F-statistic: 3.621 on 7 and 33 DF,  p-value: 0.005287
```

```
anova(mod4)
```

```
## Analysis of Variance Table
##
## Response: attendance
##                    Df   Sum Sq Mean Sq F value   Pr(>F)
## all_star_num        1  3815402 3815402  9.1693 0.004751 **
## odds_trans          1      138     138  0.0003 0.985583
## champ_trans         1  1956795 1956795  4.7026 0.037425 *
## weekday_status      1   718477  718477  1.7267 0.197899
## pop_team            1  2135561 2135561  5.1322 0.030173 *
## Follower_Count_mil  1  1274426 1274426  3.0627 0.089401 .
## holiday_close       1   646339  646339  1.5533 0.221424
## Residuals          33 13731518  416107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod5 = lm(attendance ~ all_star_num + championships + pop_team  + holiday_close:championships, data = t

summary(mod5)
```

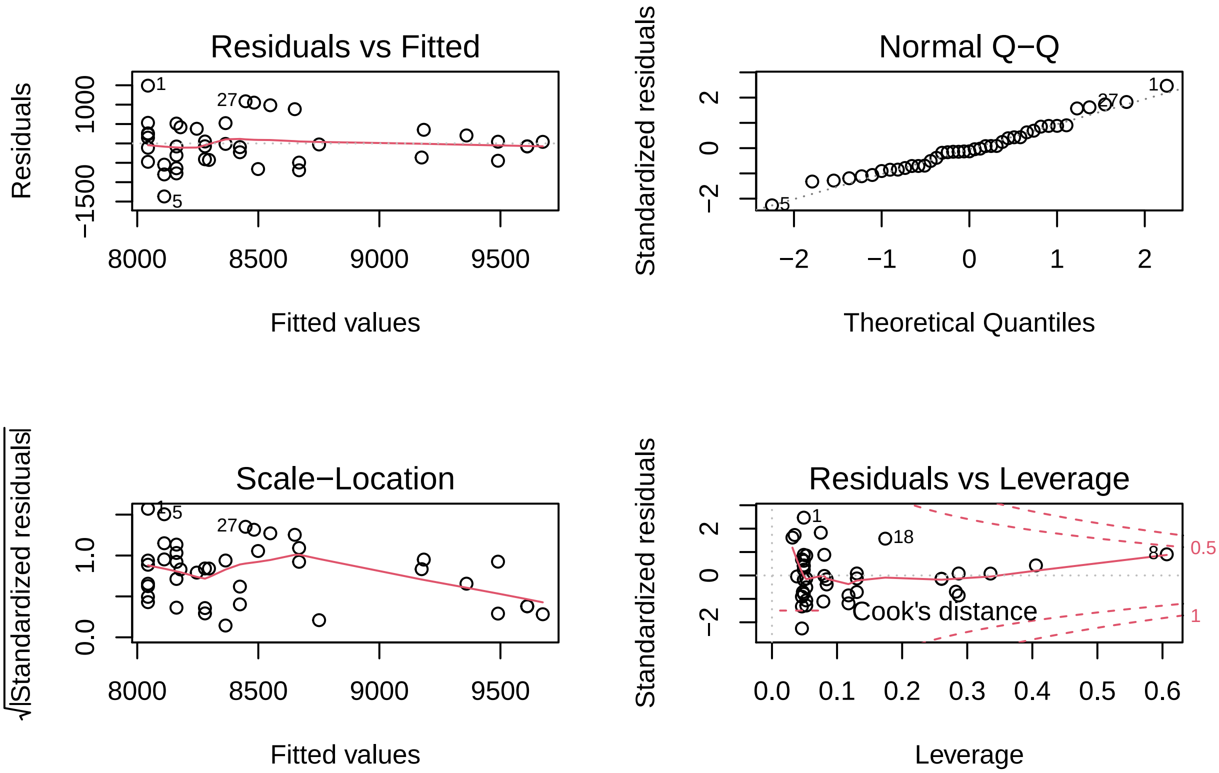
```
##
## Call:
## lm(formula = attendance ~ all_star_num + championships + pop_team +
##     holiday_close:championships, data = train_17)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1368.06  -426.92   -73.76   350.42  1489.61
##
```

```
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     8044.39     136.85  58.783  < 2e-16 ***
## all_star_num                      58.84      59.47   0.989  0.32907
## championships                     67.17      23.50   2.858  0.00705 **
## pop_team                         126.61      64.92   1.950  0.05895 .
## championships:holiday_closeTRUE   312.56     168.74   1.852  0.07219 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 618.8 on 36 degrees of freedom
## Multiple R-squared:  0.4322, Adjusted R-squared:  0.3691
## F-statistic: 6.851 on 4 and 36 DF,  p-value: 0.0003302
```

```r
anova(mod5)
```

```
## Analysis of Variance Table
##
## Response: attendance
##                             Df   Sum Sq Mean Sq F value   Pr(>F)
## all_star_num                 1  3815402 3815402  9.9640 0.003222 **
## championships                1  3916687 3916687 10.2285 0.002881 **
## pop_team                     1  1447639 1447639  3.7805 0.059698 .
## championships:holiday_close  1  1313876 1313876  3.4312 0.072188 .
## Residuals                   36 13785052  382918
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
par(mfrow = c(2, 2))
plot(mod5)
```



Predictions

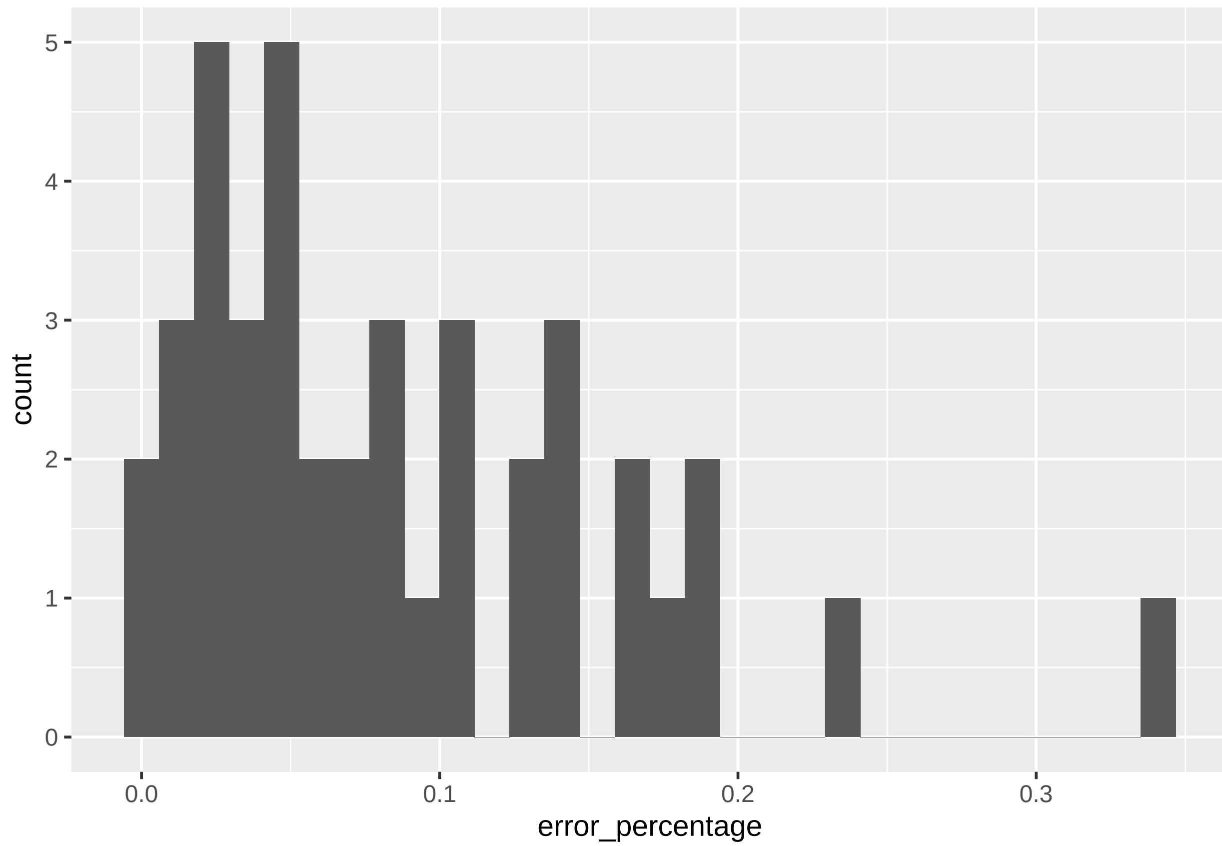
Using our model 5, which includes the number of all_stars, championships won, how many popular players with high selling jerseys are on the team, and the interaction between having holiday close to gameday and number of championships won, we will predict and check how well our model does.

```
test_18 = data[data$season == 2018,]
predictions = predict(mod5, test_18)
(prediction_differences = real_attendees$attendees - predictions)
```

```
##           1           2           3           4           5           6
##   128.61365 -2189.98786  -540.64518  -277.41681  -177.07292   363.96767
##           7           8           9          10          11          12
## -2149.97563 -1619.41471 -1025.38635  -396.38635    44.41124   722.92708
##          13          14          15          16          17          18
##  -992.24486   239.44170  -162.38635  1019.88260   -89.41471  -903.72279
##          19          20          21          22          23          24
##   246.61365 -1289.07292  -367.58876 -1006.07292 -1176.38635  -854.97563
##          25          26          27          28          29          30
##  1164.44170  -187.55830   543.94759   706.61365  1604.44170  1253.26976
##          31          32          33          34          35          36
##   904.01214   426.08907  -367.24486   -36.41802   564.61365 -1150.07292
##          37          38          39          40          41
##   -79.55830  -186.38635  -339.64518  -570.05241   950.61365
```

```r
pred_df = data.frame(predictions, attendees = real_attendees$attendees, prediction_differences)

pred_df = pred_df %>% mutate(error_percentage = abs(prediction_differences / attendees))

ggplot(pred_df, aes(x = error_percentage)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted)^2))
}

(rmse_model = rmse(pred_df$attendees, pred_df$predictions))
```

```
## [1] 893.53
```

```r
(rmse_null = rmse(pred_df$attendees, mean(pred_df$attendees)))
```

```
## [1] 974.0839
```

Trial and Error

```r
trial = test_18
trial$attendance = real_attendees$attendees
test_18
```

```
## # A tibble: 41 x 23
##    date                Opponent win_col lose_col result own_streak season
##    <dttm>              <chr>      <dbl>    <dbl> <chr>       <dbl>  <dbl>
##  1 2018-10-17 00:00:00 Denver ~       0        0 L               0   2018
##  2 2018-10-19 00:00:00 Oklahom~       0        1 W              -1   2018
##  3 2018-10-21 00:00:00 Houston~       1        1 W               1   2018
##  4 2018-10-28 00:00:00 Washing~       3        2 W               1   2018
##  5 2018-11-05 00:00:00 Minneso~       5        4 W               1   2018
##  6 2018-11-10 00:00:00 Milwauk~       6        5 W              -1   2018
##  7 2018-11-12 00:00:00 Golden ~       7        5 W               1   2018
##  8 2018-11-15 00:00:00 San Ant~       8        5 W               2   2018
##  9 2018-11-23 00:00:00 Memphis~      11        6 W              -1   2018
## 10 2018-11-28 00:00:00 Phoenix~      13        6 W               2   2018
## # ... with 31 more rows, and 16 more variables: attendance <dbl>, op_W <dbl>,
## #   op_L <dbl>, win_perc <dbl>, op_win_perc <dbl>, all_star_num <dbl>,
## #   pop_team <dbl>, odds <dbl>, days_2_holidays <dbl>, holiday_close <lgl>,
## #   championships <dbl>, Follower_Count_mil <dbl>, games_played <dbl>,
## #   border_games <lgl>, day_of_week <dbl>, weekday_status <lgl>
```

```r
real_attendees
```

```
## # A tibble: 41 x 4
##    game_number Opponent      event_datetime      attendees
##          <dbl> <chr>         <dttm>                  <dbl>
##  1           1 Denver        2018-10-18 02:30:00      8173
##  2           2 Oklahoma City 2018-10-20 02:30:00      6343
##  3           3 Houston       2018-10-22 01:00:00      8009
##  4           4 Washington    2018-10-29 01:30:00      8019
##  5           5 Minnesota     2018-11-06 03:30:00      7985
##  6           6 Milwaukee     2018-11-10 20:30:00      9159
##  7           7 Golden State  2018-11-13 03:30:00      9150
##  8           8 San Antonio   2018-11-16 03:30:00      8559
##  9           9 Memphis       2018-11-23 20:30:00      7019
## 10          10 Phoenix       2018-11-29 03:30:00      7648
## # ... with 31 more rows
```

```r
trial_mod = lm(attendance ~ all_star_num + championships + pop_team  + holiday_close:championships, data

summary(trial_mod)
```

```
##
## Call:
## lm(formula = attendance ~ all_star_num + championships + pop_team +
##     holiday_close:championships, data = trial)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1343.36  -532.51   -48.43   574.61  1750.64
```

```
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     8180.51     170.70  47.924  < 2e-16 ***
## all_star_num                    -189.16     112.64  -1.679  0.10176
## championships                    108.24      32.86   3.294  0.00222 **
## pop_team                         131.25     149.94   0.875  0.38720
## championships:holiday_closeTRUE   300.43     109.20   2.751  0.00924 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 793.7 on 36 degrees of freedom
## Multiple R-squared:  0.417,  Adjusted R-squared:  0.3522
## F-statistic: 6.437 on 4 and 36 DF,  p-value: 0.000515
```

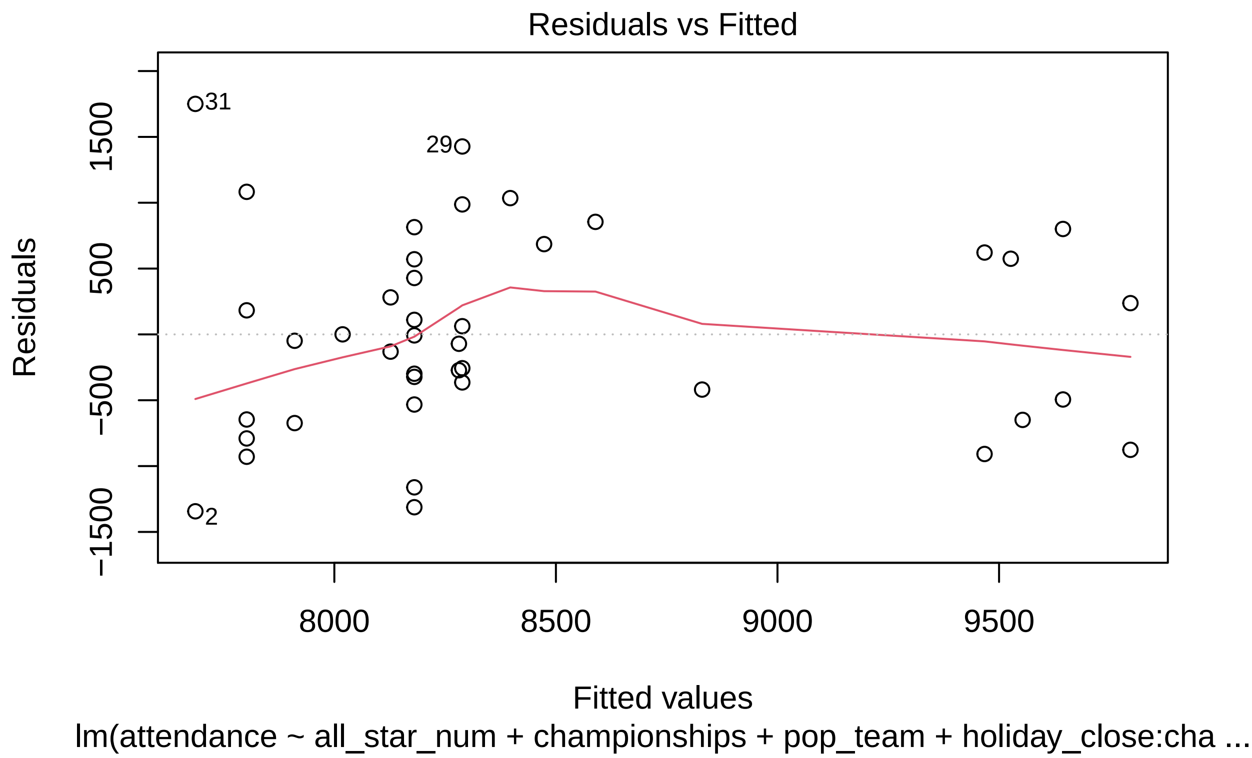
```
vif(trial_mod)
```

```
##                all_star_num               championships
##                    3.206665                    1.288950
##                    pop_team championships:holiday_close
##                    2.172403                    2.180818
```

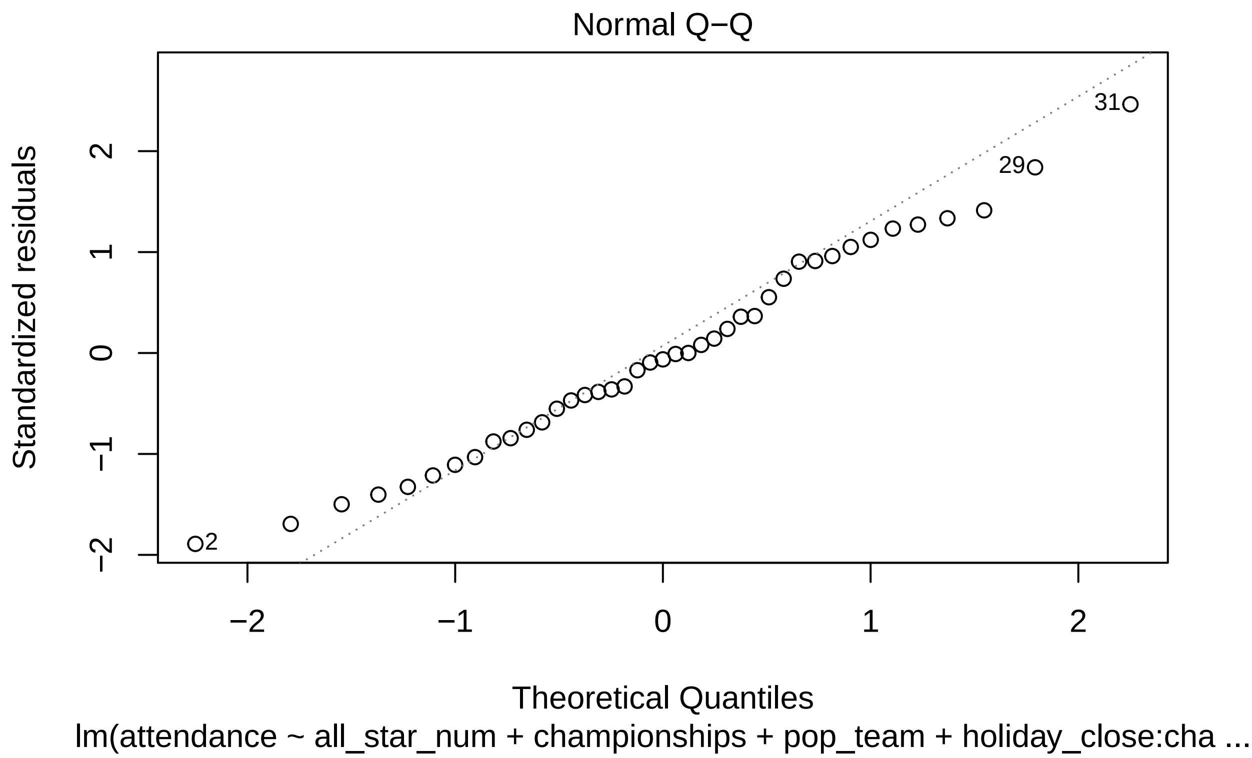
```
anova(trial_mod)
```

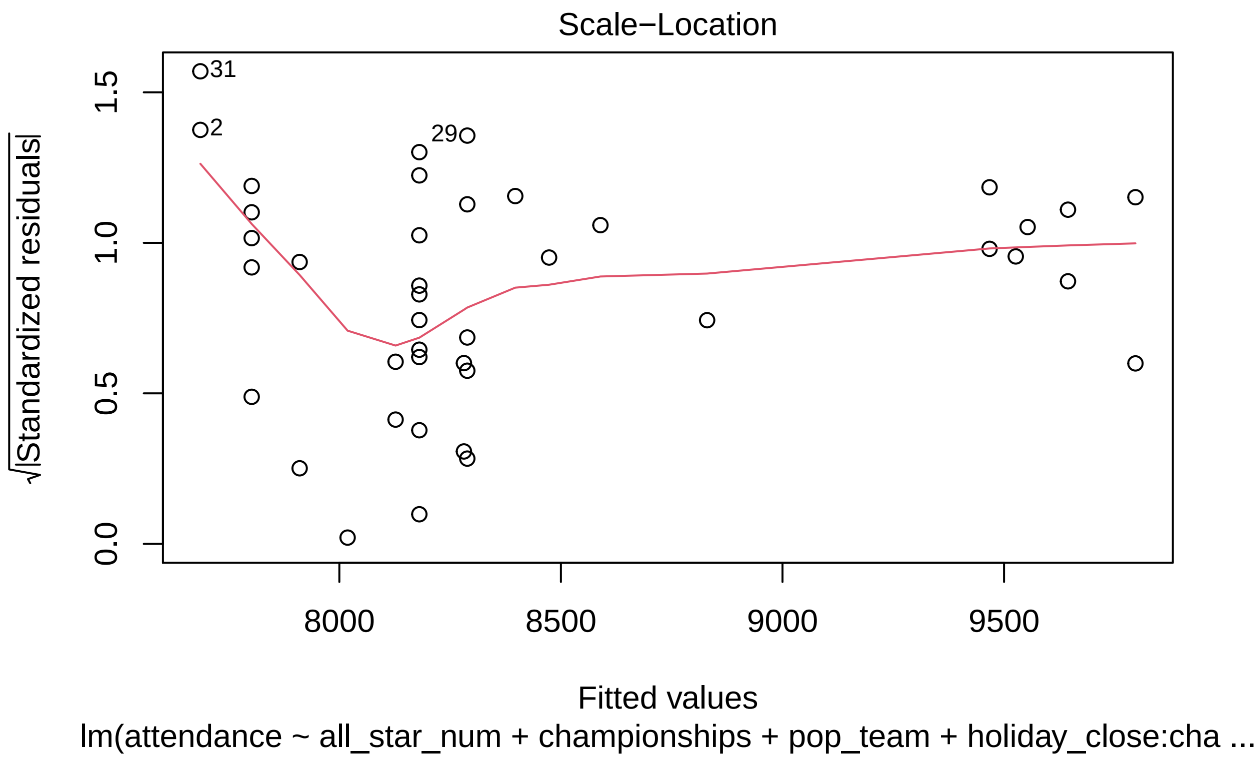
```
## Analysis of Variance Table
##
## Response: attendance
##                             Df   Sum Sq Mean Sq F value   Pr(>F)
## all_star_num                 1  3390520 3390520  5.3817 0.026134 *
## championships                1  7470746 7470746 11.8582 0.001473 **
## pop_team                     1   592169  592169  0.9399 0.338762
## championships:holiday_close  1  4768693 4768693  7.5693 0.009235 **
## Residuals                   36 22680292  630008
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(trial_mod)
```

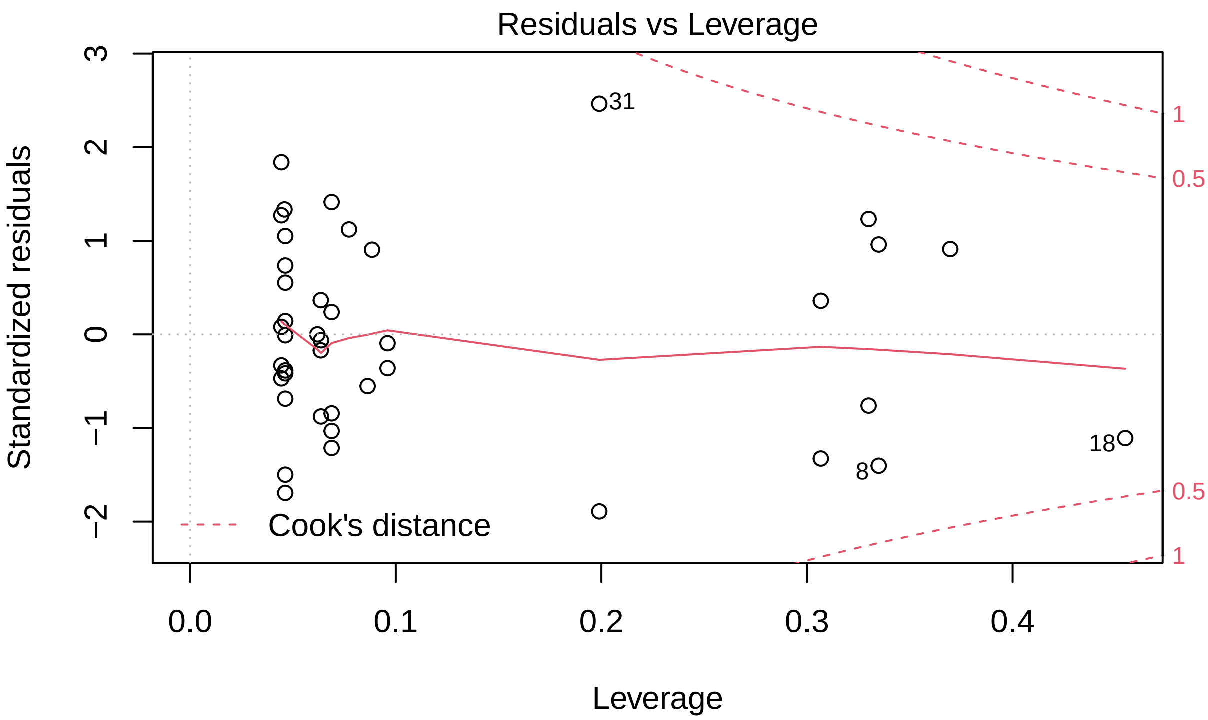


Residuals vs Fitted
Residuals
1500
500
−500
−1500
31
29
2
8000
8500
9000
9500
Fitted values
lm(attendance ~ all_star_num + championships + pop_team + holiday_close:cha ...


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(attendance ~ all_star_num + championships + pop_team + holiday_close:cha ...
2
29
31
−2
−1
0
1
2



Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
31
2
29
8000
8500
9000
9500
Fitted values
lm(attendance ~ all_star_num + championships + pop_team + holiday_close:cha ...



lm(attendance ~ all_star_num + championships + pop_team + holiday_close:cha ...