How does Travel Affect Baseball

Benjamin Ginsburg

2024-06-06

```
library(Lahman)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(baseballr)
library(retrosheet)
```

```
##
## For Retrosheet data obtained with this package:
##
## The information used here was obtained free of charge from
## and is copyrighted by Retrosheet. Interested parties may
## contact Retrosheet at "www.retrosheet.org"
```

```
library(ggplot2)
```

In baseball, teams travel for 81 games throughout the year, spending a significant amount of time on the road. This statistic raises the question of whether the distance a team travels affects their overall performance.

This project aims to investigate the relationship between how far a team travels and a teams performance over the course of a season. One facet to explore is whether there is a correlation between the total distance a team travels during the season and any negative consequences, such as a decrease in average runs scored and Wins percentage. Additionally, the project seeks to determine the impact of travel distance on the next day performance by examining how the distance traveled by a team on any given day corresponds with their performance the following day. By reviewing travel data and performance metrics, the project aims to provide insights into how travel affects baseball team's overall performance.

The `haversine_distance` function takes in a latitude and longitude for two different coordinates and then find the distance between them and converts it into miles. The function is called when determining how far two MLB stadiums are from each other.

```
haversine_distance <- function(lat1, lon1, lat2, lon2) {
  lat1 <- lat1 * pi / 180
  lon1 <- lon1 * pi / 180
  lat2 <- lat2 * pi / 180
  lon2 <- lon2 * pi / 180


  dlon <- lon2 - lon1
  dlat <- lat2 - lat1
  a <- sin(dlat/2)^2 + cos(lat1) * cos(lat2) * sin(dlon/2)^2
  c <- 2 * asin(sqrt(a))
  R <- 6371
  distance_km <- R * c

  distance_miles <- distance_km * 0.621371

  return(distance_miles)
}
```

The `teamNames` DataFrame is used to match all the the team names in the `Retrosheet` database.

```
teamNames <- c("HOU","TOR","CIN","COL","DET","LAN","SEA","MIA","CHN"
               ,"MIL","PHI","NYN","SDN","PIT","WAS","SFN","ARI","SLN"
               ,"BOS","BAL","ANA","CHA","NYA","KCA","CLE","OAK","MIN",
                "TBA","ATL","TEX")
```

The `team_data` list will store all of games each MLB team plays. The for loop will then gather all of the information from the `Retrosheet` database and put it in each respective table.

```
team_data <- list()

for (i in teamNames) {
  team_data[[i]] <- getRetrosheet("game", 2023) %>%
  filter(VisTm == i | HmTm == i) %>%
  mutate(TeamName = i, OpposingTeam =ifelse(VisTm == i,HmTm,VisTm))
}
```

This data frame contains the latitude and longitude of all the stadiums in the MLB. The data was gathered from the following Google Sheets document and rounded to 4 decimals: “https://docs.google.com/spreadsheets/d/1p0R5qqR7XjoRG2mR5E1D_trlygHSqMOUdMgMpzq0gjU/htmlview (https://docs.google.com/spreadsheets/d/1p0R5qqR7XjoRG2mR5E1D_trlygHSqMOUdMgMpzq0gjU/htmlview)”

```
stadiums_df <- data.frame(
  Team = c("HOU", "TOR", "CIN", "COL", "DET", "LAN", "SEA", "MIA", "CHN", "MIL",
           "PHI", "NYN", "SDN", "PIT", "WAS", "SFN", "ARI", "SLN", "BOS", "BAL",
           "ANA", "CHA", "NYA", "KCA", "CLE", "OAK", "MIN", "TBA", "ATL", "TEX"),
  Latitude = c(29.7573, 43.6414, 39.0979, 39.7559, 42.3391, 34.0736, 47.5914, 25.7786,
               41.9484, 43.0280, 39.9056, 40.7571, 32.7076, 40.4469, 38.8730, 37.7786,
               33.4452, 38.6226, 42.3465, 39.2839, 33.8003, 41.8299, 40.8296, 39.0517,
               41.4958, 37.7516, 44.9817, 27.7683, 33.8908, 32.7473),
  Longitude = c(-95.3554, -79.3894, -84.5086, -104.9942, -83.0486, -118.2400, -122.3324,
                -80.2194, -87.6553, -87.9712, -75.1667, -73.8458, -117.1570, -80.0057,
                -77.0074, -122.3893, -112.0667, -90.1928, -71.0970, -76.6215, -117.8827,
                -87.6338, -73.9262, -94.4803, -81.6853, -122.2005, -93.2775, -82.6486,
                -84.4678, -97.0826)
)
all_team_data <- bind_rows(team_data)
```

This function starts off by using `left_join` to add the latitude and longitude of the stadiums. It then uses `mutate` function to calculate the distance traveled for each game with the `haversine_distance` function, whether the team won or lost, how far a team travelled using the `cumsum` function, a Boolean to determine if a team won or lost, and a function to determine how many runs a team scored.

```
all_team_data <- all_team_data %>%
  left_join(stadiums_df %>% rename(AwayTeamLat = Latitude, AwayTeamLon = Longitude),
            by = c("VisTm" = "Team")) %>%
  left_join(stadiums_df %>% rename(HomeTeamLat = Latitude, HomeTeamLon = Longitude),
            by = c("HmTm" = "Team")) %>%
  mutate(GameLat = HomeTeamLat, GameLon = HomeTeamLon) %>%
  arrange(TeamName, Date) %>%
  group_by(TeamName) %>%
  mutate(PrevGameLat = lag(GameLat),
         PrevGameLon = lag(GameLon),
         DistanceTravelled = ifelse(is.na(PrevGameLat) | is.na(PrevGameLon), 0,
                                    ifelse(PrevGameLat == GameLat & PrevGameLon == GameL
on, 0,
                                           haversine_distance(PrevGameLat, PrevGameLon,
GameLat, GameLon))),
         WinLoss = ifelse(HmRuns > VisRuns & HmTm == TeamName, "W",
                          (ifelse((HmRuns < VisRuns & VisTm == TeamName), "W", "L"))),
         CumulativeTravel = cumsum(DistanceTravelled),
         TravelDay = ifelse(DistanceTravelled > 0, "Yes", "No"),
         RunsScored = ifelse(HmTm == TeamName, HmRuns, VisRuns)) %>%
  ungroup()
```

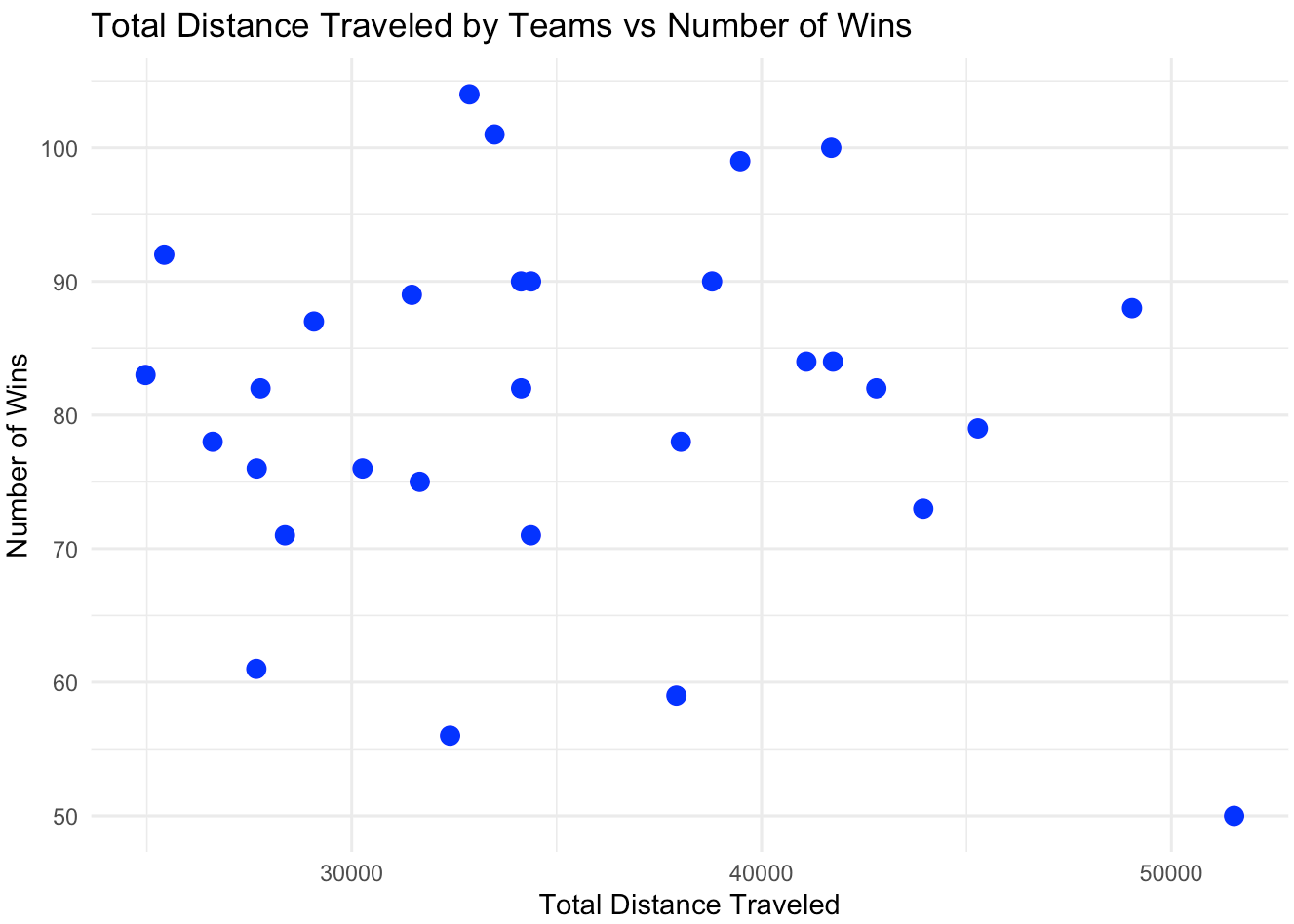
In order to show the accuracy of the the functions above, I added this chart to show the total distance of how far each team traveled over the 2023 season. I then compared this data with the actual distracted each team traveled from the MLB website and found that they were within approximately 5 miles of each other.

```
all_team_data %>%
  group_by(TeamName) %>%
  summarize(travel = sum(DistanceTravelled)) %>%
  arrange(desc(travel))
```

```
## # A tibble: 30 × 2
##    TeamName travel
##    <chr>     <dbl>
##  1 OAK      51529.
##  2 SEA      49039.
##  3 SFN      45278.
##  4 ANA      43945.
##  5 SDN      42800.
##  6 MIA      41743.
##  7 LAN      41700.
##  8 ARI      41092.
##  9 TBA      39481.
## 10 HOU      38793.
## # ℹ 20 more rows
```

After verifying the data, I created a bubble chart to show how far each team traveled compared to the total number of wins. I wanted to see if there was any kind of resemblance for how far a team traveled and their number of wins.

```
Team_Travel_Wins <- all_team_data %>%
  group_by(TeamName) %>%
  summarise(TotalDistance = sum(DistanceTravelled), Wins = sum(WinLoss == "W")) %>%
  arrange(TotalDistance)

  ggplot(Team_Travel_Wins, aes(x = TotalDistance, y = Wins, label = TeamName)) +
  geom_point(color = "blue", size = 3) +
  labs(title = "Total Distance Traveled by Teams vs Number of Wins",
       x = "Total Distance Traveled",
       y = "Number of Wins") +
  theme_minimal()
```



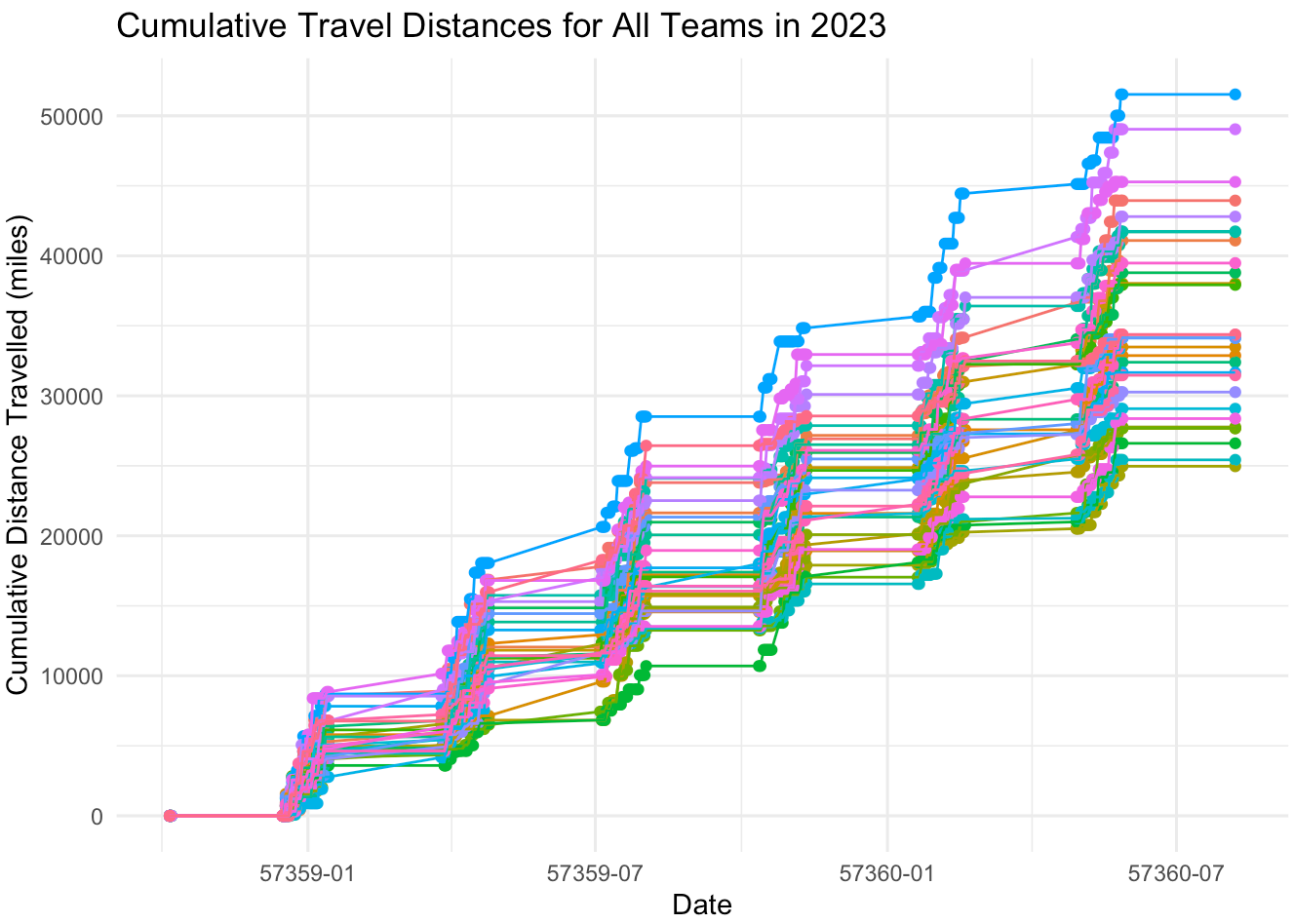
I then created a Regression Model to see if there was any statistical significance between how far a team traveled and their number of wins. The results shows that there was a p value of 0.709 which is great than 0.05, showing that there is no significance in how much a team traveled compared to their number of wins.

```
Team_Travel_Wins_Reg <- lm(Wins ~ TotalDistance, data =  Team_Travel_Wins)
  summary(Team_Travel_Wins_Reg)
```

```
##
## Call:
## lm(formula = Wins ~ TotalDistance, data = Team_Travel_Wins)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.8632  -6.3551   0.7485   8.8401  22.6851
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   85.6352244 12.5452380   6.826 2.04e-07 ***
## TotalDistance -0.0001314  0.0003489  -0.377    0.709
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.32 on 28 degrees of freedom
## Multiple R-squared:  0.00504,    Adjusted R-squared:  -0.03049
## F-statistic: 0.1418 on 1 and 28 DF,  p-value: 0.7093
```

The next graph I created was a chart that showed the how far a team traveled at a time over the course of the season. The top bar is the Oakland A's, whom traveled the most throughout the MLB season.

```
ggplot(all_team_data, aes(x = as.Date(Date, format="%Y-%m-%d"), y = CumulativeTravel, co
lor = TeamName)) +
  geom_point() +
  geom_line() +
  labs(title = "Cumulative Travel Distances for All Teams in 2023",
       x = "Date",
       y = "Cumulative Distance Travelled (miles)") +
  theme_minimal() +
  theme(legend.position = "none")
```



The regression for this graph proved that the data was statically insignificant due to the fact that the p-value was significantly higher than the 0.05 threshold.

```
travel_games <- all_team_data %>%
  filter(DistanceTravelled > 0)

travel_games_regression <-lm(RunsScored ~ DistanceTravelled, data = travel_games)
summary(travel_games_regression)
```

```
##
## Call:
## lm(formula = RunsScored ~ DistanceTravelled, data = travel_games)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8119 -2.6144 -0.6851  1.4753 15.3169
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.8164103  0.1665860   28.91   <2e-16 ***
## DistanceTravelled -0.0001283  0.0001474   -0.87    0.384
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.209 on 1138 degrees of freedom
## Multiple R-squared:  0.0006648,  Adjusted R-squared:  -0.0002133
## F-statistic: 0.7571 on 1 and 1138 DF,  p-value: 0.3844
```

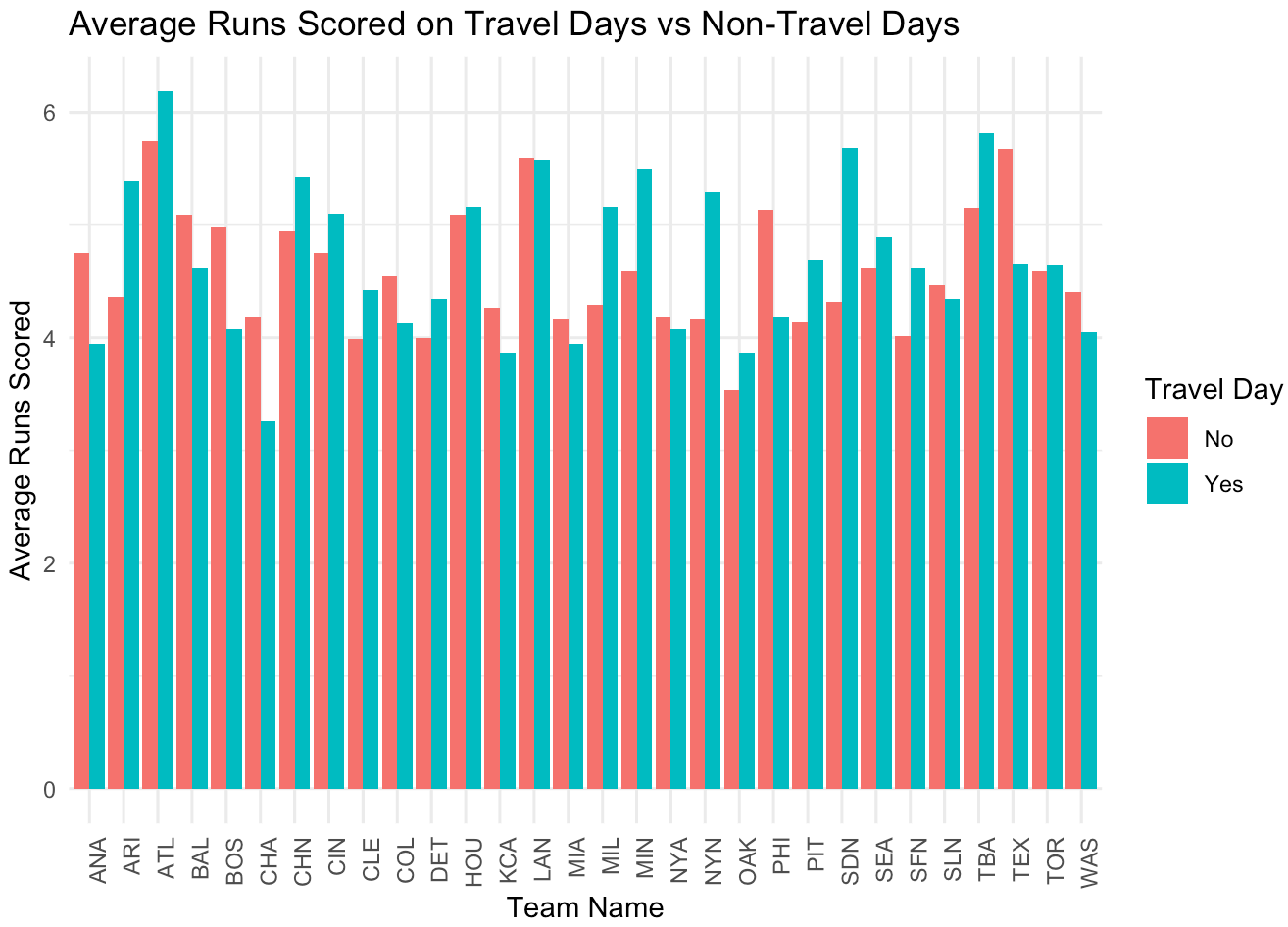
This histogram of team game groups how far a team traveled by distance and then stack teams losses over wins. Most of the columns are split evenly, however, the columns closer to the 3,000 mile marker tended to have more losses than wins, which is statically significant.

```
model4 <- all_team_data %>%
  group_by(TravelDay) %>%
  summarize(Wins = sum(WinLoss == "W"), Loss = sum(WinLoss == "L"), Percent = Wins/(Wins
+Loss), .groups = 'keep')

team_data <- all_team_data %>%
  filter(DistanceTravelled >0 )

ggplot(team_data, aes(x = DistanceTravelled, fill = WinLoss)) +
  geom_histogram(bins = 8, color = "black", position = "stack") +
  labs(title = "Distribution of Distance Traveled for Wins and Losses",
       x = "Distance Travelled (miles)",
       y = "Number of Games",
       fill = "Outcome") +
  theme_minimal() +
  scale_fill_manual(values = c("W" = "blue", "L" = "red"))
```


```
team_data <- team_data %>%
  mutate(DistanceBucket = cut(DistanceTravelled, breaks = 8, labels = FALSE))
```

I wanted to make a comparison for how teams performed on travels days versus non travel days. To do this I made a bar chart that compares each teams average runs scored on travel days versus non travel days.

```
Travel_Runs <- all_team_data %>%
  group_by(TeamName, TravelDay) %>%
  summarise(RunsScored = (sum(RunsScored)), CountGames = n(), AverageRuns = RunsScored/
CountGames,.groups = 'keep')

ggplot(Travel_Runs, aes(x = TeamName, y = AverageRuns, fill = TravelDay)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Runs Scored on Travel Days vs Non-Travel Days",
       x = "Team Name",
       y = "Average Runs Scored",
       fill = "Travel Day") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



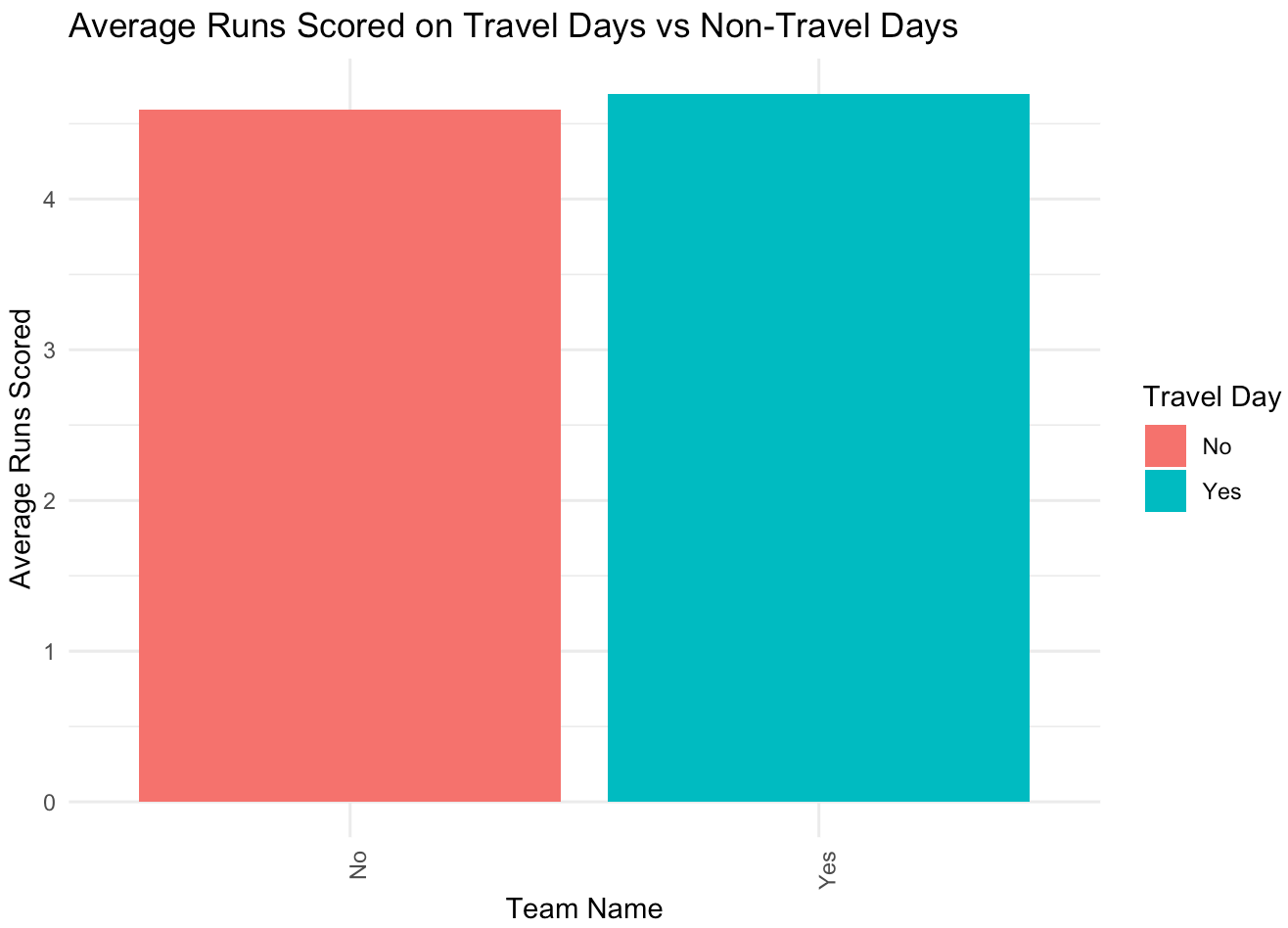
I then ran a regression model to see if there was any correlation between how many runs teams scored and having traveled the day before. The results had a p-value of 0.512 which is higher than 0.05, showing how there was no statistical significance between the data.

```
regression_model <- lm(AverageRuns ~ TravelDay, data = Travel_Runs)
summary(regression_model)
```

```
##
## Call:
## lm(formula = AverageRuns ~ TravelDay, data = Travel_Runs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4410 -0.4348 -0.0623  0.4698  1.4868
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.5903     0.1148   39.99   <2e-16 ***
## TravelDayYes   0.1072     0.1623    0.66    0.512
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6287 on 58 degrees of freedom
## Multiple R-squared:  0.007457,   Adjusted R-squared:  -0.009656
## F-statistic: 0.4358 on 1 and 58 DF,  p-value: 0.5118
```

In order to make the data easier to read, I created a bar chart with the average number of runs a team scored and if it was a travel day. The results showed that there was not that much of a difference and that there was no statistical significance between the data.

```
Travel_Runs <- all_team_data %>%
  group_by(TravelDay) %>%
  summarise(RunsScored = (sum(RunsScored)), CountGames = n(), AverageRuns = RunsScored/
CountGames,.groups = 'keep')


ggplot(Travel_Runs, aes(x = TravelDay, y = AverageRuns, fill = TravelDay)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Runs Scored on Travel Days vs Non-Travel Days",
       x = "Team Name",
       y = "Average Runs Scored",
       fill = "Travel Day") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
regression_model <- lm(AverageRuns ~ TravelDay, data = Travel_Runs)
summary(regression_model)
```

```
##
## Call:
## lm(formula = AverageRuns ~ TravelDay, data = Travel_Runs)
##
## Residuals:
## ALL 2 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.5906        NaN     NaN      NaN
## TravelDayYes   0.1068        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 1 and 0 DF,  p-value: NA
```

Through my findings, there was no significance between weather or not a team traveled the before a baseball game. The only exception to this was the fact that teams who traveled about 2,800 miles before a game, or across the country, tended to perform worse the following game. There are many outside reasons for why there is no significant impact on teams performance, but some reasons might include the investment that each team makes to player performance. Players are provided with private flights, which eliminate the need for airport time,

personal trainers, nutritionist, and many other benefits that improve performance. In a future study, I would want to look into these other characteristics that benefit players and see if they affect how the team performs when traveling.