Tips_Final_Project

Brandon Cunningham

2024-05-04

Introduction

In this paper I will be using approximately a year and a half of data collected from time sheets from my girlfriends time as a server and bartender for Olive Garden. I will be looking into trends in the data over time, in days as a server instead of a bartender, length of shifts, the day of the week, and looking into anecdotal observations such as that rainy days are better than clear days, colder days are better than warmer days, and that days where the Buffalo Bills have a game are worse than normal days. The data I was able to collect from time sheets only includes the total time worked for shift and total tips for the shift so other factors such as the total sales for the shift, number of customers severed, and section where she was serving cannot be accounted for. Lastly, the method of determining how good a shift was will be in inflation adjusted tips per hour, and will not include base pay before tips as this is a constant that is not effected by any of the factors we are interested in.

```
url_tips <- 'https://raw.githubusercontent.com/btc2628/DATA607/main/Tips_Final_Project/Olive_Garden_Tips
tips_data <- readr::read_csv(url_tips)
```

```
url_weather <- 'https://raw.githubusercontent.com/btc2628/DATA607/main/Tips_Final_Project/weather.csv'
weather_data <- readr::read_csv(url_weather)
```

```
url_2022 <- "https://fbschedules.com/2022-buffalo-bills-schedule/"
page_2022 <- read_html(url_2022)
nodes_2022 <- html_nodes(page_2022, xpath = "//script[@type='application/ld+json']")
content_2022 <- html_text(nodes_2022)
dates_2022 <- lapply(content_2022, function(json) {
  parsed <- fromJSON(json, flatten = TRUE)
  if("startDate" %in% names(parsed)) {
    return(data.frame(Date = parsed$startDate))
  } else {
    return(data.frame(Date = NA))
  }
})
```

```
url_2023 <- "https://fbschedules.com/2023-buffalo-bills-schedule/"
page_2023 <- read_html(url_2023)
nodes_2023 <- html_nodes(page_2023, xpath = "//script[@type='application/ld+json']")
content_2023 <- html_text(nodes_2023)
dates_2023 <- lapply(content_2023, function(json) {
  parsed <- fromJSON(json, flatten = TRUE)
  if("startDate" %in% names(parsed)) {
    return(data.frame(Date = parsed$startDate))
  } else {
    return(data.frame(Date = NA))
  }
```

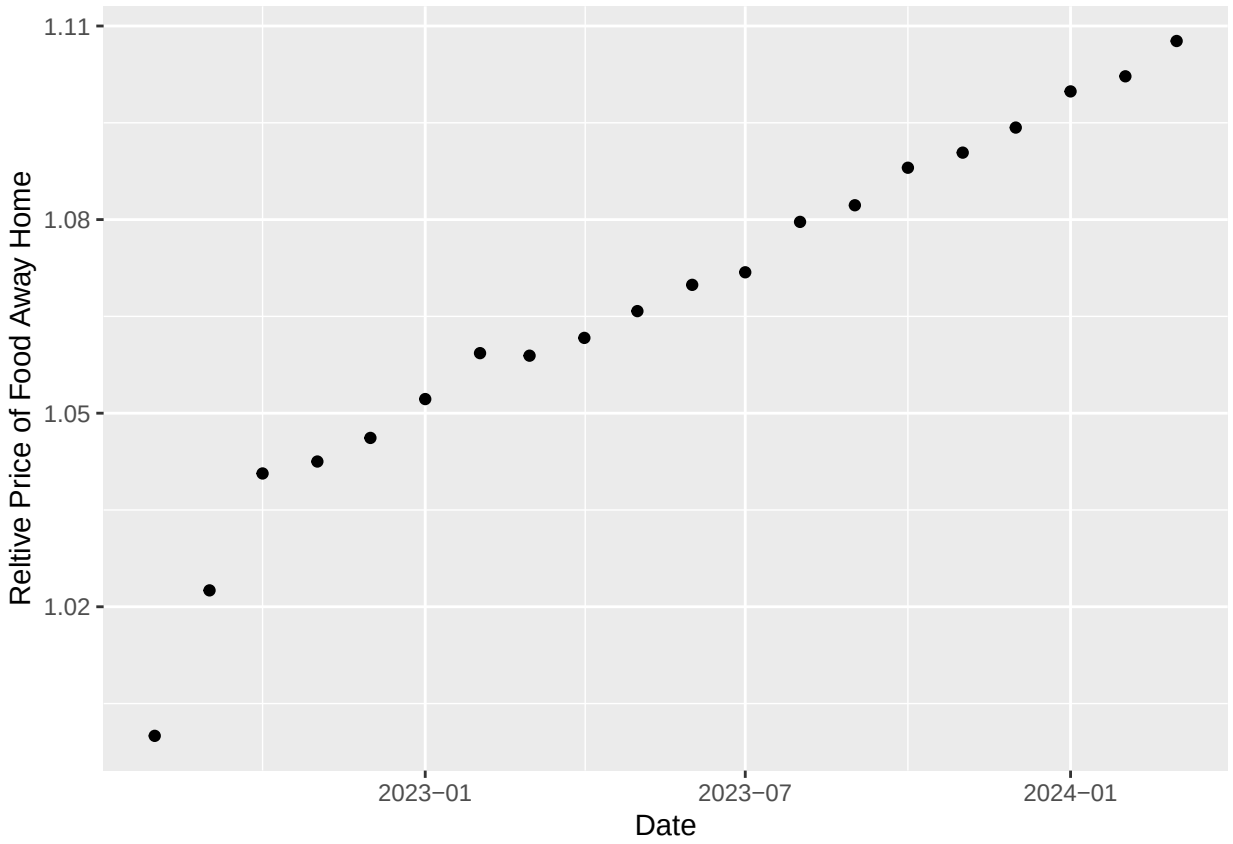
```
})
```

```
bills_dates <- bind_rows(dates_2023, dates_2022) %>%
  filter(!is.na(Date))
bills_dates$Date <- as.Date(bills_dates$Date, "%Y-%m-%d")
bills_dates$bills_game <- 1
```

```
fredr_set_key("48773d8fbfdd651713d4d48c3cbe3f92")

cpi_data <- fredr(series_id = "CUURX100SEFV", #CPI Dataset on Northeast food away from home
                  observation_start = as.Date("2022-08-21"),
                  frequency = "m",
                  units = "lin")
```

In the past few blocks of code I have loaded in all necessary datasets for analysis, this includes the tips data I personally collected, weather data collected from the National Oceanic and Atmospheric Administration, Bills game schedules for 2022 and 2023, and CPI data for food away from home in the north east from the Federal Reserve.

```
starting_cpi <- cpi_data %>%
  filter(date == as.Date("2022-8-01")) %>%
  pull(value)
cpi_data$percent <- cpi_data$value/starting_cpi
cpi_data <- cpi_data[, c('date', 'percent', 'value')]
cpi_data$date <- as.Date(cpi_data$date, "%Y-%m-%d")

ggplot(data = cpi_data, aes(x = date, y = percent)) +
  geom_point() +
  xlab("Date") +
  ylab("Reltive Price of Food Away Home")
```



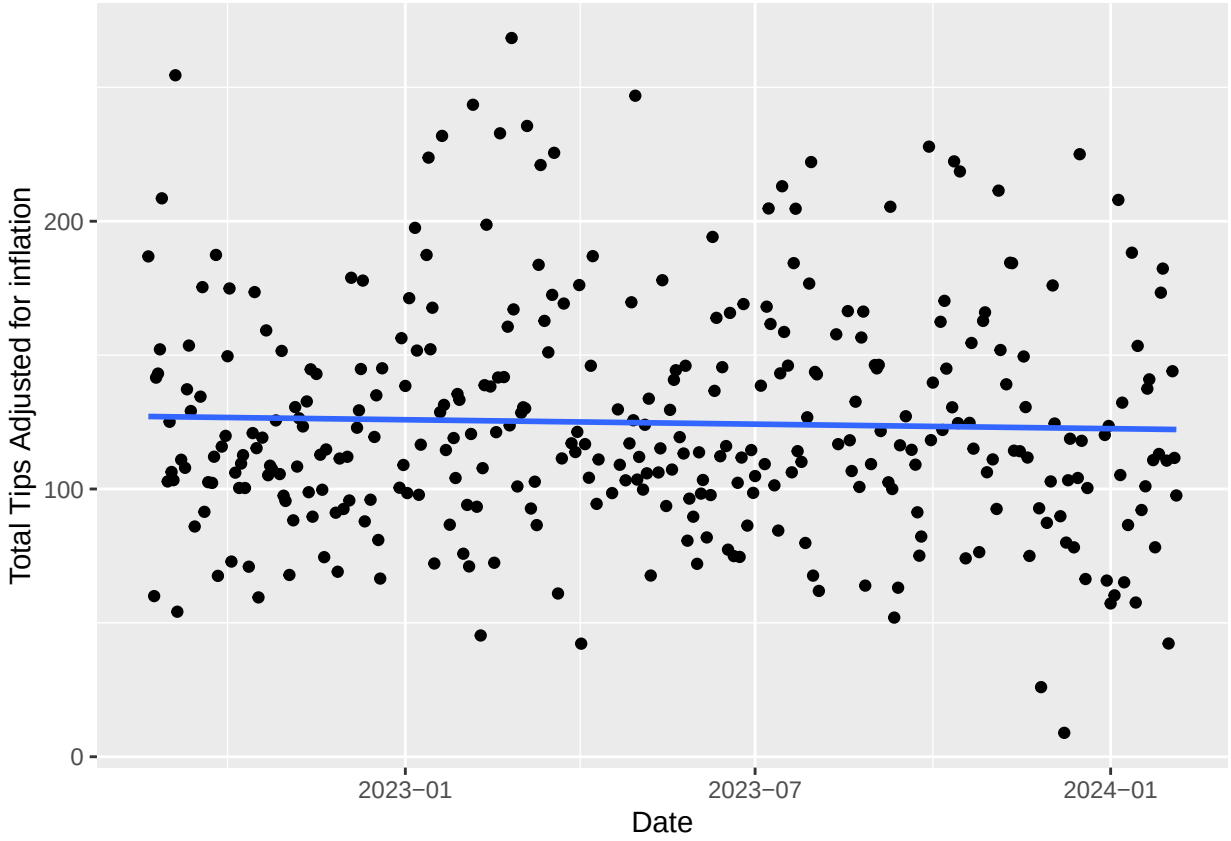
In this graph we can see the relative price of food away from home in the north east using August of 2022 as a starting point.

```
tips_data$Date <- as.Date(tips_data$Date, "%m/%d/%Y")
tips_data$Date <- format(tips_data$Date, "%Y-%m-%d")
tips_data$Date <- as.Date(tips_data$Date, "%Y-%m-%d")
```

```
bar_start <- as.Date('2023-10-01', "%Y-%m-%d")
tips_data$Bartender <- ifelse(tips_data$`Day of Week` == 'Sun', ifelse(tips_data$Date >= bar_start, 1, 0
tips_data$Tips <- as.numeric(sub("\\$", "", tips_data$Tips))
tips_data$Tips <- ifelse(tips_data$Bartender == 1, ifelse(tips_data$Tips != 0, tips_data$Tips + 5, 0),
```

In this code block I am adding $5 to every bar shift. The reasoning for this is that for every bar shift there is a 1 hour portion before the restaurant is opened where the bartender prepares for opening, and during this time the bartender is paid at regular minimum wage instead of tipped minimum wage.

```
tips_data <- tips_data %>% mutate(date_floor = floor_date(Date, "month"))
tips_data <- merge(tips_data, cpi_data, by.x="date_floor", by.y="date")

tips_data$adjusted_tips <- tips_data$Tips * (starting_cpi/tips_data$value)

tips_data <- tips_data %>% filter(Tips != 0)

ggplot(data = tips_data, aes(x = Date, y = adjusted_tips)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  xlab("Date") +
  ylab("Total Tips Adjusted for inflation")
```



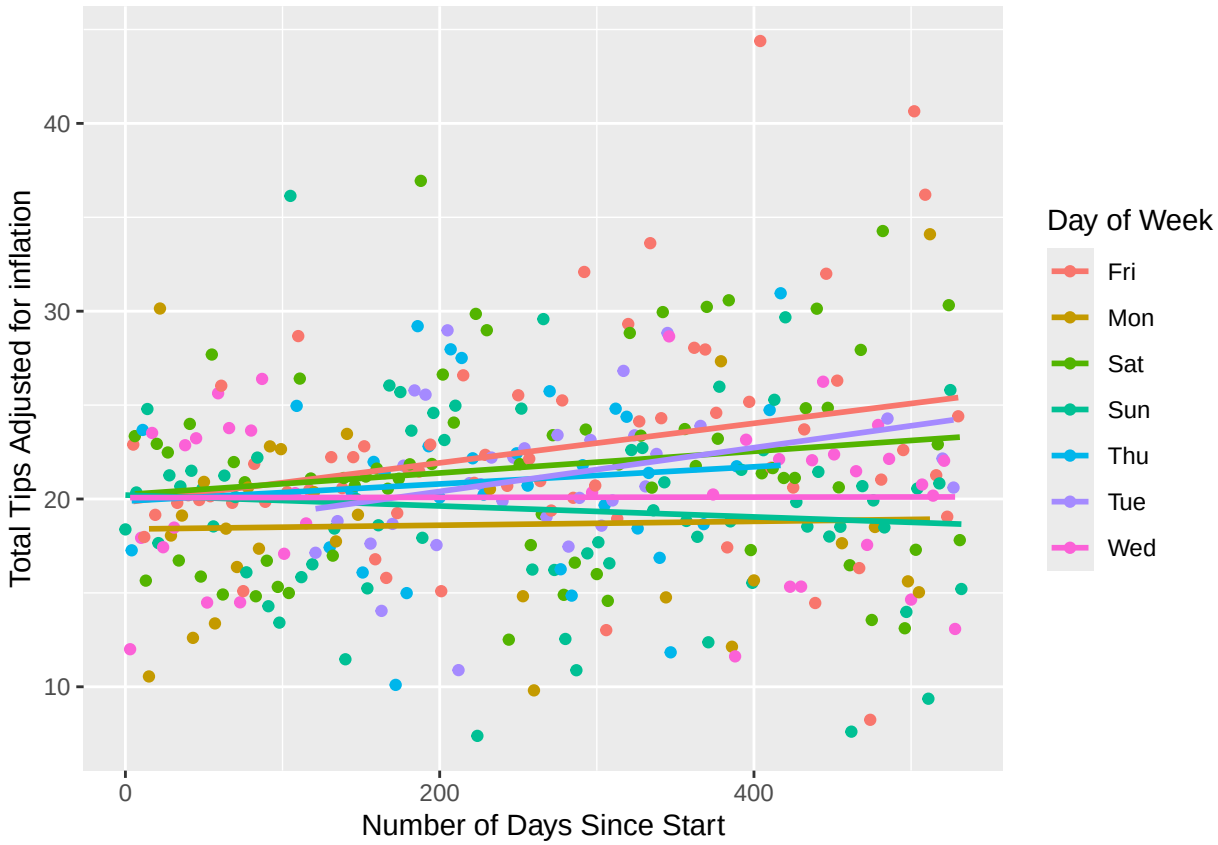
In this graph we can see all inflation adjusted tips over time, and while their is a wide spread, there is a general downward trend over time.

```
tips_bills <- merge(tips_data, bills_dates, by = "Date", all.x = TRUE)
```

```
weather_data$precipitated <- ifelse(weather_data$PRCP >= 0.1, 1, 0)
weather_data$precipitated <- ifelse(weather_data$SNOW >= 0.2, 1, weather_data$precipitated)
weather_data$TAVG[is.na(weather_data$TAVG)] <- (weather_data$TMAX + weather_data$TMIN)/2
weather_narrowed <- weather_data[, c('DATE', 'precipitated', 'PRCP', 'SNOW', 'TAVG')]
```

```
merged_data <- merge(tips_bills, weather_narrowed, by.x = "Date", by.y = "DATE")

merged_data[is.na(merged_data)] <- 0

merged_data$Date <- as.Date(merged_data$Date, "%Y-%m-%d")
start_date <- as.Date("2022-08-21")
merged_data$Date <- as.numeric(merged_data$Date - start_date)

merged_data$shift_length <- merged_data$Hours + (merged_data$Minutes/60)

merged_data$tips_per_hour <- merged_data$adjusted_tips/merged_data$shift_length
```

```
ggplot(data = merged_data, aes(x = Date, y = tips_per_hour, color=`Day of Week`)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  xlab('Number of Days Since Start') +
  ylab("Total Tips Adjusted for inflation")
```

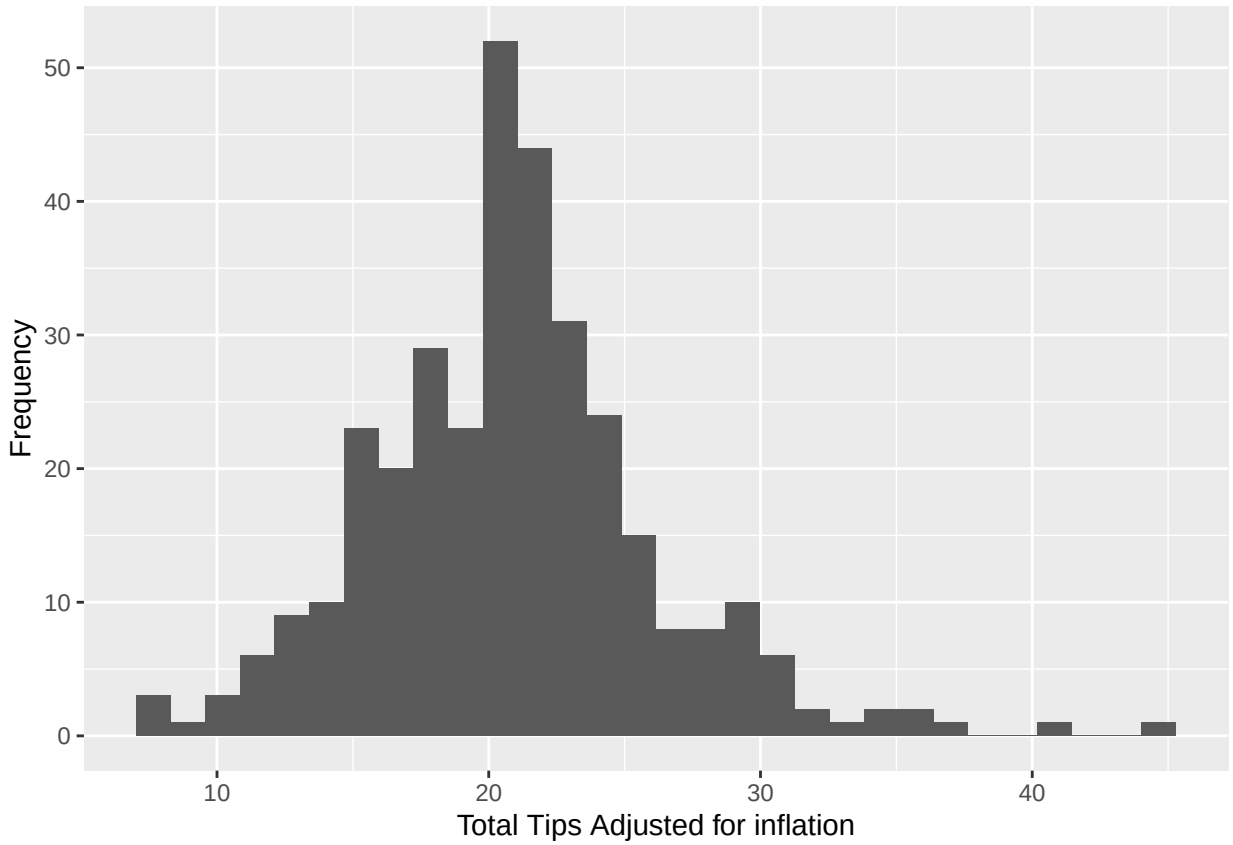


Here we can see the the same plot as before but broken down to day of week level and it becomes clear that some days are better than others, and that some days like Friday seem to get better and better over time when compared to the rest of the days.

```
tips_summary <- summary(merged_data$tips_per_hour)
tips_summary["Standard Deviation"] <- sd(merged_data$tips_per_hour)
tips_summary["Variance"] <- var(merged_data$tips_per_hour)
tips_summary
```

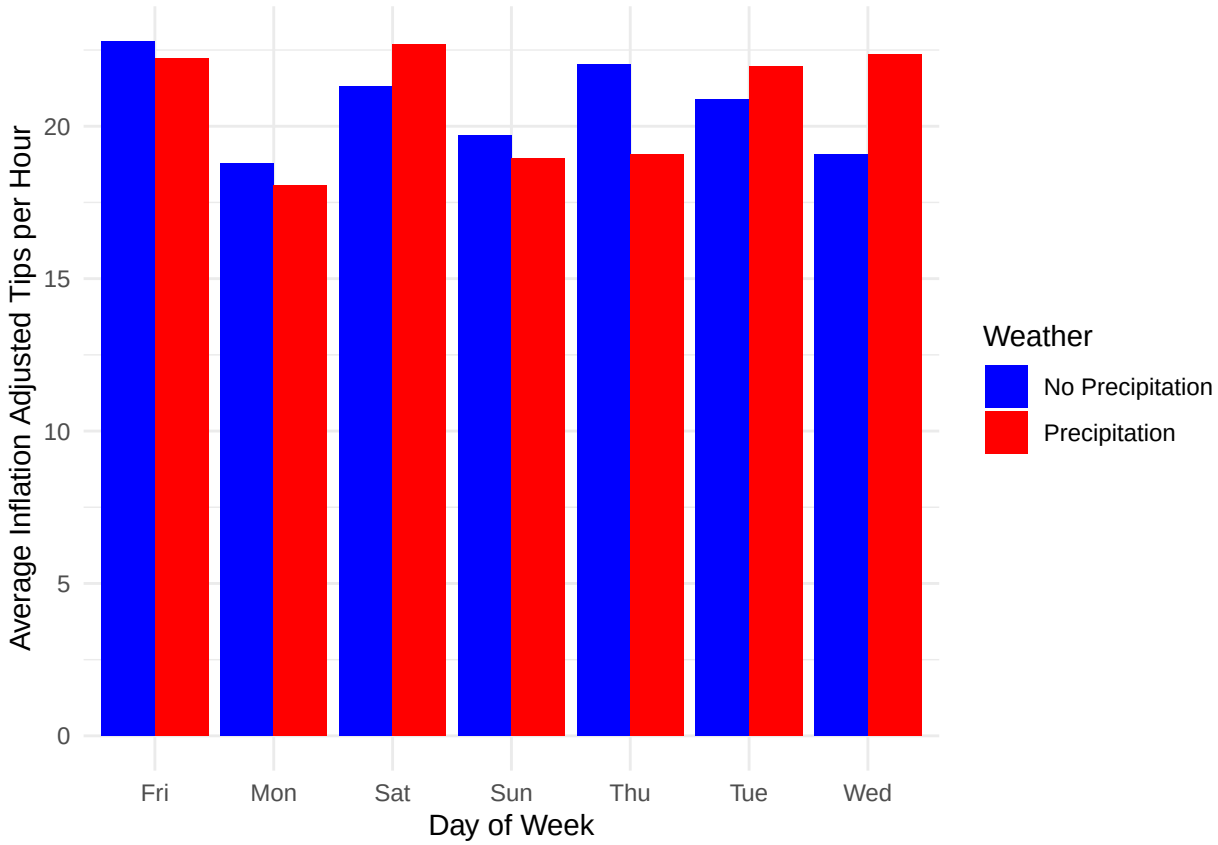
```
##               Min.            1st Qu.             Median               Mean
##              7.386             17.546             20.712             20.857
##            3rd Qu.               Max. Standard Deviation           Variance
##             23.400             44.387              5.251             27.575
```

```
ggplot(data = merged_data, aes(x = tips_per_hour)) +
  geom_histogram() +
  xlab("Total Tips Adjusted for inflation") +
  ylab("Frequency")
```



With this histogram and previous stats we can see that tips per hour is mostly normally distributed with a center around $21 with a slight right skew and a standard deviation of $5.25.

```
average_tips <- merged_data %>%
  group_by(`Day of Week`, precipitated) %>%
  summarise(average_tips_per_hour = mean(tips_per_hour, na.rm = TRUE), .groups = 'drop')

ggplot(average_tips, aes(x = `Day of Week`, y = average_tips_per_hour, fill = factor(precipitated))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(fill = "Weather", y = "Average Inflation Adjusted Tips per Hour", x = "Day of Week") +
  theme_minimal() +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"), labels = c("0" = "No Precipitation", "1" = "
```



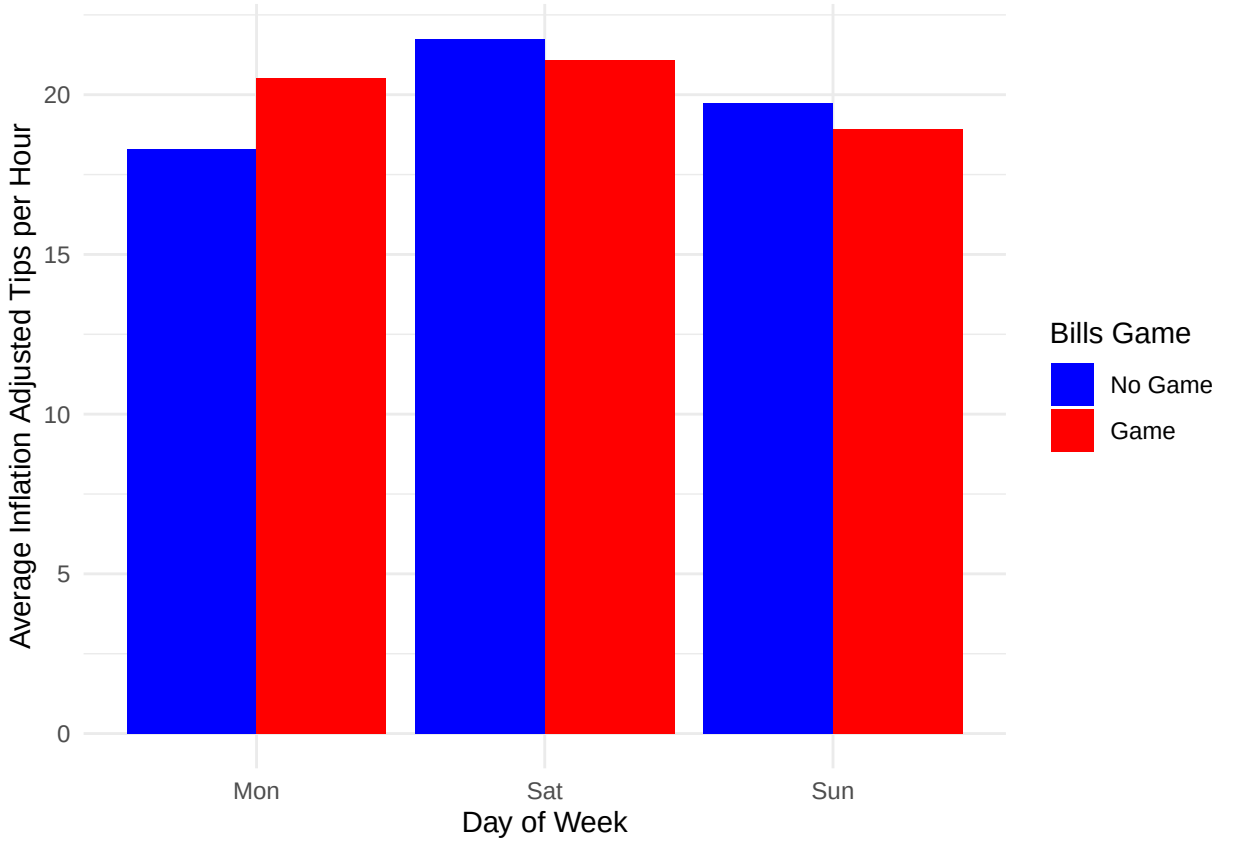

```
t_test_weather <- t.test(tips_per_hour ~ precipitated, data = merged_data)

t_test_weather
```

```
##
##  Welch Two Sample t-test
##
## data:  tips_per_hour by precipitated
## t = -0.248, df = 207.46, p-value = 0.8044
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -1.372146  1.065498
## sample estimates:
## mean in group 0 mean in group 1
##        20.80763        20.96096
```

From a visual inspection of the graph and the t-test we must conclude that there is no significant relationship between whether it rained or snowed and the amount of tips made per hour.

```
bills_tips <- merged_data %>%
  group_by(`Day of Week`, bills_game) %>%
  summarise(average_tips_per_hour = mean(tips_per_hour, na.rm = TRUE), .groups = 'drop') %>%
  filter(`Day of Week` %in% c("Sun", "Sat", "Mon"))


ggplot(bills_tips, aes(x = `Day of Week`, y = average_tips_per_hour, fill = factor(bills_game))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(fill = "Bills Game", y = "Average Inflation Adjusted Tips per Hour", x = "Day of Week") +
```

```
  theme_minimal() +
  scale_fill_manual(values = c("0" = "blue", "1" = "red"), labels = c("0" = "No Game", "1" = "Game"))
```



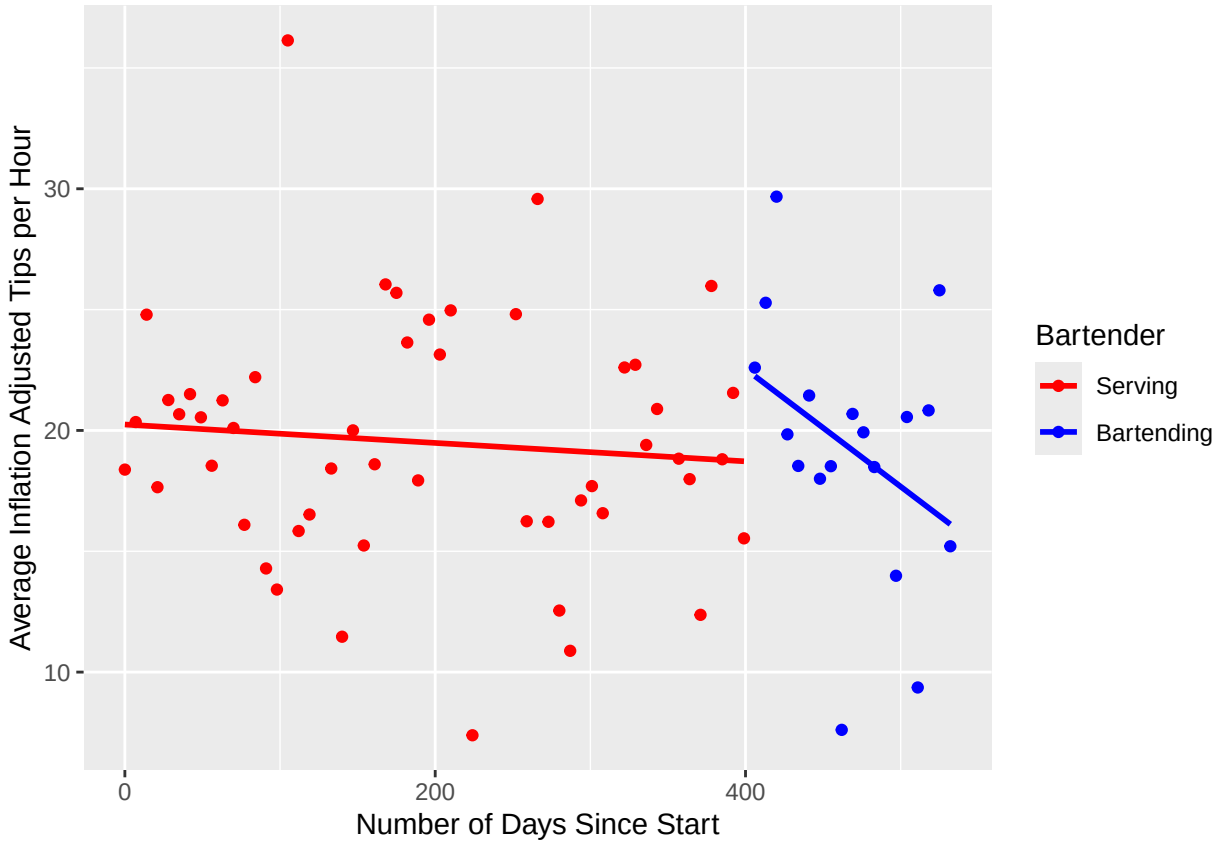

```
t_test_bills <- t.test(tips_per_hour ~ bills_game, data = merged_data)

t_test_bills
```

```
##
##  Welch Two Sample t-test
##
## data:  tips_per_hour by bills_game
## t = 1.8964, df = 36.884, p-value = 0.06575
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.1245437  3.7594220
## sample estimates:
## mean in group 0 mean in group 1
##        21.02524        19.20780
```

From a visual inspection of the graph and the t-test we must conclude that although it is close, using a 95% confidence level we must conclude that there is no significant relationship between whether there was a bills game that day and the amount of tips made per hour.

```
sunday_data <- merged_data %>% filter(`Day of Week` == "Sun")

ggplot(sunday_data, aes(x = Date, y = tips_per_hour, color = factor(Bartender))) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
```

```
  labs(color = "Bartender", y='Average Inflation Adjusted Tips per Hour', x='Number of Days Since Start
  scale_color_manual(values = c("0" = "red", "1" = "blue"), labels = c("0" = "Serving", "1" = "Bartendi
```




```
model_server <- lm(tips_per_hour ~ Date, data = filter(sunday_data, Bartender == 0))
model_bartender <- lm(tips_per_hour ~ Date, data = filter(sunday_data, Bartender == 1))
summary(model_server)
```

```
##
## Call:
## lm(formula = tips_per_hour ~ Date, data = filter(sunday_data,
##     Bartender == 0))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0050  -3.0001   0.0292   3.1951  16.2942
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.245877   1.330760  15.214   <2e-16 ***
## Date        -0.003814   0.005838  -0.653    0.517
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.049 on 49 degrees of freedom
## Multiple R-squared:  0.008637,   Adjusted R-squared:  -0.0116
## F-statistic: 0.4269 on 1 and 49 DF,  p-value: 0.5166
```

```r
summary(model_bartender)
```

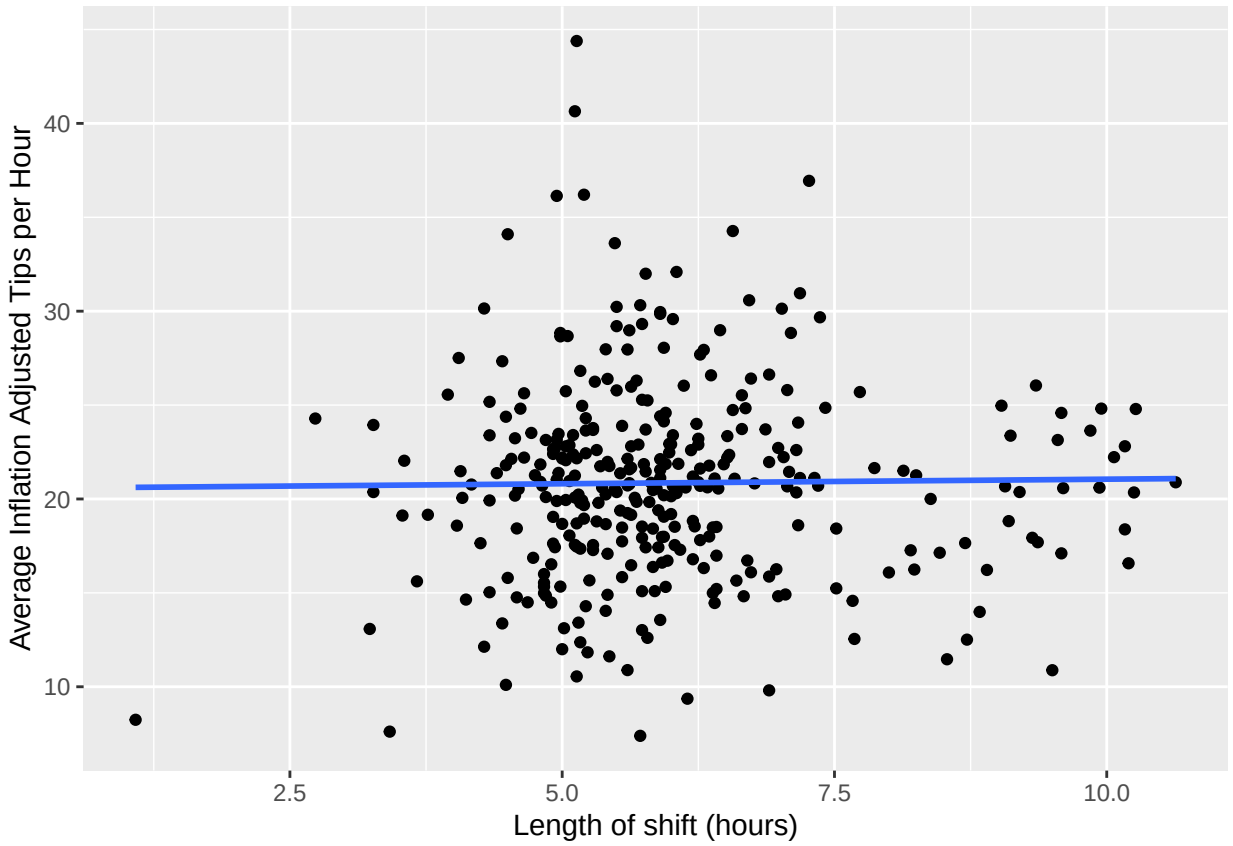
```
##
## Call:
## lm(formula = tips_per_hour ~ Date, data = filter(sunday_data,
##     Bartender == 1))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9148  -2.0006   0.1661   2.6812   9.3397
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.00343   14.62504   2.872   0.0111 *
## Date        -0.04866    0.03115  -1.562   0.1379
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.163 on 16 degrees of freedom
## Multiple R-squared:  0.1323, Adjusted R-squared:  0.07806
## F-statistic: 2.439 on 1 and 16 DF,  p-value: 0.1379
```

```r
t_test_bar <- t.test(tips_per_hour ~ Bartender, data = sunday_data)

t_test_bar
```

```
##
##  Welch Two Sample t-test
##
## data:  tips_per_hour by Bartender
## t = 0.18526, df = 28.164, p-value = 0.8544
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -2.699413  3.236397
## sample estimates:
## mean in group 0 mean in group 1
##        19.50929        19.24080
```

What this t-test tells us is that there is no significant relationship between average tips per hour on a Sunday from before and after she started working as a bartender on Sundays.

```r
ggplot(merged_data, aes(x = shift_length, y = tips_per_hour)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(y='Average Inflation Adjusted Tips per Hour', x='Length of shift (hours)')
```



The line of best fit appears to be almost perfectly flat, with a large spread on either side of the line suggesting that the length of the sift does not have any effect on the tips per hour of the shift

```
tips_model <- lm(tips_per_hour ~ Date + `Day of Week` + Bartender + bills_game + shift_length + precipi
summary(tips_model)
```

```
##
## Call:
## lm(formula = tips_per_hour ~ Date + `Day of Week` + Bartender +
##     bills_game + shift_length + precipitated + TAVG, data = merged_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2774  -3.1136  -0.2624   2.5030  21.4098
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      20.368191   1.971880  10.329  < 2e-16 ***
## Date              0.004384   0.002004   2.188  0.02940 *
## `Day of Week`Mon -3.562297   1.185487  -3.005  0.00287 **
## `Day of Week`Sat -0.970236   0.901094  -1.077  0.28241
## `Day of Week`Sun -3.137003   1.114349  -2.815  0.00518 **
## `Day of Week`Thu -1.429709   1.094299  -1.307  0.19231
## `Day of Week`Tue -1.282226   1.107880  -1.157  0.24798
## `Day of Week`Wed -2.361824   1.090969  -2.165  0.03113 *
## Bartender        -1.319896   1.616592  -0.816  0.41484
## bills_game        0.072842   1.215213   0.060  0.95224
## shift_length      0.209355   0.243984   0.858  0.39149
```

```
## precipitated      0.123550   0.617994   0.200  0.84167
## TAVG             -0.003752   0.017849  -0.210  0.83362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.141 on 322 degrees of freedom
## Multiple R-squared:  0.07585,    Adjusted R-squared:  0.04141
## F-statistic: 2.203 on 12 and 322 DF,  p-value: 0.01149
```

Here we have the linear model if we were to include all of the factors we looked at, however since we know many of them are not significant we should preform backward elimination to figure out which features are best to keep in the model and which would be best to eliminate.

```
reduced_tips_model <- step(tips_model, direction = "backward")
```

```
## Start:  AIC=1109.74
## tips_per_hour ~ Date + `Day of Week` + Bartender + bills_game +
##     shift_length + precipitated + TAVG
##
##                 Df Sum of Sq    RSS    AIC
## - bills_game     1      0.09 8511.5 1107.7
## - precipitated   1      1.06 8512.5 1107.8
## - TAVG           1      1.17 8512.6 1107.8
## - Bartender      1     17.62 8529.1 1108.4
## - shift_length   1     19.46 8530.9 1108.5
## <none>                       8511.5 1109.7
## - Date           1    126.52 8638.0 1112.7
## - `Day of Week`  6    405.51 8917.0 1113.3
##
## Step:  AIC=1107.74
## tips_per_hour ~ Date + `Day of Week` + Bartender + shift_length +
##     precipitated + TAVG
##
##                 Df Sum of Sq    RSS    AIC
## - precipitated   1      1.05 8512.6 1105.8
## - TAVG           1      1.23 8512.8 1105.8
## - Bartender      1     18.43 8530.0 1106.5
## - shift_length   1     19.51 8531.1 1106.5
## <none>                       8511.5 1107.7
## - Date           1    127.17 8638.7 1110.7
## - `Day of Week`  6    434.20 8945.8 1112.4
##
## Step:  AIC=1105.78
## tips_per_hour ~ Date + `Day of Week` + Bartender + shift_length +
##     TAVG
##
##                 Df Sum of Sq    RSS    AIC
## - TAVG           1      1.75 8514.3 1103.8
## - Bartender      1     17.72 8530.3 1104.5
## - shift_length   1     19.78 8532.4 1104.6
## <none>                       8512.6 1105.8
## - Date           1    126.50 8639.1 1108.7
## - `Day of Week`  6    439.15 8951.8 1110.6
##
## Step:  AIC=1103.85
```

```
## tips_per_hour ~ Date + `Day of Week` + Bartender + shift_length
##
##                 Df Sum of Sq    RSS    AIC
## - Bartender      1     16.49 8530.8 1102.5
## - shift_length   1     19.54 8533.9 1102.6
## <none>                       8514.3 1103.8
## - Date           1    125.82 8640.2 1106.8
## - `Day of Week`  6    438.33 8952.7 1108.7
##
## Step:  AIC=1102.5
## tips_per_hour ~ Date + `Day of Week` + shift_length
##
##                 Df Sum of Sq    RSS    AIC
## - shift_length   1     25.08 8555.9 1101.5
## <none>                       8530.8 1102.5
## - Date           1    109.53 8640.4 1104.8
## - `Day of Week`  6    555.05 9085.9 1111.6
##
## Step:  AIC=1101.48
## tips_per_hour ~ Date + `Day of Week`
##
##                 Df Sum of Sq    RSS    AIC
## <none>                       8555.9 1101.5
## - Date           1     91.21 8647.1 1103.0
## - `Day of Week`  6    540.23 9096.2 1110.0
```

```
reduced_tips_model
```

```
##
## Call:
## lm(formula = tips_per_hour ~ Date + `Day of Week`, data = merged_data)
##
## Coefficients:
##      (Intercept)              Date  `Day of Week`Mon  `Day of Week`Sat
##        21.701714          0.003399         -3.794690         -0.849540
## `Day of Week`Sun  `Day of Week`Thu  `Day of Week`Tue  `Day of Week`Wed
##        -3.162552         -1.534232         -1.403278         -2.534501
```

```
reduced_model <- lm(formula = tips_per_hour ~ Date + `Day of Week`, data = merged_data)
summary(reduced_model)
```

```
##
## Call:
## lm(formula = tips_per_hour ~ Date + `Day of Week`, data = merged_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.0714  -3.0266  -0.2007   2.5073  21.3122
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      21.701714   0.796325  27.252  < 2e-16 ***
## Date              0.003399   0.001821   1.867  0.06278 .
## `Day of Week`Mon -3.794690   1.147123  -3.308  0.00104 **
## `Day of Week`Sat -0.849540   0.884169  -0.961  0.33735
```

```
## `Day of Week`Sun -3.162552   0.884160  -3.577  0.00040 ***
## `Day of Week`Thu -1.534232   1.083527  -1.416  0.15774
## `Day of Week`Tue -1.403278   1.093397  -1.283  0.20026
## `Day of Week`Wed -2.534501   1.062789  -2.385  0.01766 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.115 on 327 degrees of freedom
## Multiple R-squared:  0.07103,    Adjusted R-squared:  0.05114
## F-statistic: 3.572 on 7 and 327 DF,  p-value: 0.00102
```

This reduced model goes to show that most of the factors looked into have no significant effect on tips made per hour, with only day of week and the intercept having significance at a 95% confidence level, and with an adjusted R-squared value of 0.05114 the only thing we can be certain of is that this model would not be a good predictor of tips for any given day.

Conclusion

What this project has taught me more than anything is to be skeptical of any anecdotes or assumptions made without supporting evidence. Coming into this project I had a firm belief that winter was a better time to be a server at Olive Garden than summer, and that the weather had some effect on the amount of customers who would show up, and therefore tips that would be made, and I was near confident that the shift to bartending was an overall bad move, however one by one each of these assumptions failed to hold up when the data was scrutinized. The main insight to gather is that serving is a highly random profession, with a standard deviation that is 25% of the median, you can expect that there will be lots of really bad shifts and really good shifts and that these will just need to be taken in stride as from the factors I was able to analyze the only way to improve your chances of making more money is to gain more experiece, and to avoid working on Sunday and Monday.