Final Report for Intro to Data Science

Hailey Skoglund, Gus Lipkin, Alex Pierstorff, and Marli

Write-up

Introduction

We have saved several datasets that explore data collected on Walt Disney World. The metadata data frame contains observations regarding the temperature, levels of precipitation, and parade times. Next, we have a collection of 14 rides, each in individual data frames that contain data observations regarding wait times for the given ride. Lastly, rides_df is a large dataframe containing all the individual rides data frames available. rides_df contains information including the name of the ride, the ride's wait time, and which park it is located in for each of the 14 rides.

Dataset Description

Our chosen dataset focuses on the affect that environmen¬¬tal conditions have on Walt Disney World attraction wait times. We explore this relationship by revealing trends between weather conditions such as temperature and precipitation levels with standby wait times. Furthermore, we analyzed the effect of certain in-park events on the ride wait times including parades, fireworks, and holiday events. Our goal was to compare data between multiple dataframes so that only relevant data is used. Each ride has its own unique dataframe and is also part of a larger dataframe containing all rides recorded for all of the parks. Furthermore, there is a metadata dataframe that has relevant data for each of the four parks. With this information, we wanted to create a new dataframe that has each ride as the row, the park it is located in, the ride opening data, if it has a splash aspect, and if it is indoors.

Research Questions

When analyzing this data, we chose to explore the following research questions: - Does hot weather increase wait times for rides with a "splash_aspect"? - Does rainy weather cause an increase in wait times for "indoor_rides"? - What is the busiest Day of the week (lubridate data) for tourists to visit WDW? - What affect do parades have on the wait times of rides? Through the analysis of this data, our goal is to create a structured travel plan for tourists to minimize the amount of time spent waiting in line during their Disney vacation.

Revised Results From The Proposal And Data Analysis

After submitting our initial Github repo via canvas, our instructor approved our topic and datasets chosen. Throughout this process, our team has displayed phenomenal communication, accountability, and equal responsibility for this project. We met often during office hours and had regimented group meetings on Microsoft Teams three times a week since the project was assigned. With this feedback, our team was able to develop a thorough understanding and complete analysis of our chosen data sets. In the following sections, we will evaluate each data analysis in detail.

```r
#Load the libraries needed
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library(forcats)
library(vistime)
```

```r
#Import metadata.csv which contains metadata on the parks
metadata <- read_csv("../data/metadata.csv")
```

```
## Rows: 3226 Columns: 190

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (46): DATE, WDW_TICKET_SEASON, SEASON, HOLIDAYN, WDWTICKETSEASON, WDWRa...
## dbl  (93): DAYOFWEEK, DAYOFYEAR, WEEKOFYEAR, MONTHOFYEAR, YEAR, HOLIDAYPX, H...
## lgl   (2): AKPRDDT2, AKFIREN
## time (49): SUNSET_WDW, MKOPEN, MKCLOSE, MKEMHOPEN, MKEMHCLOSE, MKOPENYEST, M...

##
## 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
#Format the date column to mdy format
metadata$DATE <- format(as.POSIXct(mdy(metadata$DATE) + 1, format = '%m/%d/%Y %H:%M:%S'), format='%m/%d/

#Split the metadata into five dataframes. One for each park and one for Disney World as a whole
wdw_metadata <- metadata %>%
  select(DATE, SEASON, HOLIDAYPX, HOLIDAYN, WDWMAXTEMP, WDWMINTEMP, WDWMEANTEMP, HOLIDAYJ, WEATHER_WDWPI
mk_metadata <- metadata %>%
  select(DATE, MKOPEN, MKCLOSE, MKEMHOPEN, MKEMHCLOSE, MKPRDDT1, MKPRDDT2, MKPRDNT1, MKPRDNT2, MKFIRET1
ep_metadata <- metadata %>%
  select(DATE, EPOPEN, EPCLOSE, EPEMHOPEN, EPEMHCLOSE, EPFIRET1, EPFIRET2)
hs_metadata <- metadata %>%
  select(DATE, HSOPEN, HSCLOSE, HSEMHOPEN, HSEMHCLOSE, HSPRDDT1, HSFIRET1, HSFIRET2, HSSHWNT1, HSSHWNT2)
ak_metadata <- metadata %>%
  select(DATE, AKOPEN, AKCLOSE, AKEMHOPEN, AKEMHCLOSE, AKPRDDT1, AKPRDDT2, AKSHWNT1, AKSHWNT2)

#Correct holiday proximity so that the day before and after is day 1 and not day 2 </br>
#Shorten Martin Luther King Junior Day season to MLK Day
wdw_metadata$HOLIDAYPX <- ifelse(wdw_metadata$HOLIDAYPX > 0, wdw_metadata$HOLIDAYPX - 1, wdw_metadata$HO
wdw_metadata$SEASON <- ifelse(wdw_metadata$SEASON == "MARTIN LUTHER KING JUNIOR DAY", "MLK DAY", wdw_me

#Create dataframes for temperature categories based on the mean temperature in Disney World and then bi
xhot_days <- wdw_metadata %>%
  select(DATE, WDWMEANTEMP) %>%
  filter(WDWMEANTEMP >= 85) %>%
  mutate(temp_cat = "xhot_days")

hot_days <- wdw_metadata %>%
  select(DATE, WDWMEANTEMP) %>%
  filter(WDWMEANTEMP < 85 & WDWMEANTEMP >= 79.8) %>%
  mutate(temp_cat = "hot_days")

normal_days <- wdw_metadata %>%
  select(DATE, WDWMEANTEMP) %>%
  filter(WDWMEANTEMP >= 71.3 & WDWMEANTEMP < 79.8) %>%
  mutate(temp_cat = "normal_days")

cool_days <- wdw_metadata %>%
  select(DATE, WDWMEANTEMP) %>%
  filter(WDWMEANTEMP >= 62.8 & WDWMEANTEMP < 71.3) %>%
  mutate(temp_cat = "cool_days")

xcool_days <- wdw_metadata %>%
  select(DATE, WDWMEANTEMP) %>%
  filter(WDWMEANTEMP < 62.8) %>%
  mutate(temp_cat = "xcool_days")

temp_days <- bind_rows(xhot_days, hot_days, normal_days, cool_days, xcool_days)

#Create various vectors that will be used later
park_colors <- c("darkgreen", "cornflowerblue", "chocolate1", "blueviolet")
temp_colors <- c("red", "orange", "yellow", "blue", "cyan")
```

```r
temp_list_names <- c("xhot_days", "hot_days", "normal_days", "cool_days", "xcool_days")
days_of_week <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
```

```r
#Load all the ride data csvs to their own dataframe
dwarfs_train <- read_csv("../data/7_dwarfs_train.csv")
```

```
## Rows: 319098 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime
```

```
##
## 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
alien_saucers <- read_csv("../data/alien_saucers.csv")
```

```
## Rows: 77145 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime
```

```
##
## 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
dinosaur <- read_csv("../data/dinosaur.csv")
```

```
## Rows: 288715 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime
```

```
##
## 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
expedition_everest <- read_csv("../data/expedition_everest.csv")
```

```
## Rows: 312869 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime

##
## 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.
```

```
flight_of_passage <- read_csv("../data/flight_of_passage.csv")
```

```
## Rows: 129643 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime

##
## 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.
```

```
kilimanjaro_safaris <- read_csv("../data/kilimanjaro_safaris.csv")
```

```
## Rows: 292022 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime

##
## 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.
```

```
navi_river <- read_csv("../data/navi_river.csv")
```

```
## Rows: 128580 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime

##
## 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
pirates_of_caribbean <- read_csv("../data/pirates_of_caribbean.csv")
```

```
## Rows: 368853 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime

##
## 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
rock_n_rollercoaster <- read_csv("../data/rock_n_rollercoaster.csv")
```

```
## Rows: 337659 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime

##
## 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
slinky_dog <- read_csv("../data/slinky_dog.csv")
```

```
## Rows: 79825 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime

##
## 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
soarin <- read_csv("../data/soarin.csv")
```

```
## Rows: 338988 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime
```

```
##
## 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.
```

```
spaceship_earth <- read_csv("../data/spaceship_earth.csv")
```

```
## Rows: 336875 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime
```

```
##
## 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.
```

```
splash_mountain <- read_csv("../data/splash_mountain.csv")
```

```
## Rows: 332737 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime
```

```
##
## 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.
```

```
toy_story_mania <- read_csv("../data/toy_story_mania.csv")
```

```
## Rows: 341633 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl  (2): SPOSTMIN, SACTMIN
## dttm (1): datetime
```

```
##
## 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
#Make a list of all the ride dataframes
rides <- list("dwarfs_train" = dwarfs_train,
              "alien_saucers" = alien_saucers,
              "dinosaur" = dinosaur,
              "expedition_everest" = expedition_everest,
              "flight_of_passage" = flight_of_passage,
              "kilimanjaro_safaris" = kilimanjaro_safaris,
              "navi_river" = navi_river,
              "pirates_of_caribbean" = pirates_of_caribbean,
              "rock_n_rollercoaster" = rock_n_rollercoaster,
              "slinky_dog" = slinky_dog,
              "soarin" = soarin,
              "spaceship_earth" = spaceship_earth,
              "splash_mountain" = splash_mountain,
              "toy_story_mania" = toy_story_mania)
```

```r
#Create a new time column
#Create a new column for the ride name
#Get rid of wait times that are not recorded
for (i in 1:14) {
  rides[[i]] <- rides[[i]] %>%
      mutate(time = format(ymd_hms(datetime), "%H:%M:%S"))
  rides[[i]] <- rides[[i]] %>%
    mutate(ride_name = as.factor(names(rides[i])))
  rides[[i]] <- rides[[i]] %>%
    filter(SPOSTMIN != -999 | is.na(SPOSTMIN))
}
```

```r
#Manually create a dataframe for ride metadata
ride_name <- c("dwarfs_train", "alien_saucers", "dinosaur", "expedition_everest", "flight_of_passage",
               "kilimanjaro_safaris", "navi_river", "pirates_of_caribbean", "rock_n_rollercoaster", "sli
               "soarin", "spaceship_earth", "splash_mountain", "toy_story_mania")
open_date <- as.POSIXct(c("2014/05/28", "2018/06/30", "1998/04/22", "2006/04/09", "2017/05/27",
                          "1998/04/22", "2017/05/27", "1973/12/17", "1999/07/29", "2018/06/30",
                          "2005/05/15", "1982/10/01", "1992/07/17", "2008/05/31"))
splash <- c(FALSE, FALSE, FALSE, FALSE, TRUE,
            FALSE, FALSE, TRUE,  FALSE, FALSE,
            FALSE, FALSE, TRUE,  FALSE)
indoor <- c(FALSE, FALSE, TRUE,  FALSE, TRUE,
            FALSE, TRUE,  TRUE,  TRUE,  FALSE,
            TRUE,  TRUE,  FALSE, TRUE)
age_hierarchy <- c(10, 13,  4,  8, 11,
                   5, 12,  1,  6, 14,
                   7,  2,  3,  9)
park <- c("mk", "hs", "ak", "ak", "ak",
          "ak", "ak", "mk", "hs", "hs",
          "ep", "ep", "mk", "hs")
ride_metadata <- data.frame(ride_name, open_date, age_hierarchy, splash, indoor, park)
```

```r
#Combine all ride dataframes into one large dataframe
rides_df <- rides[[1]]
for (i in 2:14) {
  rides_df <- rbind(rides_df, rides[[i]])
```

```
}
```

```
#Create temps_df by combining rides_df and temp_days from above
temps_df <- rides_df %>%
  inner_join(temp_days, by = c("date" = "DATE")) %>%
  group_by(ride_name, temp_cat) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'ride_name'. You can override using the `.groups` argument.
```

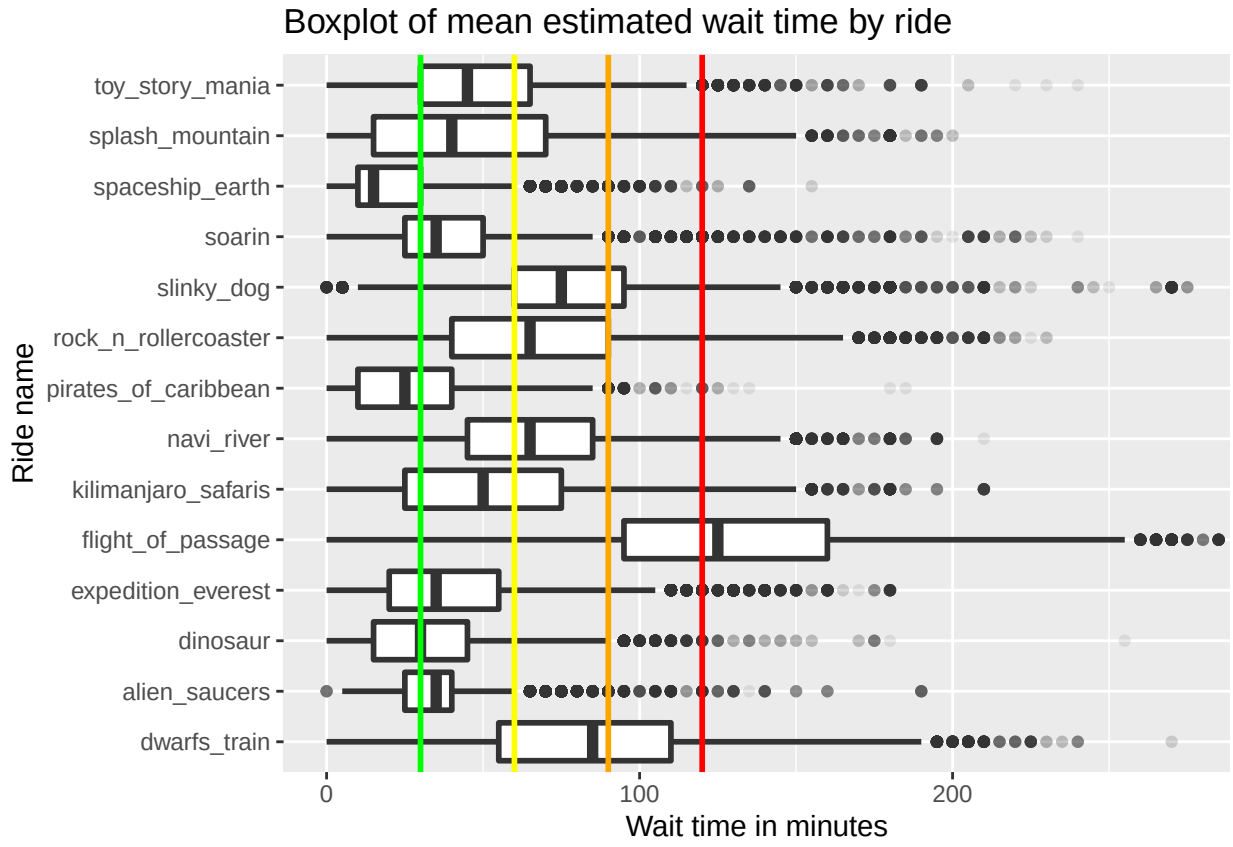
Est__time__box__plot.rmd analysis

First, we visualized the average wait time of each ride in the data set. We organized each of the rides by putting them into groups based on the park that they are located in. In this visualization, it can be observed that the data set contains five rides located in Disney's Animal Kingdom, two rides from Epcot, four rides in Hollywood Studios, and three rides found in Magic Kingdom. Immediately, it is clear that Animal Kingdom's Flight of passage ride has the longest average wait time by far. The wait time for Flight of Passage exceeds 125 minutes, making it the longest wait time for any ride in all of the Disney Parks. Other lengthy wait times include Snow White and the Seven Dwarfs Mine Train ride located in Magic Kingdom and Slinky Dog Dash located in Hollywood Studios. One may be wondering, what do all of these rides have in common? To answer this question, we turned to the opening dates of each of these rides. It can be concluded that Animal Kingdom's Flight of Passage, Magic Kingdom's Snow White and the Seven Dwarves Mine Train and Hollywood Studio's Slinky Dog Dash were all opened very recently and are each Park's newest addition to their rides available. Flight of Passage was initially opened to the public with the grand opening of Pandora on May 27, 2017. Similarly, the Seven Dwarfs Mine Train open date was May 28, 2014, which is the most recent out of the other opening dates recorded in this dataset from magic kingdom. Moreover, Hollywood studios opened Slinky Dog Dash with the grand opening of Toy Story Land on June 30, 2018. From these conclusions, we recommend avoiding the newest released rides in each park to make the most of your magical vacation at Walt Disney World instead of spending that precious time waiting in long lines.

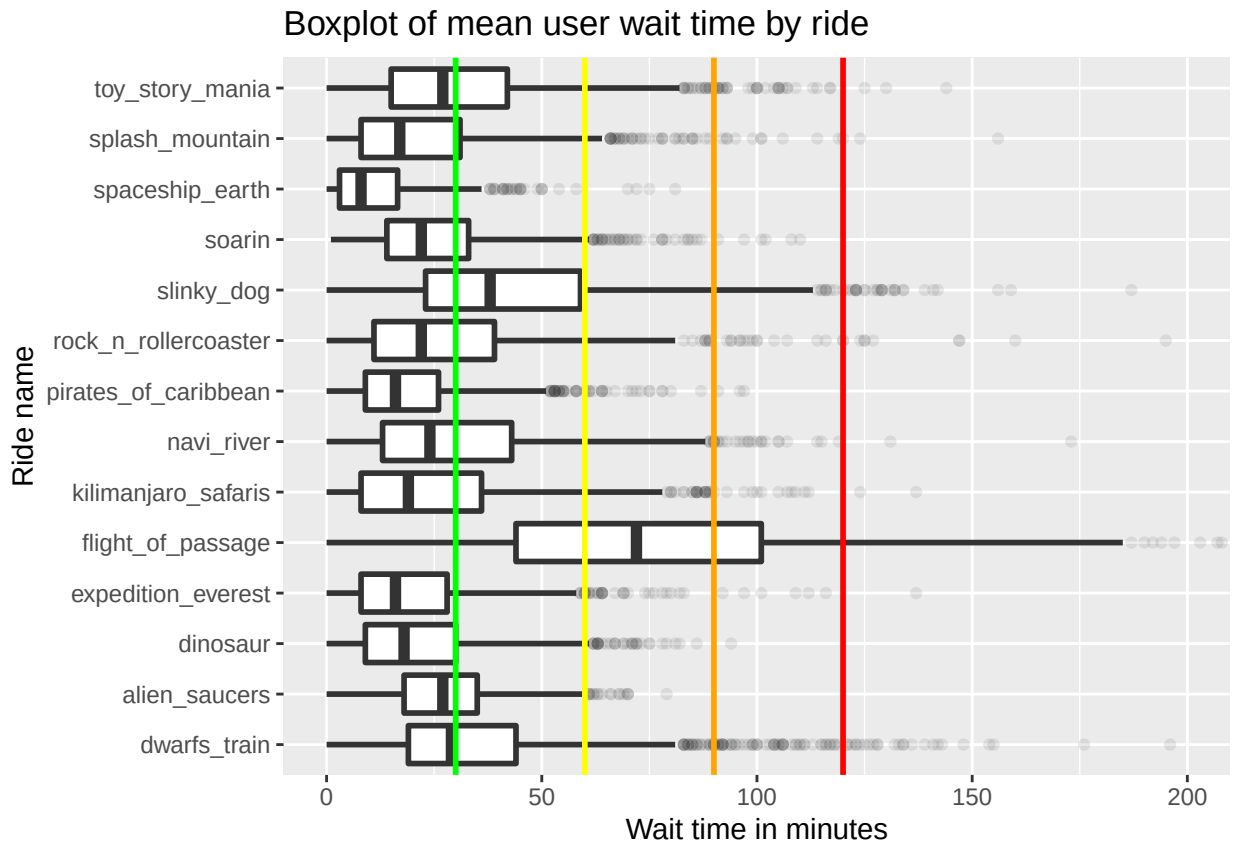
Next, we analyzed how the wait times for each ride vary over the week. Within this facet-wrapped visualization, a small dip in the middle of each week can be observed. Therefore, it can be concluded that, as a whole, there are shorter wait times in the middle of the week rather than on a weekend. On average, Wednesday is the least busy day of the week which is characterized by the shortest ride wait times. In contrast, Saturday is the busiest day of the week with the longest average wait times. From this conclusion, we recommend that tourists plan their Disney get-away on a Wednesday and avoid weekend trips if possible. Saturday is also observed as having the longest average wait time for all rides throughout the year. Saturday's longest wait times however, historically occur on the last week of December. Therefore, it is advised to avoid a weekend winter holiday on a Saturday in December to avoid waiting in extremely long lines. Historically, the month of September is characterized by unusually low average ride wait times among all parks. This dramatic dip in the influx of Disney guests in September is even mentioned in the dataset collected by Disney as a season called the "September Low." From this information we recommend visiting Disney World during September, specifically, the first week in September as it has consistently the lowest wait time throughout that month.

We determined that the fifth week in December is the busiest week of the year and has the longest wait times all year. There is also an interesting spike in the average wait times during the last week of February. On the other hand, there is a noticeable decline in wait times during the second week of September giving it the shortest average wait times all year. Lastly, it was observed that Disney's Hollywood Studios has the highest average estimated wait time by park as a portion of total wait time by day. This means that, on the whole, Hollywood Studios has the greatest wait times in comparison to the other 3 Disney parks. This is most likely due to the new release of Toy Story Land in 2018.

```
#Vertical lines mark 30, 60, 90, and 120 minutes
rides_df %>%
  filter(datetime > as.POSIXct("2018-06-30")) %>%
  ggplot(aes(x = SPOSTMIN, y = ride_name)) +
  geom_boxplot(na.rm = TRUE, outlier.shape = "circle", outlier.alpha = .1, size = 1) +
  coord_cartesian(xlim = c(0,275)) +
  geom_vline(xintercept = c(30, 60, 90, 120), color = c("green", "yellow", "orange", "red"), size = 1) +
  labs(title = "Boxplot of mean estimated wait time by ride") +
  xlab("Wait time in minutes") +
  ylab("Ride name")
```




```
#Vertical lines mark 30, 60, 90, and 120 minutes
rides_df %>% group_by(ride_name) %>%
  filter(datetime > as.POSIXct("2018-06-30")) %>%
  ggplot(aes(x = SACTMIN, y = ride_name)) +
  geom_boxplot(na.rm = TRUE, outlier.shape = "circle", outlier.alpha = .1, size = 1) +
  coord_cartesian(xlim = c(0, 200)) +
  geom_vline(xintercept = c(30, 60, 90, 120), color = c("green", "yellow", "orange", "red"), size = 1) +
  labs(title = "Boxplot of mean user wait time by ride") +
  xlab("Wait time in minutes") +
  ylab("Ride name")
```



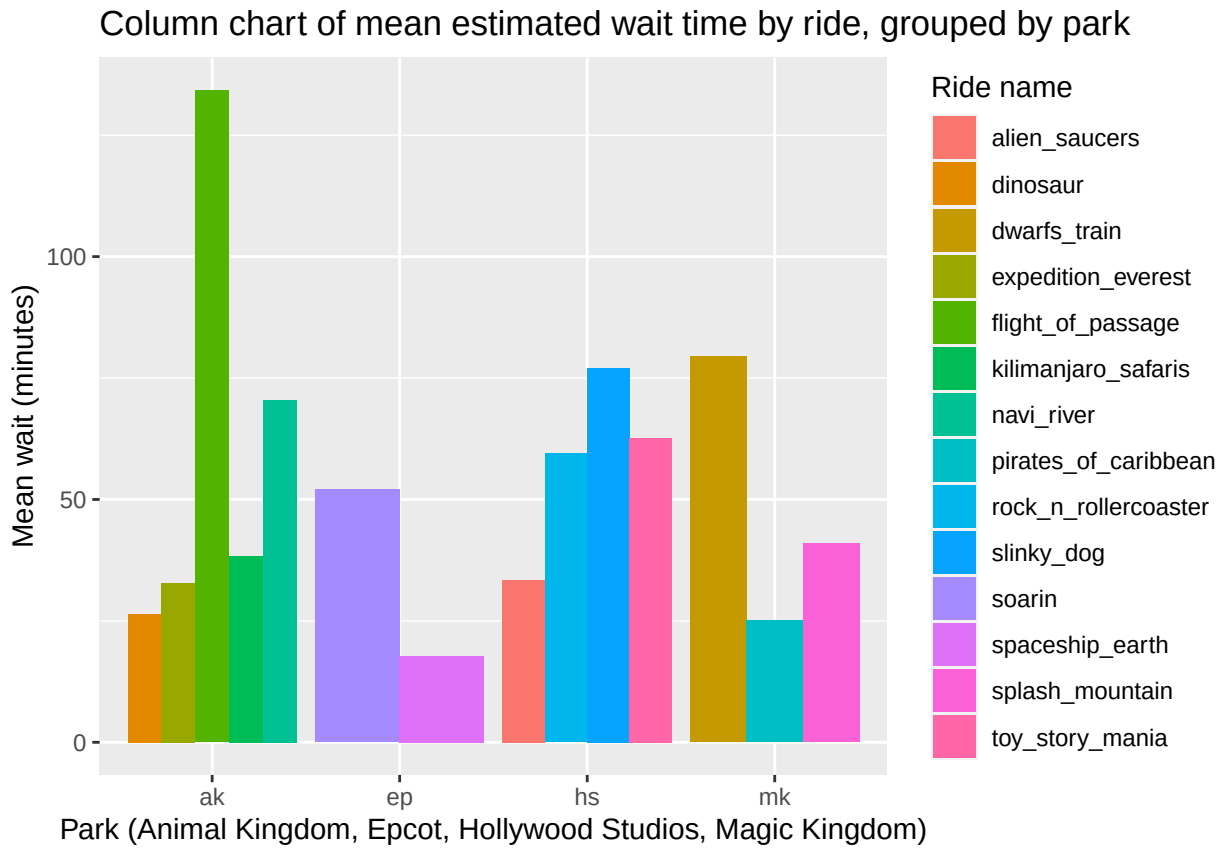

```
park_averages <- rides_df %>%
  inner_join(ride_metadata) %>%
  group_by(park) %>%
  summarise(park_averages = mean(SPOSTMIN, na.rm = TRUE))
```

```
## Joining, by = "ride_name"
```

```
rides_df %>%
  inner_join(ride_metadata) %>%
  mutate(weekday = weekdays(datetime)) %>%
  group_by(ride_name, park) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = park, y = mean_wait, fill = ride_name), position = "dodge") +
  labs(title = "Column chart of mean estimated wait time by ride, grouped by park", fill = "Ride name") +
  xlab("Park (Animal Kingdom, Epcot, Hollywood Studios, Magic Kingdom)") +
  ylab("Mean wait (minutes)")
```

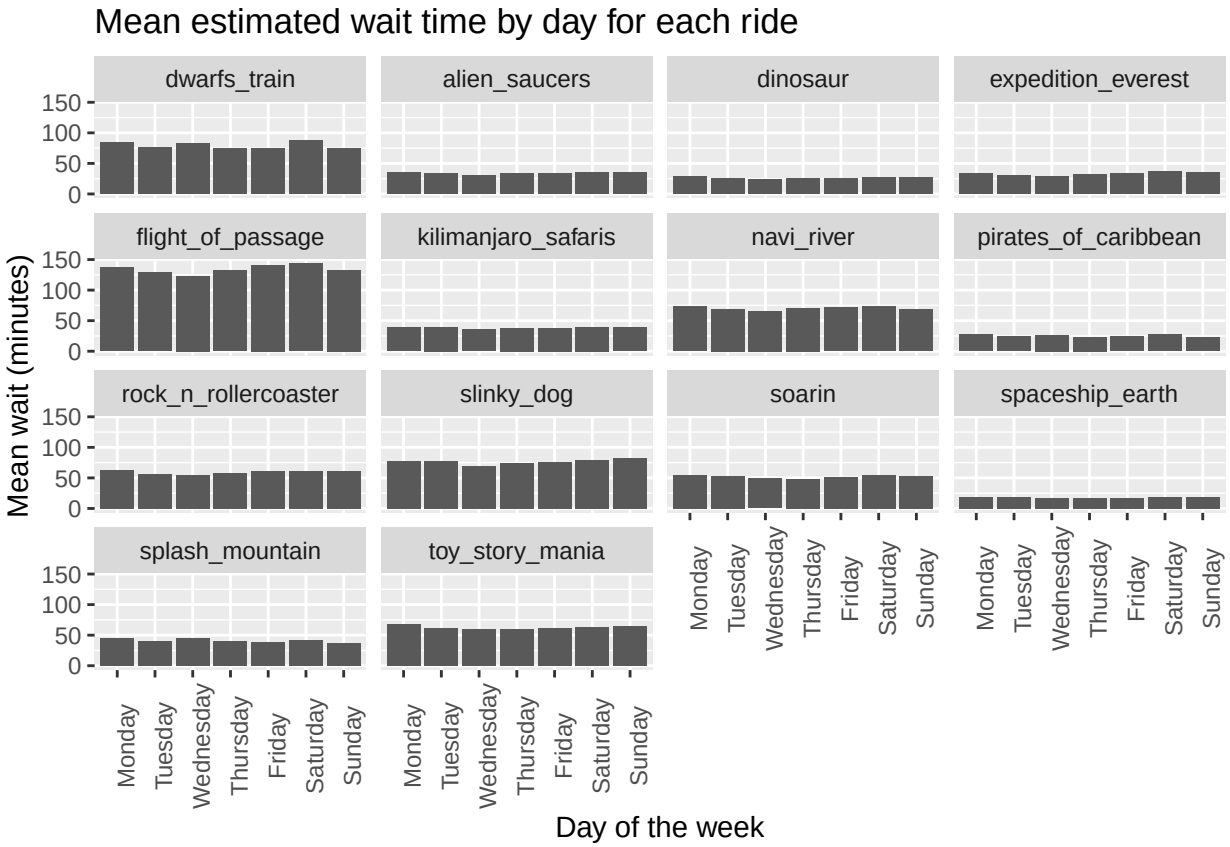
```
## Joining, by = "ride_name"
```

```
## `summarise()` has grouped output by 'ride_name'. You can override using the `.groups` argument.
```



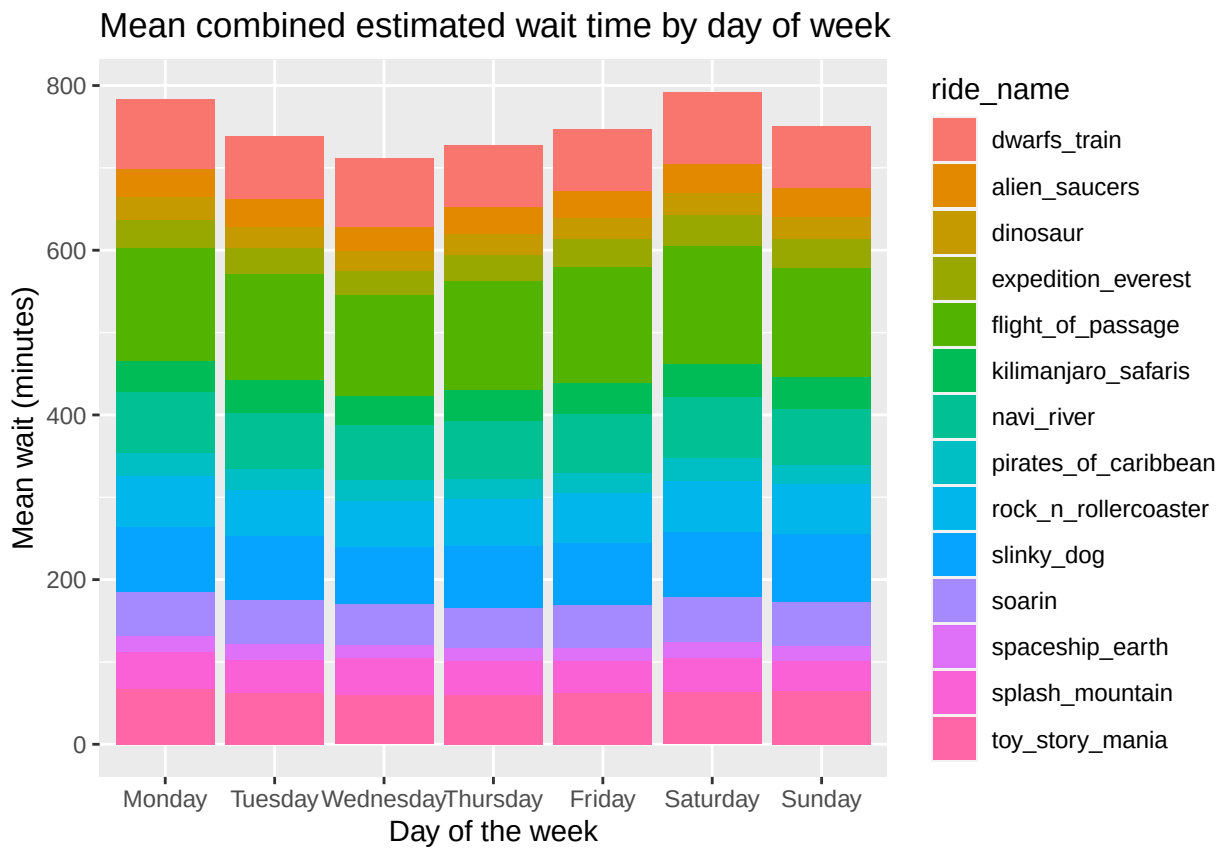

```
#All charts use the same scale
rides_df %>%
  mutate(weekday = weekdays(datetime)) %>%
  group_by(weekday, ride_name) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = ordered(weekday, levels = days_of_week), y = mean_wait)) +
  labs(title = "Mean estimated wait time by day for each ride") +
  xlab("Day of the week") +
  ylab("Mean wait (minutes)") +
  theme(axis.text.x = element_text(angle = 90)) +
  facet_wrap(~ ride_name)
```

```
## `summarise()` has grouped output by 'weekday'. You can override using the `.groups` argument.
```




```
rides_df %>%
  mutate(weekday = weekdays(datetime)) %>%
  group_by(weekday, ride_name) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot(aes(x = ordered(weekday, levels = days_of_week), y = mean_wait, fill = ride_name)) +
  geom_col() +
  labs(title = "Mean combined estimated wait time by day of week") +
  xlab("Day of the week") +
  ylab("Mean wait (minutes)")
```

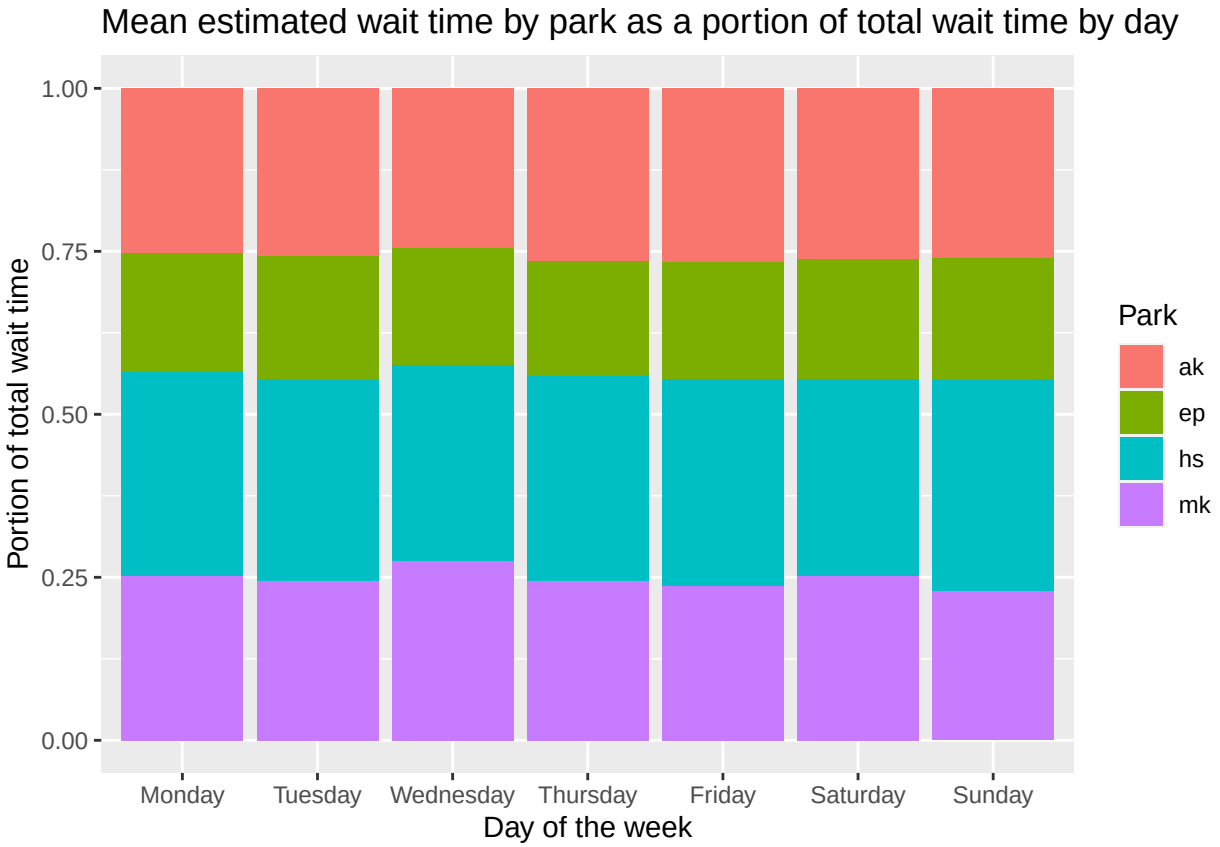
```
## `summarise()` has grouped output by 'weekday'. You can override using the `.groups` argument.
```




```
rides_df %>%
  inner_join(ride_metadata) %>%
  mutate(weekday = weekdays(datetime)) %>%
  group_by(weekday, park) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = ordered(weekday, levels = days_of_week), y = mean_wait, fill = park), position = "fi
  labs(title = "Mean estimated wait time by park as a portion of total wait time by day", fill = "Park"
  xlab("Day of the week") +
  ylab("Portion of total wait time")
```

```
## Joining, by = "ride_name"
```

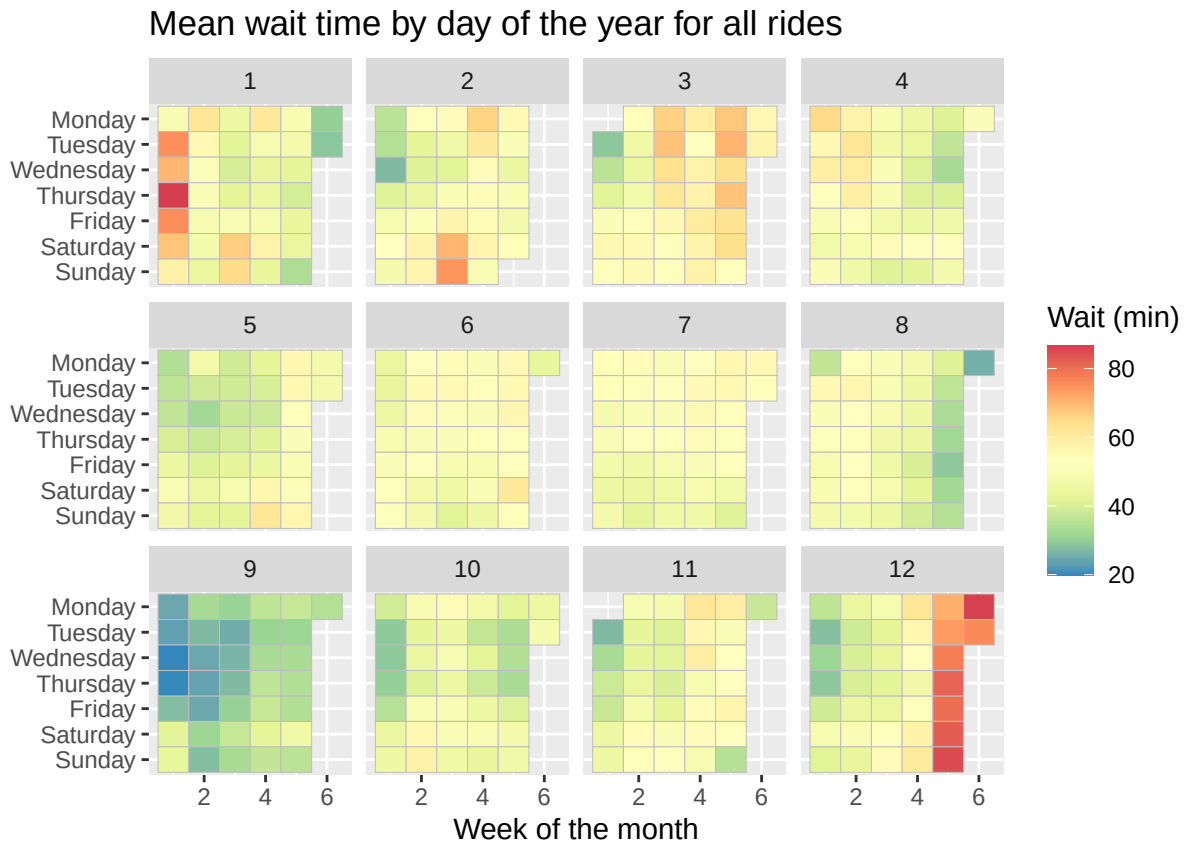
```
## `summarise()` has grouped output by 'weekday'. You can override using the `.groups` argument.
```



Many months are shown to have five weeks here. This is because across all years studied, there were times where those months did have days into a fifth week.

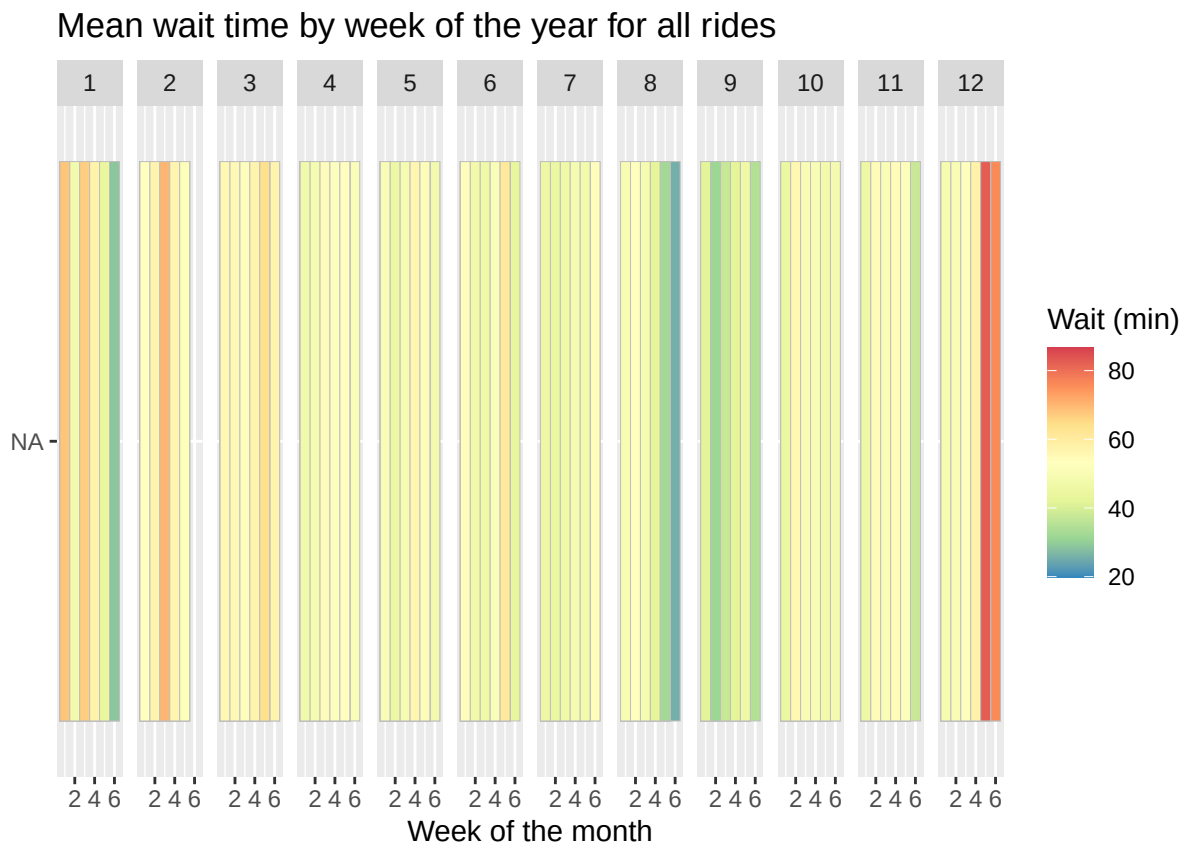
```
rides_df %>%
  mutate(year = year(as.POSIXct(mdy(date) + 1)), month = month(as.POSIXct(mdy(date) + 1)), day = weekday
         monthweek = ifelse(wday(mdy(date), week_start = 1) < wday(floor_date(mdy(date), "month"), week
                            ceiling(day((mdy(date))) / 7) + 1,
                            ceiling(day((mdy(date))) / 7))) %>%
  group_by(month, day, monthweek) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot(aes(x = monthweek, y = ordered(day, levels = rev(days_of_week)), fill = mean_wait)) +
  geom_tile(color = "grey") +
  facet_wrap(~month) +
  #scale_fill_gradient2(low = "blue", mid = "orange", high = "red", midpoint = 60) +
  scale_fill_distiller(palette = "Spectral") +
  labs(fill = "Wait (min)", title = "Mean wait time by day of the year for all rides", x = "Week of the
```

```
## `summarise()` has grouped output by 'month', 'day'. You can override using the `.groups` argument.
```




```
rides_df %>%
  mutate(year = year(as.POSIXct(mdy(date) + 1)), month = month(as.POSIXct(mdy(date) + 1)), day = wday(as
         monthweek = ifelse(wday(mdy(date), week_start = 1) < wday(floor_date(mdy(date), "month"), week
                            ceiling(day((mdy(date))) / 7) + 1,
                            ceiling(day((mdy(date))) / 7))) %>%
  group_by(month, day, monthweek) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot(aes(x = monthweek, y = ordered(day, levels = days_of_week), fill = mean_wait)) +
  geom_tile(color = "grey") +
  facet_grid(~month) +
  #scale_fill_gradient2(low = "blue", mid = "orange", high = "red", midpoint = 60) +
  scale_fill_distiller(palette = "Spectral") +
  labs(fill = "Wait (min)", title = "Mean wait time by week of the year for all rides", x = "Week of th
```

```
## `summarise()` has grouped output by 'month', 'day'. You can override using the `.groups` argument.
```



Hot_weather.rmd analysis

Hot_weather.rmd explored the relationship between average temperature levels in the Disney parks and ride wait times. From this data, we constructed several data visualizations that displayed the temperatures that were reported on each week of the month over a specified time period. Every Floridian knows how scorching the temperatures can get in the heat of the summer, so we decided to analyze how this affected the wait times specifically in indoor rides or rides with a splash or water aspect. Our analysis revealed several intriguing observations. First, we made a temperature classification system based on the average temperature peaks and troughs which gave us five distinct categories: very cold (xcool_days), cool (cool_days), comfortable (normal_days), warm (hot_days), and very hot (xhot_days). From this, we constructed an interactive graph that categorized each of the temperatures into categories based on their temperatures. We then created a visualization displaying the average estimated wait time for each ride at a given temperature. This interesting analysis revealed that guests prefer to ride indoor rides on cooler days and splash-aspect on hotter days. This may be because guests seek shelter from these varying harsh outdoor conditions.

```
#Create a chart showing the temperature buckets
temp_illustration <- data.frame(bucket = c("x < 62.8", "62.8 <= x < 71.3", "71.3 <= x < 79.8", "79.8 >=
                                name = c("xcool_days", "cool_days", "normal_days", "hot_days", "xhot_da
                                start = c("32-01-01", "62-09-18", "71-03-18", "79-09-18", "85-01-01"),
                                end = c("62-09-18", "71-03-18", "79-09-18", "85-01-01", "100-01-01"),
                                color = c("cyan", "blue", "yellow", "orange", "red"),
                                optimize_y = TRUE)


vistime(temp_illustration, groups = "name", events = "bucket", title = "Temperature buckets")
```

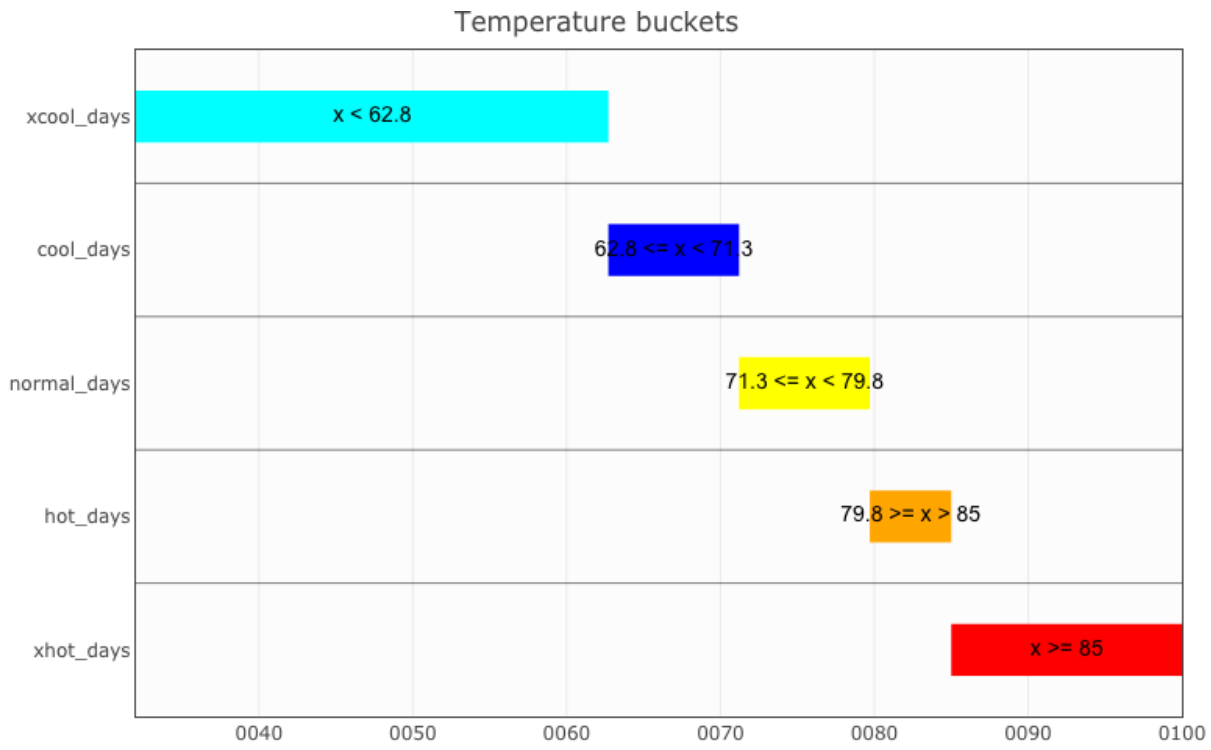
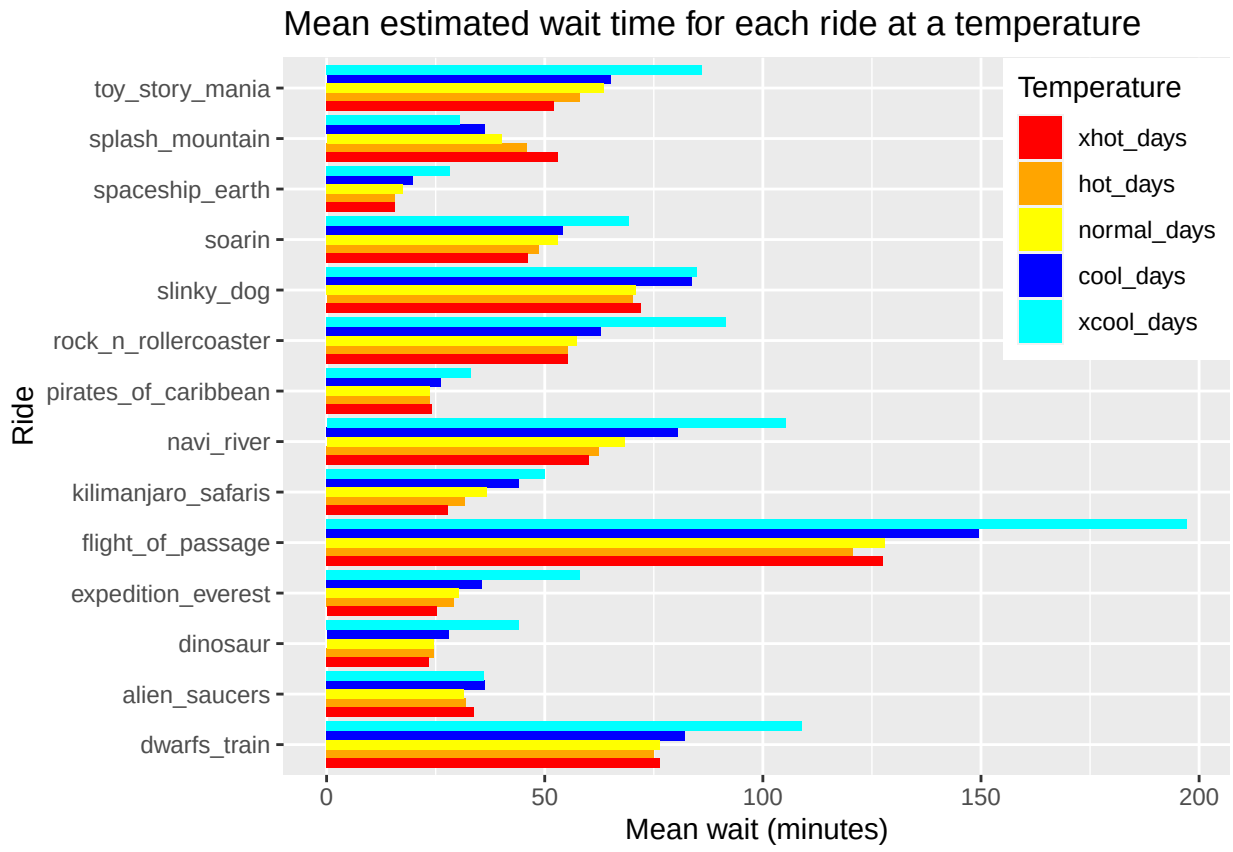


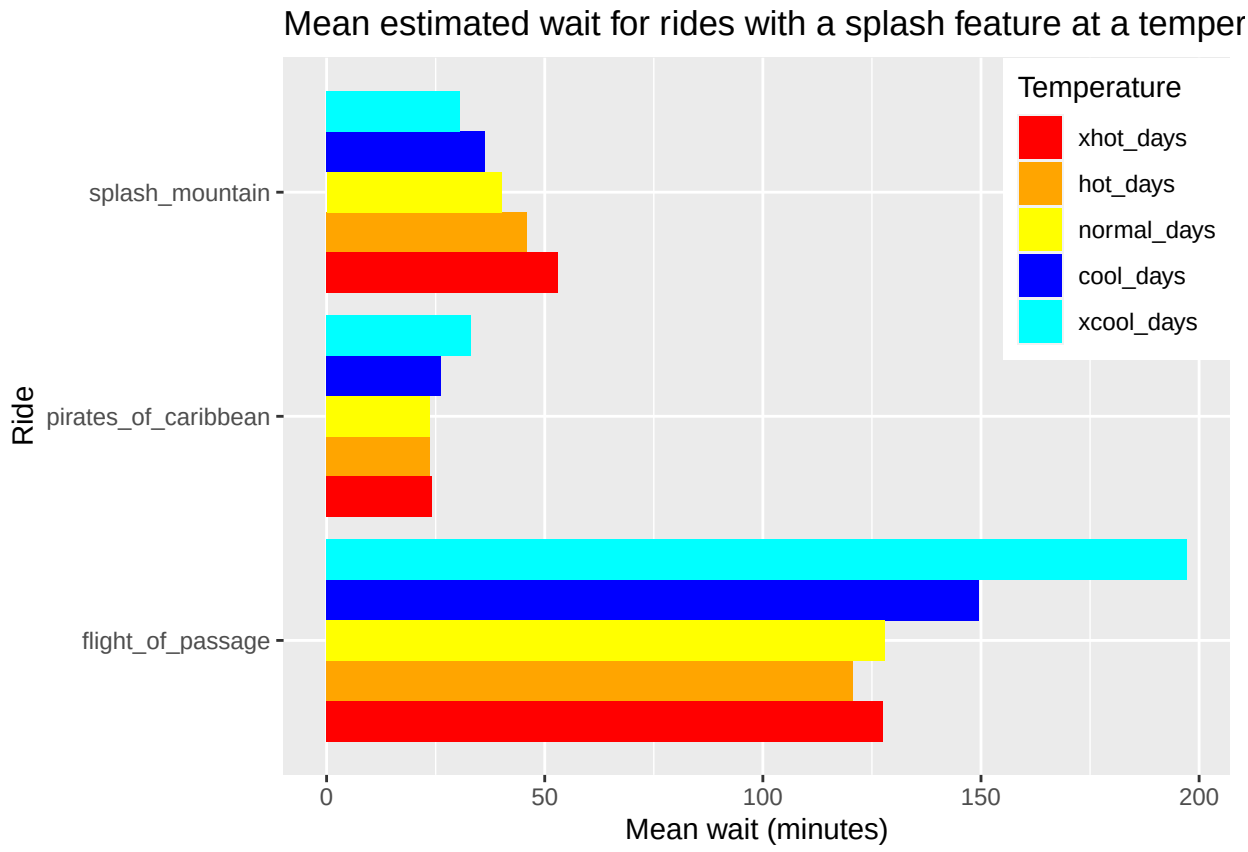
Figure 1: Temperature Buckets

```
temps_df %>%
  group_by(ride_name) %>%
  ggplot() +
  geom_col(aes(x = ride_name, y = mean_wait, fill = fct_relevel(temp_cat, temp_list_names)), position =
  scale_fill_manual(values = temp_colors) +
  coord_flip() +
  labs(fill = "Temp cat") +
  theme(legend.justification=c(1,1), legend.position=c(1,1)) +
  labs(title = "Mean estimated wait time for each ride at a temperature", fill = "Temperature") +
  ylab("Mean wait (minutes)") +
  xlab("Ride")
```



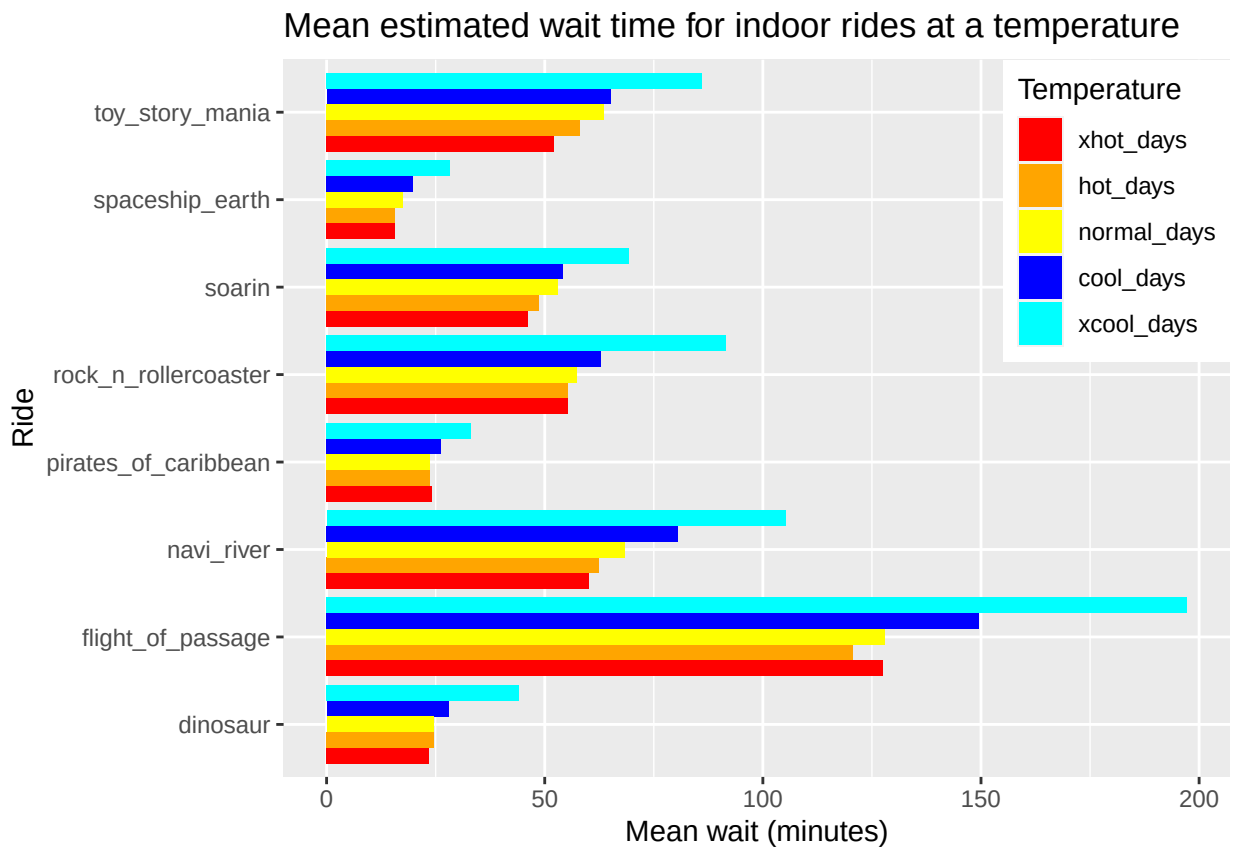

```
temps_df %>%
  inner_join(ride_metadata) %>%
  filter(splash == TRUE) %>%
  group_by(ride_name) %>%
  ggplot() +
  geom_col(aes(x = ride_name, y = mean_wait, fill = fct_relevel(temp_cat, temp_list_names)), position =
  scale_fill_manual(values = temp_colors) +
  coord_flip() +
  labs(fill = "Temp cat") +
  theme(legend.justification=c(1,1), legend.position=c(1,1)) +
  labs(title = "Mean estimated wait for rides with a splash feature at a temperature", fill = "Temperatu
  ylab("Mean wait (minutes)") +
  xlab("Ride")
```

```
## Joining, by = "ride_name"
```



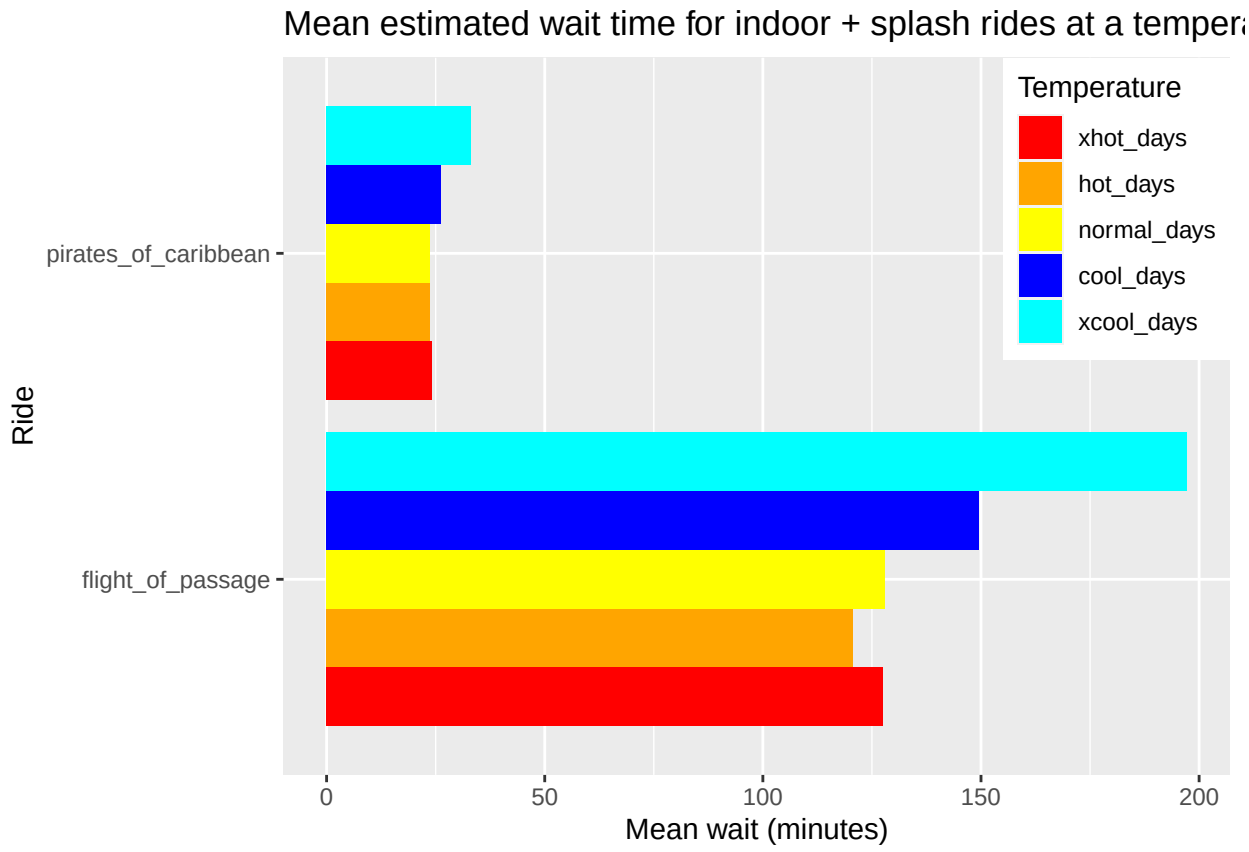

```
temps_df %>%
  inner_join(ride_metadata) %>%
  filter(indoor == TRUE) %>%
  group_by(ride_name) %>%
  ggplot() +
  geom_col(aes(x = ride_name, y = mean_wait, fill = fct_relevel(temp_cat, temp_list_names)), position =
  scale_fill_manual(values = temp_colors) +
  coord_flip() +
  labs(fill = "Temp cat") +
  theme(legend.justification=c(1,1), legend.position=c(1,1)) +
  labs(title = "Mean estimated wait time for indoor rides at a temperature", fill = "Temperature") +
  ylab("Mean wait (minutes)") +
  xlab("Ride")
```

```
## Joining, by = "ride_name"
```




```
temps_df %>%
  inner_join(ride_metadata) %>%
  filter(splash == TRUE & indoor == TRUE) %>%
  group_by(ride_name) %>%
  ggplot() +
  geom_col(aes(x = ride_name, y = mean_wait, fill = fct_relevel(temp_cat, temp_list_names)), position =
  scale_fill_manual(values = temp_colors) +
  coord_flip() +
  labs(fill = "Temp cat") +
  theme(legend.justification=c(1,1), legend.position=c(1,1)) +
  labs(title = "Mean estimated wait time for indoor + splash rides at a temperature", fill = "Temperatu
  ylab("Mean wait (minutes)") +
  xlab("Ride")
```

```
## Joining, by = "ride_name"
```



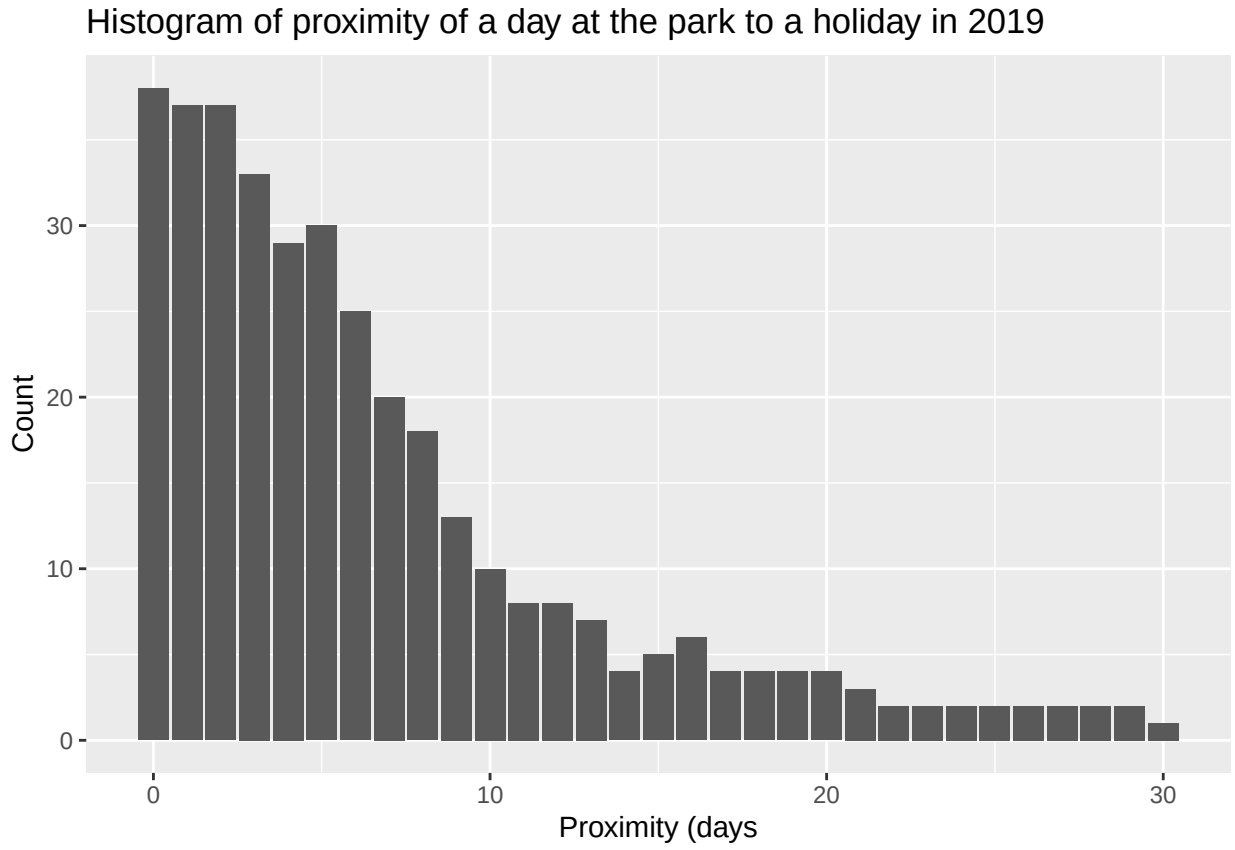
Holiday_prox.rmd analysis

This .rmd file contained data and visualizations that focus on the proximity of holiday seasons to dates and how approaching holidays affects wait times in the Walt Disney World Parks. In the visualizations constructed within this file, it can be observed that wait times of Disney rides significantly increase as a holiday nears close. Curiously, there is a dip in wait times at around 10 – 20 days away from a holiday.

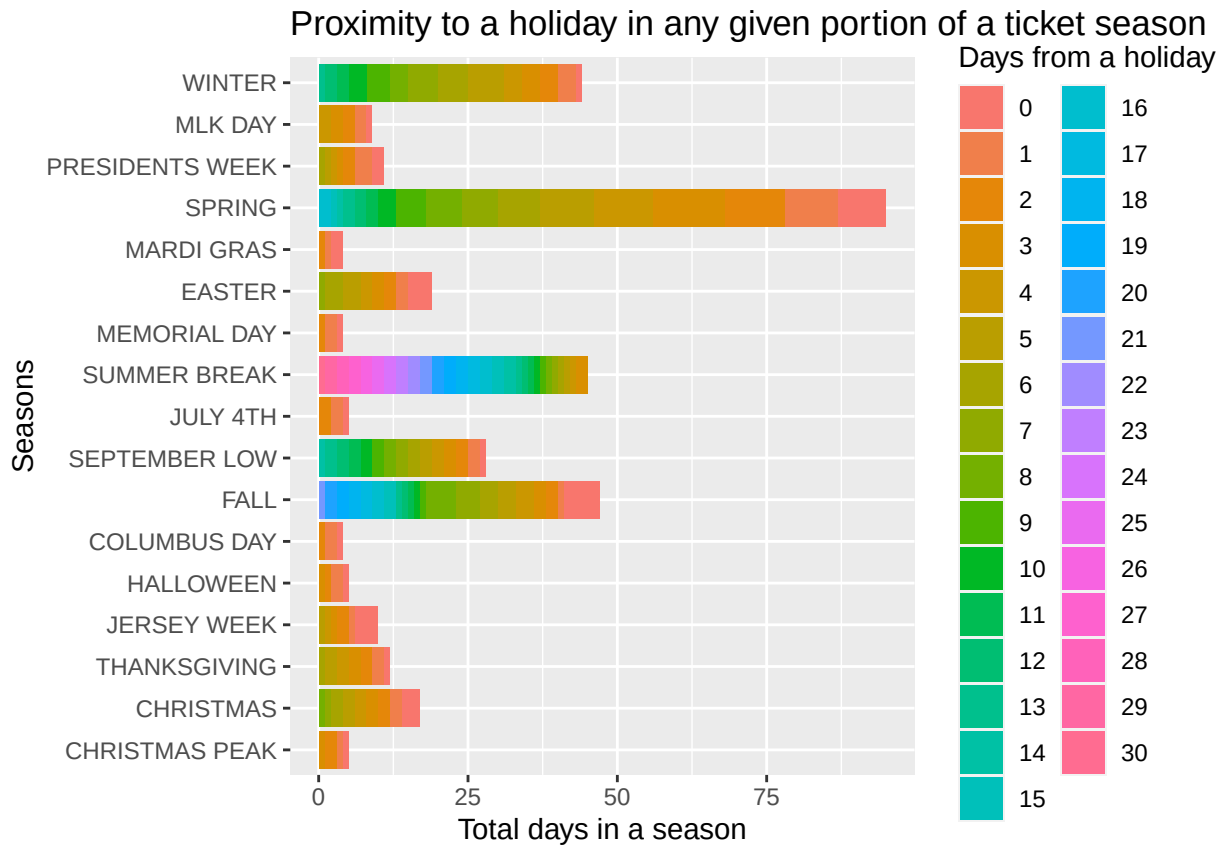
```
#Create dataframe with just days from 2019
#This is done because, in theory, the seasons are the same from year to year
wdw_metadata_2019 <- wdw_metadata %>%
  filter(year(as.Date(mdy(DATE))) == 2019)

seasons <- c("WINTER", "MLK DAY", "PRESIDENTS WEEK", "SPRING", "MARDI GRAS", "EASTER", "MEMORIAL DAY",
             "JULY 4TH", "SEPTEMBER LOW", "FALL", "COLUMBUS DAY", "HALLOWEEN", "JERSEY WEEK", "THANKSGI
```

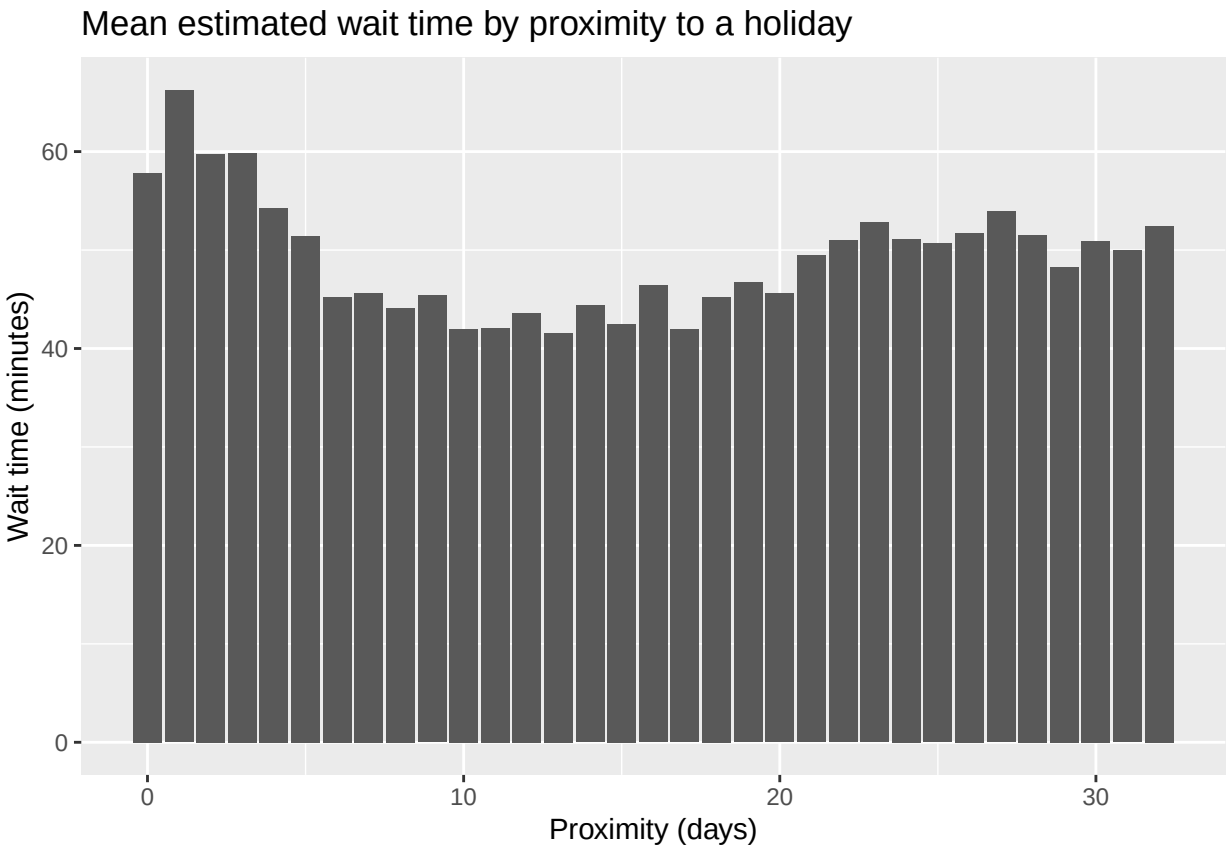
```
wdw_metadata_2019 %>%
  ggplot() +
  geom_bar(aes(x = HOLIDAYPX)) +
  labs(title = "Histogram of proximity of a day at the park to a holiday in 2019") +
  xlab("Proximity (days") +
  ylab("Count")
```




```
wdw_metadata_2019 %>%
  group_by(SEASON) %>%
  ggplot() +
  geom_bar(aes(y = ordered(SEASON, levels = rev(seasons)), fill = as.factor(HOLIDAYPX))) +
  labs(title = "Proximity to a holiday in any given portion of a ticket season", fill = "\nDays from a h
  xlab("Total days in a season") +
  ylab("Seasons")
```




```
wdw_metadata %>%
  inner_join(rides_df, by = c("DATE" = "date")) %>%
  group_by(HOLIDAYPX) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = HOLIDAYPX, y = mean_wait)) +
  labs(title = "Mean estimated wait time by proximity to a holiday") +
  xlab("Proximity (days)") +
  ylab("Wait time (minutes)")
```



parades.rmd

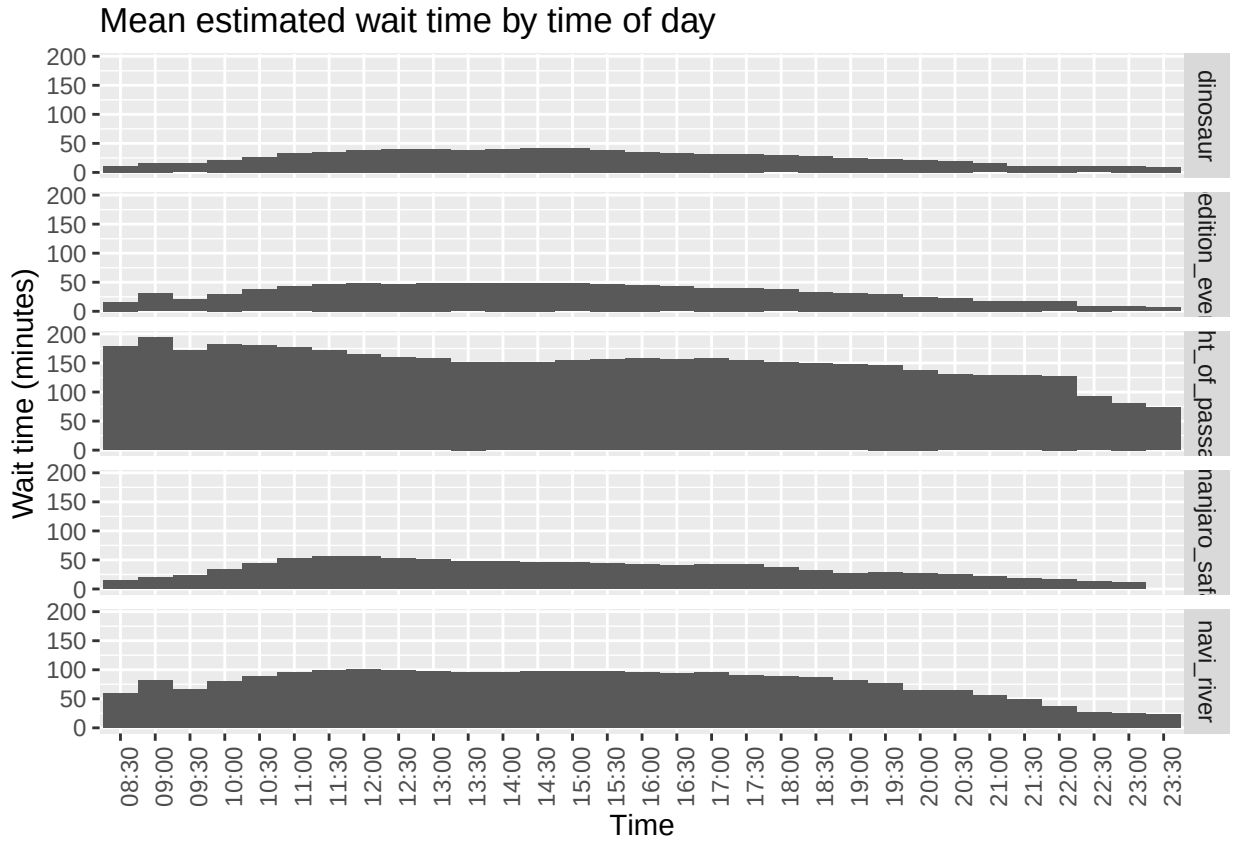
Parades.rmd looked at the average wait time for rides in proximity to the start time for parades. The correlation between wait times for rides and the start time of parades is that there is an expected drop when parades commence. So as the parades make their way through the park there will be a decrease in wait times given the location of the parade at that given time. There is an even more noticeable drop in wait times for rides that are closest to the park entrance which is where the parades usually begin. The interesting correlation observed is what follows the parade has concluded. There are large crowds associated for an extended period after the parades have finished which makes it difficult to traverse to desired locations. What happens then is that crowds will influx to the nearest rides, shops, and other attractions in their closest vicinity. This brings about an expected randomness, and an associated increase for wait times for rides in closeness to the parade routes. What can be extrapolated from this data is that if you want to avoid the crowds and wait times, then plan accordingly and potentially plan your day to account for attractions that are not in close proximity to the parade routes with respect to their associated commencement times.

```
rides_df %>%
  inner_join(ride_metadata) %>%
  filter(park == "ak") %>%
  inner_join(ak_metadata, by = c("date" = "DATE")) %>%
  mutate(time = format(round_date(ymd_hms(datetime), "30 minutes"), "%H:%M")) %>%
  filter(hm(time) > AKOPEN) %>%
  group_by(time, ride_name) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = time, y = mean_wait), width = 1) +
```

```
  theme(axis.text.x = element_text(angle = 90)) +
  facet_grid(rows = vars(ride_name)) +
  labs(title = "Mean estimated wait time by time of day") +
  xlab("Time") +
  ylab("Wait time (minutes)")
```

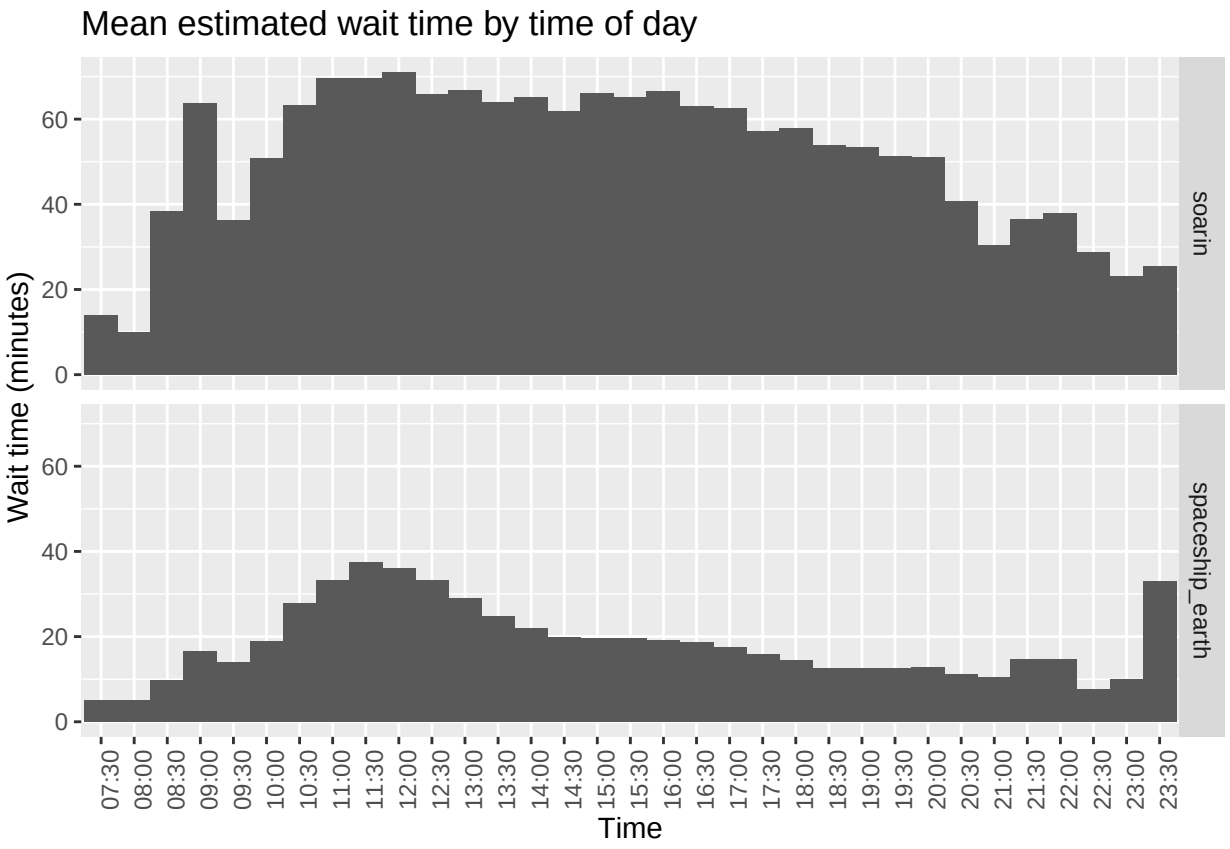
```
## Joining, by = "ride_name"
```

```
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
```



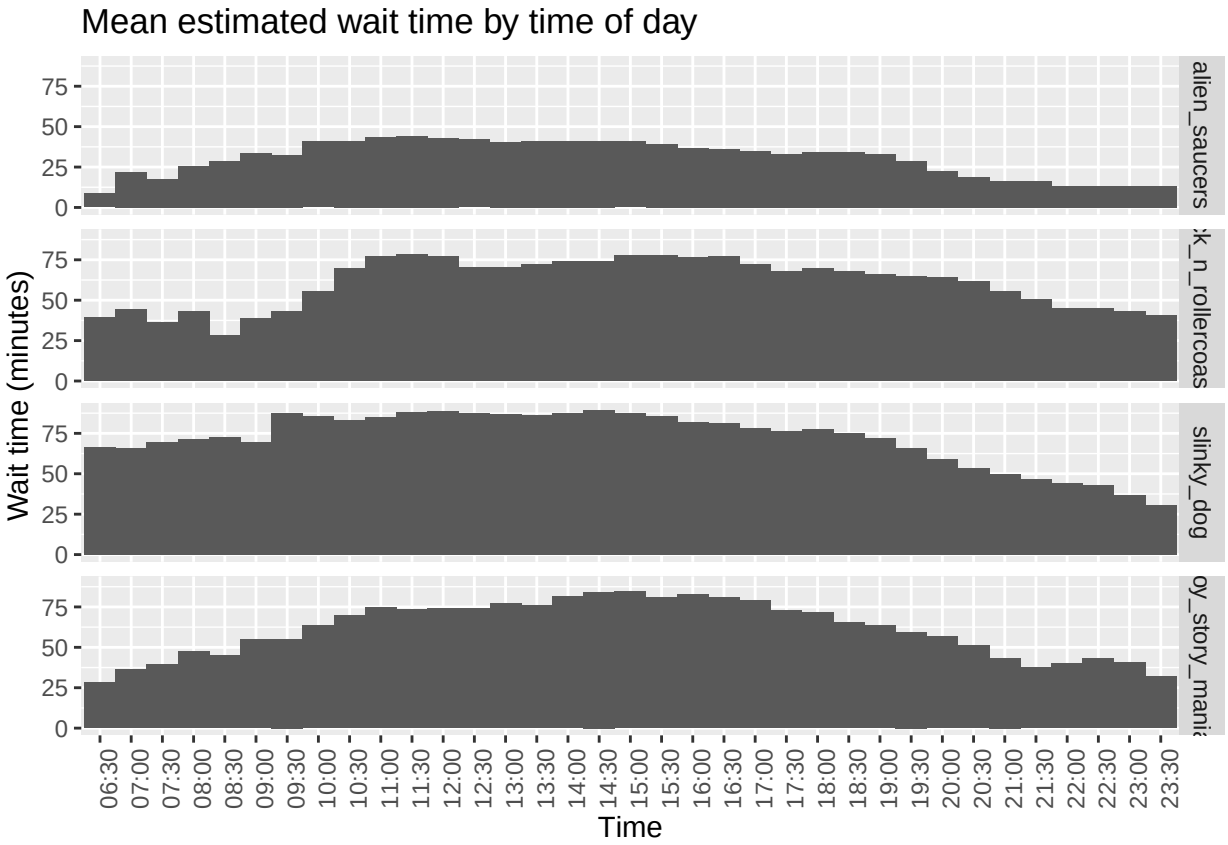

```
rides_df %>%
  inner_join(ride_metadata) %>%
  filter(park == "ep") %>%
  inner_join(ep_metadata, by = c("date" = "DATE")) %>%
  mutate(time = format(round_date(ymd_hms(datetime), "30 minutes"), "%H:%M")) %>%
  filter(hm(time) > EPOPEN) %>%
  group_by(time, ride_name) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = time, y = mean_wait), width = 1) +
  theme(axis.text.x = element_text(angle = 90)) +
  facet_grid(rows = vars(ride_name)) +
  labs(title = "Mean estimated wait time by time of day") +
  xlab("Time") +
  ylab("Wait time (minutes)")
```

```
## Joining, by = "ride_name"
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
```



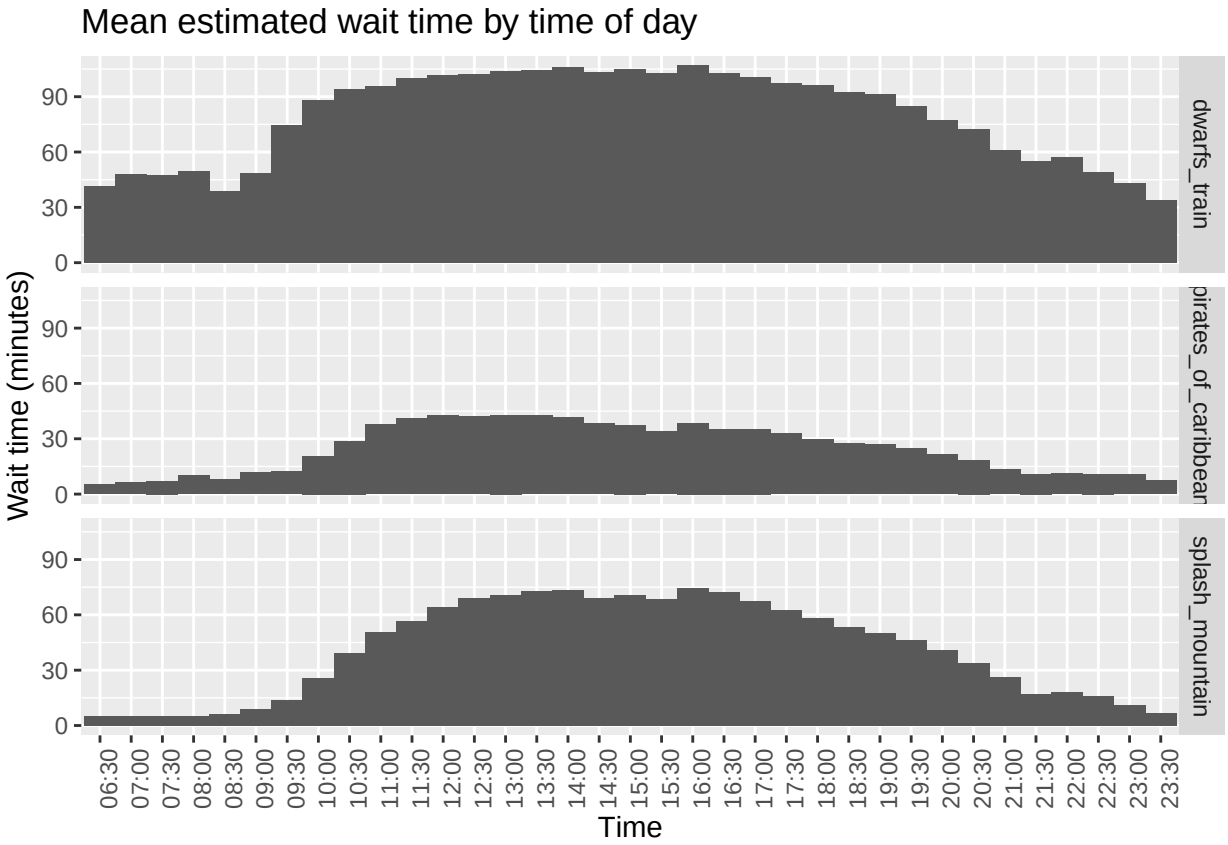

```
rides_df %>%
  inner_join(ride_metadata) %>%
  filter(park == "hs") %>%
  inner_join(hs_metadata, by = c("date" = "DATE")) %>%
  mutate(time = format(round_date(ymd_hms(datetime), "30 minutes"), "%H:%M")) %>%
  filter(hm(time) > HSOPEN) %>%
  group_by(time, ride_name) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = time, y = mean_wait), width = 1) +
  theme(axis.text.x = element_text(angle = 90)) +
  facet_grid(rows = vars(ride_name)) +
  labs(title = "Mean estimated wait time by time of day") +
  xlab("Time") +
  ylab("Wait time (minutes)")
```

```
## Joining, by = "ride_name"
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
```




```
rides_df %>%
  filter(year(mdy(date)) == 2019) %>%
  inner_join(ride_metadata) %>%
  filter(park == "mk") %>%
  inner_join(mk_metadata, by = c("date" = "DATE")) %>%
  mutate(time = format(round_date(ymd_hms(datetime), "30 minutes"), "%H:%M")) %>%
  filter(hm(time) > MKOPEN) %>%
  group_by(time, ride_name) %>%
  summarise(mean_wait = mean(SPOSTMIN, na.rm = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = time, y = mean_wait), width = 1) +
  theme(axis.text.x = element_text(angle = 90)) +
  facet_grid(rows = vars(ride_name)) +
  labs(title = "Mean estimated wait time by time of day") +
  xlab("Time") +
  ylab("Wait time (minutes)")
```

```
## Joining, by = "ride_name"
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
```



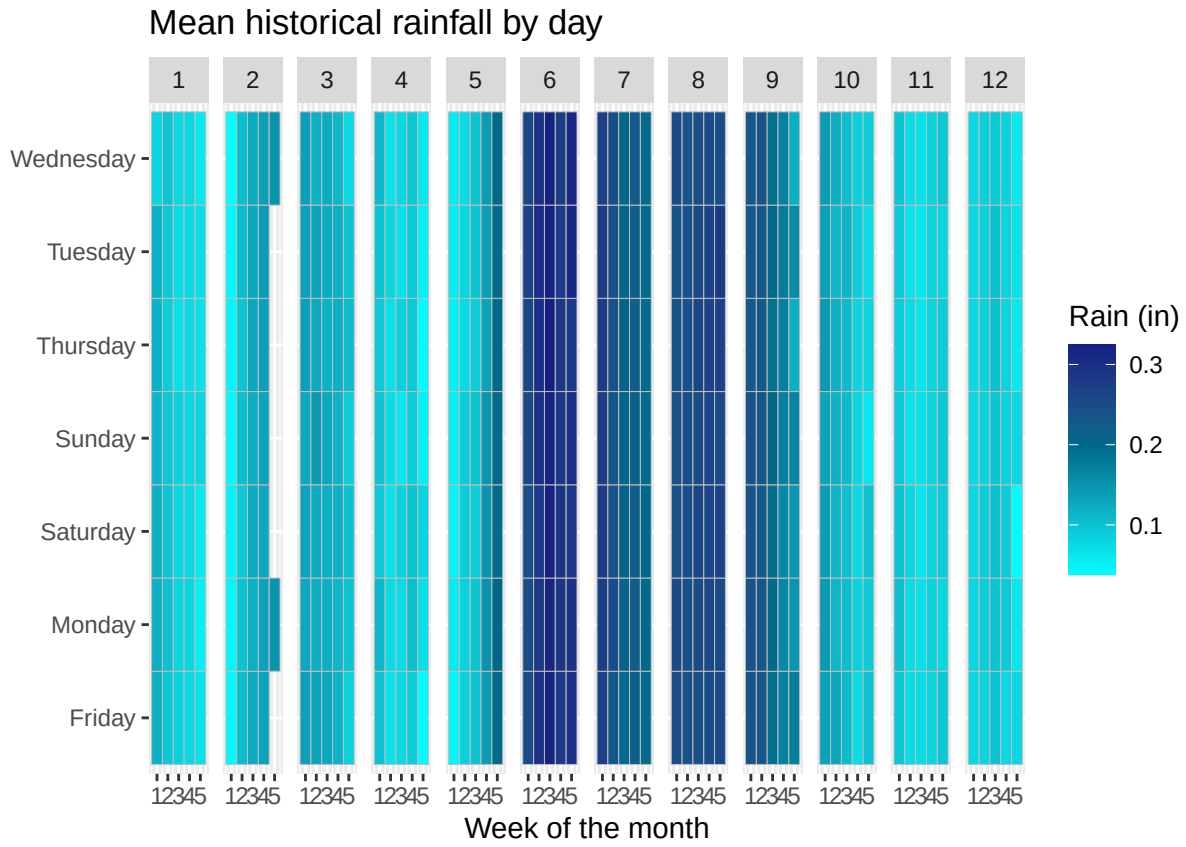
rainfall.rmd

Rainfall.rmd explored the relationship between average precipitation levels in the Disney parks and ride wait times. From this data, we constructed a data visualization that displayed the amount of rainfall, in inches, that was reported on each week on the month over a time period of several years. Our analysis revealed several interesting observations. First, during the summer months, there was an observed rainy season in the parks that had high precipitation levels. In contrast, the months September through December were characterized by significantly lower average precipitation levels. Next, our data showed that there was a distinct correlation between average rainfall levels and wait times for indoor rides. When there is significantly higher reported precipitation levels, wait times for indoor rides increase as Disney guests seek shelter during the storm. As a frequent Disney guest myself, it is always interesting to see how crowded gift shops, indoor rides, shows, and other covered venues can get under poor weather conditions. Another interesting observation drawn from rainfall.rmd was the relationship between precipitation levels and "splash-aspect" rides. Since "splash-aspect" rides are typically located outdoors such as Splash Mountain, it is mandatory that Disney shuts these rides down during severe weather or thunderstorms. Therefore, a significant decrease in wait times for "splash-aspect" rides can be noticed when there is high rainfall.

```
wdw_metadata %>%
  mutate(year = year(as.POSIXct(mdy(DATE) + 1)), month = month(as.POSIXct(mdy(DATE) + 1)), day = weekday
         monthweek = ceiling(day(mdy(DATE)) / 7)) %>%
  group_by(month, day, monthweek) %>%
  summarise(mean_rain = mean(WEATHER_WDWPRECIP, na.rm = TRUE)) %>%
  ggplot(aes(x = monthweek, y = day, fill = mean_rain)) +
  geom_tile(color = "grey") +
  facet_grid(~month) +
```

```
  scale_fill_gradient2(low = "cyan", mid = "deepskyblue4", high = "navy", midpoint = .2) +
  labs(fill = "Rain (in)", title = "Mean historical rainfall by day", x = "Week of the month", y = "")
```

```
## `summarise()` has grouped output by 'month', 'day'. You can override using the `.groups` argument.
```



Conclusion

The greatest correlation that was observed was correlated with the opening dates of new attractions at any given park. Animal Kingdom's Flight of Passage, Magic Kingdom's Snow White and the Seven Dwarves Mine Train and Hollywood Studio's Slinky Dog Dash all were opened relatively recently making them their respective park's newest attraction. It can be correlated that crowds are attracted to rides that give them a new experience to their previous Disney trips. In terms of the best weekday to plan a day at Disney, on average the middle of the week is a much safer bet than going on a weekend which could be assumed by any Disney patron based on personal experience. Specifically, Wednesday is the least busy day of the week with the shortest wait times on average. On the opposite end of the spectrum, Saturday is characterized as the busiest day of the week with the longest observed average wait times making it less than ideal for making the most of a single day. In terms of monthly averages, September has a noticeably lower average ride wait time across the board with all parks. This could be correlated with the return to school for K-12 students. The fifth week of December has been observed as the busiest week of the year with the longest recorded wait times. The December holiday season is without a doubt the easiest time for families to be able to make trips as it is synonymous with vacation time. Of the Disney parks, Hollywood Studios has the greatest average wait times which can be correlated with the opening of the attraction Toy Story Land in 2018.

Holiday_prox.rmd focused on the proximity of holiday seasons to dates and how approaching holidays affects wait times which yielded a dip in wait times at around 10 – 20 days away from a holiday. Parades.rmd

looked at the average wait time for rides in proximity to the start time for parades. The correlation between wait times for rides and the start time of parades is that there is an expected drop when parades commence. Meaning, to avoid the crowds and wait times, then plan accordingly and potentially plan your day to account for attractions that are not close to the parade routes with respect to their associated commencement times. Rainfall.rmd explored the relationship between average precipitation levels in the Disney parks and ride wait times. Our data showed that there was a distinct correlation between average rainfall levels and wait times for indoor rides which increase as Disney guests seek shelter during the storm. From our Hot_weather.rmd, analysis revealed that guests prefer to ride indoor rides on cooler days and splash-aspect on hotter days as to be expected with inclimate weather.

So what can be extrapolated from all of this is that the ideal travel itinerary would be to plan for a Wednesday in the month of September to Epcot with the intention of riding Soarin' and Spaceship Earth based off of the expected average mean times. According to our data analysis this is what would make for the efficient utilization of time with minimized wait times which can be directly correlated with a positive Disney park experience, objectively and subjectively making it the happiest place on Earth.

Discussion

This will require a summary of what you have learned about your research question. Provide suggestions for improving your analysis, and include a paragraph on what you would do differently if you were able to start over with the project or what you would do next if you were going to continue work on the project.

In the event that we were to re-do the project there are a handful of aspects that we would have handled differently, or different data that we would have analyzed which may have given a better representation of the research questions that we were attempted to answer. The intensive part of the project was restructuring the metadata which was to be expected given that there are rarely perfect or complete datasets. In terms of data format, had we mutated a column for thunderstorms and lightning strikes in the rainfall data from an outside dataset, it would have given us the necessary information to make better conclusions regarding hazardous weather. If we had been able to include more Disney rides from each of the parks, then we would have a more accurate representation of the data that we were trying to represent. This could have ultimately changed the results of our final conclusions, but we worked with the data that we were able to utilize. Another interesting point would have been made if we were able to analyze ticket sales and block-out dates for passholders. #### Answers to Our Research Questions: - Does hot weather increase wait times for rides with a "splash_aspect"? Yes! Hot weather causes outdoor splash-aspect rides to have a significantly higher wait time than on cooler days. - Does rainy weather cause an increase in wait times for "indoor_rides"? Yes! Rainy days cause there to be a distinct increase in the wait times for indoor rides. - What is the busiest Day of week (lubridate data) for tourists to visit WDW? Based on wait times, the busiest day of the week to visit WDW is Saturday. - What affect do parades have on the wait times of rides? Parades cause a slight decrease in wait times.