Cherry Blossom 10-Mile Race Report

Akshata Jha, Breanna Neuwirth, and Andrin Zuellig

December 13, 2024

##### Introduction

The primary question in this consulting project is: How does age influence athletic performance in the Cherry Blossom 10-Mile Race? This study aims to provide a comprehensive understanding and a simple answer to understand how age and other factors shape the race performance in an endurance event.

To investigate this, we worked closely with our client and by doing this we were able to scrape around 400,000 individual race records from the Cherry Blossom Race Data website. This dataset spans over 51 years worth of information such as participants’ ages, genders, finish times, paces and rankings. This gave us the unique opportunity to dive into finding and understanding the trends and patterns. We were able to explore how performance differs in across different age groups, how performance differs across genders, and much more. Our aim is to uncover meaningful insights that will be insightful for our client.

##### Background

The Cherry Blossom 10-Mile Race, held annually in Washington, D.C., has its roots as a small, community-focused event that brought people together for a fun and challenging running experience. What began as a local race has since evolved into a nationally and internationally renowned event, drawing thousands of participants from across the United States and around the globe. With its running course winding through the heart of D.C. during the peak of the cherry blossom season, the race has become a celebrated race for its camaraderie, and athleticism.

Over its 51 year, the Cherry Blossom 10-Mile Race has accumulated a wealth of data, offering an unparalleled opportunity to explore trends in athletic performance. This rich dataset not only captures individual performances but also reflects broader changes in participation, gender dynamics, and training methodologies over time.

Age, as one of the most critical factors in endurance sports, plays a central role in shaping performance. Research and historical observations suggest that endurance performance typically peaks in early adulthood, often between the late 20s and early 30s, before gradually declining as athletes age with some participants as exceptions. This project aims to analyze how age influences performance in the Cherry Blossom 10-Mile Race, identifying key trends and patterns over the years. By exploring these age-related dynamics, the project provides valuable insights into the intersection of age, athleticism, and endurance over time.

##### Data

The dataset for this project is a comprehensive collection of over 400,000 race observations spanning more than 50 years of the Cherry Blossom 10-Mile Race. It provides an extensive look into the performance and demographics of participants, offering a unique opportunity to analyze long-term trends in endurance running. The dataset includes the following detailed participant-level variables:

• Name: The name of the participant, serving as an identifier.

• PiD\_TiD: Rank within the participant’s specific division (e.g., age and gender category).

• PiS\_TiS: Overall rank within the participant’s gender group.

• Age: The participant’s age at the time of the race, a key variable for analyzing performance trends.

• Time: The official finish time recorded for each participant, reflecting their overall performance.

• Pace: The average pace per mile, calculated based on the finish time and race distance.

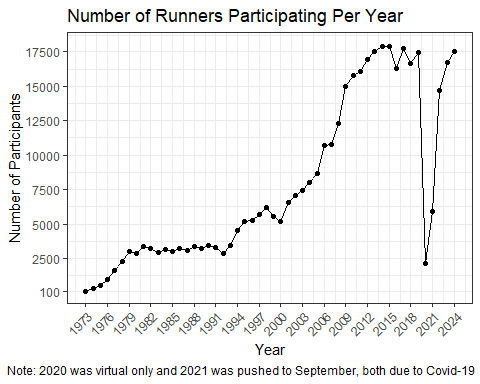
• Division: The participant’s age and gender category (e.g., “M0119” represents males aged 0-19).

• Hometown: The participant’s place of residence (note: some entries are incomplete or missing).

• Year: The year in which the race took place, allowing for the analysis of historical trends.

• Gender: The participant’s gender, categorized as “M” (Male) or “W” (Female).

This provides a comprehensive source for understanding how age, gender, and other demographic variables influence performance in endurance races.



| min | Q1 | median | Q3 | max | mean | sd | n | missing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 127 | 6530 | 15763 | 17434 | 17880 | 12356 | 5604 | 404306 | 0 |

Here we can see the number of participants over the years of the race. The first year of the race, 1973, had the smallest group of runners at 127 people. The peak number of runners, 17,880, took place in 2015. The race has become more popular over the years, attracting thousands of people.

A graph of a number of men and women participating each year

Description automatically generated

| Gender | min | Q1 | median | Q3 | max | mean | sd | n | missing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | 115 | 3105 | 5904 | 7009 | 7483 | 5096 | 2000 | 203085 | 0 |
| W | 12 | 5435 | 9563 | 10290 | 11042 | 7731 | 3280 | 201199 | 0 |

This graph shows the proportion of men and women who have participated in the race over the years. The number of women participating has steadily increased, with recent years have more women than men running. The race organizers have been making efforts to bring popularity to the women’s division of the race, which have been working. The average number of women running in the race is 7,731 and the average for men participating is 5,096.

A graph of different colored lines

Description automatically generated

| min | Q1 | median | Q3 | max | mean | sd | n | missing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 29 | 35 | 43 | 91 | 36.77 | 10.54 | 400633 | 3673 |

In this graph, we have the number of participants for different age groups across the years of the race. The majority of runners are in their 20s and 30s. There is no age restriction for the race, so the range is rather large from 2 years old to 91 years old. The average age is about 36 years old.

##### Data Collection and Challenges

This dataset was collected by scraping results from the Cherry Blossom Race website. While the client provided some guidance, the scraping process was a difficult process as the website was not designed for easy data extraction. The data was collected in multiple files, each covering a different time period, and required extensive cleaning and preprocessing to make it suitable for analysis. The data was then combined into a single dataframe for further analysis.

##### Data Cleaning and Preprocessing

After collecting the data, the following steps were taken to clean and prepare it for analysis. We removed the rows with missing values from Age, Time, Pace, and Gender. For Time and Pace, we had to correct the format by adding zeros to the hours place, when needed, which helped when converting them to a numeric format in minutes. Any implausible race times or paces (e.g., unrealistically fast finish times) were identified and excluded.There were issues with the data in the first few years of the race, along with much smaller amounts of data, so we decided to remove the years 1973-1980. The R code for these processes and for the graphs is in the Appendix.

#### Methods

In order to explore the relationship between age and performance, the following methods will be implemented. We will do visualizations for age and performance trends, gender specific insights, yearly trends, and influence of weather. We will also conduct linear regression modeling on performance, which will help quantify the relationship of these variables with performance and assess predictive ability.

#### Results

#### Conclusions

#### References

#### Appendices

##### Appendix 1: Data and Data Cleaning Code

knitr::opts\_chunk$set(echo = TRUE)  
  
library(readr)  
library(dplyr)

library(tidyverse)

library(lubridate)  
library(mosaic)

library(ggplot2)  
library(pander)

Data\_73\_21 <- read\_csv("Data\_73\_21.csv", na=c("NR", "NA"), col\_types = "ccnccccccn")  
Data\_22 <- read\_csv("Data\_22.csv", col\_types = "ncccnccccnc", na=c("NR", "NA"))

Data\_23 <- read\_csv("Data\_23.csv", col\_types = "ncccnccccnc", na=c("NR", "NA"))

Data\_24 <- read\_csv("Data\_24.csv", col\_types = "ncccnccccnc", na=c("NR", "NA"))

Data\_73\_21 <- Data\_73\_21 %>%   
 separate(Name, c("Name", "Gender"), sep = -4) %>%   
 mutate(Gender = case\_when(Gender == " (M)" ~ "M",  
 Gender == " (W)" ~ "W")) %>%   
 rename(PiS\_TiS = `PiS/TiS`, PiD\_TiD = `PiD/TiD`) %>%   
 select(1,2,9,7,4,5,6,8,10,11,3) %>%   
 select(-1)  
  
Data\_22 <- Data\_22 %>% select(-1)  
Data\_23 <- Data\_23 %>% select(-1)  
Data\_24 <- Data\_24 %>% select(-1)  
  
Dat\_Comb1 <- rbind(Data\_73\_21, Data\_22, Data\_23, Data\_24)  
  
Dat\_Comb <- Dat\_Comb1 %>%  
 filter(!is.na(Age)) %>% #Removing missing values from age  
# Process Time and add new columns  
 mutate(  
 Time = na\_if(Time, "NR"),   
 Time = na\_if(Time, ""),   
 Time = case\_when(  
 str\_detect(Time, "^\\d{1,2}:\\d{1,2}$") ~ paste0("00:", Time), # Add "00:" for missing hours  
 str\_detect(Time, "^\\d{1,2}:\\d{1,2}:\\d{1,2}$") ~ Time,   
 TRUE ~ NA\_character\_ # Replace invalid formats with NA  
 ),  
 Time = hms(Time), # Convert valid Time to hms format  
 Minutes = period\_to\_seconds(Time) / 60 # Calculate time in minutes  
 ) %>%  
 mutate(Pace\_Num = (period\_to\_seconds(ms(Pace)))/60) %>% #converting pace to numeric  
 filter(!is.na(Pace)) %>%   
 filter(!is.na(Gender)) %>%   
 filter(!is.na(Time)) %>% # Remove rows with missing Time  
 filter(Year > 1980)

This adds the precipitation and minimum temperature for that date

#library(readr)  
weather <- read\_csv("3865927.csv", show\_col\_types = F)  
  
dates <- data.frame(  
 year = 1981:2024,  
 date = as.Date(c("1981-04-05", "1982-04-04", "1983-04-03", "1984-04-01",   
 "1985-03-31", "1986-04-06", "1987-04-05", "1988-04-03",   
 "1989-04-02", "1990-04-01", "1991-03-31", "1992-04-05",   
 "1993-04-04", "1994-04-03", "1995-04-02", "1996-03-31",   
 "1997-04-06", "1998-04-05", "1999-04-04", "2000-04-02",   
 "2001-04-01", "2002-04-07", "2003-04-06", "2004-04-04",   
 "2005-04-03", "2006-04-02", "2007-04-01", "2008-04-06",   
 "2009-04-05", "2010-04-04", "2011-04-03", "2012-04-01",   
 "2013-04-07", "2014-04-06", "2015-04-05", "2016-04-03",   
 "2017-04-02", "2018-04-07", "2019-04-06", "2020-04-05",   
 "2021-09-12", "2022-04-03", "2023-04-02", "2024-04-6")))  
  
temp<-left\_join(Dat\_Comb, dates, by = c("Year" = "year"))  
sample\_n(temp, 10)

Dat\_Comb.weather <- left\_join(temp, select(weather, DATE, PRCP, TMIN), by = c("date" = "DATE"))

##### Appendix 2: Summary Statistics Code

graph1 <- Dat\_Comb1 %>%  
 group\_by(Year) %>%  
 add\_count() %>%  
 ungroup()  
   
graph1 %>% ggplot(aes(x=Year, y=n)) +  
 geom\_point() +  
 geom\_line() +  
 scale\_x\_continuous(trans="identity", limits=c(1973,2024), breaks=seq(1973,2024,3)) +  
 scale\_y\_continuous(trans="identity", limits=c(100,18000),   
 breaks=c(100, 2500,5000,7500,10000,12500,15000,17500)) +  
 labs(y="Number of Participants", title="Number of Runners Participating Per Year",   
 caption="Note: 2020 was virtual only and 2021 was pushed to September, both due to Covid-19") +  
 theme\_bw() +  
 theme(plot.caption.position = "plot",  
 plot.caption = element\_text(hjust = 0),   
 axis.text.x=element\_text(angle = 45, hjust = 1))

Dat\_Comb1 %>%   
 filter(!is.na(Gender)) %>%   
 ggplot(aes(x=Year, fill=Gender)) +  
 geom\_bar(position="fill") +  
 scale\_fill\_manual(values = c("M" = "turquoise", "W" = "salmon")) +  
 labs(y="Proportion", title = "Proportion of Men and Women Participating Each Year") +  
 geom\_abline(aes(intercept=0.5, slope=0), color="black") +  
 theme\_bw()

graph2 <- Dat\_Comb1 %>%  
 group\_by(Year, Gender) %>%   
 add\_count() %>%  
 ungroup()  
  
pander(favstats(n ~ Gender, data=graph2))

graph3 <- Dat\_Comb %>%   
 mutate(Age\_Group = case\_when(  
 Age < 20 ~ "Under 20",  
 Age >= 20 & Age < 30 ~ "20s",  
 Age >= 30 & Age < 40 ~ "30s",  
 Age >= 40 & Age < 50 ~ "40s",  
 Age >= 50 & Age < 60 ~ "50s",  
 Age >= 60 & Age < 70 ~ "60s",  
 Age >= 70 ~ "70+"))  
  
graph3 %>% ggplot(aes(x=Year, fill=Age\_Group)) +  
 geom\_bar(position = "stack") +  
 labs(y="Number of Participants",   
 title="Number of Runners Participating in Each Age Group Per Year",   
 caption = "Note: 2020 was virtual only and 2021 was pushed to September, both due to Covid-19", fill="Age Group") +  
 theme\_bw() +  
 theme(plot.caption.position = "plot",  
 plot.caption = element\_text(hjust = 0)

#### Appendix 3: Age Insights Code

This blocks ensures all race data is combined into one dataset, making it easier to clean, manipulate and visualize.

# Create age groups (if not already grouped)  
all\_data <- Dat\_Comb %>%  
 mutate(Age\_Group = case\_when(  
 Age < 20 ~ "<20",  
 Age >= 20 & Age < 30 ~ "20-29",  
 Age >= 30 & Age < 40 ~ "30-39",  
 Age >= 40 & Age < 50 ~ "40-49",  
 Age >= 50 ~ "50+"  
 ))  
  
# Calculate average race time for each age group over years  
average\_race\_times <- all\_data %>%  
 group\_by(Year, Age\_Group) %>%  
 summarize(Average\_Time = mean(Minutes, na.rm = TRUE), .groups = "drop")  
  
# Create the line graph  
ggplot(data = average\_race\_times, aes(x = Year, y = Average\_Time, color = Age\_Group)) +  
 geom\_line(size = 1) +  
 scale\_x\_continuous(breaks = seq(1981, 2024, 5), limits = c(1981, 2024)) + # Set x-axis from 1973 to 2024  
 labs(  
 title = "Average Race Times Across Age Groups (1981–2024)",  
 x = "Year",  
 y = "Average Race Time (minutes)",  
 color = "Age Group"  
 ) +  
 theme\_bw() +  
 theme(  
 text = element\_text(size = 12),  
 legend.position = "bottom"  
 )

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

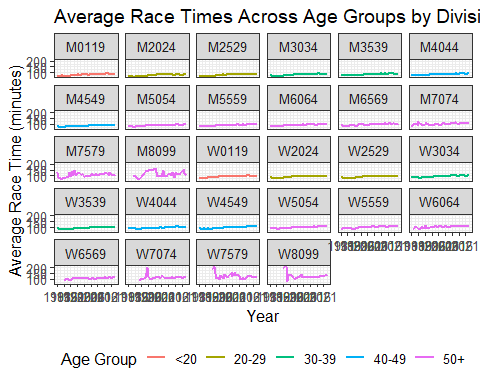
A graph of different colored lines

Description automatically generated

The grouping above makes it easier to analyze trends across age groups instead of focusing on individual ages, the insights showed that participants <20 and between 20 to 29 were faster than participants 50+

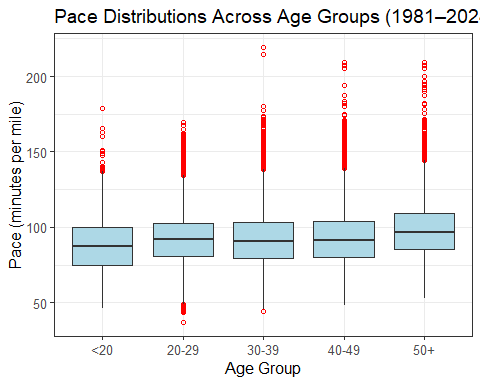
We convert the race completion times into minutes for mathematical operations. This shows the average race time for each age group, year, and division.

# Calculate average race time for each age group, year, and division  
average\_race\_times <- all\_data %>%  
 filter(!is.na(Division)) %>%   
 group\_by(Year, Division, Age\_Group) %>%  
 summarize(Average\_Time = mean(Minutes, na.rm = TRUE), .groups = "drop")  
  
# Create the faceted line graph  
ggplot(data = average\_race\_times, aes(x = Year, y = Average\_Time, color = Age\_Group)) +  
 geom\_line(size = 1) +  
 scale\_x\_continuous(breaks = seq(1981, 2024, 5), limits = c(1981, 2024)) + # Set x-axis from 1973 to 2024  
 labs(  
 title = "Average Race Times Across Age Groups by Division (1981–2024)",  
 x = "Year",  
 y = "Average Race Time (minutes)",  
 color = "Age Group"  
 ) +  
 facet\_wrap(~Division) + # Facet by Division (gender proxy)  
 theme\_bw() +  
 theme(  
 text = element\_text(size = 12),  
 legend.position = "bottom"  
 )

 This shows the average finishing time of each age group. As shown previously this plot proves again that people in these groups <20 and 20-29 have a shorter average in finishing time.

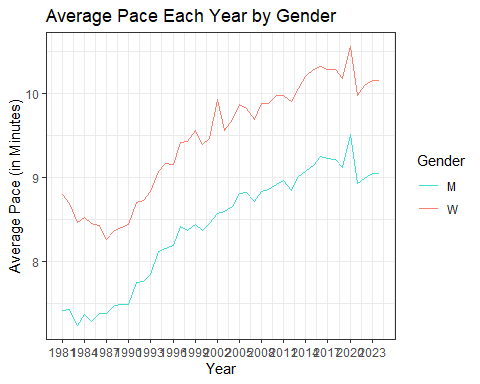
This code focuses on creating a box plot to visualize the distribution of race paces across different age groups from 1973 to 2024.

# Create the box plot  
ggplot(all\_data, aes(x = Age\_Group, y = Minutes)) +  
 geom\_boxplot(fill = "lightblue", outlier.color = "red", outlier.shape = 1) +  
 labs(  
 title = "Pace Distributions Across Age Groups (1981–2024)",  
 x = "Age Group",  
 y = "Pace (minutes per mile)"  
 ) +  
 theme\_bw() +  
 theme(  
 text = element\_text(size = 12)  
 )

 The findings of <20 group typically has higher variability, possibly due to a mix of highly competitive younger runners and casual participants. The 20-29 group tends to have the fastest median pace, confirming prior research that peak endurance occurs in the late 20s.

#### Appendix 3: Gender Insights Code

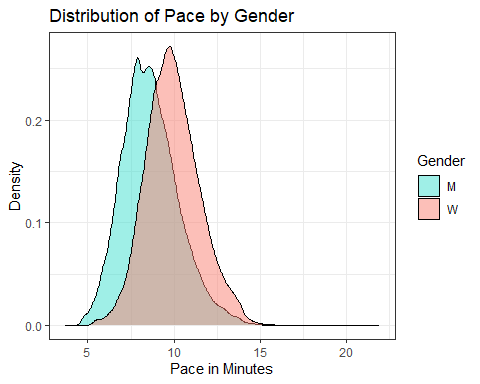
graph4 <- Dat\_Comb %>%  
 summarise(avg\_pace = mean(Pace\_Num), .by=c(Gender, Year)) %>%   
 ungroup()  
  
graph4 %>% ggplot(aes(y=avg\_pace, x=Year, color=Gender)) +  
 geom\_line() +  
 scale\_color\_manual(values = c("M" = "turquoise", "W" = "salmon")) +  
 scale\_x\_continuous(trans="identity", limits=c(1981,2024), breaks=seq(1981,2024,3)) +  
 labs(y="Average Pace (in Minutes)", title = "Average Pace Each Year by Gender") +  
 theme\_bw()



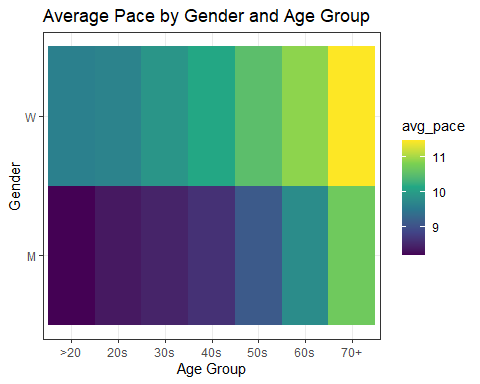
pander(favstats(avg\_pace ~ Gender, data=graph4))

| Gender | min | Q1 | median | Q3 | max | mean | sd | n | missing |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | 7.234 | 7.753 | 8.579 | 8.968 | 9.504 | 8.404 | 0.6805 | 44 | 0 |
| W | 8.264 | 8.778 | 9.62 | 9.997 | 10.56 | 9.462 | 0.6894 | 44 | 0 |

Dat\_Comb %>%  
 ggplot(aes(x=Pace\_Num, group=Gender, fill=Gender)) +  
 geom\_density(alpha=0.5) +  
 scale\_fill\_manual(values = c("M" = "turquoise", "W" = "salmon")) +  
 theme\_bw() +  
 labs(x="Pace in Minutes", title="Distribution of Pace by Gender", y="Density")

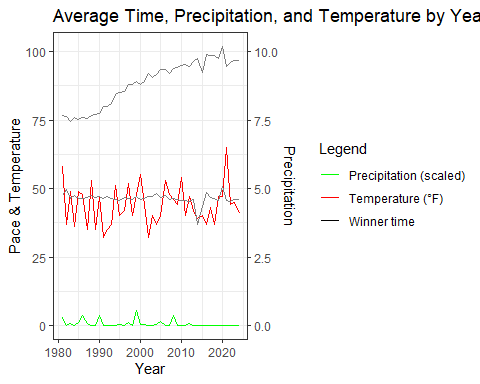


Dat\_Comb %>%  
 mutate(Age\_Group = case\_when(  
 Age < 20 ~ ">20",  
 Age >= 20 & Age < 30 ~ "20s",  
 Age >= 30 & Age < 40 ~ "30s",  
 Age >= 40 & Age < 50 ~ "40s",  
 Age >= 50 & Age < 60 ~ "50s",  
 Age >= 60 & Age < 70 ~ "60s",  
 Age >= 70 ~ "70+")) %>%   
 summarise(avg\_pace = mean(Pace\_Num), .by=c(Gender, Age\_Group)) %>%   
 ungroup() %>%   
 ggplot(aes(x=Age\_Group, y=Gender, fill=avg\_pace)) +  
 geom\_tile() +  
 labs(x="Age Group", legend="Average Pace", title="Average Pace by Gender and Age Group") +  
 theme\_bw() +  
 scale\_fill\_viridis\_c()



## Appendix 4: Weather Insights Code

avg\_per\_year <- Dat\_Comb.weather %>%  
 group\_by(Year) %>%  
 summarise(  
 Avg\_PRCP = mean(PRCP, na.rm = TRUE),  
 winner=min(Minutes, na.rm = TRUE),  
 Temperature = mean(TMIN, na.rm = TRUE),  
 Minutes = mean(Minutes, na.rm = TRUE)  
 )  
  
# Adjusted Plot  
ggplot(avg\_per\_year, aes(x = Year)) +  
 # Line for Avg\_Pace  
 geom\_line(aes(y = Minutes, color = "Average Pace (min)"), size = .3) +  
 # Line for Avg\_PRCP  
 geom\_line(aes(y = Avg\_PRCP \* 10, color = "Precipitation (scaled)"), size = 0.3) +  
 # Line for Temperature  
 geom\_line(aes(y = Temperature, color = "Temperature (°F)"), size = 0.3) +  
 # Line for Winner time  
 geom\_line(aes(y = winner, color = "Winner time"), size = 0.3) +  
 # Primary and secondary y-axis  
 geom\_line(aes(y=winner, color="Winner"), size=0.3) +  
 scale\_y\_continuous(  
 name = "Pace & Temperature",  
 sec.axis = sec\_axis(~ . / 10, name = "Precipitation")  
 ) +  
 # Custom color legend  
 scale\_color\_manual(  
 name = "Legend",  
 values = c(  
 "Precipitation (scaled)" = "green",  
 "Temperature (°F)" = "red",  
 "Winner time" = "black"  
 )  
 ) +  
 labs(  
 title = "Precipitation, Temperature, and Winner Time by Year",  
 "Average Pace (min)" = "blue",  
 "Precipitation (scaled)" = "green",  
 "Temperature (°F)" = "red",  
 "Winner Time" = "black"  
 ) +  
 labs(  
 title = "Average Time, Precipitation, and Temperature by Year",  
 x = "Year"  
 ) +  
 theme\_bw()



## Appendix 5: Regression Code

mod1 <- lm(Minutes ~ Age, data=Dat\_Comb)  
summary(mod1)

##   
## Call:  
## lm(formula = Minutes ~ Age, data = Dat\_Comb)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -53.136 -12.078 -0.677 11.073 126.842   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 85.825202 0.099526 862.34 <2e-16 \*\*\*  
## Age 0.186089 0.002595 71.72 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 17.11 on 391325 degrees of freedom  
## Multiple R-squared: 0.01297, Adjusted R-squared: 0.01297   
## F-statistic: 5143 on 1 and 391325 DF, p-value: < 2.2e-16

mod2 <- lm(Minutes ~ Age + Gender, data=Dat\_Comb)  
summary(mod2)

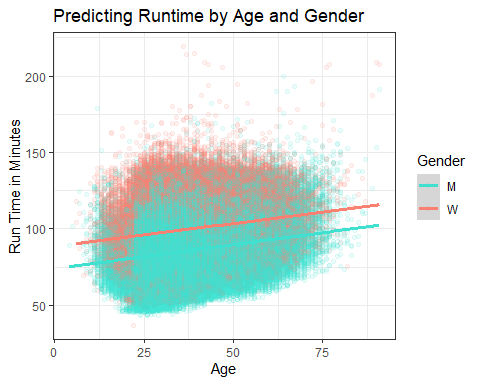
##   
## Call:  
## lm(formula = Minutes ~ Age + Gender, data = Dat\_Comb)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -58.249 -10.865 -1.079 9.703 120.041   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 74.204124 0.100228 740.4 <2e-16 \*\*\*  
## Age 0.306707 0.002413 127.1 <2e-16 \*\*\*  
## GenderW 14.080224 0.050863 276.8 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.64 on 391324 degrees of freedom  
## Multiple R-squared: 0.1746, Adjusted R-squared: 0.1746   
## F-statistic: 4.139e+04 on 2 and 391324 DF, p-value: < 2.2e-16

mod3 <- lm(Minutes ~ Age\*Gender, data=Dat\_Comb)  
summary(mod3)

##   
## Call:  
## lm(formula = Minutes ~ Age \* Gender, data = Dat\_Comb)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -58.348 -10.866 -1.082 9.706 120.048   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 73.948496 0.132661 557.425 < 2e-16 \*\*\*  
## Age 0.313292 0.003291 95.188 < 2e-16 \*\*\*  
## GenderW 14.603558 0.185055 78.915 < 2e-16 \*\*\*  
## Age:GenderW -0.014231 0.004839 -2.941 0.00327 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.64 on 391323 degrees of freedom  
## Multiple R-squared: 0.1746, Adjusted R-squared: 0.1746   
## F-statistic: 2.76e+04 on 3 and 391323 DF, p-value: < 2.2e-16

#plot(mod3)  
  
Dat\_Comb %>%   
 ggplot(aes(x=Age, y=Minutes, color=Gender)) +  
 geom\_point(alpha=0.1) +  
 geom\_smooth(method=lm, linewidth=1.2) +  
 scale\_color\_manual(values = c("M" = "turquoise", "W" = "salmon")) +  
 labs(y="Run Time in Minutes", title="Predicting Runtime by Age and Gender") +  
 theme\_bw()

## `geom\_smooth()` using formula = 'y ~ x'



# Step 1: Filter top 20 runners per year  
top\_runners <- Dat\_Comb.weather %>%  
 group\_by(Year) %>%  
 arrange(Minutes) %>%  
 slice\_head(n = 50) %>%  
 ungroup()  
  
# Step 2: Calculate average running time, temperature, and precipitation  
avg\_running\_time <- top\_runners %>%  
 group\_by(Year) %>%  
 summarize(  
 AvgTime = mean(Minutes, na.rm = TRUE),  
 AvgTemp = mean(TMIN, na.rm = TRUE),  
 AvgPrecip = mean(PRCP, na.rm = TRUE)  
 )  
  
# Step 3: Detrend running time to account for overall improvement over the years  
lm\_trend <- lm(AvgTime ~ Year, data = avg\_running\_time)  
avg\_running\_time <- avg\_running\_time %>%  
 mutate(DetrendedTime = resid(lm\_trend))  
summary(lm\_trend)

##   
## Call:  
## lm(formula = AvgTime ~ Year, data = avg\_running\_time)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.6193 -1.1076 -0.1760 0.7372 12.9387   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -12.85659 55.68918 -0.231 0.819  
## Year 0.03199 0.02781 1.150 0.256  
##   
## Residual standard error: 2.342 on 42 degrees of freedom  
## Multiple R-squared: 0.03055, Adjusted R-squared: 0.007468   
## F-statistic: 1.324 on 1 and 42 DF, p-value: 0.2565

# Step 4: Perform correlation analysis  
cor\_matrix <- avg\_running\_time %>%  
 select(DetrendedTime, AvgTemp, AvgPrecip) %>%  
 cor(use = "complete.obs")  
  
print("Correlation Matrix:")

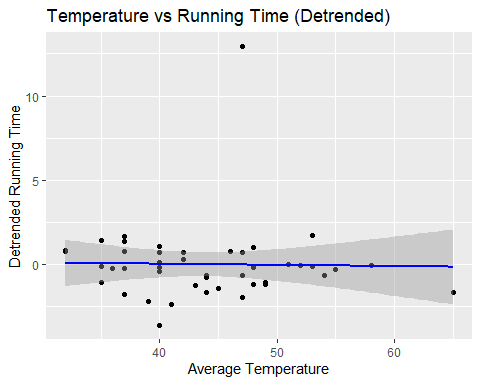
## [1] "Correlation Matrix:"

print(cor\_matrix)

## DetrendedTime AvgTemp AvgPrecip  
## DetrendedTime 1.00000000 -0.02184304 0.01166729  
## AvgTemp -0.02184304 1.00000000 0.26622083  
## AvgPrecip 0.01166729 0.26622083 1.00000000

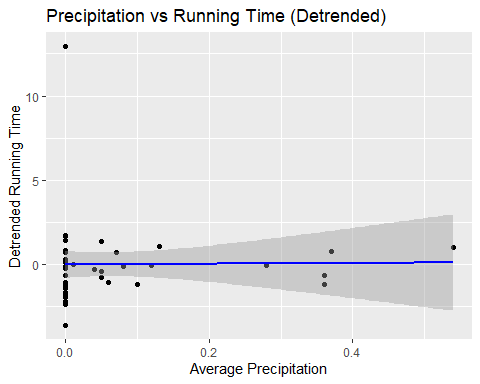
# Step 5: Visualization of relationships  
# Temperature vs Detrended Running Time  
ggplot(avg\_running\_time, aes(x = AvgTemp, y = DetrendedTime)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", col = "blue") +  
 labs(title = "Temperature vs Running Time (Detrended)",  
 x = "Average Temperature",  
 y = "Detrended Running Time")

## `geom\_smooth()` using formula = 'y ~ x'



# Precipitation vs Detrended Running Time  
ggplot(avg\_running\_time, aes(x = AvgPrecip, y = DetrendedTime)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", col = "blue") +  
 labs(title = "Precipitation vs Running Time (Detrended)",  
 x = "Average Precipitation",  
 y = "Detrended Running Time")

## `geom\_smooth()` using formula = 'y ~ x'

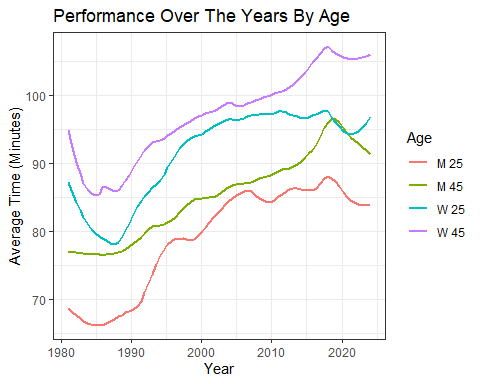


# Step 6: Linear regression analysis  
lm\_weather <- lm(DetrendedTime ~ AvgTemp + AvgPrecip, data = avg\_running\_time)  
summary(lm\_weather)

##   
## Call:  
## lm(formula = DetrendedTime ~ AvgTemp + AvgPrecip, data = avg\_running\_time)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.6315 -1.1382 -0.1751 0.7123 12.9867   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.355926 2.261270 0.157 0.876  
## AvgTemp -0.008593 0.051827 -0.166 0.869  
## AvgPrecip 0.349098 3.004812 0.116 0.908  
##   
## Residual standard error: 2.37 on 41 degrees of freedom  
## Multiple R-squared: 0.0008061, Adjusted R-squared: -0.04794   
## F-statistic: 0.01654 on 2 and 41 DF, p-value: 0.9836

# Filter data for specific age groups and genders  
filtered\_data <- Dat\_Comb.weather %>%  
 filter(Age %in% c(25, 45), Gender %in% c("M", "W")) %>%  
 mutate(  
 Age\_Group = paste(Gender, Age, sep = " ")  
 )  
  
# Summarize performance (e.g., average time in minutes) by year and age group  
performance\_by\_age <- filtered\_data %>%  
 group\_by(Year, Age\_Group) %>%  
 summarise(Average\_Time = mean(Minutes, na.rm = TRUE), .groups = "drop")  
  
# Plot performance over age with smoothed lines  
ggplot(performance\_by\_age, aes(x = Year, y = Average\_Time, color = Age\_Group)) +  
 geom\_smooth(se = FALSE, method = "loess", span=0.25,size = 1) + # Add smoothed lines  
 labs(  
 title = "Performance Over The Years By Age",  
 x = "Year",  
 y = "Average Time (Minutes)",  
 color = "Age"  
 ) +  
 theme\_bw()

## `geom\_smooth()` using formula = 'y ~ x'



top25 <- Dat\_Comb.weather %>%  
 group\_by(Year, Gender) %>%  
 arrange(Minutes) %>%  
 slice\_head(n = 25) %>%  
 ungroup()  
  
# Calculate the mean of the top 25 for each Year and Gender  
mean\_top25\_times <- top25 %>% filter(Gender %in% c("M", "W") ) %>%  
 group\_by(Year, Gender) %>%  
 summarise(mean\_minutes = mean(Minutes, na.rm = TRUE)) %>%  
 ungroup()

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

# Filter the data to exclude years before 1983 and specific years 2015 and 2020  
filtered\_data <- mean\_top25\_times %>%  
 filter(Year > 1982, !Year %in% c(2015, 2020))  
  
ggplot(filtered\_data, aes(x = Year, y = mean\_minutes, color = Gender)) +  
 geom\_point() +  
 geom\_smooth(method = "loess",scale=0.2, se = TRUE, aes(fill = Gender)) +  
 labs(  
 title = "Mean Top 25 Time by Year and Gender over the Years",  
 x = "Year",  
 y = "Mean Minutes"  
 ) +  
 theme\_bw()

## Warning in geom\_smooth(method = "loess", scale = 0.2, se = TRUE, aes(fill =  
## Gender)): Ignoring unknown parameters: `scale`

## `geom\_smooth()` using formula = 'y ~ x'

