Examining the Relationship Between Environmental Conditions and Marathon Performance in Male and Female Runners of Different Ages

Thomas. B Arnold

1. Introduction

1.1 Background Information

The following project is based on the research of Dr. Brett Romano Ely and Dr. Matthew Ely, of Providence College's Department of Health Sciences. Dr. Ely's research investigates the relationship between environmental conditions and marathon performance in male and female runners of various ages. The underlying dataset tracks the environmental conditions and best single-age performances in men and women, ages 14-85, in five major marathons run over the course of seventeen to twenty-four years. Our analysis also considers air quality, which Dr. Ely cited as a potential future direction for research; it is well-known that air quality affects cardiorespiratory capacity (Stieb et al. 2018).

Across all runners, endurance exercise performance decreases as Wet Bulb Globe Temperature (WBGT) rises (B. R. Ely et al. 2010), and this decrease in performance is more evident in marathons and other long distance competitions (M. R. Ely et al. 2007). WBGT is the weighted average of dry bulb temperature, wet bulb temperature (which factors in humidity), and black globe temperature (which factors in solar radiation) in Celsius.

Older runners tend to be particularly impacted by increased heat due to physiological thermoregulatory issues (Kenney and Munce 2003). Reduced sweat gland output, reduced skin blood flow, smaller increases in cardiac output, and decreased redistribution of blood flow from renal and splanchnic circulations may all contribute to older individuals' reduced heat stress tolerance (Kenney and Munce 2003). Over time, the effect of age on performance was worse for female runners (Ely, 2024, in progress). In the New York City marathon, both male and female "masters" runners (i.e., runners over the age of 40) tend to perform less well as temperature and humidity increase (Knechtle et al. 2021). However, older marathon runners tend to have less pace variance than younger ones (Nikolaidis et al. 2019).

While endurance performance (Besson et al. 2022) and thermoregulation processes (Yanovich, Ketko, and Charkoudian 2020) differ across sex, the effects of weather are less evident in female runners' performances (Vihma 2010). Men are more likely than women to slow over the course of a marathon (Deaner et al. 2015).

1.2 Data Sources & Collection Methods

The dataset includes marathon results and environmental condition data for Boston, Chicago, New York, Twin Cities, and Grandma's Marathons, run during periods of 17 to 24 years (1993-2016). Dr. Ely noted that the dataset ends in 2016 because advances in running shoe technology led to course records decreasing at an accelerated rate. The earliest years included in this dataset differ between cities (1993 for New York City, 1996 for Chicago, 1998 for Boston, and 2000 for Twin Cities and Grandma's).

The five races differ in some important respects reducing the comparability between samples, as reflected in the analysis in Section 3. Boston has higher qualification requirements than other marathons, such that fewer and different runners at the extremes of age are likely to compete. The Boston and Grandma's marathons are run at different times of year (April and June, respectively) than the other races (October to November). The courses also differ in elevation changes at different parts of the courses; New York, for example, has a large hill near the end of the course, which may affect runners differently based on their age or other factors.

The fastest finishing time at each age for male and female runners was compared with the course record time, measured as a percent of the course record for that runner's sex (`% CR`). The original dataset did not provide runners' actual finish times, but only their `% CR`. Using a dataset containing the course records for each year for males and females, we were able to convert the `% CR` to individual runners' finish times. An additional dataset used contained all dates of the marathons.

The source for the dataset is not clear from the materials provided. Here, we assume that the data sources are similar to those cited in Dr. Matthew Ely's 2007 study, specifically: hourly weather data from the Air Force Combat Climatology Center and marathon race results from the public domain (M. R. Ely et al. 2007). Notably, the precise geography for which weather data is collected is also not indicated, which introduces potential bias. This issue also exists with AQI data and is discussed in detail below.

Environmental condition variables follow. The temperature data, measured in Celsius, includes dry bulb temperature (`Td, C`); wet bulb temperature, which takes humidity into account (`Tw, C`); black globe temperature, which takes solar radiation into account (`Tg, C`); and Wet Bulb Globe Temperature (`WBGT`), which is a weighted average of Td, Tw, and Tg. The dataset also includes percent relative humidity (`% RH`); solar radiation in Watts per meter squared (`SR W/m2`); dew point in Celsius (`DP`); and wind speed in Km/hr (`Wind`). As noted above, the core dataset includes `Wind` speed without an indication of wind direction. As a result, we assume that higher winds make running more difficult, without respect to the direction of either the wind or the runner. The data also includes flags (`Flag`) of different colors depending on the `WBGT` and risk of heat illness: white (`WBGT` <10 C); green (`WBGT` 10-18 C); yellow (`WBGT` >18-23 C); red (`WBGT` >23-28 C); and black (`WBGT` >28 C).

In Dr. Ely's preliminary analysis of these data, races were initially divided into quintiles based on WBGT, after which regression analyses were performed to model finishing time by age, gender, and WBGT. Here, we take an approach allowing for analysis of temperature in its continuous form.

Dr. Ely's analysis also did not include air quality and noted air quality as a future direction, stating that previous work indicated that air quality may affect women more than men. We use air quality data from the `RAQSAPI` library in R to fetch air quality index (AQI) data for marathon days. The source of the data used in this package is the EPA's Air Quality System (AQS) Data Mart API v2 Air Quality System interface, containing every measured value collected by the EPA via the national ambient air monitoring program and aggregate values. We retrieve four separate parameters from AQS from which `AQI` is estimated: PM2.5 Local Conditions; PM10 Total 0-10 um STP (standard temperature and pressure); Acceptable PM2.5 AQI & Speciation Mass; and Ozone. Units are omitted because in the AQI dataset these different parameters are themselves used to estimate a single value of `AQI`, which is a unitless measure of air quality.

We note two major limitations in our use of air quality data: geographic area inaccuracy and effects of aggregation. First, for geography, all marathons cover large geographic areas, but some routes loop back towards their origin. (Chicago and Twin Cities Marathons are closer to loops; Boston, New York City, and Grandma's Marathons are more linear.) Notably, in Dr. Ely's 2007 paper, it states that the specific years of marathons used were chosen to ensure that none of the courses changed within the study period; as discussed above, we do not know the source of our current data, but assume that the courses remained the same throughout the period at issue, as with the 2007 paper. Each of the marathons also run through downtown areas. Air quality is likely to differ significantly over the course of a race based on proximity to highways, airports, etc.

Data extracted from the AQS uses the core based statistical area (CBSA), a larger geographical area than the marathon routes, incorporating suburban and exurban areas, which could lead to an underestimate of the AQI. A related limitation is that the AQS data are averaged over stations within the given CBSA because we do not know which stations are close to the marathon route, which is a potential area of refinement for future studies. This general note on geographic precision also applies to the weather variables present in the core marathon dataset, as noted above.

Turning to aggregation, data extracted from the AQS retrieves the four parameters from which AQI is estimated, discussed above (PM2.5 Local Conditions, etc.). We average the four parameters, but the effects of these parameters on cardiorespiratory performance differ. Averaging them may thus underestimate the effect of a particular type of pollution on performance. However, aggregation across parameters reduces the impact of missing data compared to using just one parameter, because we observe in the AQS dataset some missingness in individual parameter types.

Each row of the AQS dataset includes a date, station, parameter in the form of pollutant type sampled, duration of sampling, and the `AQI` value from that sample. Within one date and sometimes within one station, we aggregate dozens of estimates across the parameters and times, in part because, depending on the parameter used to estimate `AQI`, there can be high variability in `AQI` within one area in a single day, as different pollutants can be higher or lower and are not always correlated.

Additional limitations applicable to both the air quality data and Dr. Ely's environmental data follow. Because the Boston and Grandma's Marathons are run in April and June, while the others are run in October and November, different races had different environmental conditions, including `AQI`. `AQI` is typically much lower at night and in the morning, due to fluctuations in industrial and automotive activity, but `AQI` estimates are aggregated over the whole day. (We note that the AQS data provide a sample's duration but not the time the sample began to be taken.) And weather conditions change over the course of the day, with temperature typically lower in the morning. Currently, all five marathons start between 7:30 a.m. and 9:40 a.m.; we assume that this was the case during the period of time contained within the dataset. Runners also start the race at different times, with elite runners beginning earlier, and individual runners may take more or less time to complete the course. Individuals released later or spending more time running will likely experience different `AQI` and weather conditions than those released earlier and completing faster.

2. Data Preparation and Cleaning

2.1 Import, Initial Inspection, & Merging Datasets

Checks for consistency are conducted. Because variables are not properly named and labeled by default, recoding is conducted using the codebook. We ensure that `Race`, `Sex`, and `Flag` are correctly coded according to the code sheet. Categorical variables are converted to factors with labels.

Course records are merged with the core provided dataset. Actual race times are computed using `% CR` and course record times. Marathon dates and course record datasets are merged on keys `Race`, `Year`, `Date`, `Sex`. As discussed above, `AQI` are aggregated across CBSA, the day of the marathon, and the pollution parameters to create one `AQI` value for a given marathon day. `AQI` readings are merged to marathon dates and locations. Dates are also merged to help with merging of `AQI`.

For data verification, the number of observations before and after merging are checked to ensure no data loss. Spot checks are performed to confirm accurate merges. We verify that there are the same number of records for each race in our final merge dataset as in the original provided dataset and that no duplicate rows exist. Finally, we spot check the dataset to ensure that newly added variables appear as expected in association with their counterparts.

2.3 Data Cleaning

First, we further prepare the dataset by generating a variable which quantized `Age` into `Age Group` (0-17, 18-24, 25-34, 35-44, 45-59, 60-69, 70+). For the purposes of consistency in our sample, the 2013 running of the Boston Marathon is removed, as a terrorist attack occurred during the running of the marathon, which likely affected the time of those finishing subsequent to the attack.

We note that Dr. Ely's analysis indicated that ninety-two race years yielded 6,112 male finishers, ages 14 to 91, and 5,452 female finishers, ages 14 to 88. We observe different numbers of unique marathon and year pairings (the number of marathons for which data is collected) depending on the source of the data; 97 in the course records dataset, 98 in the marathon dates dataset, 98 in the ACS AQI dataset, and 96 in the core marathon dataset. This results in 96 race years in our final dataset, which does not conform to Dr. Ely's preliminary analysis, indicating a possible data processing error or misstatement. Our dataset includes 96 races but the same number of male and female finishers (6,112 and 5,452 respectively) as noted by Dr. Ely. Our data also contains the same age range for males (ages 14 to 91) as cited by Dr. Ely, but a different range for females (ages 15 to 88). These discrepancies support the idea that the 92 race years cited initially was a misstatement, and that data processing issues may exist.

Plausibility checks are also performed to ensure proper variable coding in terms of units and potential data entry errors. Histograms are examined for continuous variables but omitted for space. Categorical variables (e.g., `Sex`) are checked to ensure that they only contain the factor levels that they are allowed to take by the codebook. Plausible ranges are defined as follows: 13 to 100 years for `Age`; -10% to 400% for `% CR`; world record pace (2:00:35 for men, 2:11:53 for women) to 10 hours for `Time`; -18 to 42 degrees Celsius for `Td, C`, `Tw, C`, and `Tg, C`; 5% to 100% for `% RH`; 0 to 1500 W/m2 for `SR W/m2`; -30 to 36 degrees Celsius for `DP`; 0 to 150 Km/hr `Wind`; -18 to 42 degrees Celsius for `WBGT`; and 5 to 500 for `AQI`.

Some variables do not pass sanity checks and therefore must be evaluated further, specifically: `% CR`, with a maximum value of approximately %420; `Time`, with a maximum value of over 10 hours; `Tg, C`, with a maximum value of 44 degrees Celsius, or 111 degrees Fahrenheit; and `% RH`, with multiple values under 5%. A single `Tg, C` value of approximately 111 degrees Fahrenheit is not entirely implausible given that `Tg, C` incorporates solar energy and thus can be higher on a sunny race day than intuition might suggest. Some values of `% RH`, which should be number from 0-100, have been miscoded as between 0 and 1; we take all `% RH` values between 0 and 1 and multiply by 100 to rectify this issue.

For data cleaning, `Time` is examined extensively because of its tangible units. None of the times were faster than world record pace by gender, but there were two finishers over 10 hours (approx. 2.6 mph pace) and several over 9 (approx. 2.9 mph pace). All finishers over 10 hours are 83 years old. All finishers over 9 hours are 74 or older. These times could represent people who do not share common factors with the other runners and might therefore weaken or bias the conclusions. However, running can be defined as ambulating wherein at a given moment in a stride, neither foot is making contact with the ground; this definition has no minimum speed. Thus we do not remove any observations based on their finishing times.

None of these slow finishers are in the Boston Marathon, emphasizing its status as the most different from the other races. The five races have different qualification criteria; Boston is known to have the strictest qualifying standards. We can assume that top-level runners are running each marathon, but lower-level runners, who are slow relative to top overall finishers, but nevertheless finish first in their age and gender bracket, may qualify for some but not all marathons. This heterogeneity in marathon samples is a persistent issue in our analysis which we discuss in more depth below. We choose not to exclude the slow finishers here, but this specific issue (as well as the general issue of the marathon races not having similar samples) should be explored in more depth in an additional analysis.

2.4 Assessing & Handling Missing Data

We observe that there is missingness only in the weather variables, and that when one of these variables is missing, they are all missing within a given observation. This appears to be the case for discrete groups of observations, which we surmise to be for individual years. We note that data is missing in the year 2011 and 2012. In 2011 there are 126 observations with missing data for the Chicago Marathon, 131 observations with missing data for the New York City Marathon, and 118 observations with missing data for the Twin Cities Marathon (Minneapolis, MN). In 2012 there are 116 observations with missing data for the Grandma's Marathon (Duluth, MN). This is likely an issue with missing data in the weather data source from the original core dataset. We observe that this missingness originates from weather missing in its entirety from four dates, each of which contains multiple runner-observations as noted above.

Next we determine the missingness mechanism, i.e., whether data are Missing Completely at Random (MCAR), Missing at Random (MAR), or Missing Not at Random (MNAR). Tests are performed to confirm the type of missingness. Data is likely not MCAR, as indicated by Little's MCAR test. For each variable, a missing indicator is produced which takes values zero if missing and one if not missing. Subsequently, this missing indicator is regressed on all variables (excluding the variable from which the given missing indicator is produced). If any of the predictor variables are significant ($\alpha = 0.05$), we consider this evidence against a MAR designation for the variable from which the missing indicator was produced, and would be considered a test failure (the opposite case being test passage). When this procedure is run for our dataset, we note that each variable passes this test. This is evidence that the missing data is MAR, although we cannot be certain because the data could be MNAR due to unobserved covariates which our test procedure cannot check. We proceed under a MAR assumption.

Finally, where missing data was minimal, we consider whether to remove those records. Given that the missing data is from a relatively small subset of the sample, and the strong evidence of MAR, missing data is excluded from our analysis. Imputation was considered, but due to the small amount of missing data relative to the sample size, we opt to exclude. Additionally, data is potentially recoverable from historical sources, but there is no R package to accomplish this, and because we do not know how the geographies were defined for the original weather data, adding weather data could be more biasing than simply dropping the records.

3. Exploratory Data Analysis

3.1 Comprehensive Descriptive Statistics (Table 1)

Beginning with observations applicable to multiple variables, no variables pass the Shapiro-Wilk test for normality. This is not necessarily a problem for our future modeling strategies, as linear models do not assume normally distributed variables, but do assume normally distributed errors and linear relationships between explanatory and response variables. Q-Q plots were checked in relation to normality for variables, but are omitted for space; normality results are discussed herein. The large sample size of this dataset ($n \geq 5000$) means all statistical tests could potentially appear significant at higher rates than for a lower sample size, given the large amount of data on which to base the distributional assessment. Small deviations from normal in the sampling distribution could cause said distribution to be identified as non-normal, whereas in smaller samples, those deviations might be identified as borderline cases instead. Performance metric `% CR` (and therefore `Time`) contain outliers; as discussed above, we elect not to remove the outliers for `% CR` and `Time` values from the sample.

Full individual variable details are omitted, data is available in Table 1. Both `Age` and `Year` are non-normal, non-outlier producing, platykurtic and centered, similar to a Uniform distribution. `Age Group` for 0-17 years has a very small sample. For `% CR`, the outliers are discussed above; that it is leptokurtic and right-skewed suggests a not insignificant number of slow finishers.

`Flag` has 2,000 or more runner-observations for white, green, and yellow flags, but only 592 red and no black flag runner-observations. The red flag runner-observations represent five marathon events: Boston ’12, Chicago ’07, Twin Cities ’07, Grandma’s ’06, and Grandma’s ’16. Taken together, this suggests an issue with `Flag` as a predictor due to the imbalanced classes.

Table 1: Summary of Performance Variables

Variable	Type	Summary	Normal Distribution	Outlier(s) Present	Skewness	Kurtosis
Race [n (%)]	Categoric	Boston Marathon: 1977 (17.26%), Chicago Marathon: 2553 (22.29%), New York City Marathon: 2930 (25.58%), Twin Cities Marathon (Minneapolis, MN): 1993 (17.4%), Grandma’s Marathon (Duluth, MN): 2000 (17.46%)	NA	NA	NA	NA
Year [Mean (SD) (Quantile)]	Numeric	2006.55 (5.97) (2002 - 2011)	No	No	Centered	Platykurtic
Sex [n (%)]	Categoric	Female: 5401 (47.16%), Male: 6052 (52.84%)	NA	NA	NA	NA
Age [Mean (SD) (Quantile)]	Numeric	46.49 (17.99) (31 - 61)	No	No	Centered	Platykurtic
Age Group [n (%)]	Categoric	Age 0-17: 469 (4.09%), Age 18-24: 1329 (11.6%), Age 25-34: 1900 (16.59%), Age 35-44: 1900 (16.59%), Age 45-59: 2847 (24.86%), Age 60-69: 1780 (15.54%), Age 70+: 1228 (10.72%)	NA	NA	NA	NA
% CR [Mean (SD) (Quantile)]	Numeric	49 (45.03) (19.03 - 63.51)	No	Yes	Right-skewed	Leptokurtic

Note:
Shapiro-Wilk test for normality; Grubb’s test for outliers.

We next look to variables by `Race` (2). The Kruskal-Wallis test was performed for continued variables, the Chi-Square test was used for categorical variables, and Bonferroni correction for the P-values was applied for all. All variables other than `Sex` were statistically significantly different across `Race` from at least one other `Race` for a given variable. This suggests that the marathons are very different in terms of all variables other than `Sex`. `Age`, `% CR`, `Time`, and all environmental variables were significantly different across `Race`. For a future regression analysis, we note that these marathon samples are natural groups with potential unobserved differences between each marathon. It is appropriate to attempt to account for this by including a random effect for intercept by marathon.

Full individual variable details are omitted, data is available in Table 2. The New York City Marathon is noticeably the oldest, which could introduce bias.

Grandma’s Marathon, the only race in the summer, had significantly worse `Flag` conditions and warmer temperatures. Boston had the worst air quality. These are not all of the differences between races. As our above analysis indicates, these samples are heterogeneous, in many ways more different than they are alike.

Table 2: Summary of Variables by Race

Variable	Boston Marathon	Chicago Marathon	New York City Marathon	Twin Cities Marathon (Minneapolis, MN)	Grandma’s Marathon (Duluth, MN)	Sig.
Year [Mean (SD) (Quantile)]	2007.43 (5.18) (2003 - 2011)	2006.16 (6.04) (2001 - 2011)	2004.21 (6.89) (1998 - 2010)	2008.09 (4.89) (2004 - 2012)	2008.09 (4.9) (2004 - 2012)	****

(continued)

Variable	Boston Marathon	Chicago Marathon	New York City Marathon	Twin Cities Marathon (Minneapolis, MN)	Grandma's Marathon (Duluth, MN)	Sig.
Sex [n (%)]	Female: 933 (47.19%), Male: 1044 (52.81%)	Female: 1210 (47.4%), Male: 1343 (52.6%)	Female: 1402 (47.85%), Male: 1528 (52.15%)	Female: 922 (46.26%), Male: 1071 (53.74%)	Female: 934 (46.7%), Male: 1066 (53.3%)	ns
Age [Mean (SD) (Quantile)]	47.04 (17.3) (32 - 61)	45.82 (17.89) (30 - 61)	49.57 (18.76) (33 - 65)	44.72 (17.44) (30 - 59)	44.03 (17.51) (29 - 58)	****
% CR [Mean (SD) (Quantile)]	41.69 (34.34) (18.24 - 56.93)	51.61 (47) (20.38 - 66.78)	54.88 (55.89) (18.59 - 68.88)	45.66 (36.58) (18.62 - 63.1)	47.59 (39.93) (19.51 - 61.72)	****
Flag [n (%)]	White: 929 (46.99%), Green: 810 (40.97%), Yellow: 115 (5.82%), Red: 123 (6.22%), Black: 0 (0%)	White: 732 (30.16%), Green: 1459 (60.12%), Yellow: 120 (4.94%), Red: 116 (4.78%), Black: 0 (0%)	White: 1394 (49.8%), Green: 901 (32.19%), Yellow: 504 (18.01%), Red: 0 (0%), Black: 0 (0%)	White: 587 (31.31%), Green: 834 (44.48%), Yellow: 338 (18.03%), Red: 116 (6.19%), Black: 0 (0%)	White: 0 (0%), Green: 702 (37.26%), Yellow: 945 (50.16%), Red: 237 (12.58%), Black: 0 (0%)	****
Td, C [Mean (SD) (Quantile)]	11.79 (6.02) (8.32 - 13.83)	12.42 (6.05) (7 - 15.67)	11.73 (4.67) (7.43 - 15.1)	13.14 (5.52) (9 - 15.67)	18.86 (3.32) (16.95 - 22)	****
Tw, C [Mean (SD) (Quantile)]	7.7 (3.89) (5.38 - 8.23)	8.54 (5.68) (2.51 - 12.94)	7.57 (4.98) (2.93 - 11.52)	9.85 (5.41) (7.33 - 11.07)	14.91 (2.45) (13.67 - 16.9)	****
Tg, C [Mean (SD) (Quantile)]	24.33 (8.6) (19.48 - 28.31)	24.52 (6.3) (19.5 - 28.99)	21.36 (5.93) (18 - 24.96)	24.94 (6.52) (19.63 - 29.63)	31.63 (7.86) (27.63 - 37.67)	****
DP [Mean (SD) (Quantile)]	3.44 (4.56) (0.38 - 6.02)	4.65 (6.86) (-4.14 - 9.63)	2.74 (6.99) (-3.63 - 8.64)	5.96 (7.25) (2.68 - 9.53)	12.43 (3.17) (11.05 - 14.33)	****
WBGT [Mean (SD) (Quantile)]	11.44 (4.63) (8.69 - 12.68)	12.12 (5.76) (6.71 - 16.42)	10.74 (4.91) (6.72 - 14.13)	13.2 (5.35) (8.97 - 14.45)	18.65 (3.22) (17.14 - 21.8)	****
% RH [Mean (SD) (Quantile)]	60.63 (20.77) (45.57 - 72.75)	60.45 (10.49) (51.33 - 65.67)	55.29 (17.38) (43.82 - 61.21)	63.83 (15.59) (55.5 - 76.33)	68.03 (15.61) (55.67 - 87.36)	****
SR W/m2 [Mean (SD) (Quantile)]	652.83 (191.65) (574.05 - 800.25)	460.48 (94.56) (436.62 - 535.54)	401.15 (130.9) (309.48 - 546.19)	435.89 (138.93) (348.03 - 545.33)	676.81 (190.51) (520.03 - 833.18)	****
Wind [Mean (SD) (Quantile)]	11.98 (4.6) (8.33 - 16)	8.21 (3.19) (5.33 - 10.33)	11.22 (4.55) (9 - 14)	8.8 (3.2) (6.25 - 10)	9.16 (2.87) (7 - 11.2)	****
AQI [Mean (SD) (Quantile)]	42.7 (15.56) (33.12 - 46.09)	38.8 (12.47) (30.16 - 45.54)	32.43 (13.81) (23.26 - 37.13)	30.13 (12.83) (21.01 - 36.29)	37.23 (15.58) (30.94 - 42.29)	****

Note:

Kruskal–Wallis test for continuous variables, Chi-Square test for categorical variables. Bonferroni correction applied.

Note:

ns = P > 0.05, * = P ≤ 0.05, ** = P ≤ 0.01, *** = P ≤ 0.001, **** = P ≤ 0.0001

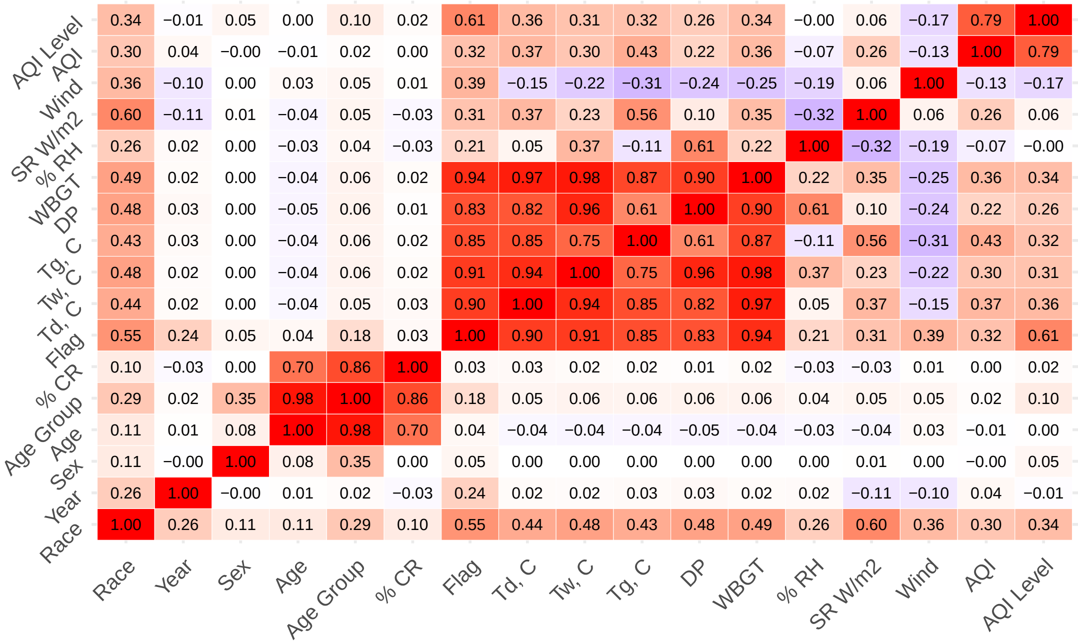
We next examine variables by levels of `Sex` (table omitted for space). Student's T-test was used for continuous variables and the Chi-Square test was used for categorical variables. No Bonferroni correction was applied because there are only two groups. There were no statistically significant differences across variables by `Sex`, with the exception of `Age`. Men were statistically significantly older, with a mean `Age` of 47.8 years compared to a mean `Age` of 45.0 years for women.

Next, we consider differences in variables by `Age Group` (table omitted for space). The Kruskal-Wallis test was performed for continued variables, the Chi-Square test was used for categorical variables, and Bonferroni correction for the P-values was applied for all. There are significant differences by `Age Group` across all variables other than `Year`, `Flag` and `AQI`. Between ages 18-69 years, the five marathons are generally comparable to each other, it is outside this range that large differences occur. The New York City Marathon has proportionally many more individuals age 70 or older (40.3% of the age 70+ sample), which is supported by the analysis in Table 2 indicating that it has the oldest average age. Interestingly, in the age 70 or older group there are many more men than women (70.0% men and 30.0% women compared to 52.8% male and 47.2% female in the total sample), a potential source of bias. Boston has the fewest individuals under 18 (6.2% of the age 0-17 sample), followed by New York City (9.8% of the age 0-17 sample). Twin Cities and Grandma's Marathons have notably higher numbers of individuals under 18 (24.5% and 30.3% of the age 0-17 sample). The significant differences in environmental variables

([2]) are likely caused by Grandma's Marathon being over-sampled for individuals under 18 years of age and taking place during the summer, and thus having the highest temperatures, as discussed above. It is notable that different age groups do not experience significantly different levels of `AQI`.

3.2 Correlation & Multicollinearity Analysis

Figure 1: Correlation Matrix



As might be expected, `WBGT` and the variables that are used to compute it are highly correlated. `WBGT` and the dew point measurement `DP` are also highly correlated; this is likely because `Tw, C`, the wet-bulb temperature, factors in humidity, of which dew point is a measure. Interestingly, `WBGT` is not highly correlated with `% RH`, despite relative humidity being a measurement of the moisture in the air, nor with `SR W/m2`, despite the solar radiation being included within `Tg, C`, itself a factor in `WBGT`. `Wind` is positively correlated with `Flag` but negatively correlated with all temperature and humidity variables. Although `Flag` is based on `WBGT`, it also appears to take into account wind conditions in a manner that is not immediately clear; this is a potential area of refinement in future study. Finally, as discussed above and shown in various tables, `Race` is moderately correlated with all environmental variables, which is a potential limitation in our study, because the races are conducted under different environmental conditions.

Variance inflation factor (VIF) is calculated for all variables. Initial checks for multicollinearity indicate perfect collinearity between some variables. Based on Figure 1, we see that these variables are likely to be the temperature variables and `DP`. We remove all variables which are highly correlated with the main or primary hypothesized explanatory variable `WBGT` from the dataset. Removing these variables alleviates indicated collinearity issues in all potential future regression analysis. One exception is that we choose to retain the `Flag` variable for use in charting below. `Flag` is purportedly a straightforward quantization of `WBGT` similar to `Age Group` and `AQI Level`, and while these variables remain in the dataset, they will not be used in modeling intended to answer the aims.

4. Specific Aim Analyses

Aim 1: Examine Effects of Increasing Age on Marathon Performance in Men and Women

Figure 2: % CR vs. Age

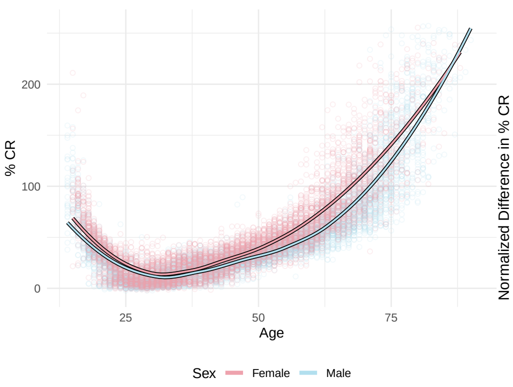
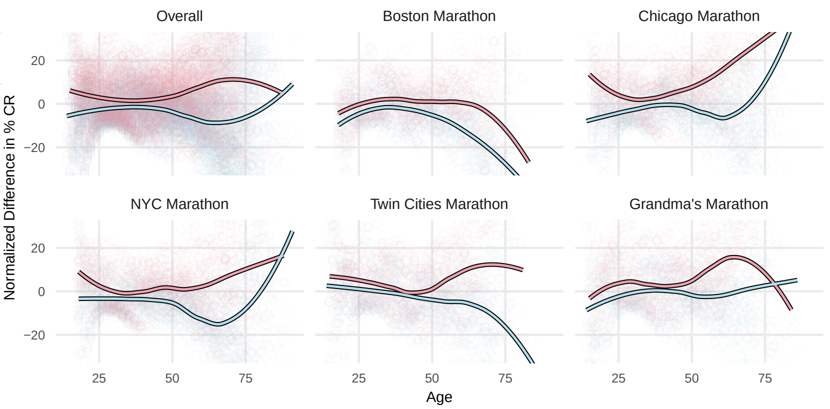


Figure 3: Normalized Difference % CR vs. Age by Race



Scatter plots of `% CR` vs. `Age`, with different colors for `Sex` and plots for each race, were created. There is an evident U-shaped, curvilinear, relationship between `Age` and `% CR` which will inform modeling decisions. Based on the visualizations we see that the curvilinear relationship between age and performance is likely more pronounced for women than men (women accelerate towards optimal finishing time faster as they approach optimal age, and decelerate faster as they age). For correlations, we refer to the correlation matrix Figure 1 and note that for bivariate correlation, there is no correlation between `Sex` and `% CR` (Correlation = 0.00). This is a consequence of marathon course records, and therefore `% CR`, being categorized separately for males and females. This does not preclude a more complex relationship between `Sex`, `Age`, and `% CR` which we will explore below. We observe a strong correlation between `Age` and `% CR` (Correlation = 0.70) which is supported by physiological theory.

We conduct preliminary linear mixed effects regression modeling to aid in answering this aim. Note that we use `Time` as our response variable rather than `% CR` because the latter is computed separately by gender, which could bias our results when including `Sex` and associated interaction terms in the model. To accurately model the complex relationship between gender and age we include quadratic terms for `Age` and both the linear interaction of `Sex` and `Age` as well as the interaction between `Sex` and quadratic `Age`. We include a random slope for `Race` and include as controls `CR` and `Year` to adjust for the differential difficulty between race courses and across years.

Specific coefficient interpretations, model diagnostics, interaction plots, and predicted value plots are omitted as outside of scope for this report. Future work should address this weakness. General interpretation of the results is as follows: our model indicates that performance follows a U-shaped curve with age for both sexes; the curve is more pronounced for women. Women reach peak performance slightly earlier (around 35.4 years) compared to men (37.1 years). As in Dr. Ely's preliminary modeling, this is an overestimate of the age of peak performance likely caused by the inflexibility of the linear model. As runners age beyond their peak, women's performance declines more rapidly than men's. Specifically, women's times increase by about 7.06 seconds per year squared, while men's increase by 6.22 seconds per year squared. This results in a widening performance gap between sexes in older age groups, despite women showing faster improvement rates in younger ages.

Aim 2: Explore the Impact of Environmental Conditions on Marathon Performance, and Whether the Impact Differs Across Age and Gender

Figure 4: Performance vs. Age by Flag

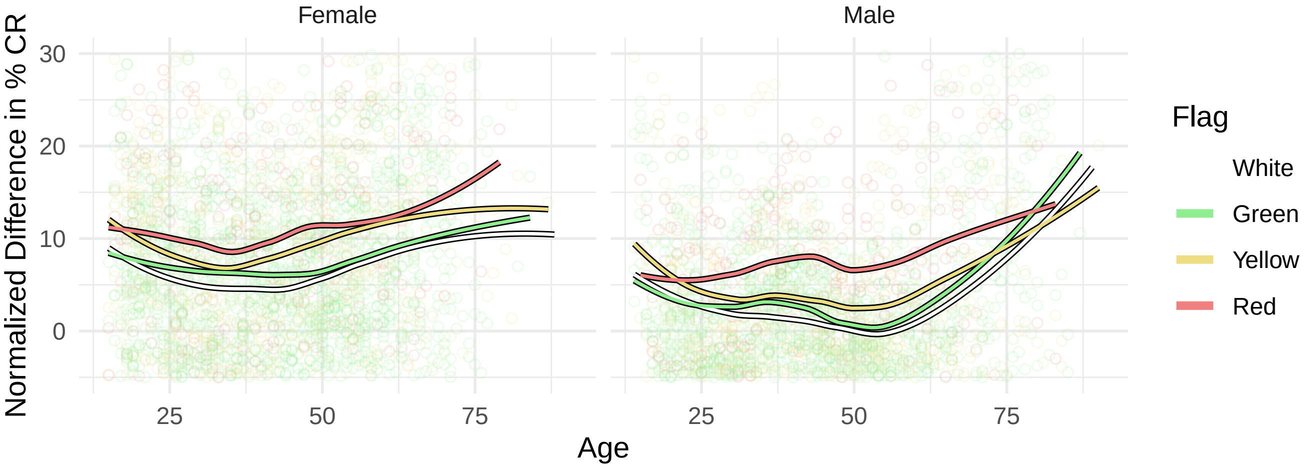
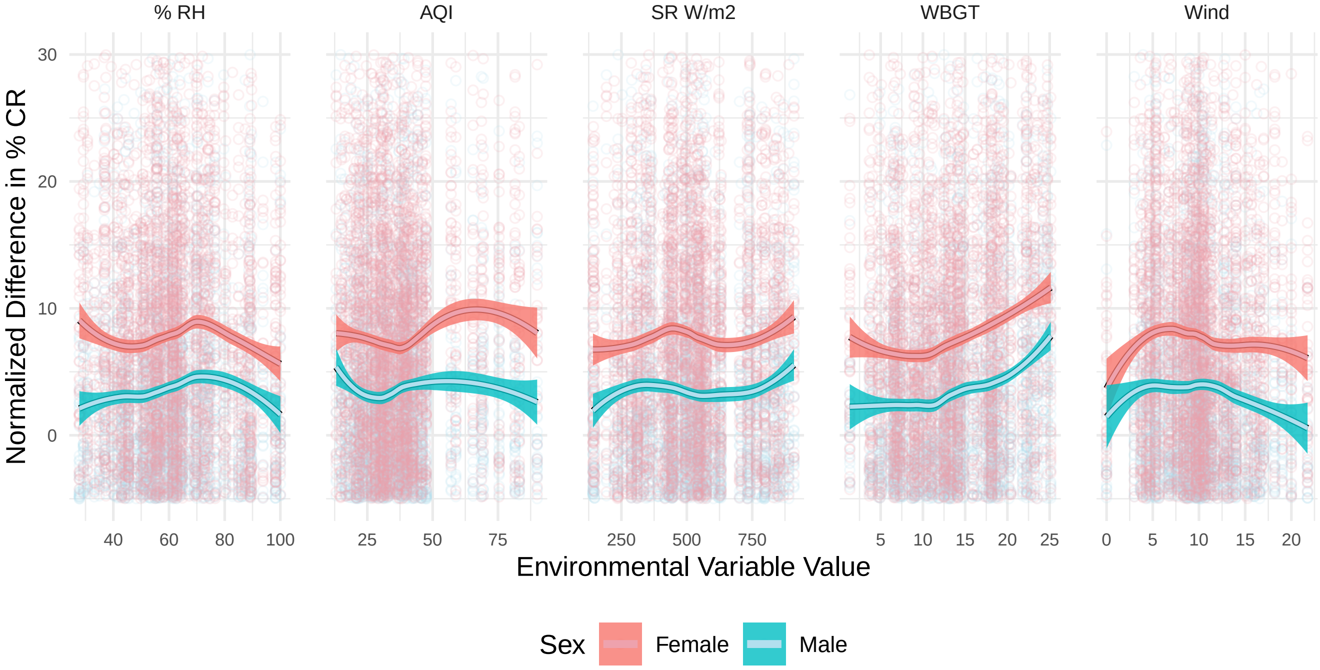


Figure 5: Performance vs. Weather by Sex



Scatter plots were created to compare `% CR` against environmental variables. We see that there are very weak relationships between the environmental variables and `% CR` (5), if any is present at all. We observe weak correlations betwen weather variables and performance Figure 1. An additional plot of `% CR` by `Age` stratified by `Flag` is generated (4). We note that this plot indicates a possible weak relationship between performance and `Flag` (and therefore likley a relationship between performance and `WBGT`).

As above, we conduct preliminary linear mixed effects regression modeling to aid in answering this aim. We produce two models, one with interactions between environmental variables and `Age` as well as between environmental variable and `Sex`, and one without. Specific coefficient interpretations, model diagnostics, interaction plots, and predicted value plots are omitted as outside of scope for this report;

future work should address this weakness. General interpretation of the results is as follows. For the model without interactions, we found that environmental conditions significantly impact marathon performance. Higher WBGT and stronger Wind slow runners down (32.34 seconds per WBGT unit and 11.36 seconds per Wind unit, respectively). Surprisingly, higher humidity is associated with slightly faster times (-4.49 seconds per 1% increase in % RH); we note that the relationship between % RH and performance is likely U-shaped. Solar radiation and air quality show minimal or non-significant effects. This model doesn't account for how these impacts might vary by age or gender.

To produce the second model, we add interaction terms for all environmental variables with both Age and Sex. Environmental impacts on marathon performance differ across Age and Sex. WBGT has a more pronounced effect on women and older runners (0.61 seconds more per year of Age per WBGT unit). Men are less affected by WBGT but more susceptible to poor air quality. Higher humidity benefits older runners more (-0.14 seconds per year of Age per 1% increase in % RH). Wind slows all runners equally (24.62 seconds per unit), regardless of age or gender.

Aim 3: Identify the Weather Parameters that Have the Largest Impact on Marathon Performance

First we observe based on Figure 1 that there are extremely weak correlations between weather variables and % CR. However, this does not preclude the relationship from being statistically significant as we see in the analysis above in Aim 2. To evaluate variable importance in a model-agnostic manner, we use a technique known as permutation importance. This analysis indicates that the most important environmental variable in predicting marathon performance (using Time as opposed to % CR for the reasons noted above) is WBGT. WBGT is 4.68 times more important than the next most important environmental variable, % RH. The remaining environmental variables, SR W/m2, Wind, and AQI are successively less important and each less important than % RH. All environmental variables are far less consequential to finishing time than Age, Sex, and other control variables (for example, WBGT is 285.87 times less important than Age).

5. Conclusion

5.1 Summary of Findings

The previous preliminary analysis on this data, performed by Dr. Ely, yielded a U-shaped relationship between age and performance, which was affected by gender, and found that WBGT affected % CR, impacted by both age and gender. Aim 1 analysis indicates marathon performance follows a U-shaped age curve, more pronounced for women. Women peak earlier (35.4 years) than men (37.1 years) and decline faster post-peak (7.06 vs 6.22 seconds/year squared). This leads to a widening gender performance gap in older ages, despite women's faster improvement in younger years. These findings emphasize the need for age and sex-specific training approaches.

Aim 2 analysis indicates environmental conditions impact marathon performance differently across age and gender. Higher WBGT and wind slow runners, with WBGT affecting women and older runners more. Men are less impacted by WBGT but more by poor air quality. Higher humidity unexpectedly benefits older runners. Wind affects all equally. These results highlight the importance of considering both environmental and individual factors in marathon preparation.

Aim 3 analysis indicates WBGT is the most crucial environmental factor for marathon performance, significantly outweighing other environmental variables. However, runner characteristics like Age and Sex remain far more influential than any environmental factor. This suggests that while race organizers should prioritize WBGT considerations, individual attributes are still the primary determinants of marathon performance. Results are discussed in depth above.

5.2 Limitations

Limitations include the following. We only have the top finisher at each age, which may produce selection bias for high performers. Data is limited for the youngest and oldest runners, with many fewer women than men over 70 years of age; and New York City and Grandma's Marathons have proportionally more 70+ year olds and 0-18 year olds, respectively, than we would expect based on their share of the total sample. Additionally, as discussed, the very slow times to finish included in the dataset could bias our conclusions.

Another limitation is that the source of the Dr. Ely data is not specified, and as noted above we assume that the sources are the same as those used in Dr. Matthew Ely's 2007 analysis. we also do not know the geography for which weather data was collected. Moreover, as noted, it is unclear why `Flag` and `WBGT` have opposite correlation directions with `Wind`. As discussed above, races differ from each other in important ways. `Race` is moderately correlated with all environmental variables, which is a potential limitation in our study, because the races are heterogeneous and conducted under different environmental conditions (discussed in detail above).

Moreover, as noted, overly general geographies and overly general time aggregation are limitations for both environmental conditions and AQI analyses, and aggregation over air quality parameters is a further limitation for AQI analysis. Weather conditions may affect an average runner more or differently. As is also noted above, experiences of weather conditions and AQI may differ for later starters and those who spend longer on the course, including because temperature and AQI generally increase through the day. As also discussed, for `AQI Level`, no days are in groups worse than moderate, which could lead to a lack of differentiation in the dataset.

We also omit (as discussed in detail above) specific coefficient interpretations, model diagnostics, interaction plots, and predicted value plots as outside of scope for this report. Additionally, we do not engage in through variable selection procedure. We also do not explore non-linear models, which may be superior to linear models in this case given the complex non-linear relationships between performance, `Age`, `Sex`, and the environmental variables. Finally, despite the good sample size, this analysis is limited by a lack of cross validation (K-fold or otherwise). These are all weaknesses that should be addressed in future work.

References

Besson, Thibault, Robin Macchi, Jeremy Rossi, Cédric YM Morio, Yoko Kunimasa, Caroline Nicol, Fabrice Vercruyssen, and Guillaume Y Millet. 2022. "Sex Differences in Endurance Running." *Sports Medicine* 52 (6): 1235–57.

Deaner, Robert O, Rickey E Carter, Michael J Joyner, and Sandra K Hunter. 2015. "Men Are More Likely Than Women to Slow in the Marathon." *Medicine and Science in Sports and Exercise* 47 (3): 607.

Ely, Brett R, Samuel N Cheuvront, Robert W Kenefick, and Michael N Sawka. 2010. "Aerobic Performance Is Degraded, Despite Modest Hyperthermia, in Hot Environments." *Med Sci Sports Exerc* 42 (1): 135–41.

Ely, Matthew R, Samuel N Cheuvront, William O Roberts, and Scott J Montain. 2007. "Impact of Weather on Marathon-Running Performance." *Medicine and Science in Sports and Exercise* 39 (3): 487–93.

Kenney, W Larry, and Thayne A Munce. 2003. "Invited Review: Aging and Human Temperature Regulation." *Journal of Applied Physiology* 95 (6): 2598–2603.

Knechtle, Beat, David Valero, Elias Villiger, José Ramón Alvero Cruz, Volker Scheer, Thomas Rosemann, and Pantelis T Nikolaidis. 2021. "Elite Marathoners Run Faster with Increasing Temperatures in Berlin Marathon." *Frontiers in Physiology* 12: 649898.

Nikolaidis, Pantelis T, Stefania Di Gangi, Hamdi Chtourou, Christoph Alexander Rüst, Thomas Rosemann, and Beat Knechtle. 2019. "The Role of Environmental Conditions on Marathon Running Performance in Men Competing in Boston Marathon from 1897 to 2018." *International Journal of Environmental Research and Public Health* 16 (4): 614.

Stieb, David M, Robin Shutt, Lisa Marie Kauri, Gail Roth, Mieczyslaw Szyszkowicz, Nina A Dobbin, LI Chen, et al. 2018. "Cardiorespiratory Effects of Air Pollution in a Panel Study of Winter Outdoor Physical Activity in Older Adults." *Journal of Occupational and Environmental Medicine* 60 (8): 673–82.

Vihma, Timo. 2010. "Effects of Weather on the Performance of Marathon Runners." *International Journal of Biometeorology* 54: 297–306.

Yanovich, R, I Ketko, and N Charkoudian. 2020. "Sex Differences in Human Thermoregulation: Relevance for 2020 and Beyond." *Physiology* 35 (3): 177–84.

Appendix I: Script Code

```r
# --- Preamble ---
# Date of last update: Oct. 7, 2024
# R Version: 4.3.1
# Package Versions:
#   tidyverse: 2.0.0
#   knitr: 1.45
#   kableExtra: 1.3.4
#   ggplot2: 3.4.3
#   naniar 1.0.0
#   visdat 0.6.0
#   car 3.1-2
#   lme4 1.1-34
#   ggpubr 0.6.0

# Knitr Engine Setup
knitr::opts_chunk$set(message=F,
                      warning=F,
                      error=F,
                      echo=F,
                      fig.pos = "H" ,
                      fig.align = 'center')

# Packages
options(kableExtra.latex.load_packages = FALSE) # Required to avoid floatrow error
library(knitr)
library(kableExtra)
library(ggplot2)
library(naniar) # For mcar_test()
library(visdat) # For vis_dat()
library(tidyverse)
library(car) # For qqPlot(), vif()
library(lme4)
library(lmerTest) # Satterthwaite approximation for computing p-values on lme4
library(ggpubr)

#library(Hmisc)
#library(vcd)


source("~/GitHub/marathon_project/_helpers.R")

setwd("~/GitHub/marathon_project/")


# Read in data
marathon_data <- read.csv("~/GitHub/marathon_project/data/project1.csv")
aqi_values <- read.csv("~/GitHub/marathon_project/data/aqi_values_ext.csv")
```

```
marathon_dates <- read.csv("~/GitHub/marathon_project/data/marathon_dates.csv")
course_record <- read.csv("~/GitHub/marathon_project/data/course_record.csv")


# Label and rename main dataset
marathon_data <- marathon_data %>%
  mutate(
    Race = factor(case_when(
      Race..0.Boston..1.Chicago..2.NYC..3.TC..4.D. ==
        0 ~ "Boston Marathon",
      Race..0.Boston..1.Chicago..2.NYC..3.TC..4.D. ==
        1 ~ "Chicago Marathon",
      Race..0.Boston..1.Chicago..2.NYC..3.TC..4.D. ==
        2 ~ "New York City Marathon",
      Race..0.Boston..1.Chicago..2.NYC..3.TC..4.D. ==
        3 ~ "Twin Cities Marathon (Minneapolis, MN)",
      Race..0.Boston..1.Chicago..2.NYC..3.TC..4.D. ==
        4 ~ "Grandma's Marathon (Duluth, MN)",
      TRUE ~ NA
    )),
    Sex = factor(case_when(
      Sex..0.F..1.M. == 0 ~ "Female",
      Sex..0.F..1.M. == 1 ~ "Male",
      TRUE ~ NA
    )),
    Flag = factor(Flag,
                  levels = c("White", "Green", "Yellow", "Red", "Black"),
                  ordered = TRUE)
  ) %>%
  rename(
    `% CR` = X.CR,
    `Td, C` = Td..C,
    `Tw, C` = Tw..C,
    `% RH` = X.rh,
    `Tg, C` = Tg..C,
    `SR W/m2` = SR.W.m2,
    `DP` = DP,
    `Wind` = Wind,
    `WBGT` = WBGT,
    `Age` = Age..yr.
  ) %>%
  select(-Race..0.Boston..1.Chicago..2.NYC..3.TC..4.D.,
         -Sex..0.F..1.M.)


# Clean and prep AQI dataset
aqi_values <- aqi_values %>%
  group_by(date_local, marathon) %>%
  dplyr::summarize(
    aqi = mean(aqi, na.rm = TRUE)) %>%
  mutate(
    `AQI Level` = case_when(
      aqi <= 50 ~ "Good",
      aqi > 50 & aqi <= 100 ~ "Moderate",
      aqi > 100 & aqi <= 150 ~ "Unhealthy for Sensitive Groups",
      aqi > 150 & aqi <= 200 ~ "Unhealthy",
      aqi > 200 & aqi <= 300 ~ "Very Unhealthy",
      aqi > 300 ~ "Hazardous"
    ),
    `AQI Level` = factor(`AQI Level`,
                         levels = c("Good",
                                    "Moderate",
                                    "Unhealthy for Sensitive Groups",
                                    "Unhealthy",
                                    "Very Unhealthy",
                                    "Hazardous"), ordered = TRUE)
  ) %>%
  ungroup() %>%
  mutate(marathon = case_when(
    marathon == "Boston" ~ "Boston Marathon",
    marathon == "Chicago" ~ "Chicago Marathon",
    marathon == "NYC" ~ "New York City Marathon",
    marathon == "Twin Cities" ~ "Twin Cities Marathon (Minneapolis, MN)",
    marathon == "Grandmas" ~ "Grandma's Marathon (Duluth, MN)",
    TRUE ~ marathon)
  )



# Prepare CR dataset
course_record <- course_record %>%
  mutate(Race = case_when(
    Race == "B" ~ "Boston Marathon",
    Race == "C" ~ "Chicago Marathon",
    Race == "NY" ~ "New York City Marathon",
    Race == "TC" ~ "Twin Cities Marathon (Minneapolis, MN)",
    Race == "D" ~ "Grandma's Marathon (Duluth, MN)",
    TRUE ~ Race)) %>%
  mutate(Gender = case_when(
    Gender == "M" ~ "Male",
    Gender == "F" ~ "Female",
    TRUE ~ Gender))

# Prepare marathon dates dataset
marathon_dates <- marathon_dates %>%
  mutate(marathon = case_when(
    marathon == "Boston" ~ "Boston Marathon",
    marathon == "Chicago" ~ "Chicago Marathon",
    marathon == "NYC" ~ "New York City Marathon",
    marathon == "Twin Cities" ~ "Twin Cities Marathon (Minneapolis, MN)",
    marathon == "Grandmas" ~ "Grandma's Marathon (Duluth, MN)",
    TRUE ~ marathon)) %>%
  mutate(date = str_trim(date))



# Merge data
data <- marathon_data %>%
  left_join(marathon_dates, by = c("Race" = "marathon", "Year" = "year")) %>%
  left_join(aqi_values, by = c("date" = "date_local", "Race" = "marathon")) %>%
  left_join(course_record, by = c("Race", "Year", "Sex" = "Gender"))


# Clean data post merge
data <- data %>%
  mutate(
    Date = as.Date(date),
    Sex = factor(Sex, levels = c("Female", "Male")),
    Race = factor(Race, levels = c("Boston Marathon",
                                   "Chicago Marathon",
                                   "New York City Marathon",
                                   "Twin Cities Marathon (Minneapolis, MN)",
                                   "Grandma's Marathon (Duluth, MN)")),
    CR = hms::as_hms(CR),
    Time = hms::round_hms(hms::as_hms((period_to_seconds(hms(CR)) *
                                           (1+(`% CR`/100)))), 6)
  ) %>%
    rename(
    `AQI` = aqi,
  ) %>%
  select(-date)


# NOTE: This cell is not evaluated or printed in the report, but commentary on
# the results is provided in prose.

# Number of observations by race
nrow(data[data$Race == "Boston Marathon", ]) ==
  nrow(marathon_data[marathon_data$Race == "Boston Marathon", ])

nrow(data[data$Race == "Chicago Marathon", ]) ==
  nrow(marathon_data[marathon_data$Race == "Chicago Marathon", ])


nrow(data[data$Race == "New York City Marathon", ]) ==
  nrow(marathon_data[marathon_data$Race == "New York City Marathon", ])
```

```r
nrow(data[data$Race == "Twin Cities Marathon (Minneapolis, MN)", ]) ==
  nrow(marathon_data[marathon_data$Race == "Twin Cities Marathon (Minneapolis, MN)", ])


nrow(data[data$Race == "Grandma's Marathon (Duluth, MN)", ]) ==
  nrow(marathon_data[marathon_data$Race == "Grandma's Marathon (Duluth, MN)", ])



# Check for duplicate rows
data %>%
  group_by(Race, Year, Sex, Age) %>%
  filter(n() > 1) %>%
  nrow()

# Make `Age Group`
data <- data %>%
  mutate(
    `Age Group` = cut(
      Age,
      breaks = c(0, 18, 25, 35, 45, 60, 70, Inf),
      labels = c("Age 0-17",
                 "Age 18-24",
                 "Age 25-34",
                 "Age 35-44",
                 "Age 45-59",
                 "Age 60-69",
                 "Age 70+"),
      include.lowest = T
    )
  )
# Reorder variables
data <- data %>%
  select(Race, Year, Sex, Age, `Age Group`, `% CR`, Flag,
         `Td, C`, `Tw, C`, `Tg, C`, DP, WBGT, everything())

# NOTE: This cell is not evaluated or printed in the report, but commentary on
# the results is provided in prose.

# Number of races in each dataset
nrow(unique(course_record[, c("Year", "Race")]))
nrow(unique(marathon_dates[, c("year", "marathon")]))
nrow(unique(aqi_values[, c("date_local", "marathon")]))
nrow(unique(marathon_data[, c("Race", "Year")]))
nrow(unique(data[, c("Race", "Year")]))

# Number of observations by race
```

```r
nrow(data[data$Race == "Boston Marathon", ])
nrow(data[data$Race == "Chicago Marathon", ])
nrow(data[data$Race == "New York City Marathon", ])
nrow(data[data$Race == "Twin Cities Marathon (Minneapolis, MN)", ])
nrow(data[data$Race == "Grandma's Marathon (Duluth, MN)", ])


# Check Race
valid_races <- c("Boston Marathon",
                 "Chicago Marathon",
                 "New York City Marathon",
                 "Twin Cities Marathon (Minneapolis, MN)",
                 "Grandma's Marathon (Duluth, MN)")
race_check <- all(data$Race %in% valid_races)
if (!race_check) {
  print("Race values invalid")
} else {
  print("Check passed")
}


# Check Sex
valid_sex <- c("Male","Female")
sex_check <- all(data$Sex %in% valid_sex)
if (!sex_check) {
  print("Sex values invalid")
} else {
  print("Check passed")
}


# Check Age (plausible range: 13 to 100)
age_check <- all(data$Age >= 13 & data$Age <= 100, na.rm = T)
if (!age_check) {
  print("Age out of plausible range (13 to 100 years)")
} else {
  print("Check passed")
}


# Check Flag
# Note: Fails check without explicit NA, missing data in Flag
valid_flags <- c("White", "Green", "Yellow", "Red", "Black", NA)
flag_check <- all(data$Flag %in% valid_flags)
if (!flag_check) {
  print("Invalid Flag values")
} else {
  print("Check passed")
}
```

```r
# Check % CR
# Note: This check fails, max % CR is ~420%, investigate.
cr_check <- all(data$`% CR` >= -10 & data$`% CR` <= 400, na.rm = T)
if (!cr_check) {
  print("% CR values out of plausible range (-10 to 400)")
} else {
  print("Check passed")
}

range(data$`% CR`) # Check range for % CR




# Check Time (plausible range: minimum is WR pace, maximum is 10 Hours)
# Note: This check fails, max time is over 10 hours
# Note: WR marathon pace is 2:00:35 for men, 2:11:53 for women
# Note: Split by men/women
time_check_m <- all(data$`Time` >= 7235 & data$`Time` <= 36000 & data$Sex=="Male",
                    na.rm = T)
if (!time_check_m) {
  print("Time values out of plausible range (WR to 10hrs)")
} else {
  print("Check passed")
}

time_check_f <- all(data$`Time` >= 7913 & data$`Time` <= 36000 & data$Sex=="Female",
                    na.rm = T)
if (!time_check_f) {
  print("Time values out of plausible range (WR to 10hrs)")
} else {
  print("Check passed")
}

range(lubridate::period_to_seconds(hms(data$Time))) # Check range for Time

hms::as_hms(range(lubridate::period_to_seconds(hms(data$Time)))[1]) # Min Time
hms::as_hms(range(lubridate::period_to_seconds(hms(data$Time)))[2]) # Max Time

# Time Subcheck: Slow finishers
data %>% dplyr::filter(Time > 36000) %>% kbl # Table for all finishers over 10hrs
data %>% dplyr::filter(Time > 32400) %>% kbl # Table for all finishers over 9hrs
# Note: 10hrs is a 2.62 mph pace, 9hrs is a 2.91 mph pace


# Time Subcheck: Fast finishers
data %>% # Table for all finishers under WR (M)
```

```r
  dplyr::filter(Time < 7235 & Sex == "Male") %>% kbl
# Note: no results, good

data %>% # Table for all finishers under WR (F)
  dplyr::filter(Time < 7913 & Sex == "Female") %>% kbl
# Note: no results, good




# Check Td, C (plausible range: -18 to 42)
td_check <- all(data$`Td, C` >= -18 & data$`Td, C` <= 42, na.rm = T)
if (!td_check) {
  print("Dry bulb temperature out of plausible range (-18 to 42C)")
} else {
  print("Check passed")
}




# Check Tw, C (plausible range: -18 to 42)
tw_check <- all(data$`Tw, C` >= -18 & data$`Tw, C` <= 42, na.rm = T)
if (!tw_check) {
  print("Wet bulb temperature out of plausible range (-18 to 42C)")
} else {
  print("Check passed")
}




# Check Tg, C (plausible range: -18 to 42)
# Note: This fails because max is 44C, which is 111F, investigate.
tg_check <- all(data$`Tg, C` >= -18 & data$`Tg, C` <= 42, na.rm = T)
if (!tg_check) {
  print("Globe temperature out of plausible range (-18 to 42C)")
} else {
  print("Check passed")
}

range(data$`Tg, C`, na.rm = T) # Check range for Tg, C


# Check % RH (plausible range: 5 and 100)
# Note: This fails because many values under 5%, investigate.
```

```r
rh_check <- all(data$`% RH` >= 5 & data$`% RH` <= 100, na.rm = T)
if (!rh_check) {
  print("Relative humidity out of range (0-100%)")
} else {
  print("Check passed")
}

range(data$`% RH`, na.rm = T) # Check range for % RH
```

```r
# Check SR W/m2 (should be non-negative, plausible max around 1500)
sr_check <- all(data$SR >= 0 & data$SR <= 1500, na.rm = T)
if (!sr_check) {
  print("Solar radiation out of plausible range (0 to 1500 W/m2)")
} else {
  print("Check passed")
}
```

```r
# Check DP (plausible range: -30 to 35)
dp_check <- all(data$DP >= -30 & data$DP <= 35, na.rm = T)
if (!dp_check) {
  print("Dew point out of plausible range (-30 to 35C)")
} else {
  print("Check passed")
}
```

```r
# Check Wind (should be non-negative, plausible max around 150 km/h)
wind_check <- all(data$Wind >= 0 & data$Wind <= 150, na.rm = T)
if (!wind_check) {
  print("Wind speed out of plausible range (0 to 150 km/h)")
} else {
  print("Check passed")
}
```

```r
# Check WBGT (plausible range: -18 to 42)
wbgt_check <- all(data$WBGT >= -18 & data$WBGT <= 42, na.rm = T)
if (!wbgt_check) {
  print("WBGT out of plausible range (-18 to 42C)")
} else {
  print("Check passed")
}
```

```r
# Check AQI (plausible range: 5 to 500)
aqi_check <- all(marathon_data$AQI >= 5 & marathon_data$AQI <= 500, na.rm = T)
if (!aqi_check) {
  print("AQI out of plausible range (5 to 500)")
} else {
  print("Check passed")
}
```

```r
# NOTE: This cell is not evaluated or printed in the report, but commentary on
# the results is provided in prose.

# Histograms for variables
hist(data$Year)
hist(data$Age)
hist(data$`% CR`)
hist(lubridate::period_to_seconds(hms(data$Time)))
hist(data$`Td, C`)
hist(data$`Tw, C`)
hist(data$`% RH`)
hist(data$`Tg, C`)
hist(data$`SR W/m2`)
hist(data$`DP`)
hist(data$`Wind`)
hist(data$`WBGT`)
hist(data$`AQI`)
```

```r
# Recoding % RH
data <- data %>% mutate(`% RH` = if_else(`% RH` <= 1, `% RH` * 100, `% RH`))
```

```r
# Remove Boston Marathon Bombing year
data <- data %>%
  filter(Race != 'Boston Marathon' | Year != 2013)
```

```r
# Visualize dataset & missingness
vis_dat(data) + ylim(0, 11600) + theme(
  plot.margin = margin(0, 0, 0, 0),
  axis.text.y = element_blank(),
  axis.title.y = element_blank()
)

# Check missing data years and marathons for patterns
missing_data <- data %>%
  filter(if_any(everything(), is.na)) %>%
  group_by(Race, Year) %>%
  summarize(Count = n(), .groups = 'drop')
```

```r
# Test for MCAR
naniar::mcar_test(data)

# Test for MAR
check_mar(data)
```

```r
# Note: using tbl-cap rather than caption= in kbl() breaks repeated title is span page

# Table 1
summary_table(data[-7:-20]) %>%
  kbl(#caption = "Summary of Variables", # Conflicts with Quarto referencing
      booktabs = T,
      longtable = T, # LONGTABLE
      escape = T,
      align = "c") %>%
  column_spec(1, width="2.2cm", latex_valign = "m") %>%
  column_spec(2, width="1.0cm", latex_valign = "m") %>%
  column_spec(3, width="6.67cm", latex_valign = "m") %>%
  column_spec(4, width="1.0cm", latex_valign = "m") %>%
  column_spec(5, width="1.0cm", latex_valign = "m") %>%
  column_spec(6, width="1.25cm", latex_valign = "m") %>%
  column_spec(7, width="1.25cm", latex_valign = "m") %>%
  kable_styling(
    font_size = 7.6, # Added for LONGTABLE
    latex_options = c(#'HOLD_position', # Removed for LONGTABLE
    #'scale_down', # Removed for LONGTABLE
    "repeat_header",  # Added for LONGTABLE
    'striped'),
    full_width = F, # Note: TRUE does not work with LONGTABLE
    position = 'center'# Added for LONGTABLE
  ) %>%
  footnote(general = "Shapiro-Wilk test for normality;
           Grubb's test for outliers.", escape = F)
```

```r
# NOTE: This cell is not evaluated or printed in the report, but commentary on
# the results is provided in prose.

# Q-Q Plots for variables
qqPlot(data$Year)
qqPlot(data$Age)
qqPlot(data$`% CR`)
qqPlot(data$`Td, C`)
qqPlot(data$`Tw, C`)
qqPlot(data$`% RH`)
qqPlot(data$`Tg, C`)
qqPlot(data$`SR W/m2`)
qqPlot(data$`DP`)
qqPlot(data$`Wind`)
qqPlot(data$`WBGT`)
qqPlot(data$`AQI`)
# Table 2
summary_table(data[,c(-5,-17:-20)], stratify_var = "Race") %>%
  kbl(#caption = "Summary of Variables by Race", # Conflicts with Quarto label
      booktabs = T,
      longtable = T, # LONGTABLE
      escape = T,
      align = "c") %>%
  column_spec(1, width="2.21cm", latex_valign = "m") %>%
  column_spec(2, width="2.4cm", latex_valign = "m") %>%
  column_spec(3, width="2.4cm", latex_valign = "m") %>%
  column_spec(4, width="2.4cm", latex_valign = "m") %>%
  column_spec(5, width="2.4cm", latex_valign = "m") %>%
  column_spec(6, width="2.4cm", latex_valign = "m") %>%
  column_spec(7, width=".35cm", latex_valign = "m") %>%
  kable_styling(
    font_size = 7.6, # Added for LONGTABLE
    latex_options = c(#'HOLD_position', # Removed for LONGTABLE
    #'scale_down', # Removed for LONGTABLE
    "repeat_header",  # Added for LONGTABLE
    'striped'),
    full_width = F, # Note: TRUE does not work with LONGTABLE
    position = 'center'# Added for LONGTABLE
  ) %>%
  footnote(general = "ns = P > 0.05,
           * = P $\\\\leq$ 0.05,
           ** = P $\\\\leq$ 0.01,
           *** = P $\\\\leq$ 0.001,
           **** = P $\\\\leq$ 0.0001
           ", escape = F) %>%
  footnote(general = "Kruskal-Wallis test for continuous variables,
           Chi-Square test for categorical variables.
           Bonferroni correction applied.", escape = F)

# Table X
summary_table(data[,c(-5,-17:-20)], stratify_var = "Sex") %>%
  kbl(caption = "Summary of Variables by Sex",
      booktabs = T,
      escape = T,
      align = "c") %>%
  column_spec(1, width="3cm", latex_valign = "m") %>%
  column_spec(2, width="8cm", latex_valign = "m") %>%
  column_spec(3, width="8cm", latex_valign = "m") %>%
  column_spec(4, width=".7cm", latex_valign = "m") %>%
 kable_styling(latex_options = c('HOLD_position',
                                 'scale_down',
                                 'striped')) %>%
  footnote(general = "ns = P > 0.05,
           * = P $\\\\leq$ 0.05,
           ** = P $\\\\leq$ 0.01,
           *** = P $\\\\leq$ 0.001,
           **** = P $\\\\leq$ 0.0001
           ", escape = F) %>%
  footnote(general = "Student's T-test for continuous variables,
           Chi-Square test for categorical variables.", escape = F)

# Table Y
summary_table(data[,c(-4,-17:-20)], stratify_var = "Age Group") %>%
  kbl(caption = "Summary of Variables by Age Group",
      booktabs = T,
      escape = T,
      align = "c") %>%
  column_spec(1, width="2.75cm", latex_valign = "m") %>%
  column_spec(2, width="4cm", latex_valign = "m") %>%
  column_spec(3, width="4cm", latex_valign = "m") %>%
  column_spec(4, width="4cm", latex_valign = "m") %>%
  column_spec(5, width="4cm", latex_valign = "m") %>%
  column_spec(6, width="4cm", latex_valign = "m") %>%
  column_spec(7, width="4cm", latex_valign = "m") %>%
  column_spec(8, width="4cm", latex_valign = "m") %>%
  column_spec(9, width=".7cm", latex_valign = "m") %>%
  kable_styling(latex_options = c('HOLD_position',
                                  'scale_down',
                                 'striped')) %>%
  footnote(general = "ns = P > 0.05,
           * = P $\\\\leq$ 0.05,
           ** = P $\\\\leq$ 0.01,
           *** = P $\\\\leq$ 0.001,
           **** = P $\\\\leq$ 0.0001
           ", escape = F) %>%
```

```r
  footnote(general = "Kruskal-Wallis test for continuous variables,
           Chi-Square test for categorical variables.
           Bonferroni correction applied.", escape = F)

# Generate psuedo correlation matrix
psuedo_cor_matrix <- psuedo_cor_mat(data[-18:-20])

# Plot the heatmap (with variable names and values)
ggplot(melt(psuedo_cor_matrix), aes(x = Var2, y = Var1, fill = value)) +
  geom_tile(color = "white") +
  geom_text(aes(label = sprintf("%.2f", value)), size = 2, color = "black") +
  scale_fill_gradient2(low = "blue",
                       high = "red",
                       mid = "white",
                       midpoint = 0,
                       limits = c(-1, 1),
                       space = "Lab",
                       name = "Correlation") +
  theme_minimal() +
  labs(x = "", y = "") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_text(angle = 45, hjust = 1, size = 7.5),
        axis.ticks.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_text(angle = 45, hjust = 1, size = 7.5),
        legend.position="none") +
  coord_fixed(ratio = .55)


# Calculate VIF
print(vif(lm(`% CR` ~ Race + Sex + Age + Flag + `Td, C` + `Tw, C` +
              `Tg, C` + `DP` + `WBGT` + `% RH` + `SR W/m2` + Wind + AQI,
             data = data)))
# Error: there are aliased coefficients in the model (perfect colinearity)


# Calculate VIF
print(vif(lm(`% CR` ~ Race + Sex + Age + `WBGT` + `% RH` +
               `SR W/m2` + Wind + AQI, data = data)))
# Note: For this combination of variables all adjusted VIF are below 2 (threshold)

data <- data %>%
  select(-`Td, C`, -`Tw, C`, -`Tg, C`, -`DP`, # Remove colinear weather vars
                         `Date`) # Remove redundant vars

# Plotting Performance vs. Age
ggplot(data, aes(x = Age, y = `% CR`, color = Sex)) +
    geom_point(alpha = .18, shape = 21, show.legend = F) +
    geom_smooth(data = data, aes(group = Sex), method = "loess", se = F,
                color = "black", linewidth = 1.3) +
    geom_smooth(data = data, aes(color = Sex), method = "loess", se = F,
                linewidth = .7, show.legend = TRUE) +
    labs(x = "Age",
         y = "% CR",
         color = "Sex") +
    theme_minimal() +
    scale_color_manual(values = c("Female" = "lightpink2", "Male" = "lightblue2")) +
    guides(color = guide_legend(override.aes = list(linewidth = 1.5))) +
  theme(legend.position = "bottom") + ylim(-5,260)



race_labels <- c(
  "Boston Marathon" = "Boston Marathon",
  "Chicago Marathon" = "Chicago Marathon",
  "New York City Marathon" = "NYC Marathon",
  "Twin Cities Marathon (Minneapolis, MN)" = "Twin Cities Marathon",
  "Grandma's Marathon (Duluth, MN)" = "Grandma's Marathon")


# calc normalized differences (in % CR for each age)
# NOTE: we considered normalizing also by sex, but % CR is already relative to sex
data_norm <- data %>%
  group_by(Age) %>%
  mutate(mean_cr = mean(`% CR`)) %>%
  ungroup() %>%
  mutate(norm_diff = `% CR` - mean_cr)

data_overall <- data_norm %>%
  mutate(Race = "Overall")

data_combined <- bind_rows(data_norm, data_overall) %>%
  mutate(Race = factor(Race, levels = c(
    "Overall",
    "Boston Marathon",
    "Chicago Marathon",
    "New York City Marathon",
    "Twin Cities Marathon (Minneapolis, MN)",
    "Grandma's Marathon (Duluth, MN)"
  )))


ggplot(data_combined, aes(x = Age, y = norm_diff, color = Sex)) +
  geom_point(alpha = 0.048, shape = 21, show.legend = F) +
  geom_smooth(data = data_combined, aes(group = Sex), method = "loess", se = F,
              color = "black", linewidth = 1) +
  geom_smooth(method = "loess", se = FALSE, linewidth = .55) +
  facet_wrap(~ Race, labeller = labeller(Race = race_labels),
             ncol = 3, dir = "h") +
  scale_color_manual(values = c("Female" = "lightpink2", "Male" = "lightblue2")) +
  labs(x = "Age",
       y = "Normalized Difference in % CR",
       color = "Sex") +
  theme_minimal() +
  theme(
    legend.position = "none",
    panel.grid.minor = element_blank(),
    axis.text = element_text(size = 6),
    axis.title = element_text(size = 6.8),
    legend.text = element_text(size = 6),
    legend.title = element_text(size = 6.8),
    strip.text = element_text(size = 7)
  ) +
  coord_cartesian(ylim = c(-30, 30))


# NOTE: Removed this plot in final revision

# Plotting Performance vs. Age by Race
ggplot(data, aes(x = Age, y = `% CR`, color = Sex)) +
  geom_point(alpha = 0, shape = 21, show.legend = F) +
  geom_smooth(data = data, aes(group = Sex), method = "loess", se = F,
              color = "black", linewidth = 1) +
  geom_smooth(data = data, aes(color = Sex), method = "loess", se = F,
              linewidth = .55, show.legend = TRUE) +
  labs(x = "Age",
       y = "% CR",
       color = "Sex") +
  theme_minimal() +
  scale_color_manual(values = c("Female" = "lightpink2", "Male" = "lightblue2")) +
  guides(color = guide_legend(override.aes = list(linewidth = 1.5))) +
  facet_wrap(~ Race, labeller = labeller(Race = race_labels)) +
  theme(legend.position = "none",
        plot.margin = margin(t = 0,
                             r = 0,
                             b = 0,
                             l = 0,
                             unit = "mm")) + ylim(-5,250)


# Convert `Time` & `CR` to Second for Modeling
data$Time <- lubridate::period_to_seconds(hms(data$Time))
data$CR <- lubridate::period_to_seconds(hms(data$CR))

# Aim 1: Model
aim1m1 <- lmer(Time ~ CR + Year + Sex + Age + I(Age^2)
                + Sex * Age + Sex * I(Age^2) + (1 | Race), data = data)
```

```
# Plotting Performance vs. Age by Flag
ggplot(data=subset(drop_na(data_norm)),
       aes(x = Age, y = norm_diff, color = Flag)) +
  geom_point(alpha = .14, shape = 21, show.legend = F) +
  geom_smooth(data = subset(drop_na(data_norm)),
                            aes(group = Flag), method = "loess", se = F,
              color = "black", linewidth = 1.1) +
  geom_smooth(data = subset(drop_na(data_norm)),
                            aes(color = Flag), method = "loess", se = F,
              linewidth = .6, show.legend = TRUE) +
    facet_wrap(~ Sex) +
  labs(x = "Age",
       y = "Normalized Difference in % CR",
       color = "Flag") +
  theme_minimal() +
  scale_color_manual(values = c("White" = "white",
                                "Green" = "lightgreen",
                                "Yellow" = "lightgoldenrod",
                                "Red" = "lightcoral")) +
  guides(color = guide_legend(override.aes = list(linewidth = 1.5))) +
  theme(legend.position = "right",
          plot.margin = margin(t = 0,  # Top margin
                             r = 1,  # Right margin
                             b = 1,  # Bottom margin
                             l = 1,  # Left margin
                             unit = "mm")) + ylim (-5,30)



env_vars_long <- data_norm %>%
  select(Sex, norm_diff, WBGT, `% RH`, `SR W/m2`, Wind, AQI) %>%
  pivot_longer(
    cols = c(WBGT, `% RH`, `SR W/m2`, Wind, AQI),
    names_to = "Variable",
    values_to = "Value"
  )

ggplot(env_vars_long, aes(x = Value, y = norm_diff, color = Sex)) +
  geom_point(alpha = .16, shape = 21, show.legend = TRUE) +
  geom_smooth(aes(group = Sex),
              method = "loess",
              se = FALSE,
              color = "black",
              linewidth = 1.2) +
  geom_smooth(aes(fill = Sex),
              method = "loess",
              se = TRUE,
```

```r
              linewidth = 0.8,
              alpha = 0.8) +
  facet_wrap(~ Variable, scales = "free_x", ncol = 5) +
  labs(x = "Environmental Variable Value",
       y = "Normalized Difference in % CR",
       color = "Sex") +
  theme_minimal() +
  scale_color_manual(values = c("Female" = "lightpink2", "Male" = "lightblue2")) +
  guides(color = guide_legend(override.aes = list(linewidth = 1.5))) +
  theme(
    strip.text = element_text(size = 8),
    axis.text = element_text(size = 7),
    legend.position = "bottom",
    panel.spacing = unit(1, "lines")
  ) + ylim (-5,30)


# Aim 2: Model 1
aim2m1 <- lmer(Time ~ CR + Year + Sex + Age + I(Age^2) + Sex * Age + Sex * I(Age^2) +
                 WBGT + `% RH` + `SR W/m2` + Wind + AQI + (1 | Race),
              data = data)




# Aim 2: Model 2
aim2m2 <- lmer(Time ~ CR + Year + Sex + Age + I(Age^2) + Sex * Age + Sex * I(Age^2) +
                 WBGT + WBGT * Sex + WBGT * Age +
                 `% RH` + `% RH` * Sex + `% RH` * Age +
                 `SR W/m2` +`SR W/m2` * Sex + `SR W/m2` * Age +
                 Wind + Wind * Sex + Wind * Age +
                 AQI + AQI * Sex + AQI * Age + (1 | Race),
              data = data)

# Run permutation importance
imp <- variable_importance_reg(aim2m2, data, "Time", n_iterations = 100)

saveRDS(imp, "imp.rds")
imp <- readRDS("imp.rds")
# Table of Variable Importance
kbl(imp[-1:-3])

# Plot of Variable Importance
ggplot(imp, aes(x = reorder(Variable, rev(`Relative Importance`)),
                y = `Relative Importance`)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  #coord_flip() +
  labs(title = "Variable Importance",
       x = "Variable",
       y = "Relative Importance") +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 0, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

Appendix II: Functions Code

```
# Helper Functions for Marathon Project EDA

# --- Preamble ---
# Date of last update: Oct. 7, 2024
# R Version: 4.3.1
# Package Versions:
#   outliers: 0.15
#   tidyverse: 2.0.0
#   reshape2: 1.4.4
#   moments: 0.14.1
#   vcd: 1.4-12


# Libraries for functions
suppressMessages(library(tidyverse))
library(outliers) # for grubbs.test()
library(reshape2) # for melt()
library(moments) # for skewness() and kurtosis()
library(vcd) # for assocstats()









# -- BEGIN Psuedo-correlation Matrix Section --

# Helper function to calculate Eta^2
eta_squared <- function(aov_model) {
  #' This function calculates the Eta^2 stat from an anova model. Eta^2 is a measure
  #' of effect size. Eta^2 describes the proportion of the total variance attributable
  #' to a given factor
  #'
  #' @param aov_model An aov object
  #' @return A numeric value of Eta^2 (proportion of the total variance explained)
  #'
  sum_of_squares_model <- summary(aov_model)[[1]]$"Sum Sq"[1]
  sum_of_squares_total <- sum(summary(aov_model)[[1]]$"Sum Sq")
  eta_sq <- sum_of_squares_model / sum_of_squares_total
  return(eta_sq)
}


# Main function to compute correlations or psuedo-correlations
# NOTE: Below still does not work with logical variables, but now fixed for NAs
```

```r
psuedo_cor_mat <- function(data) {
  #' Builds a pseudo correlation matrix for a given dataset, flexible for numerical
  #' and categorical variables. Uses Pearson correlation for numerical-numerical pairs,
  #' Cramer's V for categorical-categorical pairs, point biserial correlation for
  #' numerical-binary categorical pairs, and the square root of Eta^2 for
  #' numerical-multi-category pairs.
  #'
  #' @param data A df with any mix of numerical and categorical variables.
  #' @return A symmetric matrix with correlation coefficients
  #' (or equivalent measures) for all variable pairs.
  #'
  variables <- names(data)
  n <- length(variables)
  cor_matrix <- matrix(NA, n, n, dimnames = list(variables, variables))

  for (i in 1:n) {
    for (j in i:n) {
      if (i == j) {
        cor_matrix[i, j] <- 1
      } else {
        var_i <- data[[i]]
        var_j <- data[[j]]

        if (is.numeric(var_i) && is.numeric(var_j)) {
          # Pearson correlation for continuous-continuous
          cor_matrix[i, j] <- cor_matrix[j, i] <- cor(var_i, var_j,
                                                      use = "pairwise.complete.obs")

        } else if (is.factor(var_i) && is.factor(var_j)) {
          # Remove NAs for both variables
          valid_idx <- !is.na(var_i) & !is.na(var_j)
          var_i_clean <- var_i[valid_idx]
          var_j_clean <- var_j[valid_idx]

          # Drop unused levels
          var_i_clean <- droplevels(var_i_clean)
          var_j_clean <- droplevels(var_j_clean)

          # Cramer's V for categorical-categorical
          if (length(var_i_clean) > 0 && length(var_j_clean) > 0) {
            cramers_v <- sqrt(assocstats(table(var_i_clean, var_j_clean))$cramer)
            cor_matrix[i, j] <- cor_matrix[j, i] <- cramers_v
          } else {
            cor_matrix[i, j] <- cor_matrix[j, i] <- NA
          }

        } else {
          # Continuous-categorical pairs
          if (is.numeric(var_i) && (is.factor(var_j) || is.character(var_j))) {
            continuous <- var_i
            factor_var <- as.factor(var_j)
          } else if ((is.factor(var_i) || is.character(var_i)) && is.numeric(var_j)) {
            continuous <- var_j
            factor_var <- as.factor(var_i)
          } else {
            next  # Skip if variables are not appropriate types
          }

          # Remove NAs in both variables
          valid_idx <- !is.na(continuous) & !is.na(factor_var)
          continuous <- continuous[valid_idx]
          factor_var <- factor_var[valid_idx]

          # Drop unused levels
          factor_var <- droplevels(factor_var)

          if (length(unique(factor_var)) == 2) {
            # Point biserial correlation for binary factors
            factor_numeric <- as.numeric(factor_var) - 1
            cor_matrix[i, j] <- cor_matrix[j, i] <- cor(continuous, factor_numeric)
          } else if (length(unique(factor_var)) > 2) {
            # Eta-squared for multi-category factors
            if (length(factor_var) > 0) {
              aov_result <- aov(continuous ~ factor_var)
              eta_sq <- eta_squared(aov_result)
              cor_matrix[i, j] <- cor_matrix[j, i] <- sqrt(eta_sq)
            } else {
              cor_matrix[i, j] <- cor_matrix[j, i] <- NA
            }
          } else {
            # Cannot compute correlation if factor_var has insufficient levels
            cor_matrix[i, j] <- cor_matrix[j, i] <- NA
          }
        }
      }
    }
  }
  return(cor_matrix)
}

# -- END Psuedo-correlation Matrix Section --
# Function to generate summary table aka "Table 1"
# NOTE: Does not like the following variable types: Date, `hms` num / difftime
# NOTE: Will likely need custom subroutines to handle dates / hms / difftime
summary_table <- function(df, stratify_var = NULL) {
  #' Create summary table for a data frame with option for stratification. It
  #' summarizes numeric variables by mean, standard deviation, and IQR.
  #' Categorical variables by count and percentage. Optionally stratifies by a
  #' specified variable.
  #'
  #' @param df A data frame
  #' @param stratify_var (Optional) A variable on which to stratify the summaries.
  #'
  #' @return A data frame (or tibble) containing the summarized statistics.

  # Internal function to handle the actual summarization
  summarize_internal <- function(df, stratify_var = NULL, is_recursive_call = FALSE) {

    # --- SUMMARIZER SECTION ---
    # Function to summarize continuous variable
    summarize_continuous <- function(data, var) {
      var_data <- data[[var]]
      mean_sd <- paste0(round(mean(var_data, na.rm = TRUE), 2),
                        " (",
                        round(sd(var_data, na.rm = TRUE), 2),
                        ")")
      quantiles <- quantile(var_data, probs = c(0.25, 0.75), na.rm = TRUE)
      quantile_range <- paste0("(",
                               round(quantiles[1], 2),
                               " - ",
                               round(quantiles[2], 2),
                               ")")
      paste(mean_sd, quantile_range)
    }

    # Func to summarize categorical variable
    summarize_categorical <- function(data, var) {
      var_data <- table(data[[var]])
      percentages <- prop.table(var_data) * 100
      paste(names(var_data), ": ",
            var_data, " (",
            round(percentages, 2),
            "%)",
            sep = "",
            collapse = ", ")
    }

    # Handler for strata
    variable_names <- names(df)
    if (!is.null(stratify_var) && stratify_var %in% variable_names) {
      variable_names <- variable_names[variable_names != stratify_var]
    }

    # Summarize all into a list for table
    summary_list <- map(variable_names, ~ {
      var_type <- ifelse(is.numeric(df[[.x]]) | is.integer(df[[.x]]),
                         "Numeric", "Categorical")
      summarizer <- ifelse(var_type == "Numeric",
                           summarize_continuous,
                           summarize_categorical)
      summary <- summarizer(df, .x)
      variable_name <- ifelse(var_type == "Numeric",
                              paste(.x, "[Mean (SD) (Quantile)]"),
                              paste(.x, "[n (%)]"))
      tibble(Variable = variable_name, Type = var_type, Summary = summary)
    })

    # Bind to table
    summary_table <- bind_rows(summary_list)

    # --- NORMALITY AND OUTLIERS SECTION ---
    # Perform Shapiro-Wilk and Grubbs tests only if it's not a recursive call
    # and no stratify_var is given
    if (!is_recursive_call && is.null(stratify_var)) {
      # Get numeric variables
      numeric_vars <- variable_names[sapply(df[variable_names],
                                            function(x) is.numeric(x) | is.integer(x))]

      # Init object for S-W / Grubbs results
      test_results <- map(numeric_vars, ~ {
        x <- df[[.x]]
        x <- na.omit(x)
        n <- length(x)
        normality <- NA
        outlier <- NA
        skewness_label <- NA
        kurtosis_label <- NA
```

```
# Shapiro-Wilk test (Note: now accommodates n > 5000)
# NOTE: Consider in future -- KS test, AD test, Jarque-Bera test
if (n >= 3) {
  if (n > 5000) {
    shapiro_test <- shapiro.test(sample(x, 4999))
  } else {
    shapiro_test <- shapiro.test(x)
  }
  normality <- ifelse(shapiro_test$p.value < 0.05, "No", "Yes")
} else {
  normality <- "Insufficient data"
}

# Grubbs test
if (n >= 3) {
  grubbs_test <- grubbs.test(x)
  outlier <- ifelse(grubbs_test$p.value < 0.05, "Yes", "No")
} else {
  outlier <- "Insufficient data"
}

# Skewness and Kurtosis
if (n >= 2) {
  skewness <- skewness(x)
  kurtosis <- kurtosis(x)

  # Skewness label
  if (abs(skewness) < 0.5) {
    skewness_label <- "Centered"
  } else if (skewness > 0.5) {
    skewness_label <- "Right-skewed"
  } else {
    skewness_label <- "Left-skewed"
  }

  # Kurtosis label
  if (kurtosis < 2.5) {
    kurtosis_label <- "Platykurtic"
  } else if (kurtosis > 3.5) {
    kurtosis_label <- "Leptokurtic"
  } else {
    kurtosis_label <- "Mesokurtic"
  }
} else {
  skewness_label <- "Insufficient data"
  kurtosis_label <- "Insufficient data"
}

tibble(Variable = .x,
             `Normal Distribution` = normality,
             `Outlier(s) Present` = outlier,
             `Skewness` = skewness_label,
             `Kurtosis` = kurtosis_label)
    })

    # Bind test results into df
    test_results_df <- bind_rows(test_results)

    # Get actual variable names from summary_table for merge
    # Note: this could break in some datasets if the variable names include "["
    # therefore a more general solution later is preferred
    summary_table$ActualVariable <- gsub(" \\[.*\\]", "", summary_table$Variable)

    # Merge summary_table with test_results_df
    summary_table <- left_join(summary_table,
                               test_results_df,
                               by = c("ActualVariable" = "Variable"))

    # Remove temp matching ActualVariable column
    summary_table <- summary_table %>% dplyr::select(-ActualVariable)
  } else if (!is.null(stratify_var)) {
    # --- STRATA SECTION ---

    # Stratification handling
    stratified_summaries <- df %>%
      group_by(!!sym(stratify_var)) %>%
      do(summarize_internal(., stratify_var = NULL, is_recursive_call = TRUE))

    summary_table <- stratified_summaries %>%
      ungroup() %>%
      dplyr::select(-group_cols()) %>% # This doesn't seem necessary, why'd I do it?
      pivot_wider(names_from = !!sym(stratify_var), values_from = Summary)

    # Function for significance testing (numeric variables)
    test_continuous <- function(data, var, group_var) {
      formula <- as.formula(paste0("`", var, "` ~ `", group_var, "`")) #BUGFIX: spaces
      num_groups <- length(unique(data[[group_var]]))
      if (num_groups == 2) {
        test_result <- t.test(formula, data = data)
        p_value <- test_result$p.value
      } else {
        test_result <- kruskal.test(formula, data = data)
        p_value <- test_result$p.value
      }
      p_value
    }

    # Function for significance testing (categorical variables)
```

```r
test_categorical <- function(data, var, group_var) {
  table_data <- table(data[[var]], data[[group_var]])
  table_data <- table_data[rowSums(table_data) > 0, # This prevents NaN error
                           colSums(table_data) > 0, # in ChiSq test.
                           drop = FALSE]
  test_result <- chisq.test(table_data)
  p_value <- test_result$p.value
  p_value
}

# Perform significance testing (if stratify_var is specified)
significance_tests <- map(variable_names, ~ {
  var_type <- ifelse(is.numeric(df[[.x]]) | is.integer(df[[.x]]),
                     "Numeric", "Categorical")
  tester <- ifelse(var_type == "Numeric",
                   test_continuous,
                   test_categorical)
  p_value <- tester(df, .x, stratify_var)
  tibble(Variable = .x, `P-value` = p_value)
})

significance_table <- bind_rows(significance_tests)

# Adjust p-values using Bonferroni correction
significance_table$`Adjusted P-value` <- p.adjust(significance_table$`P-value`,
                                                  method = "bonferroni")

# Get actual variable names from summary_table for merge
# Note: this could break in some datasets if the variable names include "["
# therefore a more general solution later is preferred
summary_table$ActualVariable <- gsub(" \\[.*\\]", "", summary_table$Variable)

# Merge summary_table with significance_table on actual variable names
summary_table <- left_join(summary_table,
                           significance_table,
                           by = c("ActualVariable" = "Variable"))

# Remove rows corresponding to stratify_var
summary_table <- summary_table %>% filter(ActualVariable != stratify_var)

# Create 'Sig.' column based on Adjusted P-value
summary_table$`Sig.` <- case_when(
  summary_table$`Adjusted P-value` <= 0.0001 ~ "****",
  summary_table$`Adjusted P-value` <= 0.001 ~ "***",
  summary_table$`Adjusted P-value` <= 0.01  ~ "**",
  summary_table$`Adjusted P-value` <= 0.05  ~ "*",
  TRUE                                      ~ "ns"
)
```

```
      # Remove temp matching ActualVariable, adjusted/unadjusted P-Values
      summary_table <- dplyr::select(summary_table,
                                     c(-ActualVariable,
                                       -`P-value`,
                                       -`Adjusted P-value`))
    }

    # Remove "Type" column for stratified tables
    if (!is.null(stratify_var) && !is_recursive_call) {
      summary_table <- summary_table %>% select(-Type)
    }

    return(summary_table)
  }

  # Call internal function to initiate procedure
  summarize_internal(df, stratify_var)
}
```

```
# Function to count number of variables where missing data occours
count_missing_vars <- function(df) {
  #' Calc the number of variables in a data frame that have at least one missing
  #' value.
  #'
  #' @param df A data frame
  #'
  #' @return A data frame with a single row and column indicating the count
  #' of variables with any missing values.
  #'
  missing_count <- df %>%
    summarise_all(~ sum(is.na(.))) %>%
    pivot_longer(everything()) %>%
    filter(value > 0) %>%
    nrow()
```

```r
  return(data.frame(Num_Missing_Vars = missing_count))
}

# Count by variable type
summarize_variables_types <- function(data) {
  #' Summarizes the types of variables in a dataset by type. Possible types:
  #' Numeric, or Categorical (factors or logical).
  #'
  #' @param data A data frame
  #' @return A table with the counts of variables categorized by type
  #'
  var_types <- sapply(data, function(x) {
    if(is.numeric(x) | is.integer(x)) {
      return("Numeric")
    }
    else {
      return("Categorical")
    }
  })

  var_counts <- table(var_types)
  return(var_counts)
}
```

```
# Function to find groups of correlated variables
findCorrelatedGroups <- function(corr_matrix, threshold) {
  #' Find Groups of Correlated Variables
  #'
  #' This function identifies groups of variables in a correlation matrix that
  #' exceed a specified correlation threshold.
  #'
  #' @param corr_matrix A square symmetric matrix representing variable correlations.
  #' @param threshold A numeric threshold for defining significant correlation.
  #' @return A list of numeric vectors, each containing indices of a group of
  #' correlated variables.
  correlation <- abs(corr_matrix)
  diag(correlation) <- 0 # blank out diagonal to ignore self-correlation
  groups <- list()
  visited <- rep(FALSE, ncol(correlation))

  for (i in 1:ncol(correlation)) {
    if (!visited[i]) {
      # Find index of variables correlated w/ the current var
      high_cor_vars <- which(correlation[, i] > threshold)
      if (length(high_cor_vars) > 0) {
        group <- unique(c(i, high_cor_vars)) # include current var in the group
        groups[[length(groups) + 1]] <- group
        visited[group] <- TRUE
      }
      else {
        visited[i] <- TRUE
      }
    }
  }
  return(groups)
}



# Function to drop select highly correlated vars
reduceDataset <- function(df, corr_matrix, threshold) {
  #' Reduce Dataset by Dropping Highly Correlated Variables
  #'
  #' This function reduces a dataset by identifying and dropping highly correlated
  #' variables based on the provided correlation matrix and threshold.
  #'
  #' @param df A dataframe containing the dataset to be reduced.
  #' @param corr_matrix A square symmetric matrix of correlations between
  #' variables in `df`.
  #' @param threshold A numeric threshold to identify high correlations.
  #' @return A dataframe with reduced variables.
  groups <- findCorrelatedGroups(corr_matrix, threshold)
  to_drop <- numeric()

  for (group in groups) {
    # Subroutine to select which to drop (inverse of variance explained)
    variances <- sapply(group, function(index) {
      # Below is to fix bug with fread/data.table (depending on how data is read in)
      if (is.data.table(df)) {
        var(as.vector(df[[index]]))
      } else {
        var(df[, index, drop = FALSE])
      }
    })

    # Drop variable with minimum variance
    min_variance_index <- which.min(variances)

    to_drop_var <- group[min_variance_index]

    to_drop <- c(to_drop, to_drop_var)
  }

  to_drop <- unique(to_drop)

  # only drop columns that exist in the dataset (bug fix)
  to_drop <- to_drop[to_drop <= ncol(df)]

  if (length(to_drop) > 0) {
    reduced_df <- dplyr::select(df, -to_drop)
  }
  else {
    reduced_df <- df # If nothing to drop, return the original dataset (bug fix)
  }

  return(reduced_df)
}
```

```
check_mar <- function(data) {
  #' Check Missing At Random (MAR) in Data
  #'
```

```r
#' This function tests each variable in the dataset for missingness being at
#' random, conditional on all other variables, using logistic regression models.
#'
#' @param data A dataframe with variables to test for MAR.
#' @return A list where each entry corresponds to a variable in `data` and
#' contains results of MAR tests.


all_vars <- names(data)

# Init object for results
results <- list()

for (var in all_vars) {
  # Generate missing indicator
  missing_var_name <- paste0("missing_", make.names(var))
  data[[missing_var_name]] <- as.integer(is.na(data[[var]]))

  # formula for GLM
  formula_vars <- sapply(all_vars[all_vars != var], function(v) paste0("`", v, "`"))
  formula_str <- paste0(missing_var_name, " ~ ", paste(formula_vars, collapse = " + "))
  formula <- as.formula(formula_str)

  model <- tryCatch(
    {
      glm(formula, data = data, family = binomial())
    },
    error = function(e) {
      return(NULL)
    }
  )

  if (!is.null(model)) {
    # check for sig. variables
    summary_model <- summary(model)
    pvalues <- summary_model$coefficients[, "Pr(>|z|)"]
    significant_vars <- names(pvalues[pvalues < 0.05 & !is.na(pvalues)])

    # removve intercept from significant vars (if present)
    significant_vars <- significant_vars[significant_vars != "(Intercept)"]

    # rm backticks from significant vars for cleaner output
    significant_vars <- gsub("`", "", significant_vars)

    results[[var]] <- list(
      is_mar = length(significant_vars) == 0,
      significant_vars = significant_vars
    )
  } else {
      results[[var]] <- list(
        is_mar = NA,
        significant_vars = NA,
        error = "Error in fitting GLM"
      )
    }

    # remove temporary missing indicator col
    data[[missing_var_name]] <- NULL
  }

  return(results)
}




calculate_mse <- function(actual, predicted) {
  #' Calculate Mean Squared Error
  #'
  #' This function computes the mean squared error between actual and predicted
  #' numerical values.
  #'
  #' @param actual A numeric vector of actual values.
  #' @param predicted A numeric vector of predicted values.
  #' @return Numeric value representing the mean squared error.

  # inputs must be numeric
  actual <- as.numeric(actual)
  predicted <- as.numeric(predicted)

  # calc squared differences
  squared_diff <- (actual - predicted)^2

  # Remove NA / infinite values
  valid_diff <- squared_diff[is.finite(squared_diff) & !is.na(squared_diff)]

  # calc MSE
  mse <- sum(valid_diff) / length(valid_diff)

  return(mse)
}



variable_importance_reg <- function(model, data, target_column, n_iterations = 20) {
  #' Calculate Variable Importance for Linear Regression Models
  #'
  #' This function estimates the importance of each predictor variable in a
  #' regression model by measuring the increase in prediction error after
  #' permuting each predictor variable.
  #'
  #' @param model A regression model object (e.g., lm, lmer).
  #' @param data A dataframe containing the data used in the model.
  #' @param target_column The name of the target (dependent) variable in `data`.
  #' @param n_iterations The number of iterations to perform for estimating
  #' importance (default is 20).
  #' @return A dataframe with variables and their relative importance scores.

  predictor_vars <- all.vars(formula(model))[-1]
  original_preds <- as.numeric(predict(model, data))
  original_mse <- calculate_mse(data[[target_column]], original_preds)

  importance_scores <- matrix(0, nrow = n_iterations, ncol = length(predictor_vars))
  colnames(importance_scores) <- predictor_vars

  for (i in 1:n_iterations) {
    for (var in predictor_vars) {
      data_shuffled <- data
      data_shuffled[[var]] <- sample(data_shuffled[[var]])

      shuffled_preds <- as.numeric(predict(model, data_shuffled))
      shuffled_mse <- calculate_mse(data_shuffled[[target_column]], shuffled_preds)

      importance_scores[i, var] <- shuffled_mse - original_mse
    }
  }

  avg_importance <- abs(colMeans(importance_scores))
  se_importance <- apply(importance_scores, 2, sd) / sqrt(n_iterations)

  result <- data.frame(
    Variable = predictor_vars,
    Importance = avg_importance,
    SE = se_importance
  )

  result <- result[order(-result$Importance), ]
  result$`Relative Importance` <- result$Importance / max(result$Importance) * 100

  return(result)
}
```