MinneMUDAC DS - Novice Questions

Noah Lee

Big Brother Big Sisters Twin Cities the largest and oldest youth award-winning mentoring organization in the greater Twin Cities. Each year, we match up youth (Littles age 8-13) and their families with caring adults (Bigs) who volunteer as mentors. Through a variety of community-based, school-based, and workplace-based mentoring programs, and together with our community, we want every youth to have a mentor, be affirmed in who they are, and explore who they want to be.

Question: What are things present in 'successful matches'? - matches ongoing, lasting long or closed 'successfully'?

Preprocessing: (THIS IS STUFF FROM MY PREVIOUS NOTEBOOKS - IGNORE OR LOOK BACK AT IT LATER - MAINLY DELETING IRRELEVANT COLUMNS AND DATA TRANSFORMATION)

```
df <- read.csv('../Data/Novice.csv')
extract_binary_indicators <- function(df) {
  # Initialize new columns with FALSE (0)
  interest_categories <- c("has_interests", "personality_compatibility", "has_proximity",
                           "has_commitment", "has_experience", "has_preference",
                           "has_challenges", "has_goals")

  for (category in interest_categories) {
    df[[category]] <- FALSE  # Initialize with FALSE for all rows
  }

  # Define keywords for each category
  keywords <- list(
    has_interests = c("outdoors", "hiking", "biking", "fishing", "camping", "parks", "nature", "gardenin
    personality_compatibility = c("outgoing", "talkative", "bubbly", "energetic", "enthusiastic", "char
                                  "shy", "reserved", "quiet", "introverted", "soft-spoken", "calm",
                                  "adventurous", "curious", "exploratory", "open to new things",
                                  "friendly", "kind", "sweet", "thoughtful", "empathetic",
                                  "funny", "goofy", "humorous", "light-hearted",
                                  "mature", "respectful", "responsible", "thoughtful",
                                  "active", "sporty", "energetic", "athletic",
                                  "creative", "imaginative", "artistic", "crafty",
                                  "patient", "calm", "steady", "nurturing"),
    has_proximity = c("miles", "minutes", "close", "far", "convenient", "driving", "traffic", "commute"
    has_commitment = c("long-term", "committed", "consistent", "reliable", "short-term", "temporary", "
    has_experience = c("child experience", "nanny", "teacher", "coach", "mentor", "social work", "couns
    has_preference= c("age", "younger", "older", "in 20s", "gender", "male", "female", "religion", "Chr
    has_challenges = c("behavioral challenges", "ADHD", "unmedicated", "redirection", "mental health",
    has_goals = c("self-esteem", "confidence", "self-image", "leadership", "decision-making", "independ
  )
```

```r
  # Check if Rationale.for.Match column exists in the dataframe
  if (!"Rationale.for.Match" %in% names(df)) {
    warning("Column 'Rationale.for.Match' not found in dataframe. No keywords will be extracted.")
    # Return dataframe with all FALSE values
    return(df)
  }

  # Process each row
  for (i in 1:nrow(df)) {
    rationale <- df$Rationale.for.Match[i]

    # Skip if rationale is NA or empty
    if (is.na(rationale) || rationale == "") {
      next
    }

    # Check for keywords in each category
    for (category in names(keywords)) {
      category_keywords <- keywords[[category]]
      for (keyword in category_keywords) {
        if (grepl(keyword, rationale, ignore.case = TRUE)) {
          df[[category]][i] <- TRUE
          break  # Once we find a match, no need to check other keywords in this category
        }
      }
    }
  }

  # Convert logical columns to factors (0/1)
  for (category in interest_categories) {
    df[[category]] <- as.factor(as.integer(df[[category]]))
  }

  return(df)
}

# Apply the function to your dataframe
df <- extract_binary_indicators(df)
df$Match.ID.18Char <- NULL
df$Little.ID <- NULL
df$Big.ID <- NULL
df$Big..Military <- NULL
df$Big.Employer <- NULL
df$Closure.Details <- NULL
df$Big.Open.to.Cross.Gender.Match <- NULL
df$Big.Contact..Interest.Finder...Sports <- NULL
df$Big.Contact..Interest.Finder...Places.To.Go <- NULL
df$Big.Contact..Interest.Finder...Hobbies <- NULL
df$Big.Contact..Interest.Finder...Entertainment <- NULL
df$Big.Contact..Interest.Finder...Hobbies <- NULL
df$Big.Contact..Created.Date <- NULL
df$Big.Enrollment..Created.Date <- NULL
df$Little.Contact..Interest.Finder...Sports <- NULL
```

```
df$Little.Contact..Interest.Finder...Outdoors <- NULL
df$Little.Contact..Interest.Finder...Arts <- NULL
df$Little.Contact..Interest.Finder...Places.To.Go <- NULL
df$Little.Contact..Interest.Finder...Hobbies <- NULL
df$Little.Contact..Interest.Finder...Entertainment <- NULL
df$Little.Contact..Interest.Finder...Other.Interests <- NULL
df$Little.Other.Interests <- NULL
df$Little.Contact..Interest.Finder...Career <- NULL
df$Little.Contact..Interest.Finder...Personality <- NULL
df$Little.Contact..Interest.Finder...Three.Wishes <- NULL
df$Little.Other.Interests <- NULL
df$Rationale.for.Match <- NULL
df$Big.County[df$Big.County == ""] <- NA
df$Match.Activation.Date <- as.Date(df$Match.Activation.Date, format="%Y-%m-%d")
df$Big.Approved.Date <- as.Date(df$Big.Approved.Date, format="%Y-%m-%d")
df$Big.Acceptance.Date <- as.Date(df$Big.Acceptance.Date, format="%Y-%m-%d")
df$Match.Closure.Meeting.Date <- as.Date(df$Match.Closure.Meeting.Date, format="%Y-%m-%d")
df$Big.Birthdate <- as.Date(df$Big.Birthdate, format="%Y-%m-%d")
df$Little.Birthdate <- as.Date(df$Little.Birthdate, format="%Y-%m-%d")
df$Little.Interview.Date <- as.Date(df$Little.Interview.Date, format="%Y-%m-%d")
#Function to check if Big and Little ethnicities share any keywords
check_ethnicity_match <- function(df) {
  # Create a new column to store the matching result
  df$Ethnicity_Match <- FALSE

  # Loop through each row
  for (i in 1:nrow(df)) {
    # Get the Big and Little race/ethnicity values
    big_race <- df$Big.Race.Ethnicity[i]
    little_race <- df$Little.Participant..Race.Ethnicity[i]

    # Skip if either value is NA
    if (is.na(big_race) || is.na(little_race)) {
      df$Ethnicity_Match[i] <- NA
      next
    }

    # Convert to character (in case they're factors)
    big_race <- as.character(big_race)
    little_race <- as.character(little_race)

    # Split strings by semicolons to handle multiple ethnicities
    big_races <- unlist(strsplit(big_race, ";"))
    little_races <- unlist(strsplit(little_race, ";"))

    # Clean up any leading/trailing spaces
    big_races <- trimws(big_races)
    little_races <- trimws(little_races)

    # Check if there's any match
    match_found <- FALSE
    for (b in big_races) {
      for (l in little_races) {
```

```
        # Extract keywords to compare (simplify the comparison)
        keywords <- c("White", "Black", "Asian", "Hispanic", "Indian", "Alaska",
                      "Middle Eastern", "North African", "Other")

        # Check for each keyword
        for (keyword in keywords) {
          if (grepl(keyword, b, ignore.case = TRUE) &&
              grepl(keyword, l, ignore.case = TRUE)) {
            match_found <- TRUE
            break
          }
        }
        if (match_found) break
      }
      if (match_found) break
    }

    # Assign the result
    df$Ethnicity_Match[i] <- match_found
  }

  return(df)
}

df <- check_ethnicity_match(df)
df$Big.Race.Ethnicity <- NULL
df$Little.Participant..Race.Ethnicity <- NULL
df$Stage <- factor(ifelse(df$Stage == "Closed", "Closed", "Active"))
df[df == ""] <- NA
df$Big.Languages[df$Big.Languages == ""] <- NA
df$Big.Gender <- factor(df$Big.Gender,
                        levels = c("Female", "Male"),
                        labels = c("Female", "Male"))

df$Program <- as.factor(df$Program)
df$Program.Type <- as.factor(df$Program.Type)
df$Big.Level.of.Education <- NULL
df$Big.Languages <- NULL
df$Big.Car.Access <- NULL
df$Big.Contact..Preferred.Communication.Type <- NULL
df$Big.Contact..Former.Big.Little <- NULL
df$Big.Contact..Volunteer.Availability <- NULL
df$Little.RTBM.Date.in.MF <- NULL
df$Little.Contact..Language.s..Spoken <- NULL
df$Little.Acceptance.Date <- NULL
df$Little.Application.Received <- NULL
df$Little.Moved.to.RTBM.in.MF <- NULL
df$Little.Mailing.Address.Census.Block.Group <- NULL
df$Little.Acceptance.Date <- NULL
df$Big.Home.Census.Block.Group <- NULL
df$Big.Employer.School.Census.Block.Group <- NULL
df$Little.Gender <- NULL
df$Little.Birthdate <- NULL
```

```r
df$Little.RTBM.in.Matchforce <- NULL
df$Little.Interview.Date <- NULL
df$Big.Acceptance.Date <- NULL
df$Big.Assessment.Uploaded <- NULL
df$Big.Days.Interview.to.Match <- NULL
df$Big.Days.Interview.to.Acceptance <- NULL
consolidate_counties <- function(county_data, min_frequency = 50) {
  consolidated <- county_data
  county_counts <- table(county_data[county_data != ""])
  rare_counties <- names(county_counts[county_counts < min_frequency])
  consolidated[consolidated %in% rare_counties] <- "Other"
  # Convert to factor with meaningful levels
  consolidated <- factor(consolidated)

  return(consolidated)
}

df$County_Factor <- consolidate_counties(df$Big.County)
summary(df$County_Factor)
```

```
##      Anoka      Dakota   Hennepin      Other      Ramsey Washington       NA's
##        139         157       1485        152         592         95        655
```

```r
df$Big.County <- NULL
# Function to categorize text fields based on keywords
categorize_text <- function(text_vector, category_rules, default_category = "Other") {
  result <- rep(default_category, length(text_vector))

  if (any(is.na(text_vector))) {
    result[is.na(text_vector)] <- NA
  }

  text_vector <- tolower(trimws(text_vector))

  for (category_name in names(category_rules)) {
    keywords <- category_rules[[category_name]]

    # Check if any keyword appears in each entry
    match_indices <- sapply(text_vector, function(text) any(grepl(paste(keywords, collapse = "|"), text

    # Assign the category where matches occur
    result[match_indices] <- category_name
  }

  return(factor(result))
}

# Define category rules for each text field
closure_reason_rules <- list(
  "Scheduling_Issues" = c("schedule", "time", "availability", "busy", "time constraint"),
  "Relationship_Problems" = c("relationship", "conflict", "disagree", "personal", "not compatible", "in
  "Relocation" = c("move", "moved", "relocation", "relocate", "different city", "different state"),
  "Family_Issues" = c("family", "parent", "guardian", "parental"),
```

```r
  "School_Issues" = c("school", "academic", "education", "grade", "graduated", "graduate"),
  "Health_Issues" = c("health", "illness", "medical", "sick", "disease", "covid", "deceased"),
  "Behavior_Issues" = c("behavior", "conduct", "attitude", "disciplin"),
  "Program_Requirements" = c("requirement", "qualify", "eligibility", "criteria", "guideline", "infract
  "Success" = c("success", "successful")
)

occupation_rules <- list(
  "Business_Finance" = c("account", "financ", "budget", "analyst", "bank", "economic", "market", "busine
  "Education" = c("teach", "professor", "instructor", "education", "academic", "school", "college", "uni
  "Healthcare" = c("doctor", "nurse", "medical", "health", "dental", "therapist", "clinic", "hospital",
  "Technology" = c("software", "developer", "engineer", "IT", "computer", "tech", "program", "web", "da
  "Legal" = c("lawyer", "attorney", "legal", "law", "judge", "paralegal"),
  "Arts_Media" = c("artist", "design", "writer", "media", "journalist", "creative", "music", "film", "a
  "Service_Industry" = c("retail", "sales", "service", "hospitality", "restaurant", "customer", "child"
  "Trades_Labor" = c("construct", "mechanic", "carpenter", "electric", "plumb", "repair", "builder", "l
  "Student" = c("student", "graduate", "undergrad"),
  "Unknown" = c("unknown"),
  "Retired" = c("retire")
)

df$Closure_Reason_Category <- categorize_text(df$Closure.Reason, closure_reason_rules)
df$Occupation_Category <- categorize_text(df$Big.Occupation, occupation_rules)
summary(df$Closure_Reason_Category)
```

```
##        Family_Issues         Health_Issues                 Other
##                  684                   173                     2
##  Program_Requirements Relationship_Problems            Relocation
##                  168                   343                   297
##     Scheduling_Issues         School_Issues               Success
##                  401                   326                    95
##                 NA's
##                  786
```

```r
summary(df$Occupation_Category)
```

```
##       Arts_Media Business_Finance        Education       Healthcare
##              103              777              169              278
##            Legal            Other          Retired Service_Industry
##              115              160               29              358
##          Student       Technology     Trades_Labor          Unknown
##              500              234               36              191
##             NA's
##              325
```

```r
df$Closure.Reason <- NULL
df$Big.Occupation <- NULL
df$Big.Days.Acceptance.to.Match <- abs(df$Big.Days.Acceptance.to.Match)

# Sort the original DataFrame in place
df <- df[order(df$Match.Activation.Date), ]
# Create a factor variable with two levels
```

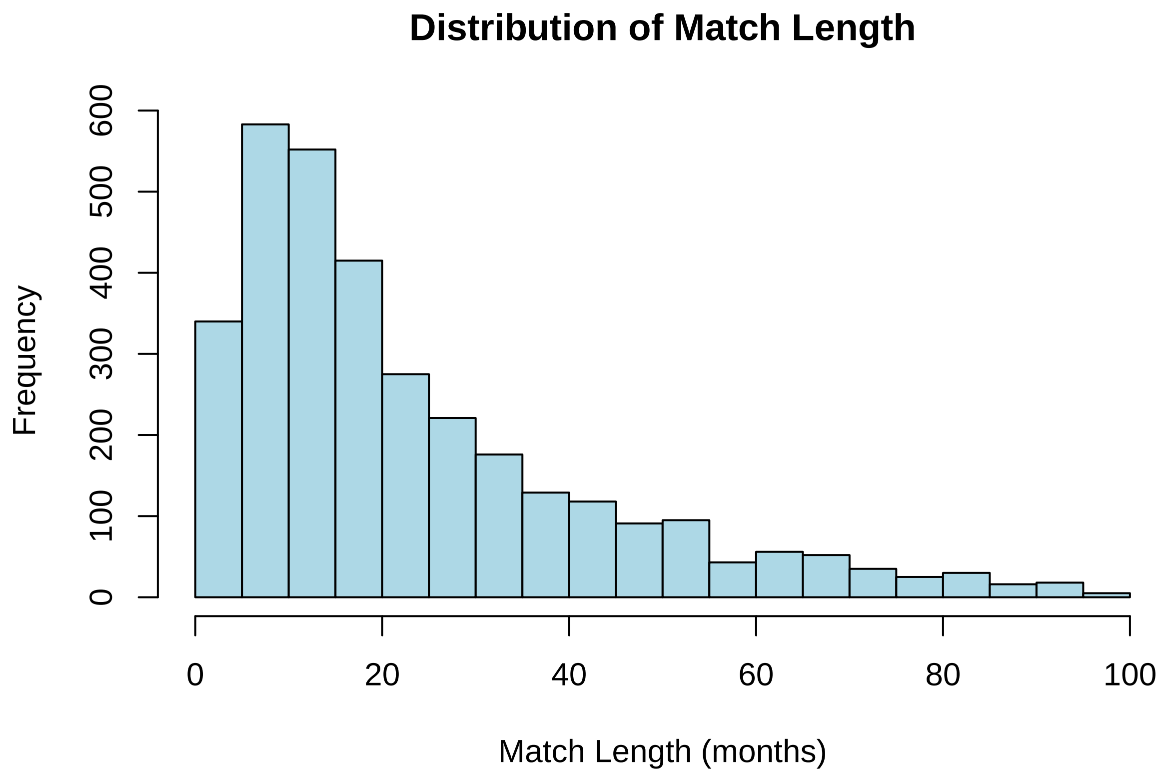
```r
df$Big.Enrollment..Record.Type <- factor(
  ifelse(df$Big.Enrollment..Record.Type == "CB Volunteer Enrollment",
         "CB Volunteer Enrollment",
         "Others")
)
# Create new categorical variable from Big.Contact..Marital.Status
df$Big.Contact..Marital.Status <- factor(
  case_when(
    df$Big.Contact..Marital.Status == "Single" ~ "Single",
    !is.na(df$Big.Contact..Marital.Status) ~ "Not Single",
    TRUE ~ NA_character_
  ),
  levels = c("Single", "Not Single")
)
df$Stage <- ifelse(df$Stage == "Closed", 1, 0)
```

Understand and analyze the response variable distributions: Match Length:

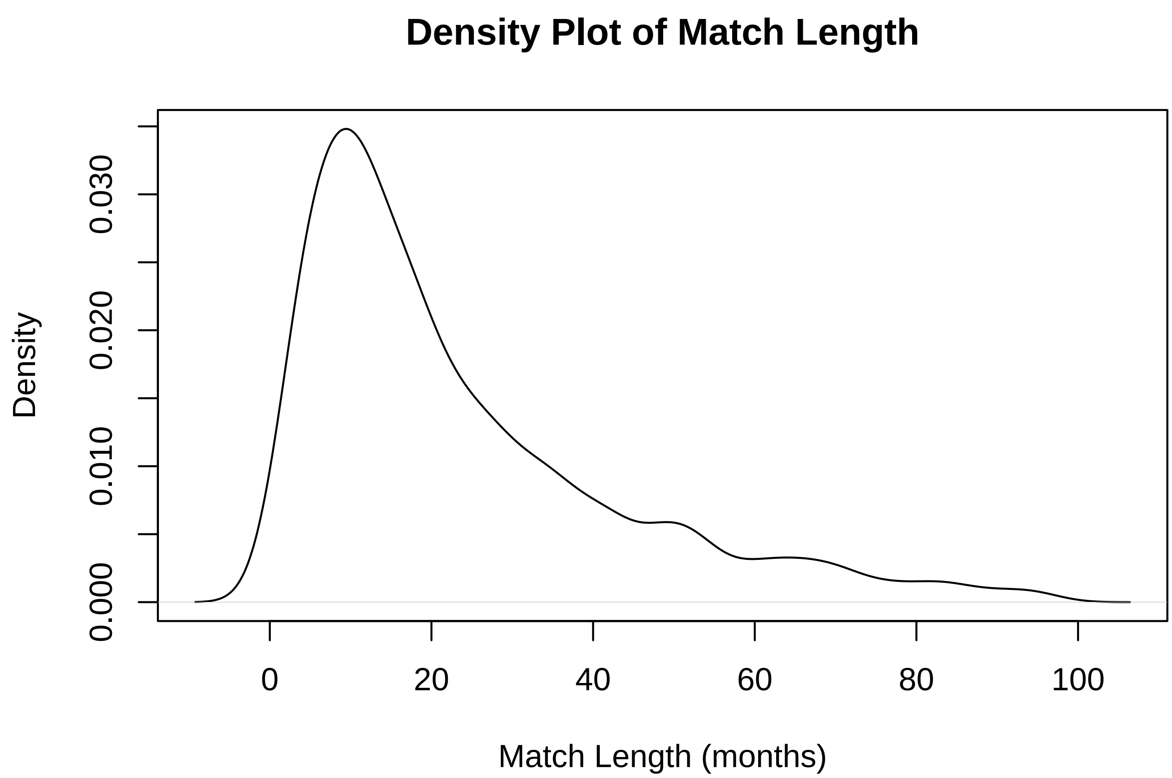
```r
summary(df$Match.Length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    9.10   16.80   23.38   32.20   97.20
```

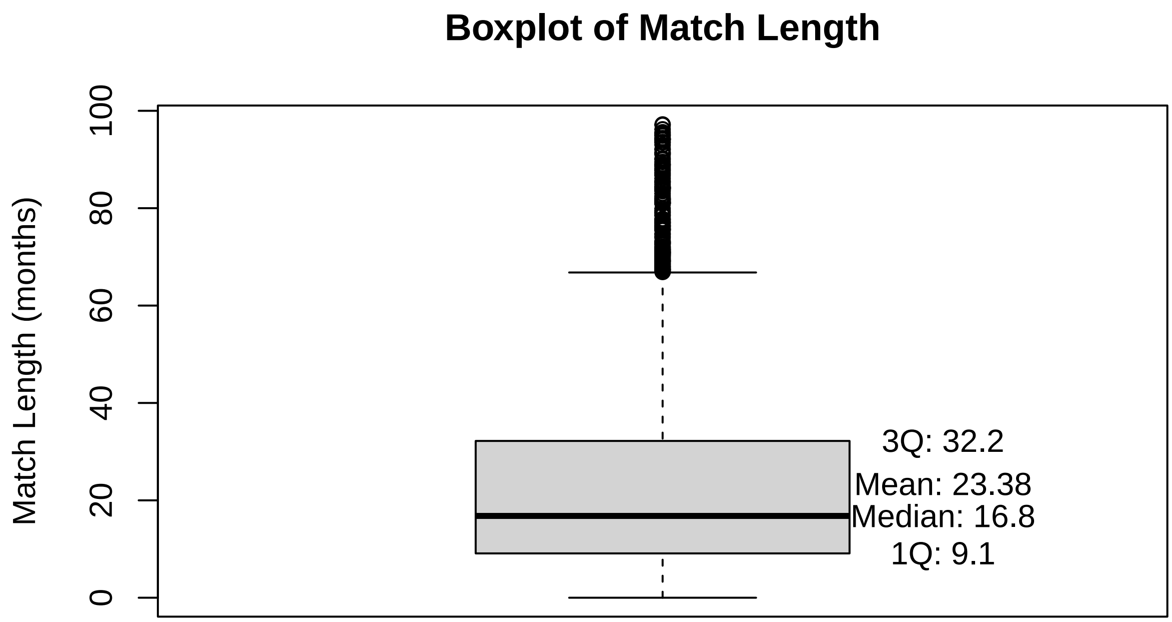
```r
# Histogram to visualize distribution
hist(df$Match.Length,
     main="Distribution of Match Length",
     xlab="Match Length (months)",
     col="lightblue",
     breaks=20)
```



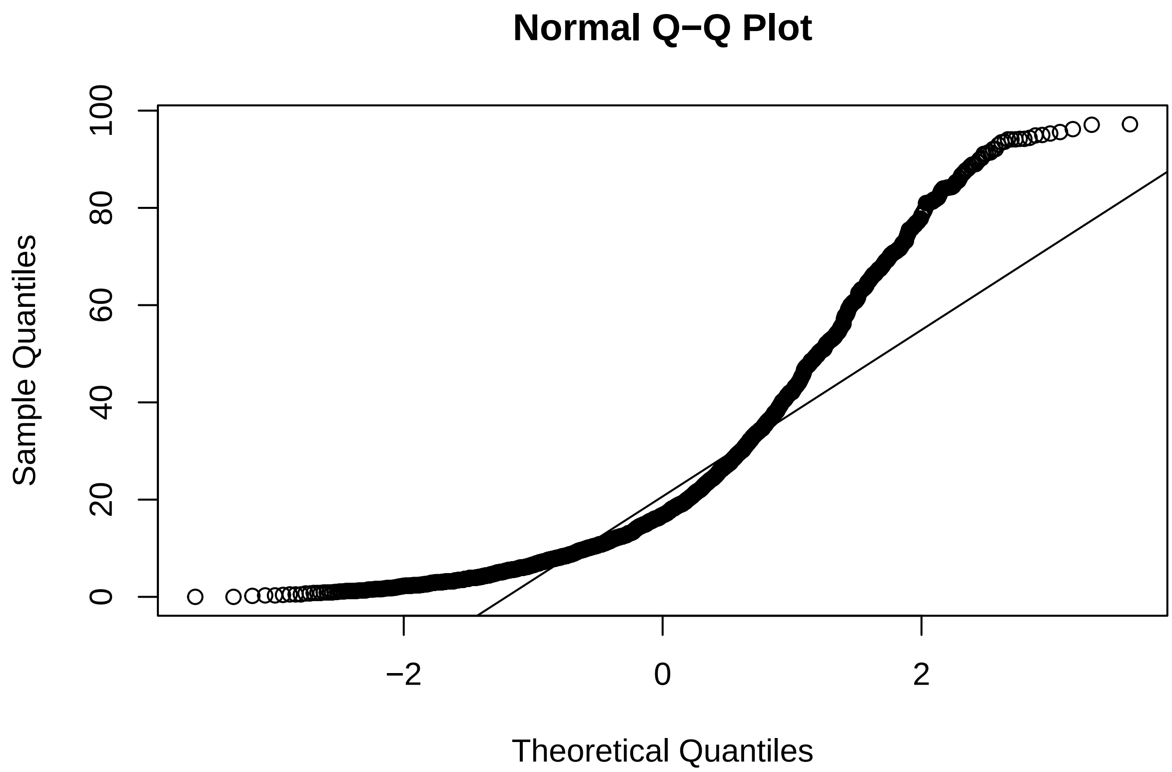

```
# Density plot
plot(density(df$Match.Length),
     main="Density Plot of Match Length",
     xlab="Match Length (months)")
```




```r
# Boxplot
boxplot(df$Match.Length,
        main="Boxplot of Match Length",
        ylab="Match Length (months)")

stats <- summary(df$Match.Length)
mean_value <- mean(df$Match.Length)
q1 <- stats["1st Qu."]
median_value <- stats["Median"]
q3 <- stats["3rd Qu."]

text(x = 1.3, y = mean_value, labels = paste("Mean:", round(mean_value, 2)), col = "black")
text(x = 1.3, y = q1, labels = paste("1Q:", round(q1, 2)), col = "black")
text(x = 1.3, y = median_value, labels = paste("Median:", round(median_value, 2)), col = "black")
text(x = 1.3, y = q3, labels = paste("3Q:", round(q3, 2)), col = "black")
```




```r
# Check for normality
qqnorm(df$Match.Length)
qqline(df$Match.Length) # Looks exponentially distributed
```



Definitely not normally distributed as expected. Maybe Log transform? Survival analysis?

```
table(df$Closure_Reason_Category)
```

```
##
##         Family_Issues          Health_Issues                  Other
##                   684                    173                      2
##  Program_Requirements Relationship_Problems             Relocation
##                   168                    343                    297
##     Scheduling_Issues          School_Issues                Success
##                   401                    326                     95
```

Only 95 defined as successful.

How do the response variable distributions vary across Program Type?

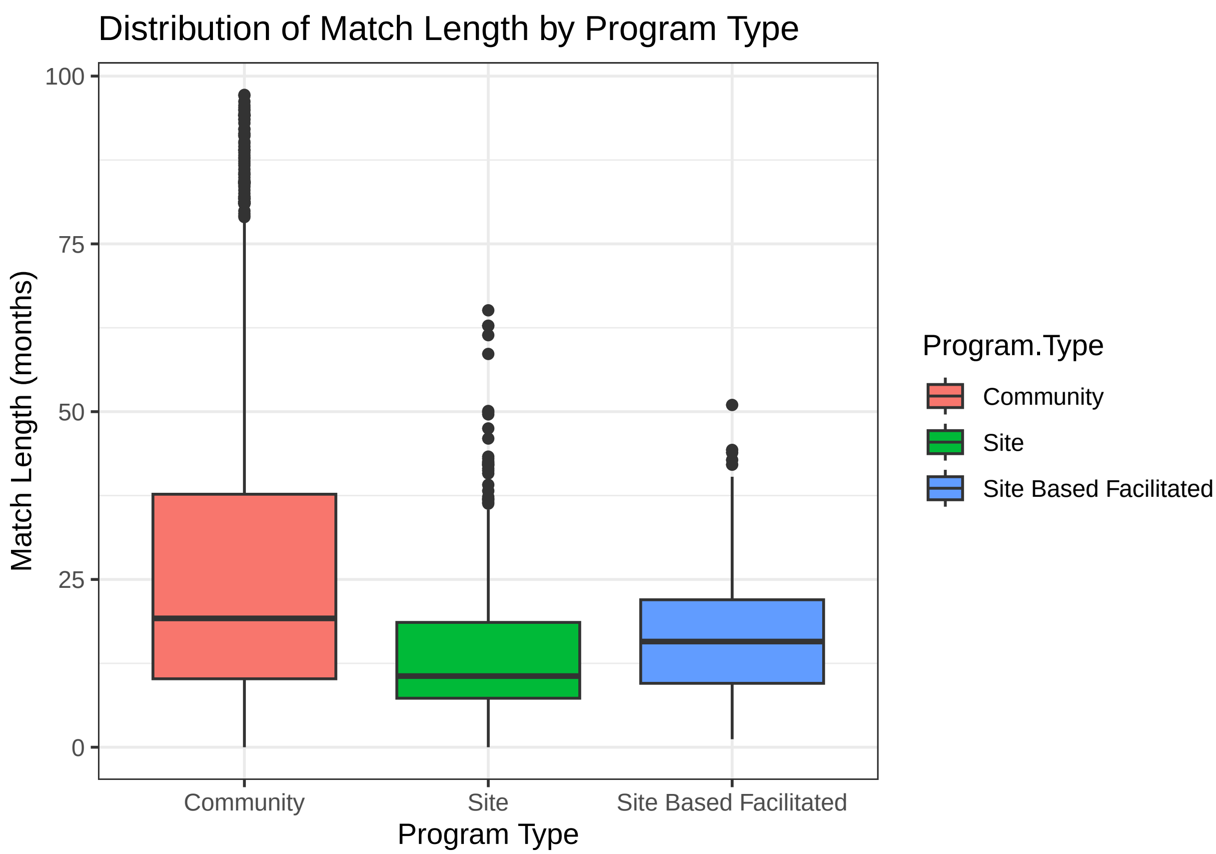
```
table(df$Program.Type)
```

```
##
##               Community                     Site Site Based Facilitated
##                    2420                      570                    282
##         Site Based Plus
##                       3
```

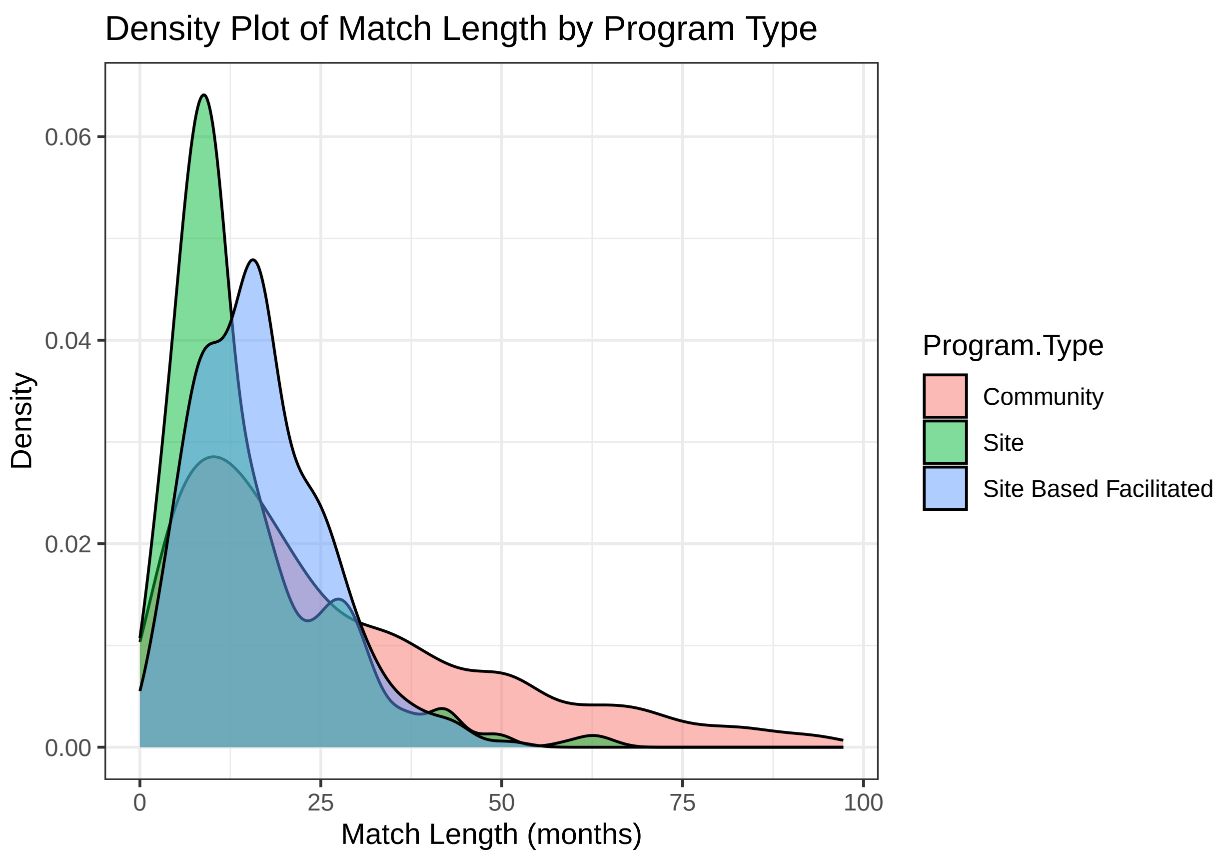
```r
df$Program.Type[df$Program.Type == "Site Based Plus"] <- NA # too little records to consider
df$Program.Type <- droplevels(df$Program.Type)
df_filtered <- df %>% filter(!is.na(Program.Type))
# Summary statistics by Program Type
summary_stats <- df %>%
  group_by(Program.Type) %>%
  summarise(
    Mean = mean(Match.Length, na.rm = TRUE),
    Median = median(Match.Length, na.rm = TRUE),
    SD = sd(Match.Length, na.rm = TRUE),
    Q1 = quantile(Match.Length, 0.25, na.rm = TRUE),
    Q3 = quantile(Match.Length, 0.75, na.rm = TRUE)
  )
print(summary_stats)
```

```
## # A tibble: 4 x 6
##   Program.Type            Mean Median    SD    Q1    Q3
##   <fct>                  <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 Community               26.3   19.2 21.3  10.2   37.7
## 2 Site                    14.2   10.6 10.8   7.3   18.6
## 3 Site Based Facilitated  16.7   15.8  9.02  9.52  22.0
## 4 <NA>                    33.4   27.2 26.9  18.6   45
```

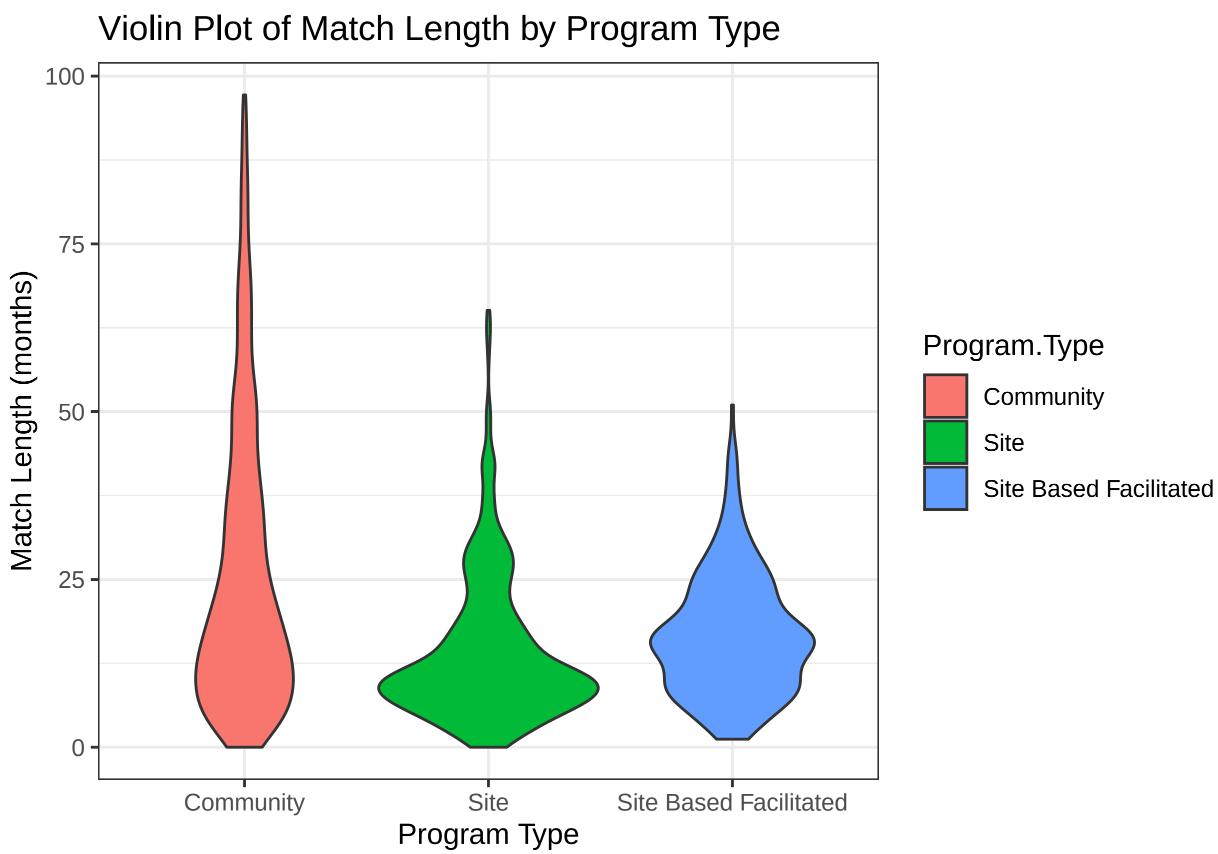
```r
# Boxplot to visualize distribution of Match.Length by Program.Type
ggplot(df_filtered, aes(x = Program.Type, y = Match.Length, fill = Program.Type)) +
  geom_boxplot(na.rm = TRUE) +
  labs(
    title = "Distribution of Match Length by Program Type",
    x = "Program Type",
    y = "Match Length (months)"
  ) +
  theme_bw()
```




```r
# Density plot to compare distributions
ggplot(df_filtered, aes(x = Match.Length, fill = Program.Type)) +
  geom_density(alpha = 0.5, na.rm = TRUE) +
  labs(
    title = "Density Plot of Match Length by Program Type",
    x = "Match Length (months)",
    y = "Density"
  ) +
  theme_bw()
```




```
# Violin plot for a more detailed view
ggplot(df_filtered, aes(x = Program.Type, y = Match.Length, fill = Program.Type)) +
  geom_violin(na.rm = TRUE) +
  labs(
    title = "Violin Plot of Match Length by Program Type",
    x = "Program Type",
    y = "Match Length (months)"
  ) +
  theme_bw()
```




```r
# ANOVA test
anova_result <- aov(Match.Length ~ Program.Type, data = df)
summary(anova_result)
```

```
##                 Df  Sum Sq Mean Sq F value Pr(>F)
## Program.Type     2   80777   40388   111.1 <2e-16 ***
## Residuals     3269 1188634     364
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3 observations deleted due to missingness
```

```r
# Chi-Square Test of Independence for Closure Reason
chi_square_result <- chisq.test(table(df$Closure_Reason_Category, df$Program.Type))
```

```
## Warning in chisq.test(table(df$Closure_Reason_Category, df$Program.Type)):
## Chi-squared approximation may be incorrect
```

```r
print(chi_square_result)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(df$Closure_Reason_Category, df$Program.Type)
## X-squared = 781.42, df = 16, p-value < 2.2e-16
```

```r
# Kruskal-Wallis test (non-parametric alternative)
kruskal.test(Match.Length ~ Program.Type, data = df)
```

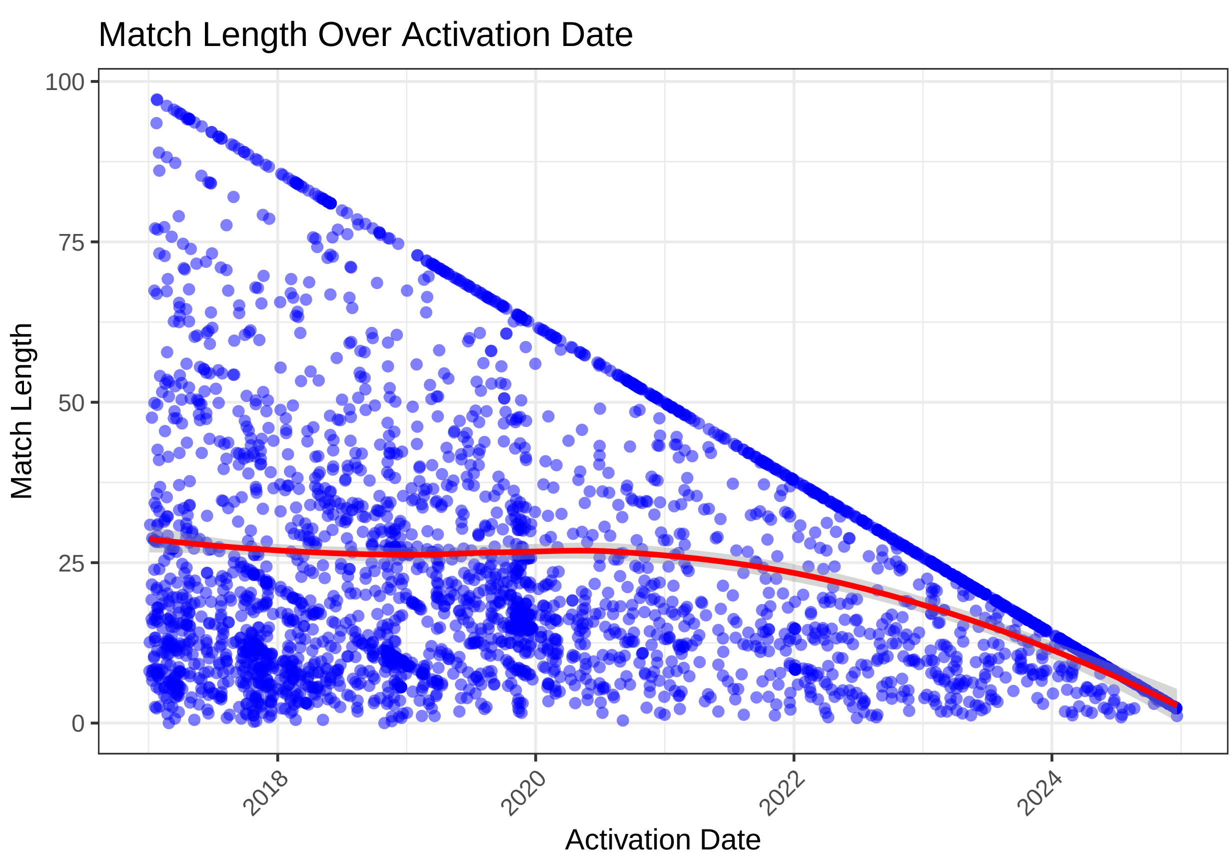
```
##
##  Kruskal-Wallis rank sum test
##
## data:  Match.Length by Program.Type
## Kruskal-Wallis chi-squared = 181.38, df = 2, p-value < 2.2e-16
```

Program.Type a significant predictor of Match Length and Closure Reason

Response distributions over time

```r
ggplot(df, aes(x = Match.Activation.Date, y = Match.Length)) +
  geom_point(alpha = 0.5, color = "blue") +  # Scatter plot with transparency
  geom_smooth(method = "loess", color = "red", se = TRUE) +  # Trend line
  labs(title = "Match Length Over Activation Date",
       x = "Activation Date",
       y = "Match Length") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

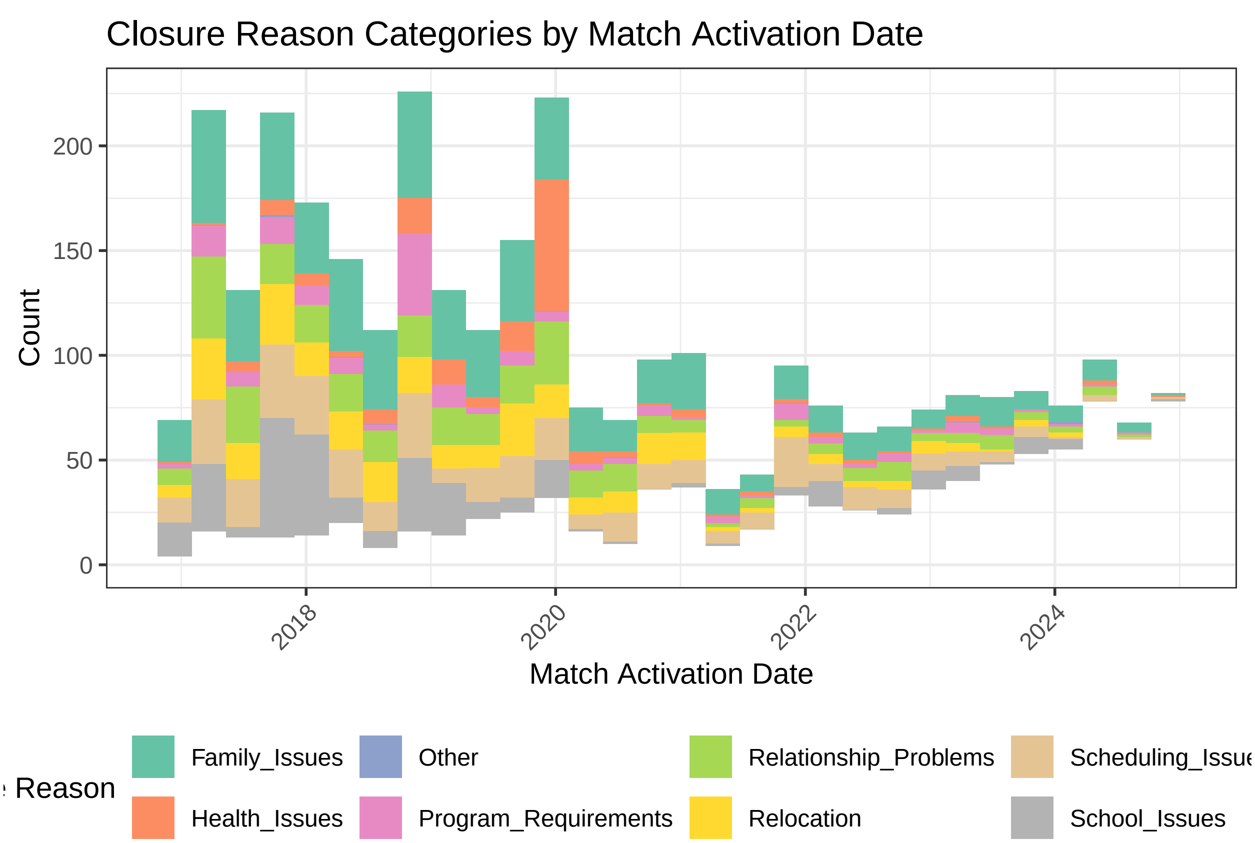
```
## `geom_smooth()` using formula = 'y ~ x'
```



Clearly older matches have an advantage - look into survival analysis and Cox potential hazards model.

```
ggplot(df, aes(x = Match.Activation.Date, fill = Closure_Reason_Category)) +
  geom_histogram(position = "stack", bins = 30) +
  labs(
    title = "Closure Reason Categories by Match Activation Date",
    x = "Match Activation Date",
    y = "Count",
    fill = "Closure Reason"
  ) +
  scale_fill_brewer(palette = "Set2") +
  theme_bw() +
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

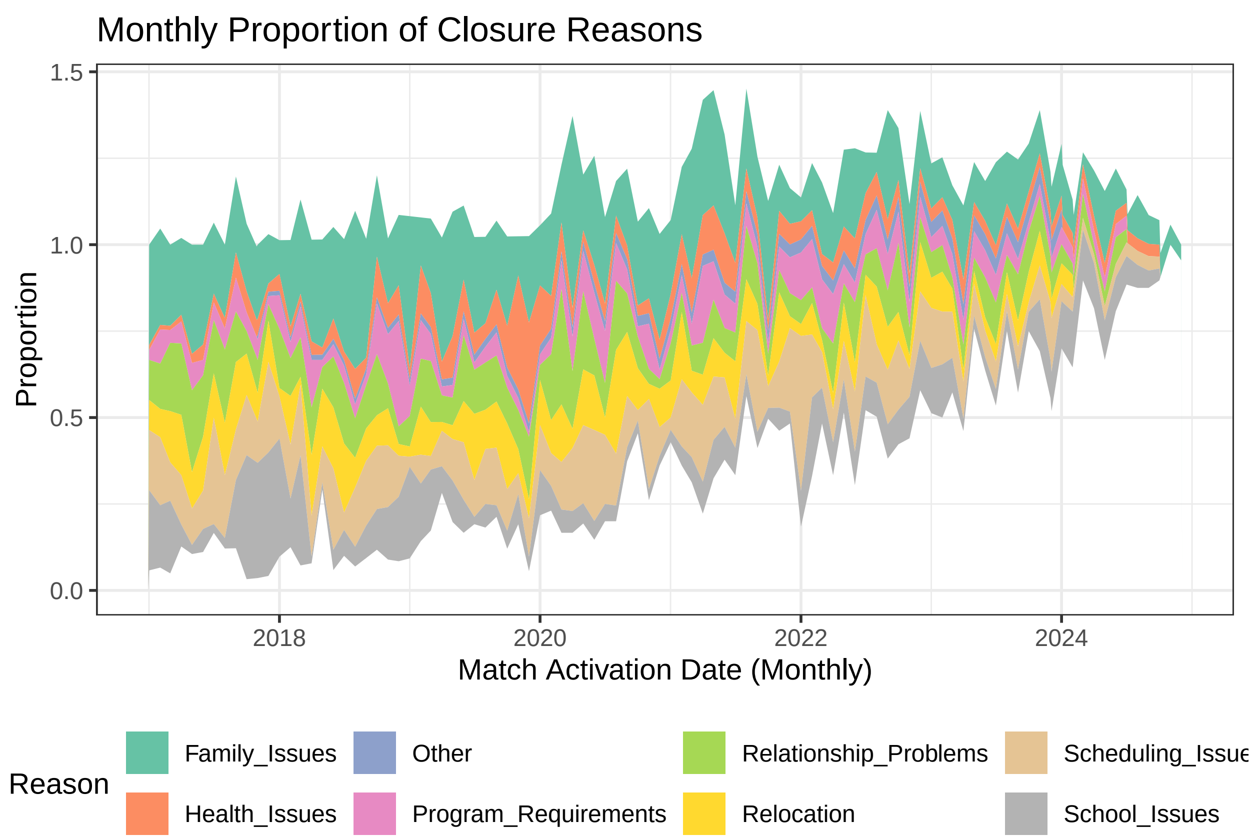
```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
```




```
df_monthly <- df %>%
  mutate(Month = floor_date(Match.Activation.Date, "month")) %>%
  group_by(Month, Closure_Reason_Category) %>%
  summarise(Count = n(), .groups = "drop") %>%
  group_by(Month) %>%
```

```r
  mutate(Proportion = Count / sum(Count))

ggplot(df_monthly, aes(x = Month, y = Proportion, fill = Closure_Reason_Category)) +
  geom_area() +
  labs(
    title = "Monthly Proportion of Closure Reasons",
    x = "Match Activation Date (Monthly)",
    y = "Proportion",
    fill = "Closure Reason"
  ) +
  scale_fill_brewer(palette = "Set2") +
  theme_bw() +
  theme(legend.position = "bottom")
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
```



Seems to stay relatively consistent - with a rise in 'NA' values in the bottom due to less match closures.

```r
table(df$Stage)
```

```
##
##     0    1
##   789 2486
```

2486 match closures, 789 active matches

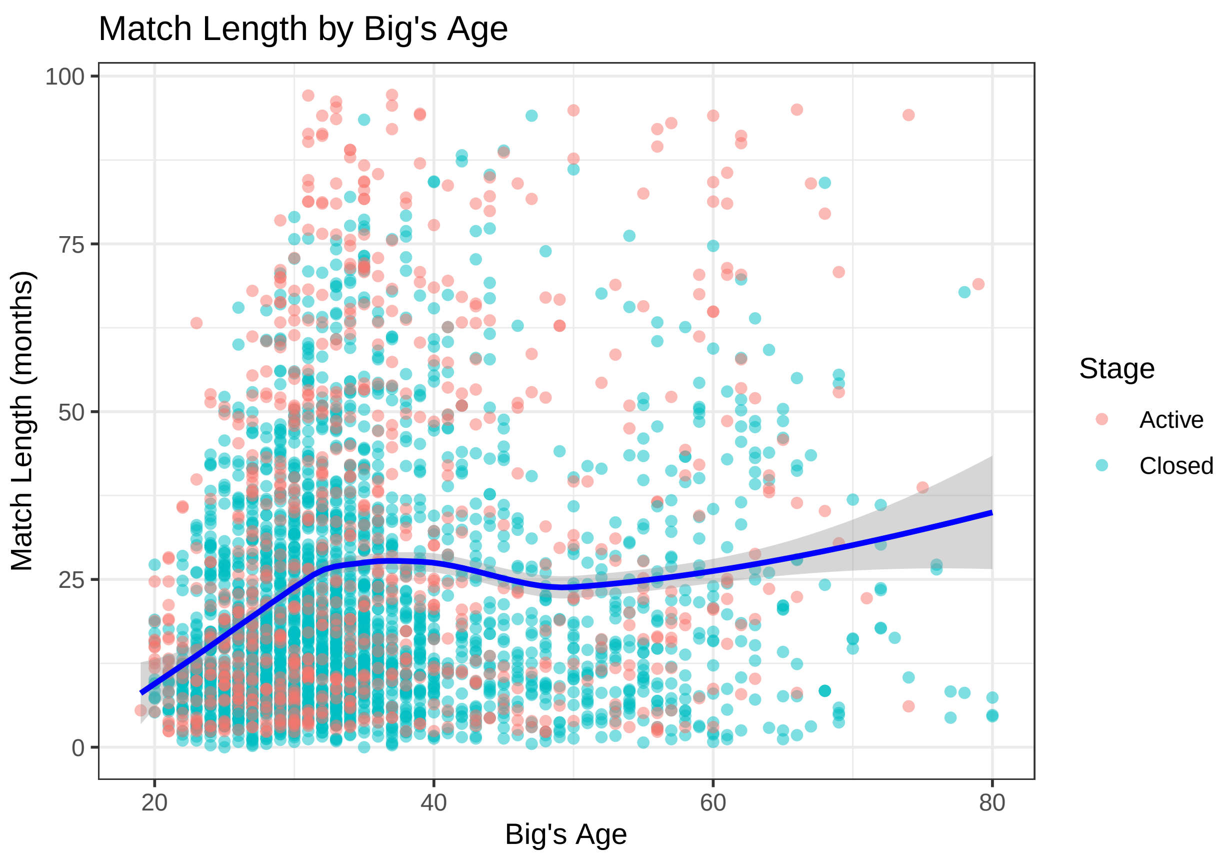
```r
str(df)
```

```
## 'data.frame':    3275 obs. of  26 variables:
##  $ Stage                       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Big.Age                     : int  78 37 35 59 48 38 41 44 50 26 ...
##  $ Big.Approved.Date           : Date, format: NA NA ...
##  $ Big.Gender                  : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 2 1 1 ...
##  $ Big.Birthdate               : Date, format: "1946-06-01" "1987-10-01" ...
##  $ Program                     : Factor w/ 34 levels "Alumni 2021",..: 22 9 9 22 22 22 9 22 22 22 ..
##  $ Program.Type                : Factor w/ 3 levels "Community","Site",..: 2 1 1 2 2 2 1 2 2 2 ...
##  $ Match.Activation.Date       : Date, format: "2017-01-03" "2017-01-04" ...
##  $ Match.Closure.Meeting.Date  : Date, format: NA NA ...
##  $ Big.Enrollment..Record.Type : Factor w/ 2 levels "CB Volunteer Enrollment",..: NA NA NA NA NA NA N
##  $ Big.Days.Acceptance.to.Match: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Big.Re.Enroll               : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Big.Contact..Marital.Status : Factor w/ 2 levels "Single","Not Single": NA NA NA NA NA NA NA NA N
##  $ Match.Length                : num  8.1 12.6 30.9 16.2 7.4 19.2 47.6 21.6 28.8 6.6 ...
##  $ has_interests               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ personality_compatibility   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ has_proximity               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ has_commitment              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ has_experience              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ has_preference              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ has_challenges              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ has_goals                   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Ethnicity_Match             : logi  TRUE TRUE FALSE FALSE FALSE TRUE ...
##  $ County_Factor               : Factor w/ 6 levels "Anoka","Dakota",..: 6 4 3 3 3 4 5 4 1 3 ...
##  $ Closure_Reason_Category     : Factor w/ 9 levels "Family_Issues",..: 1 1 7 1 1 5 1 7 6 4 ...
##  $ Occupation_Category         : Factor w/ 12 levels "Arts_Media","Business_Finance",..: 12 10 2 6 6
```

What influence do the various Big and/or Little demographic variables have on the response variable distributions?

```r
# Analysis of Big Age vs Match Length
ggplot(df, aes(x = Big.Age, y = Match.Length, color = factor(Stage, labels = c("Active", "Closed")))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", color = "blue") +
  labs(
    title = "Match Length by Big's Age",
    x = "Big's Age",
    y = "Match Length (months)",
    color = "Stage"  # Legend title
  ) +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



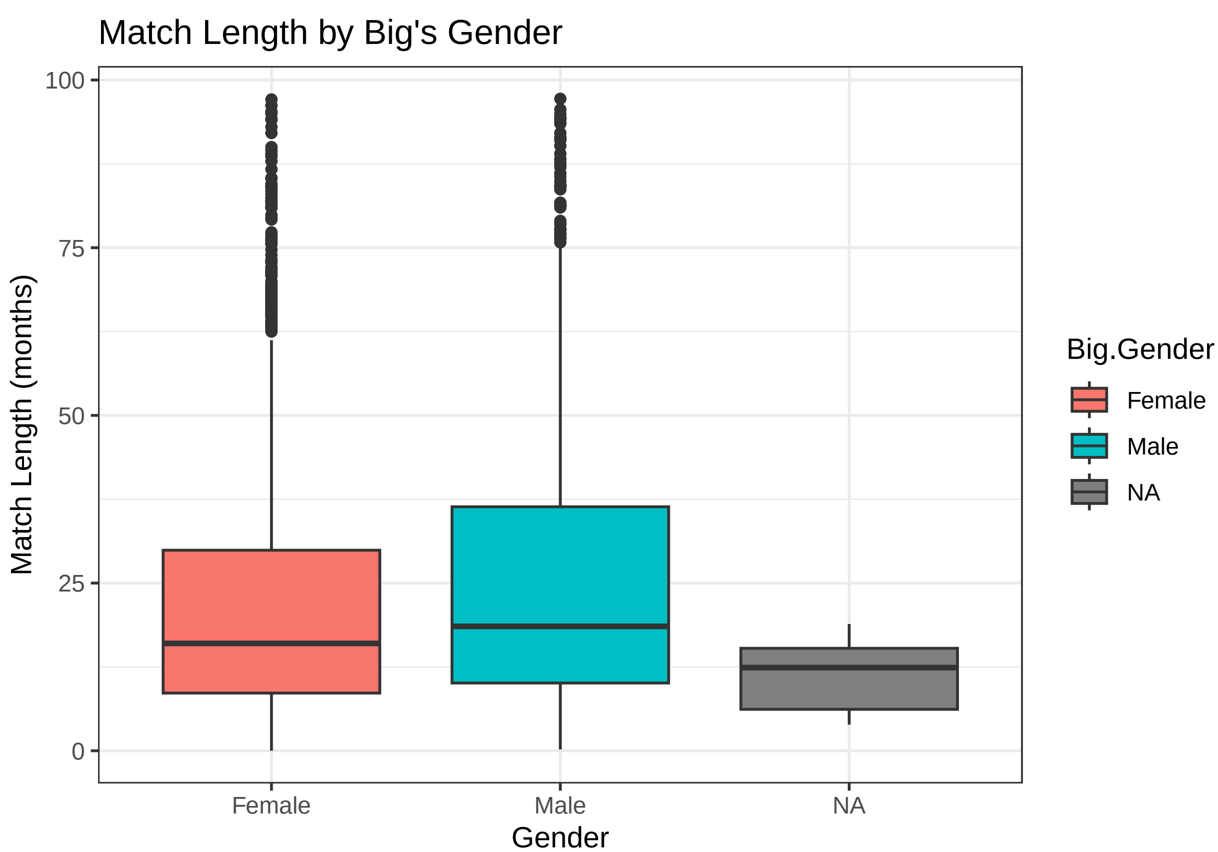

```
# Age group analysis
df$Age_Group <- cut(df$Big.Age,
                    breaks = c(0, 25, 35, 45, 55, 65, 100),
                    labels = c("18-25", "26-35", "36-45", "46-55", "56-65", "65+"))

age_summary <- df %>%
  group_by(Age_Group) %>%
  summarise(
    Mean_Length = mean(Match.Length, na.rm = TRUE),
    Median_Length = median(Match.Length, na.rm = TRUE),
    Count = n()
  )
age_summary
```

```
## # A tibble: 6 x 4
##   Age_Group Mean_Length Median_Length Count
##   <fct>           <dbl>         <dbl> <int>
## 1 18-25            14.2          10.9   454
## 2 26-35            24.0          18.1  1678
## 3 36-45            26.7          19.1   621
## 4 46-55            21.4          15.3   267
## 5 56-65            29.7          22.8   198
## 6 65+              29.6          23.4    57
```

```r
# Big Gender analysis
gender_summary <- df %>%
  group_by(Big.Gender) %>%
  summarise(
    Mean_Length = mean(Match.Length, na.rm = TRUE),
    Median_Length = median(Match.Length, na.rm = TRUE),
    Count = n()
  )
gender_summary
```

```
## # A tibble: 3 x 4
##   Big.Gender Mean_Length Median_Length Count
##   <fct>            <dbl>         <dbl> <int>
## 1 Female            22.1          16    1955
## 2 Male              25.4          18.6  1306
## 3 <NA>              11.2          12.4    14
```

```r
# Box plot of match length by Big's gender
ggplot(df, aes(x = Big.Gender, y = Match.Length, fill = Big.Gender)) +
  geom_boxplot() +
  labs(title = "Match Length by Big's Gender",
       x = "Gender",
       y = "Match Length (months)") +
  theme_bw()
```



```r
# Statistical test for gender difference
t.test(Match.Length ~ Big.Gender, data = df)
```

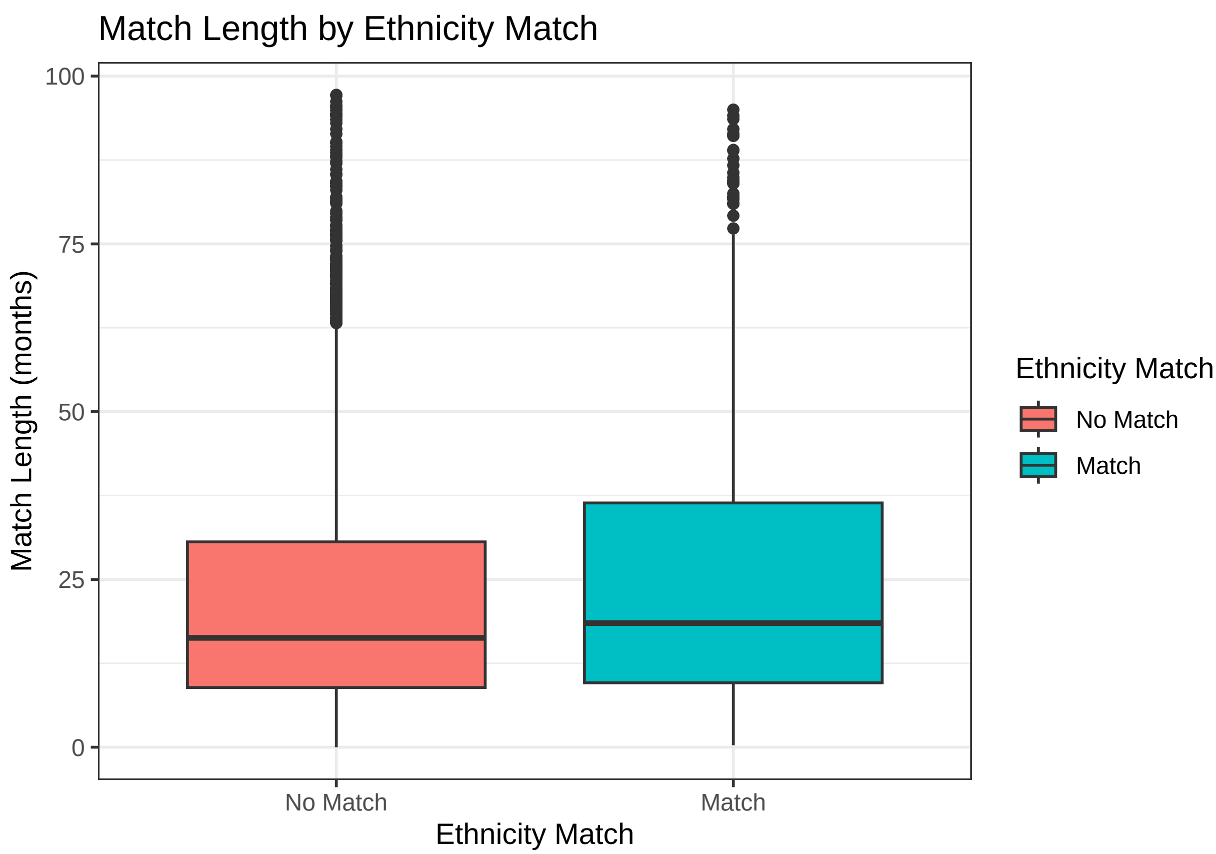
```
##
##  Welch Two Sample t-test
##
## data:  Match.Length by Big.Gender
## t = -4.6081, df = 2615.2, p-value = 4.259e-06
## alternative hypothesis: true difference in means between group Female and group Male is not equal to
## 95 percent confidence interval:
##  -4.703042 -1.895265
## sample estimates:
## mean in group Female   mean in group Male
##             22.11478             25.41394
```

Statistically discernable difference in match length and gender. Longer for male bigs.

```r
# Ethnicity match analysis
ethnicity_summary <- df %>%
  group_by(Ethnicity_Match) %>%
  summarise(
    Mean_Length = mean(Match.Length, na.rm = TRUE),
    Median_Length = median(Match.Length, na.rm = TRUE),
    Count = n()
  )
ethnicity_summary
```

```
## # A tibble: 2 x 4
##   Ethnicity_Match Mean_Length Median_Length Count
##   <lgl>                 <dbl>         <dbl> <int>
## 1 FALSE                  22.6          16.3  2322
## 2 TRUE                   25.2          18.5   953
```

```r
# Box plot for ethnicity match
ggplot(df %>% filter(!is.na(Ethnicity_Match)),
       aes(x = factor(Ethnicity_Match), y = Match.Length, fill = factor(Ethnicity_Match))) +
  geom_boxplot() +
  labs(title = "Match Length by Ethnicity Match",
       x = "Ethnicity Match",
       y = "Match Length (months)") +
  scale_x_discrete(labels = c("FALSE" = "No Match", "TRUE" = "Match")) +
  scale_fill_discrete(name = "Ethnicity Match", labels = c("No Match", "Match")) +
  theme_bw()
```



Statsitically discerinible difference for match length and ethnicity match - longer if same ethnicity.

```r
# Interest and proximity analysis
interest_summary <- df %>%
  group_by(has_interests) %>%
  summarise(
    Mean_Length = mean(Match.Length, na.rm = TRUE),
    Median_Length = median(Match.Length, na.rm = TRUE),
    Count = n()
  )
interest_summary
```

```
## # A tibble: 2 x 4
##   has_interests Mean_Length Median_Length Count
##   <fct>               <dbl>         <dbl> <int>
## 1 0                    21.5          15.4  1003
## 2 1                    24.2          17.8  2272
```

```r
proximity_summary <- df %>%
  group_by(has_proximity) %>%
  summarise(
    Mean_Length = mean(Match.Length, na.rm = TRUE),
    Median_Length = median(Match.Length, na.rm = TRUE),
    Count = n()
  )
proximity_summary
```

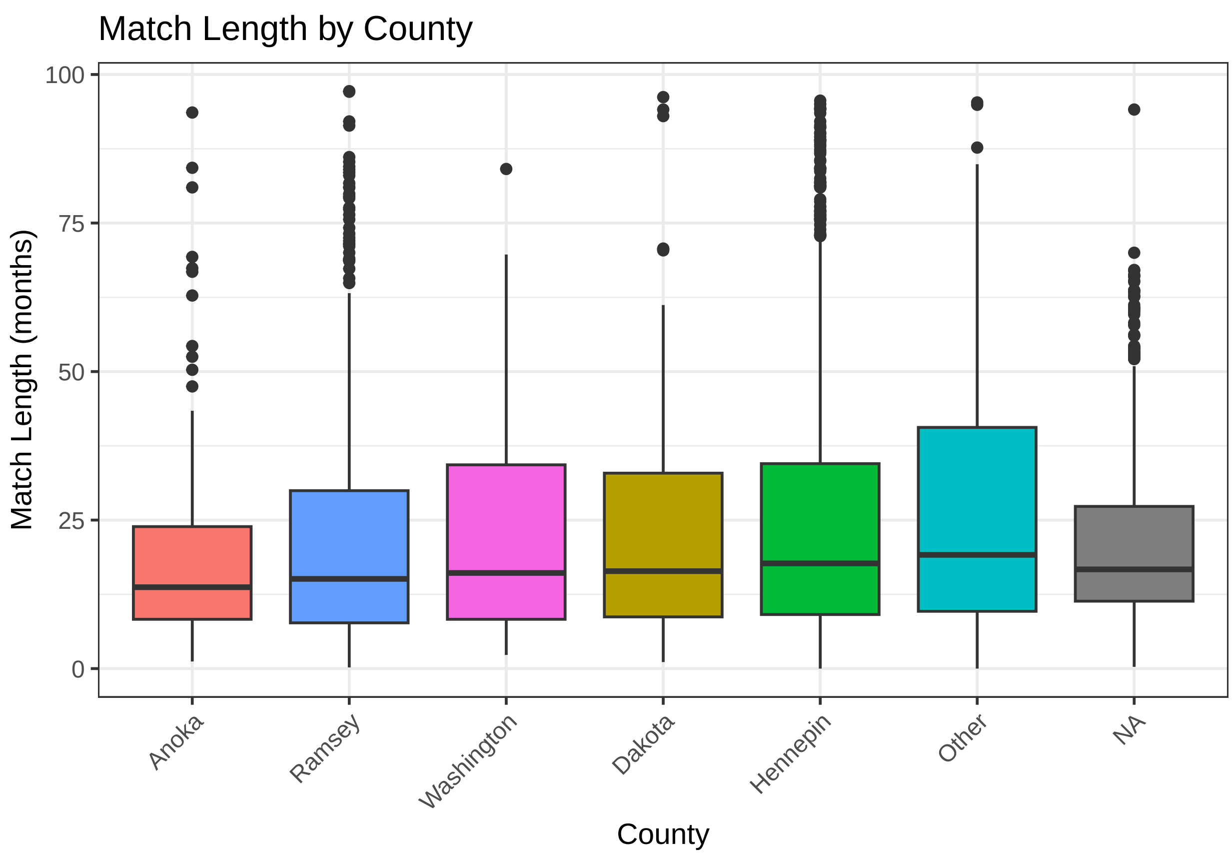
```
## # A tibble: 2 x 4
##   has_proximity Mean_Length Median_Length Count
##   <fct>               <dbl>         <dbl> <int>
## 1 0                    22.4          15.6  1929
## 2 1                    24.9          19.1  1346
```

Statistically discernible difference for close distance - longer for closer.

```r
# County analysis
county_summary <- df %>%
  group_by(County_Factor) %>%
  summarise(
    Mean_Length = mean(Match.Length, na.rm = TRUE),
    Median_Length = median(Match.Length, na.rm = TRUE),
    Count = n()
  ) %>%
  arrange(desc(Mean_Length))
county_summary
```

```
## # A tibble: 7 x 4
##   County_Factor Mean_Length Median_Length Count
##   <fct>               <dbl>         <dbl> <int>
## 1 Other                26.4          19.2   152
## 2 Hennepin             24.7          17.7  1485
## 3 Washington           24.0          16.1    95
## 4 Dakota               23.0          16.4   157
## 5 Ramsey               22.1          15.1   592
## 6 <NA>                 21.7          16.7   655
## 7 Anoka                19.4          13.7   139
```

```r
# Boxplot for counties
ggplot(df, aes(x = reorder(County_Factor, Match.Length, FUN = median),
               y = Match.Length, fill = County_Factor)) +
  geom_boxplot() +
  labs(title = "Match Length by County",
       x = "County",
       y = "Match Length (months)") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```

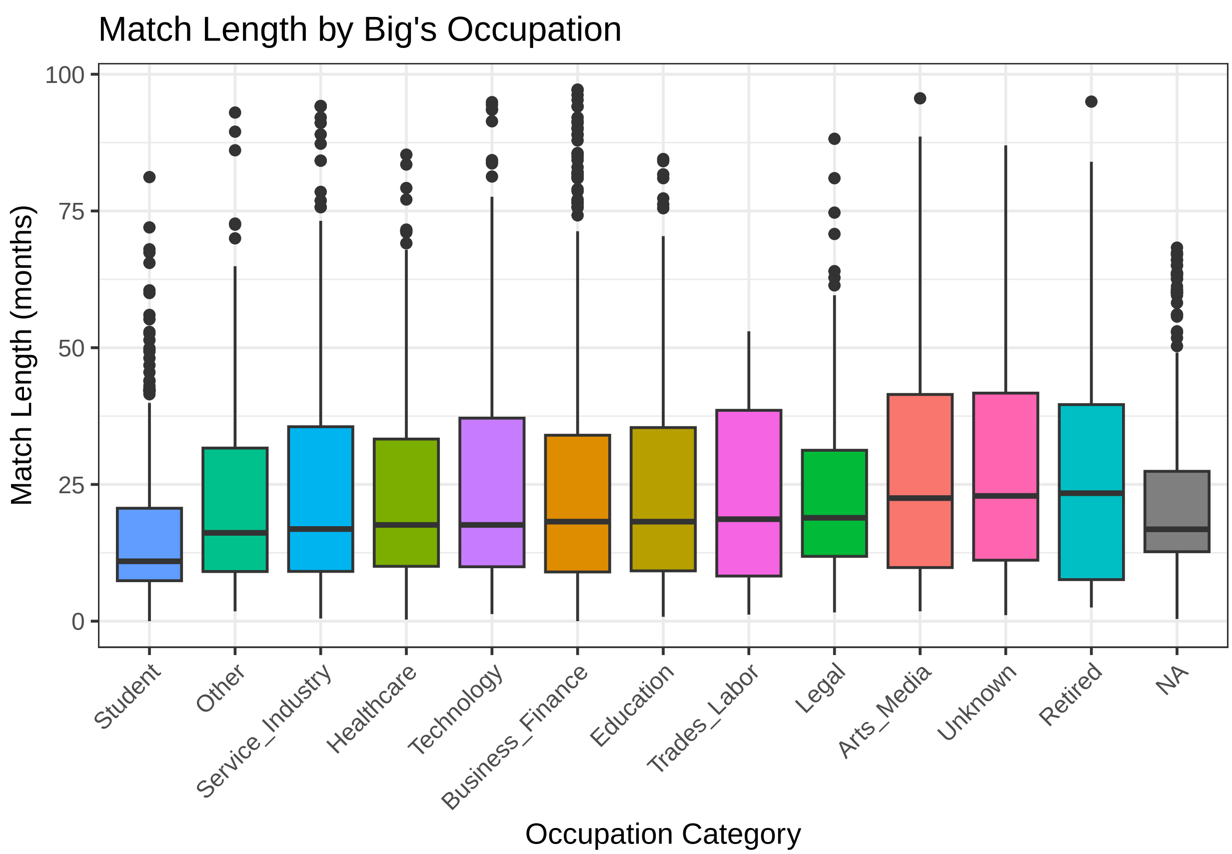



```
# Occupation analysis
occupation_summary <- df %>%
  group_by(Occupation_Category) %>%
  summarise(
    Mean_Length = mean(Match.Length, na.rm = TRUE),
    Median_Length = median(Match.Length, na.rm = TRUE),
    Count = n()
  ) %>%
  arrange(desc(Mean_Length))
occupation_summary
```

```
## # A tibble: 13 x 4
##    Occupation_Category Mean_Length Median_Length Count
##    <fct>                     <dbl>         <dbl> <int>
##  1 Retired                    30.8          23.4    29
##  2 Arts_Media                 29.2          22.5   103
##  3 Unknown                    29.0          22.9   191
##  4 Technology                 25.6          17.6   234
##  5 Business_Finance           24.8          18.2   777
##  6 Education                  24.7          18.2   169
##  7 Legal                      24.5          18.9   115
##  8 Service_Industry           24.3          16.8   358
##  9 Healthcare                 23.7          17.6   278
## 10 Other                      23.0          16.2   160
## 11 Trades_Labor               22.7          18.6    36
## 12 <NA>                       22.4          16.8   325
```

```
## 13 Student                    15.7          11.0   500
```

```r
ggplot(df, aes(x = reorder(Occupation_Category, Match.Length, FUN = median),
               y = Match.Length, fill = Occupation_Category)) +
  geom_boxplot() +
  labs(title = "Match Length by Big's Occupation",
       x = "Occupation Category",
       y = "Match Length (months)") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```



Students on average have shorter matches.

```r
df_marital <- df %>% filter(!is.na(Big.Contact..Marital.Status))
marital_summary <- df_marital %>%
  group_by(Big.Contact..Marital.Status) %>%
  summarise(
    Mean_Length = mean(Match.Length, na.rm = TRUE),
    Median_Length = median(Match.Length, na.rm = TRUE),
    Count = n()
  )
marital_summary
```

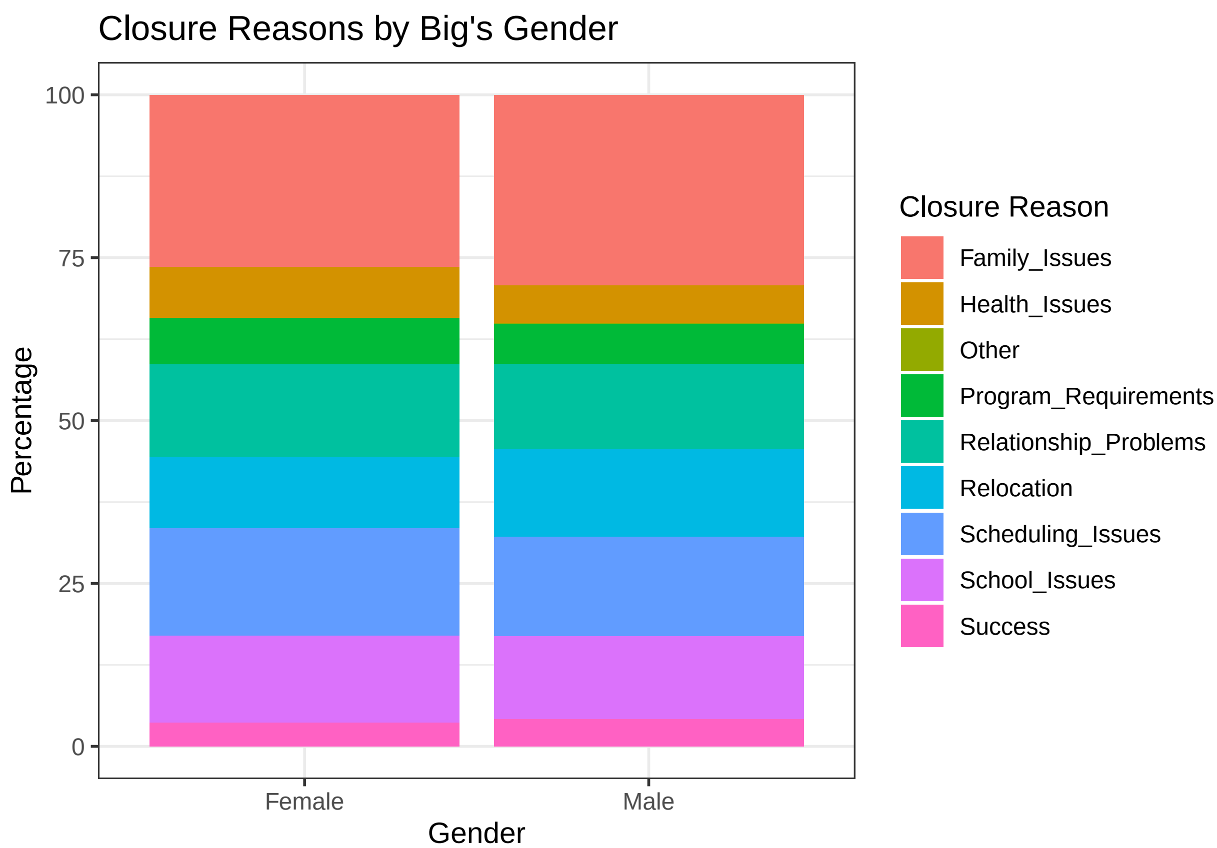
```
## # A tibble: 2 x 4
##   Big.Contact..Marital.Status Mean_Length Median_Length Count
```

```
##   <fct>                              <dbl>         <dbl> <int>
## 1 Single                              18.1          14     651
## 2 Not Single                          19.3          15.6   664
```

Of those which are available, not single bigs have longer match lengths - more stability?

Analysis of closure reasons by demographics

```
age_closure <- df %>%
  filter(!is.na(Age_Group), !is.na(Closure_Reason_Category)) %>%
  group_by(Age_Group, Closure_Reason_Category) %>%
  summarise(Count = n(), .groups = "drop") %>%
  group_by(Age_Group) %>%
  mutate(Percentage = Count / sum(Count) * 100)

gender_closure <- df %>%
  filter(!is.na(Big.Gender), !is.na(Closure_Reason_Category)) %>%
  group_by(Big.Gender, Closure_Reason_Category) %>%
  summarise(Count = n(), .groups = "drop") %>%
  group_by(Big.Gender) %>%
  mutate(Percentage = Count / sum(Count) * 100)

# Visualize closure reasons by gender
ggplot(gender_closure, aes(x = Big.Gender, y = Percentage, fill = Closure_Reason_Category)) +
  geom_bar(stat = "identity", position = "stack") +
  labs(title = "Closure Reasons by Big's Gender",
       x = "Gender",
       y = "Percentage",
       fill = "Closure Reason") +
  theme_bw() +
  theme(legend.position = "right")
```



Roughly equal - more family issues for males? more health issues for females?

```r
# Statistical tests for demographic effects on match length
# ANOVA for County effect
county_anova <- aov(Match.Length ~ County_Factor, data = df)
summary(county_anova)
```

```
##                 Df  Sum Sq Mean Sq F value  Pr(>F)
## County_Factor    5    6919  1383.8    3.27 0.00601 **
## Residuals     2614 1106111   423.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 655 observations deleted due to missingness
```

```r
# ANOVA for Occupation effect
occupation_anova <- aov(Match.Length ~ Occupation_Category, data = df)
summary(occupation_anova)
```

```
##                       Df  Sum Sq Mean Sq F value Pr(>F)
## Occupation_Category   11   44119    4011   10.28 <2e-16 ***
## Residuals           2938 1145926     390
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 325 observations deleted due to missingness
```

```r
# ANOVA for Age Group effect
age_anova <- aov(Match.Length ~ Age_Group, data = df)
summary(age_anova)
```

```
##               Df  Sum Sq Mean Sq F value Pr(>F)
## Age_Group      5   56802   11360   30.58 <2e-16 ***
## Residuals   3269 1214353     371
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Simple regression model
model <- lm(Match.Length ~ Big.Age + Big.Gender + Program.Type + Ethnicity_Match +
             has_interests + has_proximity + County_Factor + Occupation_Category,
           data = df %>% filter(!is.na(Ethnicity_Match)))
summary(model)
```

```
##
## Call:
## lm(formula = Match.Length ~ Big.Age + Big.Gender + Program.Type +
##     Ethnicity_Match + has_interests + has_proximity + County_Factor +
##     Occupation_Category, data = df %>% filter(!is.na(Ethnicity_Match)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.880 -13.920  -4.624   9.533  70.135
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         22.18579    3.27152   6.782 1.47e-11 ***
## Big.Age                              0.21479    0.04274   5.025 5.39e-07 ***
## Big.GenderMale                       2.31206    0.81238   2.846 0.004462 **
## Program.TypeSite                   -15.62586    1.47546 -10.590  < 2e-16 ***
## Program.TypeSite Based Facilitated -13.75678    1.68240  -8.177 4.55e-16 ***
## Ethnicity_MatchTRUE                  2.16703    0.85823   2.525 0.011630 *
## has_interests1                      -4.24546    1.01872  -4.167 3.18e-05 ***
## has_proximity1                      -1.88121    0.93953  -2.002 0.045361 *
## County_FactorDakota                  1.21252    2.35242   0.515 0.606294
## County_FactorHennepin                4.87486    1.80528   2.700 0.006973 **
## County_FactorOther                   8.68833    2.34243   3.709 0.000212 ***
## County_FactorRamsey                  4.30132    1.92835   2.231 0.025797 *
## County_FactorWashington              2.94781    2.68533   1.098 0.272420
## Occupation_CategoryBusiness_Finance -4.42108    2.15762  -2.049 0.040559 *
## Occupation_CategoryEducation        -4.22587    2.56528  -1.647 0.099613 .
## Occupation_CategoryHealthcare       -5.93931    2.42439  -2.450 0.014360 *
## Occupation_CategoryLegal            -3.18413    2.89462  -1.100 0.271430
## Occupation_CategoryOther            -6.40089    2.61407  -2.449 0.014407 *
## Occupation_CategoryRetired          -7.10637    4.34019  -1.637 0.101683
## Occupation_CategoryService_Industry -5.97413    2.30503  -2.592 0.009603 **
## Occupation_CategoryStudent          -4.21006    2.45583  -1.714 0.086594 .
## Occupation_CategoryTechnology       -5.97554    2.46760  -2.422 0.015522 *
## Occupation_CategoryTrades_Labor     -9.08501    4.13433  -2.197 0.028078 *
## Occupation_CategoryUnknown          -0.73212    2.47418  -0.296 0.767329
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.57 on 2534 degrees of freedom
##   (717 observations deleted due to missingness)
## Multiple R-squared:  0.111,  Adjusted R-squared:  0.103
## F-statistic: 13.76 on 23 and 2534 DF,  p-value: < 2.2e-16
```

Very poor predictive performance. But many important predictors.

Looking at interests

```r
df_with_indicators <- df
summary_indicators <- df_with_indicators %>%
  summarise(across(c(has_interests, personality_compatibility, has_proximity,
                    has_commitment, has_experience, has_preference,
                    has_challenges, has_goals),
                  ~sum(as.integer(as.character(.)) == 1, na.rm = TRUE)))
summary_indicators
```

```
##   has_interests personality_compatibility has_proximity has_commitment
## 1          2272                      2332          1346             67
##   has_experience has_preference has_challenges has_goals
## 1            267            460            167      1001
```

```r
# Calculate correlation with match length
indicator_correlations <- df_with_indicators %>%
  select(Match.Length, has_interests, personality_compatibility, has_proximity,
         has_commitment, has_experience, has_preference,
         has_challenges, has_goals) %>%
  mutate(across(has_interests:has_goals, ~as.numeric(as.character(.)))) %>%
  cor(use = "pairwise.complete.obs")

print(indicator_correlations["Match.Length", ])
```

```
##              Match.Length             has_interests personality_compatibility
##               1.000000000               0.064779285               0.048492680
##             has_proximity            has_commitment            has_experience
##               0.062594137               0.040693589               0.005982212
##            has_preference            has_challenges                 has_goals
##               0.058519911               0.041053917               0.052933802
```

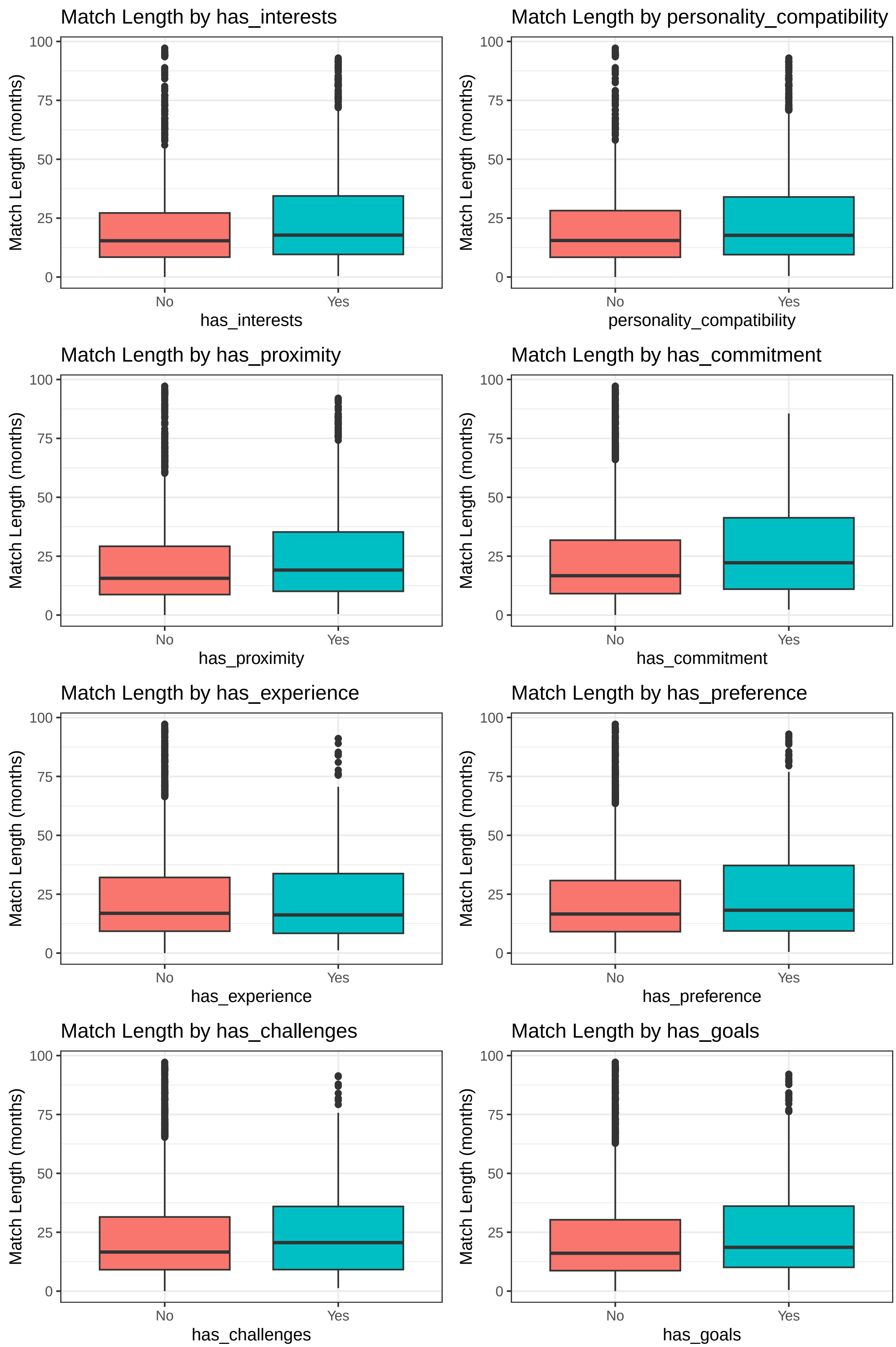
```r
# Visualize the distribution of match length by each indicator
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
# Function to create box plots
create_boxplot <- function(df, var) {
  ggplot(df, aes_string(x = var, y = "Match.Length", fill = var)) +
    geom_boxplot() +
    labs(title = paste("Match Length by", var),
         x = var,
         y = "Match Length (months)") +
    theme_bw() +
    theme(legend.position = "none") +
    scale_x_discrete(labels = c("0" = "No", "1" = "Yes"))
}

# Create a list of plots
plot_list <- lapply(c("has_interests", "personality_compatibility", "has_proximity",
                      "has_commitment", "has_experience", "has_preference",
                      "has_challenges", "has_goals"),
                    function(var) create_boxplot(df_with_indicators, var))
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
# Arrange plots in a grid
grid.arrange(grobs = plot_list, ncol = 2)
```

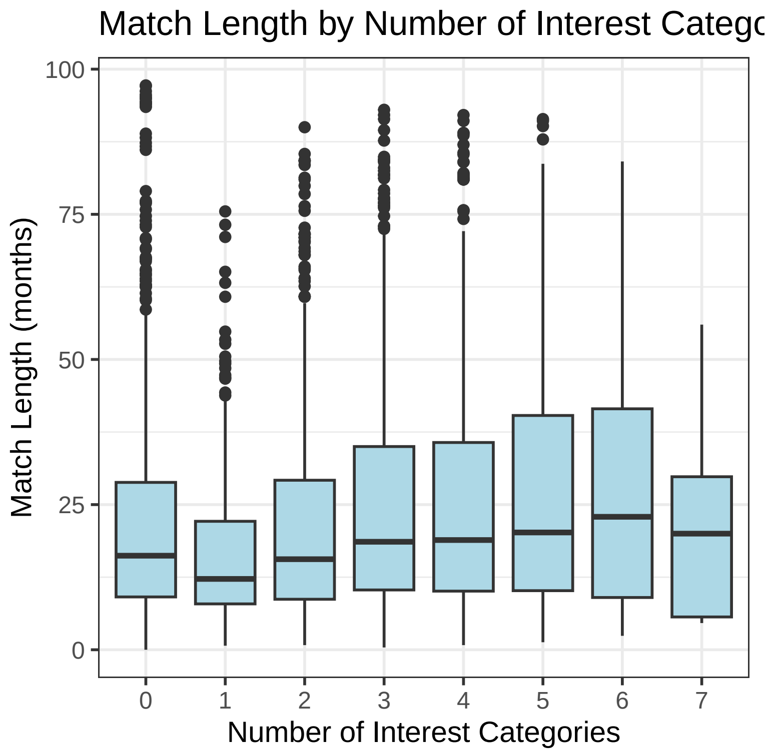

Match Length by has_interests
Match Length (months)
100
75
50
25
0
No
Yes
has_interests
Match Length by personality_compatibility
Match Length (months)
personality_compatibility
Match Length by has_proximity
Match Length (months)
has_proximity
Match Length by has_commitment
Match Length (months)
has_commitment
Match Length by has_experience
Match Length (months)
has_experience
Match Length by has_preference
Match Length (months)
has_preference
Match Length by has_challenges
Match Length (months)
has_challenges
Match Length by has_goals
Match Length (months)
has_goals

has_experience not statistically discernible and present in data much.

```r
# Analyze the impact of different interest combinations
df_with_indicators$interest_count <- rowSums(sapply(df_with_indicators[, c("has_interests",
                                                                           "personality_compatibility",
                                                                           "has_proximity",
                                                                           "has_commitment",
                                                                           "has_experience",
                                                                           "has_preference",
                                                                           "has_challenges",
                                                                           "has_goals")],
                                                  function(x) as.integer(as.character(x))))

# Analyze relationship between number of interest categories and match length
interest_count_summary <- df_with_indicators %>%
  group_by(interest_count) %>%
  summarise(
    Mean_Length = mean(Match.Length, na.rm = TRUE),
    Median_Length = median(Match.Length, na.rm = TRUE),
    Count = n()
  )

print(interest_count_summary)
```

```
## # A tibble: 8 x 4
##   interest_count Mean_Length Median_Length Count
##            <dbl>       <dbl>         <dbl> <int>
## 1              0        23.2          16.2   616
## 2              1        16.5          12.2   332
## 3              2        21.6          15.6   587
## 4              3        25.0          18.6   821
## 5              4        25.0          18.9   703
## 6              5        27.8          20.2   172
## 7              6        28.9          22.9    37
## 8              7        21.6          20         7
```

```r
# Visualize relationship between interest count and match length
ggplot(df_with_indicators, aes(x = factor(interest_count), y = Match.Length)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Match Length by Number of Interest Categories Mentioned",
       x = "Number of Interest Categories",
       y = "Match Length (months)") +
  theme_bw()
```




```r
# Test statistical significance
interest_model <- lm(Match.Length ~ has_interests + personality_compatibility + has_proximity +
                       has_commitment + has_experience + has_preference +
                       has_challenges + has_goals, data = df_with_indicators)
summary(interest_model)
```

```
##
## Call:
## lm(formula = Match.Length ~ has_interests + personality_compatibility +
##     has_proximity + has_commitment + has_experience + has_preference +
##     has_challenges + has_goals, data = df_with_indicators)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.594 -13.787  -6.180   8.527  76.120
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 21.0796     0.6903  30.537  < 2e-16 ***
## has_interests1               1.4299     1.0058   1.422  0.15522
## personality_compatibility1  -0.2839     0.9997  -0.284  0.77643
## has_proximity1               1.3575     0.8153   1.665  0.09601 .
## has_commitment1              5.0878     2.4266   2.097  0.03610 *
## has_experience1             -0.4376     1.2980  -0.337  0.73603
## has_preference1              2.7382     0.9985   2.742  0.00613 **
## has_challenges1              2.8724     1.6242   1.768  0.07708 .
## has_goals1                   1.1681     0.8177   1.428  0.15325
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 19.62 on 3266 degrees of freedom
## Multiple R-squared:  0.01101,    Adjusted R-squared:  0.00859
## F-statistic: 4.546 on 8 and 3266 DF,  p-value: 1.614e-05
```

Successful match?

```r
# A match is successful if:
# 1. It's active
# 2. It has a long duration
# 3. Closure reason is "Success"

df$successful_match <- FALSE

# Active matches (no closure date)
df$successful_match[df$Stage == 0] <- TRUE
sum(df$successful_match[df$Stage == 0])
```

```
## [1] 789
```

```r
# Matches with "Success" as closure reason
df$successful_match[df$Closure_Reason_Category == "Success"] <- TRUE

# Long duration matches (define long as above the 80th percentile)
long_duration_threshold <- quantile(df$Match.Length, 0.8, na.rm = TRUE)
df$successful_match[df$Match.Length > long_duration_threshold &
                      !is.na(df$Match.Length)] <- TRUE

# Success rate
success_rate <- mean(df$successful_match, na.rm = TRUE)
print(paste("Overall success rate:", round(success_rate * 100, 1), "%"))
```

```
## [1] "Overall success rate: 36.8 %"
```

A third of the matches are successful (maybe slightly biased because many matches just started).

analyzing factors associated with successful matches

```r
age_group_success <- aggregate(successful_match ~ Age_Group, data = df, FUN = mean)
age_group_success$count <- aggregate(successful_match ~ Age_Group, data = df, FUN = length)$successful_r
age_group_success <- age_group_success[order(-age_group_success$successful_match),]
print("Success rate by mentor age group:")
```

```
## [1] "Success rate by mentor age group:"
```

```r
print(age_group_success)
```

```
##   Age_Group successful_match count
## 5     56-65        0.5303030   198
## 6       65+        0.4210526    57
```

```
## 3     36-45        0.4154589   621
## 4     46-55        0.3558052   267
## 2     26-35        0.3551847  1678
## 1     18-25        0.2819383   454
```

```r
gender_success <- aggregate(successful_match ~ Big.Gender, data = df, FUN = mean)
gender_success$count <- aggregate(successful_match ~ Big.Gender, data = df, FUN = length)$successful_ma
print("Success rate by mentor gender:")
```

```
## [1] "Success rate by mentor gender:"
```

```r
print(gender_success)
```

```
##   Big.Gender successful_match count
## 1     Female        0.3468031  1955
## 2       Male        0.3996937  1306
```

```r
program_type_success <- aggregate(successful_match ~ Program.Type, data = df, FUN = mean)
program_type_success$count <- aggregate(successful_match ~ Program.Type, data = df, FUN = length)$succes
print("Success rate by program type:")
```

```
## [1] "Success rate by program type:"
```

```r
print(program_type_success)
```

```
##             Program.Type successful_match count
## 1              Community        0.4491736  2420
## 2                   Site        0.1210526   570
## 3 Site Based Facilitated        0.1702128   282
```

```r
ethnicity_match_success <- aggregate(successful_match ~ Ethnicity_Match, data = df, FUN = mean)
ethnicity_match_success$count <- aggregate(successful_match ~ Ethnicity_Match, data = df, FUN = length)$
print("Success rate by ethnicity match:")
```

```
## [1] "Success rate by ethnicity match:"
```

```r
print(ethnicity_match_success)
```

```
##   Ethnicity_Match successful_match count
## 1           FALSE        0.3608958  2322
## 2            TRUE        0.3861490   953
```

```r
occupation_success <- aggregate(successful_match ~ Occupation_Category, data = df, FUN = mean)
occupation_success$count <- aggregate(successful_match ~ Occupation_Category, data = df, FUN = length)$
occupation_success <- occupation_success[order(-occupation_success$successful_match),]
print("Success rate by occupation category:")
```

```
## [1] "Success rate by occupation category:"
```

```r
print(occupation_success)
```

```
##    Occupation_Category successful_match count
## 1           Arts_Media        0.5048544   103
## 7              Retired        0.4827586    29
## 10          Technology        0.4658120   234
## 6                Other        0.4625000   160
## 2     Business_Finance        0.4555985   777
## 11         Trades_Labor        0.4444444    36
## 8     Service_Industry        0.4329609   358
## 3            Education        0.4023669   169
## 4           Healthcare        0.3812950   278
## 12             Unknown        0.3507853   191
## 5                Legal        0.3304348   115
## 9              Student        0.1760000   500
```

```r
county_success <- aggregate(successful_match ~ County_Factor, data = df, FUN = mean)
county_success$count <- aggregate(successful_match ~ County_Factor, data = df, FUN = length)$successful_
county_success <- county_success[order(-county_success$successful_match),]
print("Success rate by county:")
```

```
## [1] "Success rate by county:"
```

```r
print(county_success)
```

```
##   County_Factor successful_match count
## 6    Washington        0.4526316    95
## 2        Dakota        0.4522293   157
## 4         Other        0.4473684   152
## 3      Hennepin        0.4060606  1485
## 1         Anoka        0.3669065   139
## 5        Ramsey        0.3226351   592
```

```r
# Create a function to check success rate for binary factors
check_binary_factor <- function(factor_name) {
  formula <- as.formula(paste("successful_match ~", factor_name))
  success_rate <- aggregate(formula, data = df, FUN = mean)
  success_rate$count <- aggregate(formula, data = df, FUN = length)$successful_match
  print(paste("Success rate by", factor_name, ":"))
  print(success_rate)
}

compatibility_factors <- c("has_interests", "personality_compatibility", "has_proximity",
                           "has_commitment", "has_experience", "has_preference",
                           "has_challenges", "has_goals")

for (factor in compatibility_factors) {
  check_binary_factor(factor)
}
```

```
## [1] "Success rate by has_interests :"
```

```
##   has_interests successful_match count
## 1             0        0.1874377  1003
## 2             1        0.4480634  2272
## [1] "Success rate by personality_compatibility :"
##   personality_compatibility successful_match count
## 1                         0        0.2205726   943
## 2                         1        0.4279588  2332
## [1] "Success rate by has_proximity :"
##   has_proximity successful_match count
## 1             0        0.3240021  1929
## 2             1        0.4316493  1346
## [1] "Success rate by has_commitment :"
##   has_commitment successful_match count
## 1              0        0.3678304  3208
## 2              1        0.3880597    67
## [1] "Success rate by has_experience :"
##   has_experience successful_match count
## 1              0        0.3713431  3008
## 2              1        0.3333333   267
## [1] "Success rate by has_preference :"
##   has_preference successful_match count
## 1              0        0.3687389  2815
## 2              1        0.3652174   460
## [1] "Success rate by has_challenges :"
##   has_challenges successful_match count
## 1              0        0.3658301  3108
## 2              1        0.4131737   167
## [1] "Success rate by has_goals :"
##   has_goals successful_match count
## 1         0        0.3192612  2274
## 2         1        0.4795205  1001
```

```r
# Logistic regression to identify key predictors of success
df$successful_match_numeric <- as.numeric(df$successful_match)

model <- glm(successful_match_numeric ~ Big.Age + Big.Gender + Program.Type +
              Ethnicity_Match + County_Factor + Occupation_Category + Age_Group +
              has_interests + personality_compatibility + has_proximity +
              has_commitment + has_experience + has_preference + has_challenges + has_goals,
            family = binomial(link = "logit"), data = df)

summary_model <- summary(model)
print("Logistic regression results (key predictors of successful matches):")
```

```
## [1] "Logistic regression results (key predictors of successful matches):"
```

```r
print(summary_model)
```

```
##
## Call:
## glm(formula = successful_match_numeric ~ Big.Age + Big.Gender +
##     Program.Type + Ethnicity_Match + County_Factor + Occupation_Category +
##     Age_Group + has_interests + personality_compatibility + has_proximity +
```

```
##     has_commitment + has_experience + has_preference + has_challenges +
##     has_goals, family = binomial(link = "logit"), data = df)
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                            0.67667    0.56879   1.190 0.234180
## Big.Age                               -0.01087    0.01699  -0.640 0.522450
## Big.GenderMale                         0.13734    0.09306   1.476 0.139996
## Program.TypeSite                      -1.69806    0.20777  -8.173 3.02e-16 ***
## Program.TypeSite Based Facilitated    -1.10760    0.21263  -5.209 1.90e-07 ***
## Ethnicity_MatchTRUE                    0.07325    0.09801   0.747 0.454863
## County_FactorDakota                   -0.19314    0.26451  -0.730 0.465273
## County_FactorHennepin                 -0.07488    0.20888  -0.358 0.719991
## County_FactorOther                     0.52310    0.27392   1.910 0.056173 .
## County_FactorRamsey                   -0.19670    0.22423  -0.877 0.380382
## County_FactorWashington                0.00562    0.30363   0.019 0.985232
## Occupation_CategoryBusiness_Finance -0.10871    0.23083  -0.471 0.637677
## Occupation_CategoryEducation          -0.43354    0.27689  -1.566 0.117400
## Occupation_CategoryHealthcare         -0.54122    0.25938  -2.087 0.036926 *
## Occupation_CategoryLegal              -0.34035    0.32465  -1.048 0.294483
## Occupation_CategoryOther              -0.17175    0.28176  -0.610 0.542165
## Occupation_CategoryRetired            -0.75857    0.49583  -1.530 0.126041
## Occupation_CategoryService_Industry -0.33769    0.24663  -1.369 0.170944
## Occupation_CategoryStudent            -0.96035    0.29106  -3.299 0.000969 ***
## Occupation_CategoryTechnology         -0.22809    0.26459  -0.862 0.388647
## Occupation_CategoryTrades_Labor       -0.43299    0.44613  -0.971 0.331776
## Occupation_CategoryUnknown            -0.61054    0.26966  -2.264 0.023567 *
## Age_Group26-35                        -0.79274    0.22273  -3.559 0.000372 ***
## Age_Group36-45                        -0.53516    0.32862  -1.629 0.103414
## Age_Group46-55                        -0.53302    0.50011  -1.066 0.286515
## Age_Group56-65                         0.14377    0.64518   0.223 0.823660
## Age_Group65+                           0.36789    0.86174   0.427 0.669445
## has_interests1                         0.69799    0.13793   5.061 4.18e-07 ***
## personality_compatibility1             0.04914    0.14030   0.350 0.726135
## has_proximity1                        -0.26331    0.10265  -2.565 0.010316 *
## has_commitment1                       -0.10960    0.30899  -0.355 0.722809
## has_experience1                        0.06184    0.17791   0.348 0.728130
## has_preference1                       -0.22296    0.12736  -1.751 0.080004 .
## has_challenges1                       -0.19366    0.19433  -0.997 0.318996
## has_goals1                             0.35596    0.10105   3.522 0.000428 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3422.5  on 2557  degrees of freedom
## Residual deviance: 2990.2  on 2523  degrees of freedom
##   (717 observations deleted due to missingness)
## AIC: 3060.2
##
## Number of Fisher Scoring iterations: 4
```

```
significant_predictors <- summary_model$coefficients[summary_model$coefficients[,4] < 0.05,]
print("Significant predictors of match success:")
```

```
## [1] "Significant predictors of match success:"
```

```r
print(significant_predictors)
```

```
##                                      Estimate Std. Error   z value     Pr(>|z|)
## Program.TypeSite                   -1.6980576  0.2077714 -8.172720 3.015124e-16
## Program.TypeSite Based Facilitated -1.1076024  0.2126302 -5.209055 1.898049e-07
## Occupation_CategoryHealthcare      -0.5412176  0.2593806 -2.086577 3.692639e-02
## Occupation_CategoryStudent         -0.9603478  0.2910634 -3.299445 9.687612e-04
## Occupation_CategoryUnknown         -0.6105386  0.2696589 -2.264114 2.356709e-02
## Age_Group26-35                     -0.7927387  0.2227270 -3.559239 3.719303e-04
## has_interests1                      0.6979888  0.1379274  5.060552 4.180455e-07
## has_proximity1                     -0.2633115  0.1026538 -2.565043 1.031629e-02
## has_goals1                          0.3559563  0.1010526  3.522485 4.275208e-04
```

```r
print("Key insights and recommendations for Big Brothers Big Sisters Twin Cities:")
```

```
## [1] "Key insights and recommendations for Big Brothers Big Sisters Twin Cities:"
```

```r
print("1. Most important factors for successful matches:")
```

```
## [1] "1. Most important factors for successful matches:"
```

Survival analysis on successful matches

```r
library(survival)
cox_model <- coxph(Surv(Match.Length, successful_match_numeric) ~ Big.Gender + Big.Age + Program.Type +
summary(cox_model)
```

```
## Call:
## coxph(formula = Surv(Match.Length, successful_match_numeric) ~
##     Big.Gender + Big.Age + Program.Type + Occupation_Category +
##         has_interests + has_proximity + has_goals + Ethnicity_Match,
##     data = df)
##
##   n= 2933, number of events= 1133
##    (342 observations deleted due to missingness)
##
##                                         coef exp(coef)  se(coef)      z
## Big.GenderMale                     -0.104354  0.900907  0.062718 -1.664
## Big.Age                            -0.013861  0.986235  0.003317 -4.179
## Program.TypeSite                    0.448508  1.565973  0.161246  2.782
## Program.TypeSite Based Facilitated  0.968432  2.633812  0.162894  5.945
## Occupation_CategoryBusiness_Finance  0.151272  1.163313  0.149220  1.014
## Occupation_CategoryEducation        0.082318  1.085801  0.185952  0.443
## Occupation_CategoryHealthcare       0.091506  1.095824  0.170451  0.537
## Occupation_CategoryLegal            0.182979  1.200789  0.217569  0.841
## Occupation_CategoryOther            0.469176  1.598676  0.182331  2.573
## Occupation_CategoryRetired          0.175258  1.191553  0.316390  0.554
```

```
## Occupation_CategoryService_Industry  0.202537  1.224505  0.161001  1.258
## Occupation_CategoryStudent           0.057110  1.058773  0.191770  0.298
## Occupation_CategoryTechnology        0.137113  1.146958  0.170378  0.805
## Occupation_CategoryTrades_Labor      0.598988  1.820276  0.290639  2.061
## Occupation_CategoryUnknown          -0.303514  0.738220  0.185867 -1.633
## has_interests1                       0.995285  2.705496  0.097376 10.221
## has_proximity1                       0.039054  1.039827  0.065783  0.594
## has_goals1                           0.138750  1.148837  0.065117  2.131
## Ethnicity_MatchTRUE                 -0.088930  0.914910  0.065825 -1.351
##                                     Pr(>|z|)
## Big.GenderMale                       0.09614 .
## Big.Age                             2.93e-05 ***
## Program.TypeSite                     0.00541 **
## Program.TypeSite Based Facilitated  2.76e-09 ***
## Occupation_CategoryBusiness_Finance  0.31070
## Occupation_CategoryEducation         0.65799
## Occupation_CategoryHealthcare        0.59137
## Occupation_CategoryLegal             0.40034
## Occupation_CategoryOther             0.01008 *
## Occupation_CategoryRetired           0.57963
## Occupation_CategoryService_Industry  0.20840
## Occupation_CategoryStudent           0.76585
## Occupation_CategoryTechnology        0.42096
## Occupation_CategoryTrades_Labor      0.03931 *
## Occupation_CategoryUnknown           0.10248
## has_interests1                       < 2e-16 ***
## has_proximity1                       0.55272
## has_goals1                           0.03311 *
## Ethnicity_MatchTRUE                  0.17670
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                     exp(coef) exp(-coef) lower .95 upper .95
## Big.GenderMale                         0.9009     1.1100    0.7967    1.0187
## Big.Age                                0.9862     1.0140    0.9798    0.9927
## Program.TypeSite                       1.5660     0.6386    1.1416    2.1480
## Program.TypeSite Based Facilitated     2.6338     0.3797    1.9139    3.6244
## Occupation_CategoryBusiness_Finance    1.1633     0.8596    0.8683    1.5585
## Occupation_CategoryEducation           1.0858     0.9210    0.7542    1.5633
## Occupation_CategoryHealthcare          1.0958     0.9126    0.7846    1.5305
## Occupation_CategoryLegal               1.2008     0.8328    0.7839    1.8393
## Occupation_CategoryOther               1.5987     0.6255    1.1183    2.2854
## Occupation_CategoryRetired             1.1916     0.8392    0.6409    2.2153
## Occupation_CategoryService_Industry    1.2245     0.8167    0.8931    1.6788
## Occupation_CategoryStudent             1.0588     0.9445    0.7271    1.5418
## Occupation_CategoryTechnology          1.1470     0.8719    0.8213    1.6017
## Occupation_CategoryTrades_Labor        1.8203     0.5494    1.0298    3.2176
## Occupation_CategoryUnknown             0.7382     1.3546    0.5128    1.0627
## has_interests1                         2.7055     0.3696    2.2354    3.2744
## has_proximity1                         1.0398     0.9617    0.9140    1.1829
## has_goals1                             1.1488     0.8704    1.0112    1.3052
## Ethnicity_MatchTRUE                    0.9149     1.0930    0.8042    1.0409
##
## Concordance= 0.668  (se = 0.009 )
```

```
## Likelihood ratio test= 223  on 19 df,   p=<2e-16
## Wald test            = 195.7  on 19 df,   p=<2e-16
## Score (logrank) test = 202.6  on 19 df,   p=<2e-16
```

Surprisingly very good model. Program type, big age and shared interest seem to be the most telling signs of match length by succesful match.

**Kaplan Meier curve function not working rn but I will try to add that later

Model but on strictly whether the match is still ongoing or not

```r
library(survival)
cox_model <- coxph(Surv(Match.Length, Stage) ~ Big.Gender + Big.Age + Program.Type + Occupation_Category
summary(cox_model)
```

```
## Call:
## coxph(formula = Surv(Match.Length, Stage) ~ Big.Gender + Big.Age +
##     Program.Type + Occupation_Category + has_interests + has_proximity +
##     has_goals + Ethnicity_Match, data = df)
##
##   n= 2933, number of events= 2184
##    (342 observations deleted due to missingness)
##
##                                          coef exp(coef)  se(coef)      z
## Big.GenderMale                      -0.101870  0.903147  0.045205 -2.253
## Big.Age                             -0.008539  0.991498  0.002467 -3.461
## Program.TypeSite                     0.928326  2.530271  0.080138 11.584
## Program.TypeSite Based Facilitated   0.664334  1.943197  0.091173  7.286
## Occupation_CategoryBusiness_Finance  0.106664  1.112561  0.128740  0.829
## Occupation_CategoryEducation         0.227743  1.255762  0.150173  1.517
## Occupation_CategoryHealthcare        0.328225  1.388501  0.139103  2.360
## Occupation_CategoryLegal             0.081515  1.084930  0.163342  0.499
## Occupation_CategoryOther             0.103856  1.109441  0.158008  0.657
## Occupation_CategoryRetired           0.029030  1.029456  0.286448  0.101
## Occupation_CategoryService_Industry  0.263533  1.301520  0.135711  1.942
## Occupation_CategoryStudent           0.255885  1.291604  0.142811  1.792
## Occupation_CategoryTechnology        0.109617  1.115850  0.146321  0.749
## Occupation_CategoryTrades_Labor      0.337969  1.402097  0.235873  1.433
## Occupation_CategoryUnknown           0.218426  1.244117  0.143590  1.521
## has_interests1                      -0.093124  0.911080  0.057134 -1.630
## has_proximity1                       0.124789  1.132910  0.054188  2.303
## has_goals1                          -0.156659  0.854996  0.054848 -2.856
## Ethnicity_MatchTRUE                 -0.032101  0.968409  0.047145 -0.681
##                                     Pr(>|z|)
## Big.GenderMale                      0.024228 *
## Big.Age                             0.000538 ***
## Program.TypeSite                     < 2e-16 ***
## Program.TypeSite Based Facilitated  3.18e-13 ***
## Occupation_CategoryBusiness_Finance 0.407374
## Occupation_CategoryEducation        0.129384
## Occupation_CategoryHealthcare       0.018296 *
## Occupation_CategoryLegal            0.617746
## Occupation_CategoryOther            0.510998
## Occupation_CategoryRetired          0.919276
```

```
## Occupation_CategoryService_Industry 0.052154 .
## Occupation_CategoryStudent          0.073170 .
## Occupation_CategoryTechnology       0.453765
## Occupation_CategoryTrades_Labor     0.151902
## Occupation_CategoryUnknown          0.128216
## has_interests1                      0.103118
## has_proximity1                      0.021284 *
## has_goals1                          0.004287 **
## Ethnicity_MatchTRUE                 0.495941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                     exp(coef) exp(-coef) lower .95 upper .95
## Big.GenderMale                         0.9031     1.1072    0.8266    0.9868
## Big.Age                                0.9915     1.0086    0.9867    0.9963
## Program.TypeSite                       2.5303     0.3952    2.1625    2.9606
## Program.TypeSite Based Facilitated     1.9432     0.5146    1.6252    2.3234
## Occupation_CategoryBusiness_Finance    1.1126     0.8988    0.8645    1.4319
## Occupation_CategoryEducation           1.2558     0.7963    0.9356    1.6855
## Occupation_CategoryHealthcare          1.3885     0.7202    1.0572    1.8237
## Occupation_CategoryLegal               1.0849     0.9217    0.7877    1.4943
## Occupation_CategoryOther               1.1094     0.9014    0.8140    1.5122
## Occupation_CategoryRetired             1.0295     0.9714    0.5872    1.8048
## Occupation_CategoryService_Industry    1.3015     0.7683    0.9975    1.6981
## Occupation_CategoryStudent             1.2916     0.7742    0.9763    1.7088
## Occupation_CategoryTechnology          1.1159     0.8962    0.8376    1.4865
## Occupation_CategoryTrades_Labor        1.4021     0.7132    0.8831    2.2261
## Occupation_CategoryUnknown             1.2441     0.8038    0.9389    1.6485
## has_interests1                         0.9111     1.0976    0.8146    1.0190
## has_proximity1                         1.1329     0.8827    1.0188    1.2599
## has_goals1                             0.8550     1.1696    0.7679    0.9520
## Ethnicity_MatchTRUE                    0.9684     1.0326    0.8829    1.0622
##
## Concordance= 0.613  (se = 0.007 )
## Likelihood ratio test= 420.4  on 19 df,   p=<2e-16
## Wald test            = 467.6  on 19 df,   p=<2e-16
## Score (logrank) test = 504.2  on 19 df,   p=<2e-16
```