STEG SURVEY ANALYSIS

1. Data Cleaning

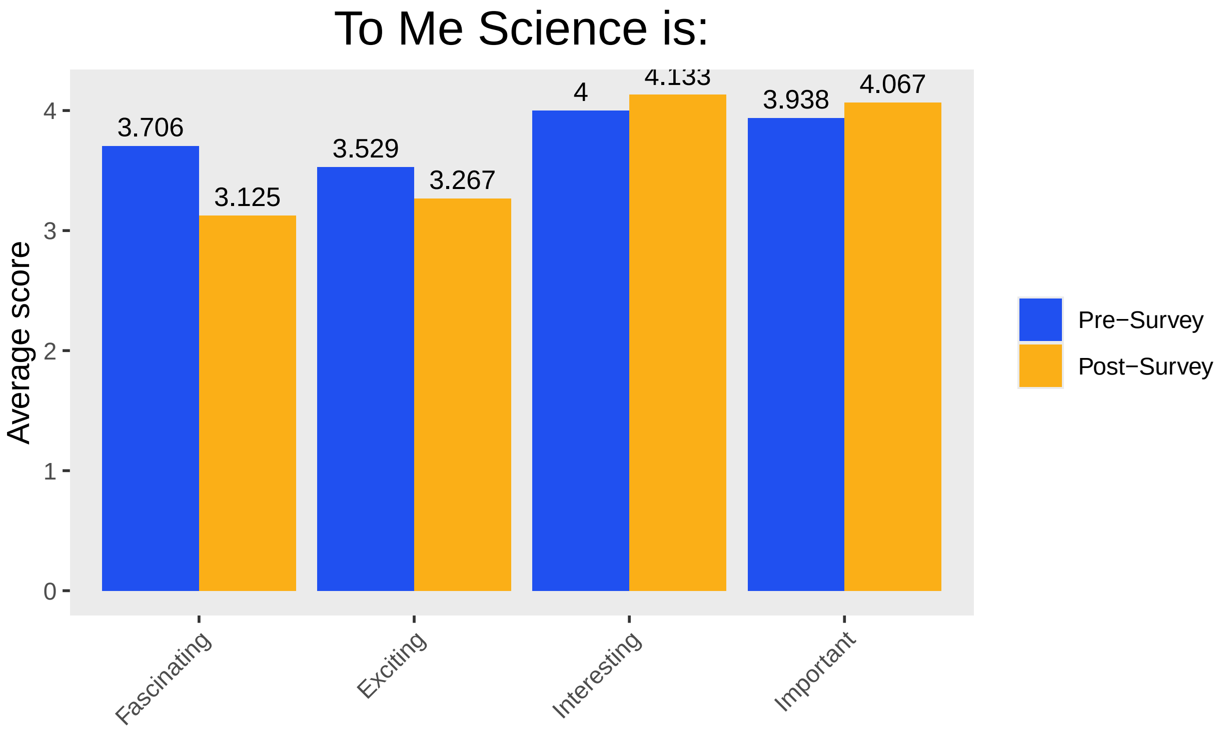
```r
#read in the data

data <- read.csv("Nate_Reverse_Pre&Post_STEG_Survey_Fall 2023.csv")
# cleanup the column names
colnames(data) <- gsub(pattern = ".*(Pre|Post).*", replacement = "PrePost", x = colnames(data))
colnames(data) <- gsub(pattern = ".*(Name).*", replacement = "Name", x = colnames(data))
colnames(data) <- gsub(pattern = "To.me..", replacement = "", x = colnames(data))
colnames(data) <- gsub(pattern = ".is....", replacement = "_", x = colnames(data))
colnames(data) <- gsub(pattern =
                         "a.CAREER.in.science..technology..and.engineering",
                       replacement = "STE", x = colnames(data))
colnames(data) <- gsub(pattern = "a.CAREER.in.geography", replacement = "GEO",
                       x = colnames(data))
# Seperate the data into pre and post
pre_data <- data[data$PrePost == "PRE", ]
post_data <- data[data$PrePost == "POST", ]

# remove the first column
pre_data <- pre_data[, -1]
post_data <- post_data[, -1]

# Corrected code
colnames(pre_data)[-1] <- paste0(colnames(pre_data)[-1], "_PRE")
colnames(post_data)[-1] <- paste0(colnames(post_data)[-1], "_POST")
```

2. Graphs

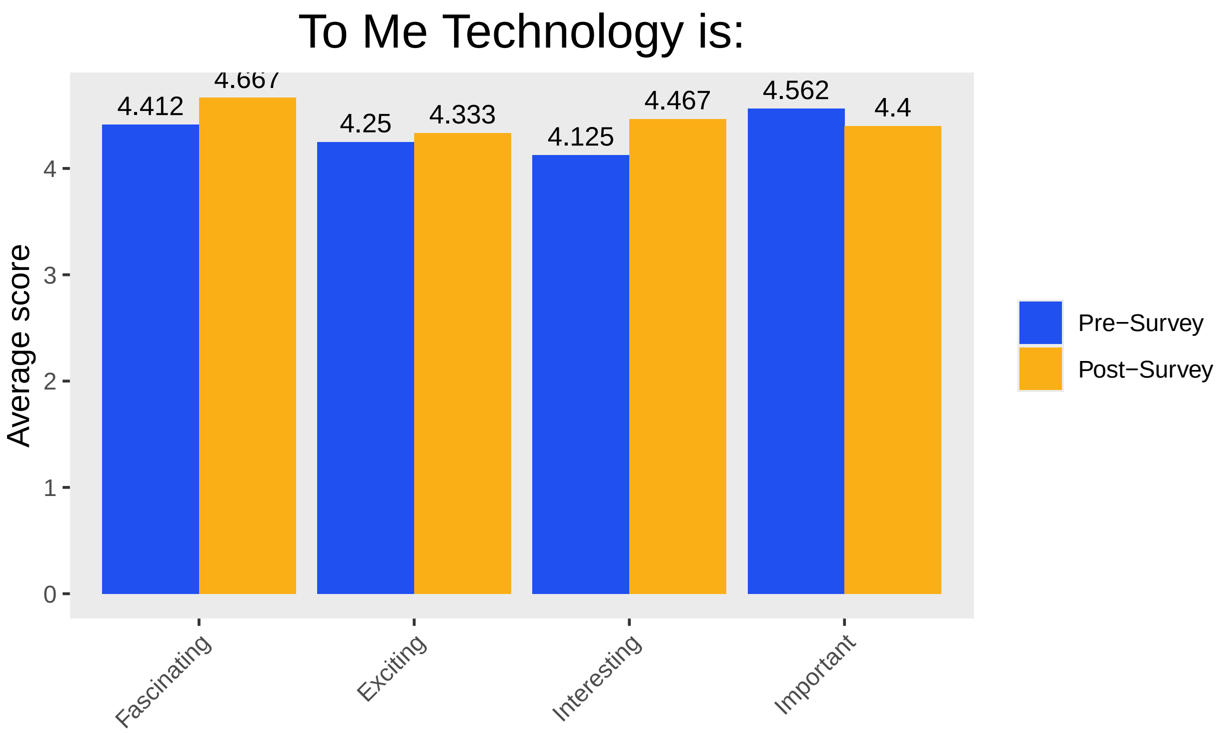
```r
# Get the mean for each column and collect it in a data frame
mean_data <- data.frame(
  colnames(pre_data),
  sapply(pre_data, function(x) mean(as.numeric(x), na.rm = TRUE))
)
# Rename cols
colnames(mean_data) <- c("Question", "Mean_PRE")
mean_data$Mean_POST <- sapply(post_data, function(x) mean(as.numeric(x), na.rm = TRUE))
# Removing the names row
mean_data <- mean_data[-1, ]

# Show means
mean_data[, -1]
```

```
##                              Mean_PRE Mean_POST
## SCIENCE_Fascinating_PRE     3.705882  3.125000
## SCIENCE_Exciting_PRE        3.529412  3.266667
## SCIENCE_Interesting_PRE     4.000000  4.133333
## SCIENCE_Important_PRE       3.937500  4.066667
## TECHNOLOGY_Fascinating_PRE  4.411765  4.666667
## TECHNOLOGY_Exciting_PRE     4.250000  4.333333
## TECHNOLOGY_Interesting_PRE  4.125000  4.466667
## TECHNOLOGY_Important_PRE    4.562500  4.400000
## ENGINEERING_Fascinating_PRE 3.933333  3.076923
## ENGINEERING_Exciting_PRE    3.692308  3.076923
## ENGINEERING_Interesting_PRE 3.692308  3.285714
## ENGINEERING_Important_PRE   3.916667  3.583333
## GEOGRAPHY_Fascinating_PRE   3.461538  3.230769
## GEOGRAPHY_Exciting_PRE      3.214286  2.769231
## GEOGRAPHY_Interesting_PRE   3.153846  3.230769
## GEOGRAPHY_Important_PRE     3.769231  3.307692
## STE_Fascinating_PRE         3.500000  3.928571
## STE_Exciting_PRE            3.733333  3.846154
## STE_Interesting_PRE         3.857143  4.076923
## STE_Important_PRE           3.857143  4.307692
## GEO_Fascinating_PRE         3.625000  3.400000
## GEO_Exciting_PRE            3.352941  3.571429
## GEO_Interesting_PRE         3.687500  3.285714
## GEO_Important_PRE           4.125000  3.571429
```

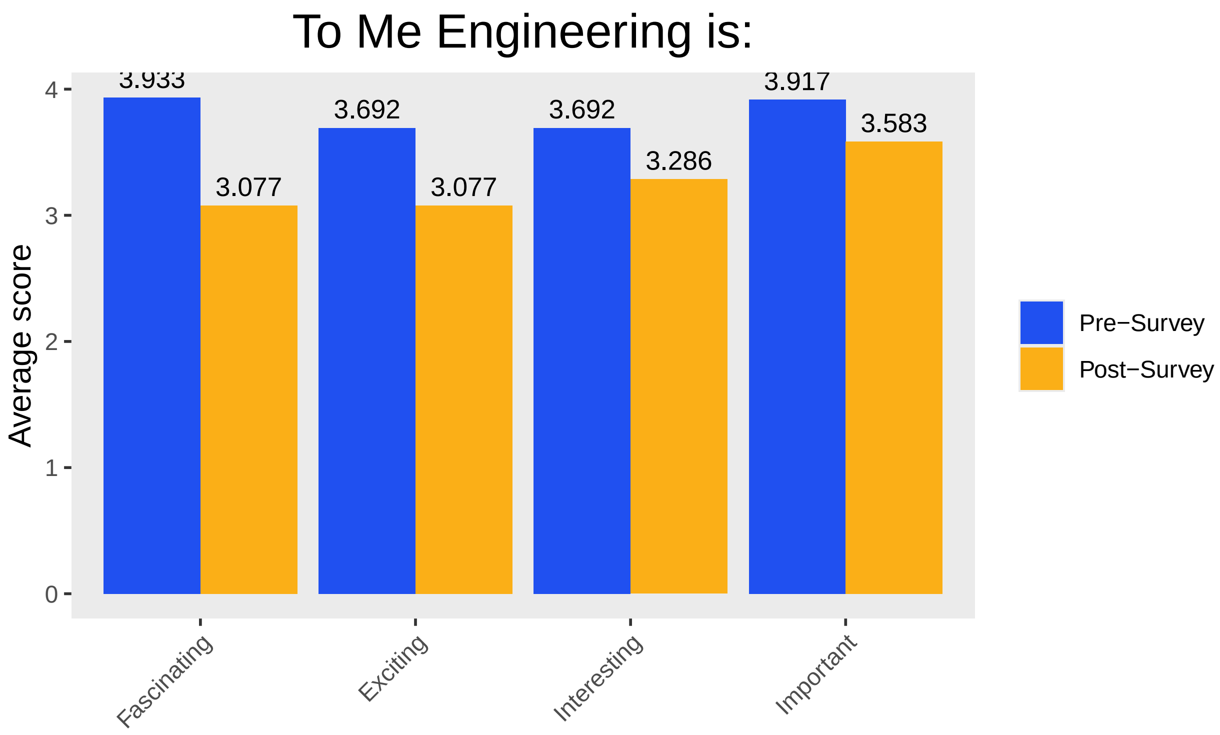
```r
# Generating plots to comapre pre and post data
# Load necessary libraries
library(ggplot2)
library(tidyr)
library(stringr)


plot_figures <- function(data, grep_string, title = NULL) {
  # Filter data based on the grep_string
  subset <- data[grepl(grep_string, data$Question), ]
  colnames(subset) <- c("Question", "Pre-Survey", "Post-Survey")

  # Convert data to long format
  subset_long <- pivot_longer(subset,
                              cols = c("Pre-Survey", "Post-Survey"),
                              names_to = "Type",
                              values_to = "Mean")

  # Create a mapping for the original questions to the new labels
  label_mapping <- c("Fascinating", "Exciting", "Interesting", "Important")
  names(label_mapping) <- unique(subset_long$Question)

  # Ensure the bars appear in the order of label_mapping
  subset_long$Question <- factor(subset_long$Question, levels = names(label_mapping))

  # Ensure Pre-Survey appears to the left of Post-Survey
  subset_long$Type <- factor(subset_long$Type, levels = c("Pre-Survey", "Post-Survey"))

  # Create a title
  if (is.null(title)){
  title <- paste("To Me", str_to_title(grep_string), "is:")
  }
  ggplot(subset_long, aes(x = Question, y = Mean, fill = Type)) +
    geom_bar(stat = "identity", position = "dodge") +
    labs(title = title, x = "", y = "Average score") +
    scale_fill_manual(values = c("Pre-Survey" = "#2050f0", "Post-Survey" = "#fbaf17")) +
    scale_x_discrete(labels = label_mapping) +
    theme(
      plot.title = element_text(hjust = 0.5, size = 18),  # Center the title
      panel.grid.major = element_blank(),  # Remove major grid lines
      panel.grid.minor = element_blank(),  # Remove minor grid lines
      legend.title = element_blank(),  # Remove legend title
      plot.margin = margin(20, 0, 20, 10),
      axis.text.x = element_text(angle = 45, hjust = 1),
      axis.title.x = element_text(size = 12),  # Increase x-axis label size
      axis.title.y = element_text(size = 12)
    ) +
    geom_text(aes(label = round(Mean, 3)),  # Add text labels with rounded mean values
              position = position_dodge(width = 0.9), vjust = -0.5, size = 3.5)
}

plot_figures(mean_data, "SCIENCE")
```



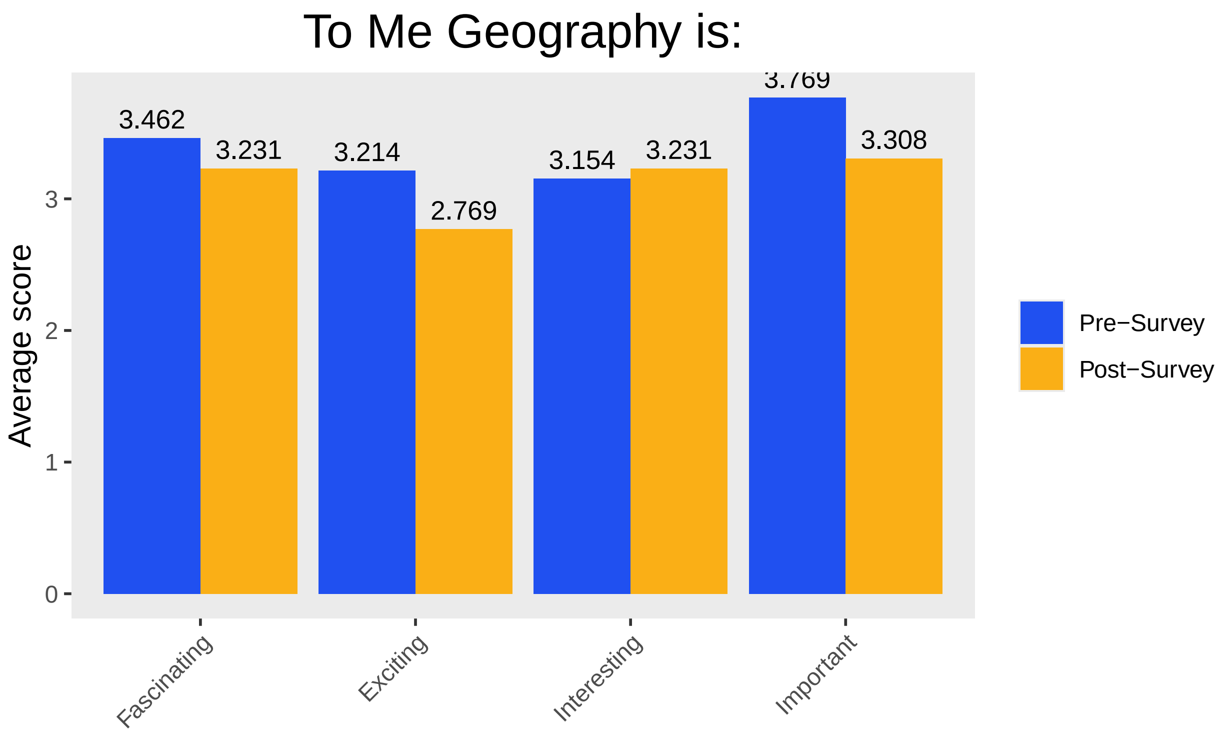
```
plot_figures(mean_data, "TECHNOLOGY")
```



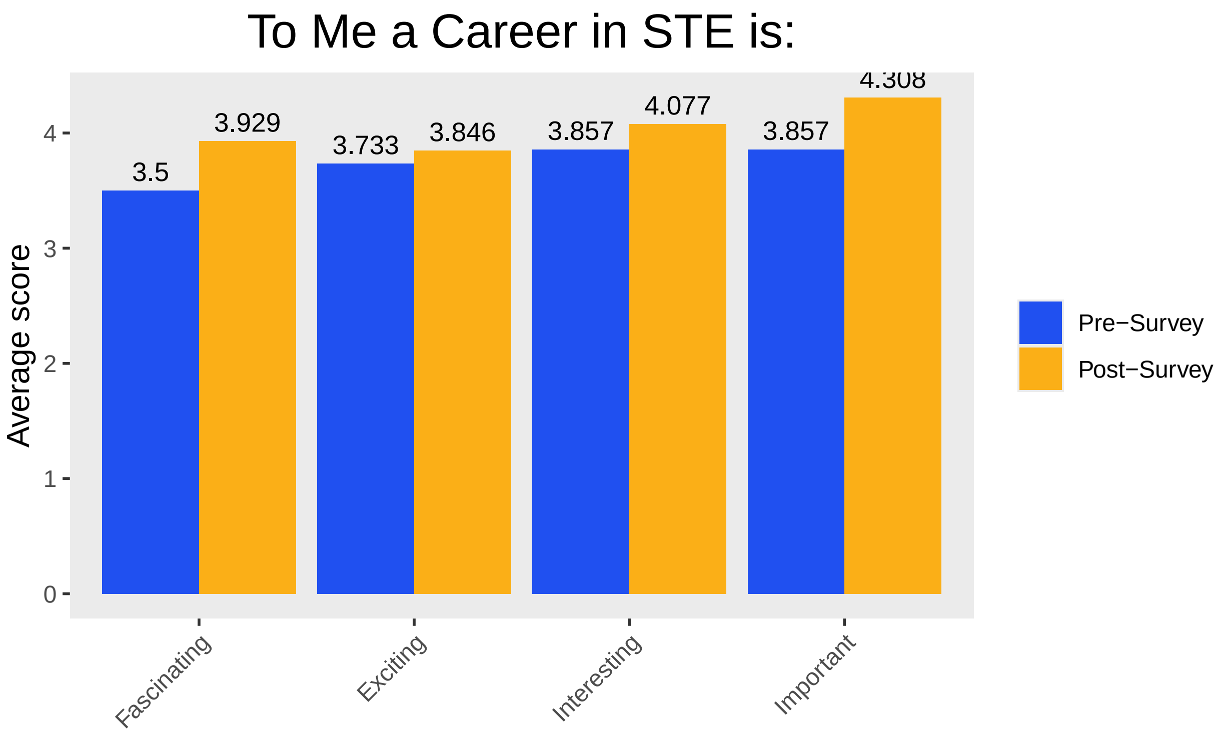
```
plot_figures(mean_data, "ENGINEERING")
```



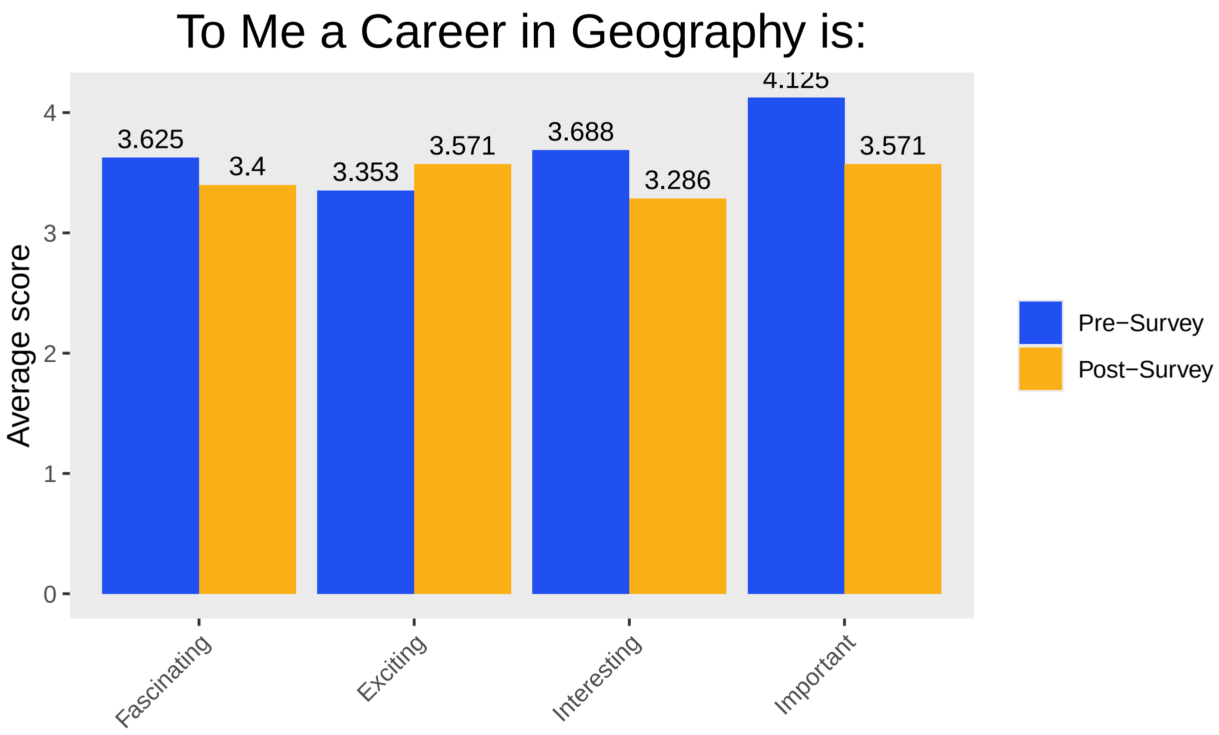
```
plot_figures(mean_data, "GEOGRAPHY")
```



```
plot_figures(mean_data, "STE", "To Me a Career in STE is:")
```

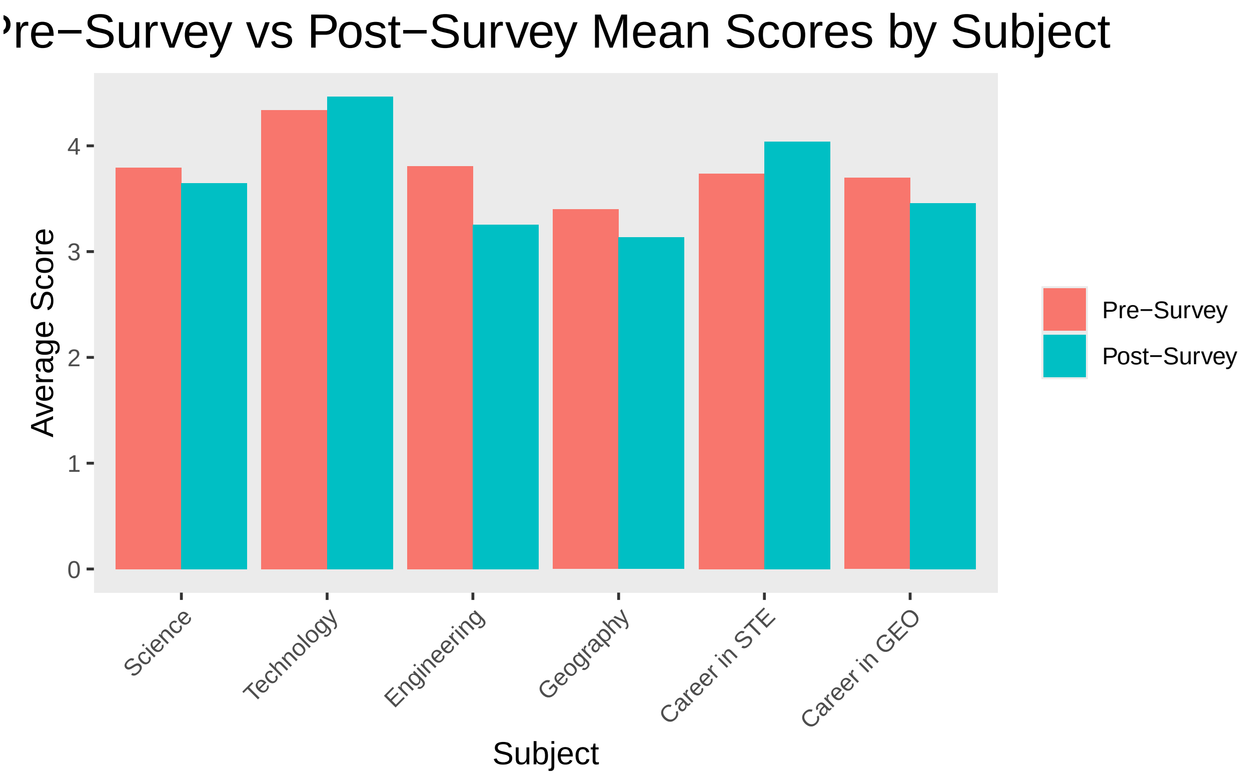


```
plot_figures(mean_data, "GEO_", "To Me a Career in Geography is:")
```



3. Plotting the mean scores for each subject

```r
# making a new dataframe where the cols are pre and post and rows are the categories of the questions
# Creating the subject_data data frame
subject_data <- data.frame(
  `Pre-Survey` = numeric(6),
  `Post-Survey` = numeric(6),
  row.names = c("Science", "Technology", "Engineering", "Geography", "Career in STE", "Career in GEO")
)

# SCIENCE
subject_data["Science", "Pre-Survey"] <- mean(mean_data[grepl("SCIENCE", mean_data$Question), "Mean_PRE
subject_data["Science", "Post-Survey"] <- mean(mean_data[grepl("SCIENCE", mean_data$Question), "Mean_PO

# TECHNOLOGY
subject_data["Technology", "Pre-Survey"] <- mean(mean_data[grepl("TECHNOLOGY", mean_data$Question), "Mea
subject_data["Technology", "Post-Survey"] <- mean(mean_data[grepl("TECHNOLOGY", mean_data$Question), "M

# ENGINEERING
subject_data["Engineering", "Pre-Survey"] <- mean(mean_data[grepl("ENGINEERING", mean_data$Question), "
subject_data["Engineering", "Post-Survey"] <- mean(mean_data[grepl("ENGINEERING", mean_data$Question),

# GEOGRAPHY
subject_data["Geography", "Pre-Survey"] <- mean(mean_data[grepl("GEOGRAPHY", mean_data$Question), "Mean_
subject_data["Geography", "Post-Survey"] <- mean(mean_data[grepl("GEOGRAPHY", mean_data$Question), "Mea

# STE
subject_data["Career in STE", "Pre-Survey"] <- mean(mean_data[grepl("STE", mean_data$Question), "Mean_P
subject_data["Career in STE", "Post-Survey"] <- mean(mean_data[grepl("STE", mean_data$Question), "Mean_

# GEO
subject_data["Career in GEO", "Pre-Survey"] <- mean(mean_data[grepl("GEO_", mean_data$Question), "Mean_
subject_data["Career in GEO", "Post-Survey"] <- mean(mean_data[grepl("GEO_", mean_data$Question), "Mean

# Reshape data into long format
subject_data_long <- data.frame(
  Subject = rep(rownames(subject_data), times = 2),
  Survey_Type = factor(rep(c("Pre-Survey", "Post-Survey"), each = nrow(subject_data)), levels = c("Pre-
  Mean_Score = c(subject_data$`Pre-Survey`, subject_data$`Post-Survey`)
)

# Ensure the order of subjects on the x-axis
subject_data_long$Subject <- factor(subject_data_long$Subject, levels = rownames(subject_data))

# Create the double bar graph
ggplot(subject_data_long, aes(x = Subject, y = Mean_Score, fill = Survey_Type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Pre-Survey vs Post-Survey Mean Scores by Subject",
       x = "Subject",
       y = "Average Score") +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18),  # Center the title
    panel.grid.major = element_blank(),  # Remove major grid lines
    panel.grid.minor = element_blank(),  # Remove minor grid lines
    legend.title = element_blank(),  # Remove legend title
    plot.margin = margin(20, 0, 20, 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title.x = element_text(size = 12),  # Increase x-axis label size
    axis.title.y = element_text(size = 12)
  )
```



```r
subject_data[, 3:4]
```

```
##               Pre-Survey Post-Survey
## Science         3.793199    3.647917
## Technology      4.337316    4.466667
## Engineering     3.808654    3.255723
## Geography       3.399725    3.134615
## Career in STE   3.736905    4.039835
## Career in GEO   3.697610    3.457143
```

4. T-Test

```r
# Perform a paired t-test for each subject, including only students with complete data

# Identify common students based on the "Name" column
common_students <- intersect(pre_data$Name, post_data$Name)

# Filter pre_data and post_data for common students only
pre_data_filtered <- pre_data[pre_data$Name %in% common_students, ]
post_data_filtered <- post_data[post_data$Name %in% common_students, ]

# Define the updated paired_t function
paired_t <- function(pre_data, post_data, subject) {
  # Extract the pre-test and post-test columns matching the subject
  pre_cols <- grep(subject, colnames(pre_data), value = TRUE)
  post_cols <- grep(subject, colnames(post_data), value = TRUE)

  # Ensure the same number of columns are found for both pre and post data
  if (length(pre_cols) != length(post_cols)) {
    stop("Mismatch in the number of columns found for pre and post data.")
  }

  # Combine pre_data and post_data on 'Name'
  combined_data <- merge(pre_data[, c("Name", pre_cols)], post_data[, c("Name", post_cols)], by = "Name

  # Identify rows where all pre and post data for the subject are complete (no NA)
  valid_rows <- complete.cases(combined_data)
  complete_data <- combined_data[valid_rows, ]

  # Check if there are enough data points to perform t-test
  if (nrow(complete_data) == 0) {
    stop("No complete data available to perform t-test for subject: ", subject)
  }

  # Flatten the pre and post data into vectors
  pre <- as.numeric(unlist(complete_data[, pre_cols]))
  post <- as.numeric(unlist(complete_data[, post_cols]))

  # Perform the paired t-test
  t.test(pre, post, paired = TRUE)$p.value
}

# List of subjects
subjects <- c("SCIENCE", "TECHNOLOGY", "ENGINEERING", "GEOGRAPHY", "STE", "GEO_")

# Initialize a data frame to store p-values
t_test_results <- data.frame(Subject = character(), P_Value = numeric(), stringsAsFactors = FALSE)

for (subject in subjects) {
  p_value <- tryCatch({
    paired_t(pre_data_filtered, post_data_filtered, subject)
  }, error = function(e) {
    NA  # If there's an error (e.g., no data), return NA
  })
  t_test_results <- rbind(t_test_results, data.frame(Subject = subject, P_Value = p_value))
}

# Display the t-test results
#rename STE to Career in STE
t_test_results$Subject[t_test_results$Subject == "STE"] <- "Career in STE"
t_test_results$Subject[t_test_results$Subject == "GEO_"] <- "Career in GEO"
print(t_test_results)
```

```
##          Subject     P_Value
## 1        SCIENCE 0.295957470
## 2     TECHNOLOGY 0.321394280
## 3   ENGINEERING 0.005226938
## 4     GEOGRAPHY 0.006262083
## 5 Career in STE 0.182462768
## 6 Career in GEO 0.429452013
```