Methods for Interpreting Meaningful Change in Sport Science

Nico Philipp PhD - 2025-02-27

Introduction

Athletes and coaches need robust methods to determine whether observed performance changes are **real** and **meaningful**, rather than **random variation**.
This document presents **four statistical approaches** to quantifying meaningful change in sports science:

Methods Covered

1. **Standard Error of Measurement (SEM) & Minimal Detectable Change (MDC)**
2. **Coefficient of Variation (CV) Method**
3. **Smallest Worthwhile Change (SWC)**
4. **Model Statistic Method**

Each method is supported with **example R code and visualizations**.

Load Packages & Simulate Data

```r
library(tidyverse)
library(knitr)
library(kableExtra)
options(kableExtra.latex.load_packages = FALSE)
library(ggforce)
library(broom)
library(tinytex)
```

```r
#-------------------------------------------------------------
set.seed(123)  # Set seed for reproducibility

# Create the data frame
df <- expand.grid(Athlete = paste0("Athlete ", 1:15),
                  TimePoint = c(1, 2)) %>%
  mutate(Calf_Strength_1_N = rnorm(n(), mean = 1500, sd = 200),
         Calf_Strength_2_N = rnorm(n(), mean = 1500, sd = 125),
         Calf_Strength_3_N = rnorm(n(), mean = 1500, sd = 130))

# Print first few rows
head(df)
```

```
##     Athlete TimePoint Calf_Strength_1_N Calf_Strength_2_N Calf_Strength_3_N
## 1 Athlete 1         1          1387.905          1553.308          1549.353
## 2 Athlete 2         1          1453.965          1463.116          1434.698
## 3 Athlete 3         1          1811.742          1611.891          1456.683
## 4 Athlete 4         1          1514.102          1609.767          1367.585
## 5 Athlete 5         1          1525.858          1602.698          1360.667
## 6 Athlete 6         1          1843.013          1586.080          1539.459
```

```
#-------------------------------------------------------------
# Pivot the data frame to long format
df_long <- df %>%
  pivot_longer(cols = starts_with("Calf_Strength_"),
               names_to = "Trial",
               values_to = "Calf_Strength") %>%
  mutate(Trial = as.numeric(str_extract(Trial, "\\d+")),
         TimePoint = rep(c(1, 2), each = 15 * 3))  # Restore TimePoint

#--------------------------------------------------------------
#time point 1 only to calculate baseline variability

df_baseline <- df_long %>%
  filter(TimePoint == 1)

#--------------------------------------------------------------
#change score data frame
df_change <- df_long %>%
  group_by(Athlete, TimePoint) %>%
  summarise(Mean_Calf_Strength = mean(Calf_Strength), .groups = "drop") %>%
  pivot_wider(names_from = TimePoint, values_from = Mean_Calf_Strength, names_prefix = "TimePoint_") %>%
  mutate(Change_Score = TimePoint_2 - TimePoint_1,  # Compute absolute change
         Percent_Change = (Change_Score / TimePoint_1) * 100) #Percent change

# View results
head(df_change)
```

```
## # A tibble: 6 x 5
##   Athlete   TimePoint_1 TimePoint_2 Change_Score Percent_Change
##   <fct>           <dbl>       <dbl>        <dbl>          <dbl>
## 1 Athlete 1       1497.       1617.        120.            8.01
## 2 Athlete 2       1451.       1504.         53.5           3.69
## 3 Athlete 3       1627.       1297.       -330.          -20.3
## 4 Athlete 4       1497.       1587.         90.0           6.01
## 5 Athlete 5       1496.       1459.        -37.4          -2.50
## 6 Athlete 6       1656.       1440.       -217.          -13.1
```

Example 1: Standard Error of Measurement (SEM) / Mindimal Difference Needed to Be Considerd "Meaningful" (MDC)

Our first example is based on the paper by Weir from 2005 (Weir, 2005). The standard error of measurement (SEM) represents measurement error or within-subject variability between repeated measures (3 baseline trials in our case). The SEM can be acquired from an ANOVA by taking the square-root of the mean-squared error for the residuals. The code below applies this to our simulated calf strength data.

Concept

- **SEM measures within-subject variability between repeated trials.**
- **MDC determines the minimum change needed to be statistically significant beyond measurement error.**

```
#-------------------------------------------------------------
#Weir, 2005 - SEM & MD
#-------------------------------------------------------------

# Fit a standard ANOVA model
model <- aov(`Calf_Strength` ~ `Trial`, data = df_baseline)

# Summary of the ANOVA model
summary(model)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Trial        1  16901   16901   0.872  0.356
## Residuals   43 833066   19374
```

```
# Extract residual variance (Mean Squared Error)
residual_variance <- summary(model)[[1]]["Residuals", "Mean Sq"]

# Compute Standard Error of Measurement (SEM)
SEM <- sqrt(residual_variance)

# Compute Minimal Detectable Change (MDC) at 90% confidence
MDC <- SEM * 1.65 * sqrt(2)

# Print results
cat("Standard Error of Measurement (SEM):", SEM, "\n")
```

```
## Standard Error of Measurement (SEM): 139.1892
```

```
cat("Minimal Detectable Change (MDC 95% CI):", MDC, "\n")
```

```
## Minimal Detectable Change (MDC 95% CI): 324.7913
```

```
#----------------------------------------------------------------
#change meaningful?
# Add SEM and MDC to df_change and check for significant positive/negative changes
df_change_sem_md <- df_long %>%
  group_by(Athlete, TimePoint) %>%
  summarise(Mean_Calf_Strength = mean(Calf_Strength), .groups = "drop") %>%
  pivot_wider(names_from = TimePoint, values_from = Mean_Calf_Strength,
              names_prefix = "TimePoint_") %>%
  mutate(Change_Score = TimePoint_2 - TimePoint_1,
         Percent_Change = (Change_Score / TimePoint_1) * 100,
         SEM = SEM,  # Add SEM column
         MDC = MDC,  # Add MDC column
         Change_Category = case_when(
           Change_Score > SEM  ~ "Significant Increase",  # Improved beyond MDC
           Change_Score < -SEM ~ "Significant Decrease",  # Worsened beyond MDC
           abs(Change_Score) <= SEM ~ "Within Normal Variability" # Normal range
         ))

#print data frame

print_df_change_sem_md <- df_change_sem_md %>%
  mutate(
    Change_Score = as.numeric(Change_Score),
    SEM = as.numeric(SEM),
    MDC = as.numeric(MDC)
  ) %>%
  mutate(across(where(is.numeric), round, 2)) %>%
  select(Athlete, Change_Score, SEM, MDC, Change_Category)

head(print_df_change_sem_md)
```

```
## # A tibble: 6 x 5
##   Athlete   Change_Score   SEM   MDC Change_Category
##   <fct>            <dbl> <dbl> <dbl> <chr>
## 1 Athlete 1        120.   139.  325. Within Normal Variability
## 2 Athlete 2         53.5  139.  325. Within Normal Variability
## 3 Athlete 3       -330.   139.  325. Significant Decrease
## 4 Athlete 4         90.0  139.  325. Within Normal Variability
## 5 Athlete 5        -37.4  139.  325. Within Normal Variability
## 6 Athlete 6       -217.   139.  325. Significant Decrease
```

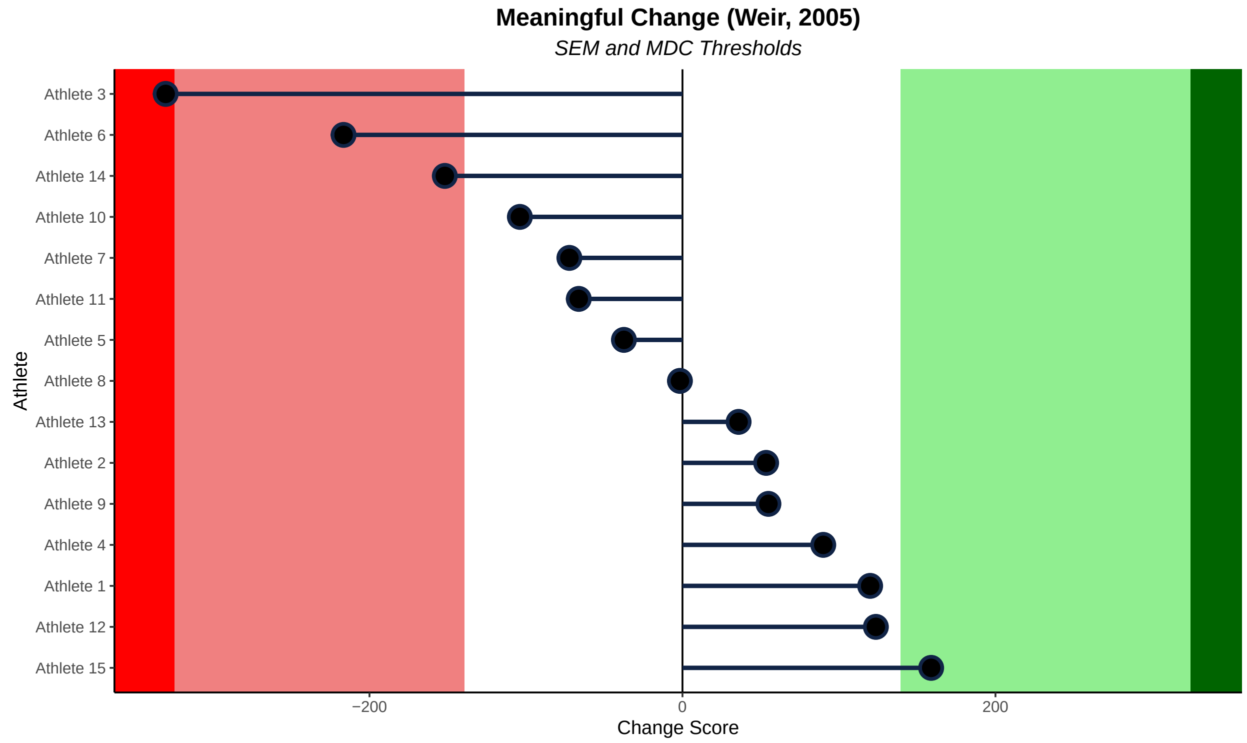
Compared to other methods such as calcualting a coefficient of variation percentage, this methods highlights the trial to trial variability in the data in the actual units of measure (Newtons in our case). **Our data shows that at baseline, across our group, we may expect 139 Newtons of variability**, with changing having to exceed this variabiltiy to be considered "real" or "meaningful". The MD is calculated by constructing a confidence interval around the SEM and presents **a much more robust threshold.**

Table to highlight results

In this example we can see that 3 athletes had negative change greater than the pre-test SEM, while one athlete saw positive change greater than the SEM. Only 1 athlete saw change greater thant he MDC, which is a more robust and less sensitive threshold compared to just using the SEM.

Plot to highlight results

```
#plot
# Reorder athletes based on Change Score (largest to smallest)
df_change_sem_md <- df_change_sem_md %>%
  mutate(Athlete = fct_reorder(Athlete, Change_Score, .desc = TRUE))  # Orders

# Create the lollipop plot with sorting
sem_mdc_plot <- ggplot(df_change_sem_md) +
  aes(x = Change_Score, y = Athlete) +

  # Add shaded regions:
  geom_rect(aes(xmin = -MDC, xmax = -SEM, ymin = -Inf, ymax = Inf),
            fill = "lightcoral", alpha = 0.3) +  # Light orange (Negative SEM to Negative MDC)
  geom_rect(aes(xmin = -Inf, xmax = -MDC, ymin = -Inf, ymax = Inf),
            fill = "red", alpha = 0.3) +  # Red (Less than Negative MDC)
  geom_rect(aes(xmin = SEM, xmax = MDC, ymin = -Inf, ymax = Inf),
            fill = "lightgreen", alpha = 0.3) +  # Light green (Positive SEM to Positive MDC)
  geom_rect(aes(xmin = MDC, xmax = Inf, ymin = -Inf, ymax = Inf),
            fill = "darkgreen", alpha = 0.3) +  # Dark green (Greater than Positive MDC)

  # Add vertical reference lines for SEM and MDC
  geom_vline(xintercept = 0) +

  # Add lollipop sticks (segments)
  geom_segment(aes(x = 0, xend = Change_Score, y = Athlete, yend = Athlete),
               color = "#112446", linewidth = 1.2) +

  # Add lollipop heads (points)
  geom_point(aes(x = Change_Score, y = Athlete),
             size = 5, color = "#112446", fill = "black", shape = 21, stroke = 1.5) +

  # Labels and theme
  labs(
    title = "Meaningful Change (Weir, 2005)",
    subtitle = "SEM and MDC Thresholds"
  ) +
  xlab("Change Score") +
  theme_classic() +
  theme(
    plot.title = element_text(size = 14L, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 12L, face = "italic", hjust = 0.5)
  )
print(sem_mdc_plot)
```



Example 2: Coefficient of Variation (CV)

The Coefficient of Variation (CV) method assesses whether an athlete's performance change is meaningful by comparing the percentage change in performance to the baseline variability (CV).

Concept

- **CV assesses change relative to baseline variability**
- **If % change > CV, it is considered beyond normal variability**

```
#--------------------------------------------------------------------------
#Coefficient of Variation
#--------------------------------------------------------------------------

# Calculate Coefficient of Variation (CV) in df_baseline
df_baseline <- df_baseline %>%
  group_by(Athlete) %>%
  mutate(CV_Calf_Strength = (sd(Calf_Strength) / mean(Calf_Strength)) * 100) %>%
  ungroup() %>%
  #select(Athlete, CV_Calf_Strength) %>%  # Keep only necessary columns for merging
  distinct()  # Ensure only one row per Athlete

#-------------------------------------------------------------------------

df_change_cv <- df_long %>%
  group_by(Athlete, TimePoint) %>%
  summarise(Mean_Calf_Strength = mean(Calf_Strength), .groups = "drop") %>%
  pivot_wider(names_from = TimePoint, values_from = Mean_Calf_Strength,
              names_prefix = "TimePoint_") %>%
  mutate(Change_Score = TimePoint_2 - TimePoint_1,
         Percent_Change = (Change_Score / TimePoint_1) * 100,
         SEM = SEM,  # Add SEM column
         MDC = MDC,  # Add MDC column
         Change_Category = case_when(
           Change_Score > SEM  ~ "Significant Increase",
           Change_Score < -SEM ~ "Significant Decrease",
           abs(Change_Score) <= SEM ~ "Within Normal Variability"
         )) %>%
  left_join(df_baseline %>% select(Athlete, CV_Calf_Strength) %>%
              distinct(), by = "Athlete") %>%  # Merge CV_Calf_Strength
  mutate(CV_Comparison = case_when(
    abs(Percent_Change) > CV_Calf_Strength ~ "Beyond Normal Variability",
    TRUE ~ "Within Normal Variability"
  ))

#print data frame

print_df_change_cv <- df_change_cv %>%
  mutate(
    Percent_Change = as.numeric(Percent_Change),
    CV_Calf_Strength = as.numeric(CV_Calf_Strength),
  ) %>%
  mutate(across(where(is.numeric), round, 2)) %>%
  select(Athlete, Percent_Change, CV_Calf_Strength, CV_Comparison)

head(print_df_change_cv)
```

```
## # A tibble: 6 x 4
##   Athlete   Percent_Change CV_Calf_Strength CV_Comparison
##   <fct>              <dbl>            <dbl> <chr>
## 1 Athlete 1           8.01             6.3  Beyond Normal Variability
## 2 Athlete 2           3.69             1    Beyond Normal Variability
## 3 Athlete 3         -20.3             10.9  Beyond Normal Variability
## 4 Athlete 4           6.01             8.15 Within Normal Variability
## 5 Athlete 5          -2.5              8.26 Within Normal Variability
## 6 Athlete 6         -13.1              9.87 Beyond Normal Variability
```

Plot to highlight results

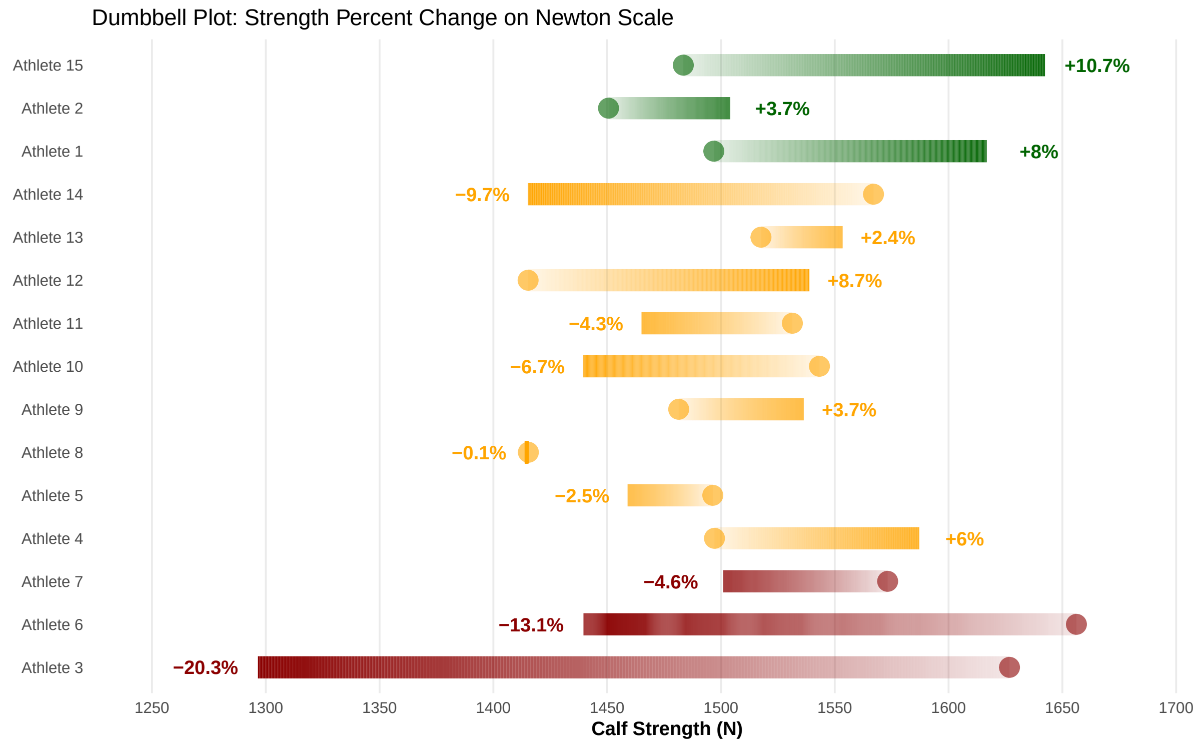
Dumbbell Plot to visualize change between two timepoints adapted from Horsley (2022)

https://insidethenumbers.netlify.app/post/change-scores/

```
#plot

# Define colors for changes
df_change_cv <- df_change_cv %>%
  mutate(colour = case_when(
    Percent_Change < -CV_Calf_Strength ~ "darkred",  # Significant Decrease (Red)
```

```r
    Percent_Change > CV_Calf_Strength ~ "darkgreen",  # Significant Increase (Green)
    Percent_Change < 0 & Percent_Change >= -CV_Calf_Strength ~ "orange",  # Small Decrease (Orange)
    Percent_Change > 0 & Percent_Change <= CV_Calf_Strength ~ "orange"  # Small Increase (Orange)
  )) %>%

  # Order by color: Green (top), Orange (middle), Red (bottom)
  mutate(colour = factor(colour, levels = c("darkgreen", "orange", "darkred")),
         Athlete = fct_reorder(Athlete, as.numeric(colour), .desc = TRUE))
# Sort according to color levels

# Create dumbbell plot
ggplot(df_change_cv) +
  # Dumbbell line connecting TimePoint_1 to TimePoint_2
  geom_link(aes(x = TimePoint_1, xend = TimePoint_2, y = Athlete, yend = Athlete,
                colour = colour, alpha = after_stat(index)),
            show.legend = FALSE, linewidth = 6, n = 500) +

  # Start point (TimePoint_1)
  geom_point(aes(x = TimePoint_1, y = Athlete, colour = colour), shape = 19,
             size = 5, alpha = 0.6) +

  # End point (TimePoint_2)
  geom_point(aes(x = TimePoint_2, y = Athlete, colour = colour), shape = 32,
             size = 5, alpha = 0.6) +

  # Define x-axis limits & breaks
  scale_x_continuous(limits = c(min(df_change$TimePoint_1, df_change$TimePoint_2) - 50,
                                max(df_change$TimePoint_1, df_change$TimePoint_2) + 50),
                     breaks = scales::pretty_breaks(n = 8)) +

  scale_colour_identity() +  # Keep defined colors

  # Classic minimal theme
  theme_minimal() +
  labs(x = "Calf Strength (N)", title = "Dumbbell Plot: Strength Percent Change on Newton Scale") +
  theme(panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_blank(),
        axis.title.x = element_text(face = "bold"),
        axis.title.y = element_blank()) +

  # Large negative changes (significant decrease)
  geom_text(data = subset(df_change_cv, Percent_Change < -CV_Calf_Strength),
            aes(x = TimePoint_2, y = Athlete, fontface = "bold",
                label = paste0(round(Percent_Change, 1), "%")), color = "darkred",
            nudge_x = -23) +

  # Large positive changes (significant increase)
  geom_text(data = subset(df_change_cv, Percent_Change > CV_Calf_Strength),
            aes(x = TimePoint_2, y = Athlete, fontface = "bold",
                label = paste0("+", round(Percent_Change, 1), "%")), color = "darkgreen",
            nudge_x = 23) +

  # Small negative changes (within normal variability)
```

```
geom_text(data = subset(df_change_cv, Percent_Change < 0 & Percent_Change >= -CV_Calf_Strength),
          aes(x = TimePoint_2, y = Athlete, fontface = "bold",
              label = paste0(round(Percent_Change, 1), "%")), color = "orange",
          nudge_x = -20) +

# Small positive changes (within normal variability)
geom_text(data = subset(df_change_cv, Percent_Change > 0 & Percent_Change <= CV_Calf_Strength),
          aes(x = TimePoint_2, y = Athlete, fontface = "bold",
              label = paste0("+", round(Percent_Change, 1), "%")),
          color = "orange", nudge_x = 20)
```



Smallest Worthwhile Change (SWC)

Harry et al. recently wrote on this approach in JSCR (Harry et al. 2024). While somewhat arbitrary, to calculate the SWC, a constant of 0.2 is used to establish the SWC threshold for trained populations or athletes. Earlier literature has suggested that the SWC approach may be excessively sensitive, which is not ideal for athlete populations with a wide range of responses to training stimuli. This is further highlighted in our example below.

Concept

- **SWC is 0.2 × SD of baseline performance.**
- **Overly sensitive for detecting small changes.**

```r
# Step 1: Calculate Individual SD for Each Athlete
df_swc <- df_long %>%
  group_by(Athlete) %>%
  summarise(SD_Athlete = sd(Calf_Strength), .groups = "drop") %>%
  mutate(SWC = 0.2 * SD_Athlete)  # Apply SWC formula for trained athletes

# Step 2: Compute Individual Change Scores & Merge SWC
df_change_swc <- df_long %>%
  group_by(Athlete, TimePoint) %>%
  summarise(Mean_Calf_Strength = mean(Calf_Strength), .groups = "drop") %>%
  pivot_wider(names_from = TimePoint, values_from = Mean_Calf_Strength,
              names_prefix = "TimePoint_") %>%
  mutate(Change_Score = TimePoint_2 - TimePoint_1) %>%
  left_join(df_swc, by = "Athlete") %>%  # Merge athlete-specific SWC values
  mutate(SWC_Category = case_when(
    abs(Change_Score) > SWC ~ "Meaningful Change",
    TRUE ~ "Trivial Change"
  ))

# print data frame
print_df_change_swc <- df_change_swc %>%
  mutate(
    Change_Score = as.numeric(Change_Score),
    SWC = as.numeric(SWC),
  ) %>%
  mutate(across(where(is.numeric), round, 2)) %>%
  select(Athlete, Change_Score, SWC, SWC_Category)

head(print_df_change_swc)
```

```
## # A tibble: 6 x 4
##   Athlete   Change_Score   SWC SWC_Category
##   <fct>            <dbl> <dbl> <chr>
## 1 Athlete 1        120.   36.2 Meaningful Change
## 2 Athlete 2         53.5  12.2 Meaningful Change
## 3 Athlete 3       -330.   47.8 Meaningful Change
## 4 Athlete 4         90.0  19.8 Meaningful Change
## 5 Athlete 5        -37.4  17.2 Meaningful Change
## 6 Athlete 6       -217.   35.7 Meaningful Change
```

Calculate Baseline Typical Error

We can calculate the typical error in our data from our baseline assessments to use in our interpretation and viusals. This aids us in making an overly sensitive method more robust.

```r
#-------------------------------------------------------
#calculate te
#-------------------------------------------------------

# Step 1: Filter only TimePoint 1 and reshape to wide format
df_baseline_te <- df_baseline %>%
  filter(TimePoint == 1) %>%
  pivot_wider(names_from = Trial, values_from = Calf_Strength, names_prefix = "Trial_")

# Step 2: Compute trial-to-trial differences across all athletes
change_1_to_2 <- df_baseline_te$Trial_2 - df_baseline_te$Trial_1
change_2_to_3 <- df_baseline_te$Trial_3 - df_baseline_te$Trial_2

# Step 3: Compute Typical Error (TE)
te_list <- list(change_1_to_2, change_2_to_3)  # Store differences in a list
group_te <- round(mean(sapply(te_list, sd), na.rm = TRUE) / sqrt(2), 1)  # Compute TE

# Print result
cat("Group-Level Typical Error (TE):", group_te, "\n")
```

```
## Group-Level Typical Error (TE): 140.2
```

```
#Notice that the TE is very similar to the SEM we calcualted earlier
#-------------------------------------------------------------------
```

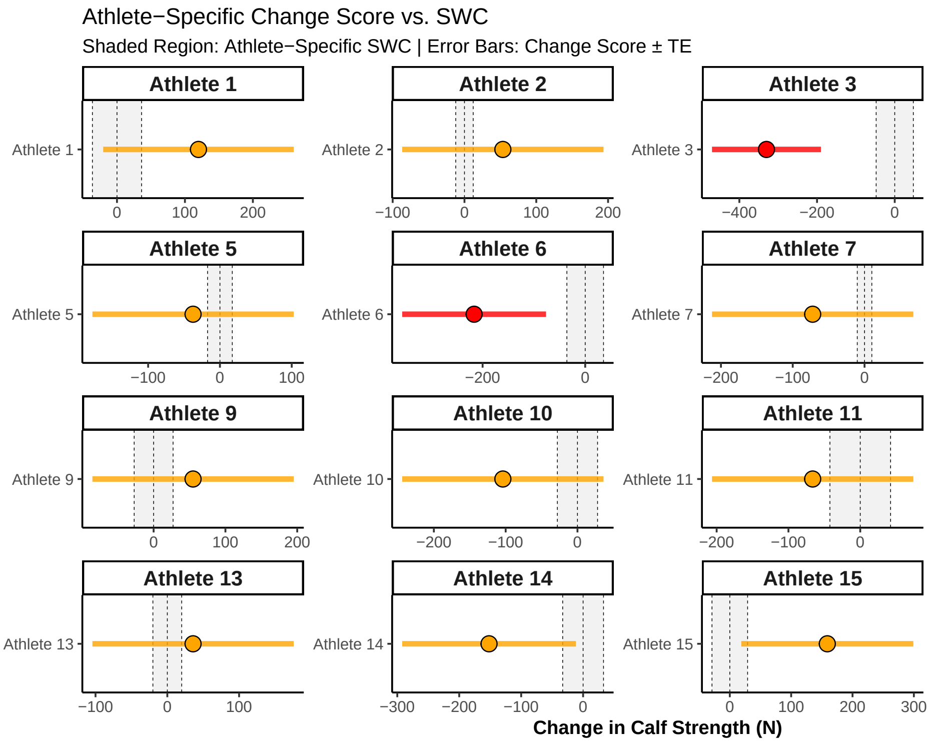
Plot to highlight results

To combat this excessive sensitivity and to reduce false positives from SWC, we calculate TE, which accounts for variability between trials.

In the plot below we flag “meaningful change” as change scores that are greater than the SWC and error bars that do not contact the athlete-specific SWC area (shaded in grey). This is more robust, and shows that only 2 atheltes may have experienced “meaningful” change in calf strength.

```
df_change_swc <- df_change_swc %>%
  group_by(Athlete) %>%
  mutate(
    Upper_Range = Change_Score + group_te,  # Change Score + SEM
    Lower_Range = Change_Score - group_te,  # Change Score - SEM

    # Determine if Change Score ± SEM crosses the athlete-specific SWC
    Line_Contacts_SWC = (Lower_Range < SWC & Upper_Range > -SWC),

    # Assign colors based on SWC crossing condition
    Change_Color = case_when(
      Change_Score > 0 & !Line_Contacts_SWC ~ "green",  # Positive & does NOT touch SWC
      Change_Score < 0 & !Line_Contacts_SWC ~ "red",    # Negative & does NOT touch SWC
      TRUE ~ "orange"  # Line contacts SWC
    )
  ) %>%
  ungroup()  # Ensure calculations are done per athlete

#--------

ggplot(df_change_swc) +

  # Minimal theme for cleaner look
  theme_classic() +
```

```r
# Add athlete-specific shaded vertical region for SWC range
geom_rect(aes(
  xmin = -SWC, xmax = SWC, ymin = -Inf, ymax = Inf
), fill = "grey", alpha = 0.2) +

# Add reference dashed lines at -SWC, 0 (baseline), and +SWC for each athlete
geom_vline(aes(xintercept = 0), linetype = "dashed", size = 0.2) +
geom_vline(aes(xintercept = -SWC), linetype = "dashed", size = 0.2) +
geom_vline(aes(xintercept = SWC), linetype = "dashed", size = 0.2) +

# Add horizontal lineranges for Change Score ± SEM (error bars)
geom_linerange(aes(xmin = Lower_Range, xmax = Upper_Range, y = Athlete,
                   colour = Change_Color), size = 1.5, alpha = 0.8) +

# Ensure colors are applied correctly
scale_colour_identity() +

# Add points for Change Score
geom_point(aes(x = Change_Score, y = Athlete, fill = Change_Color), shape = 21, size = 4) +

# Ensure fill colors match categories
scale_fill_identity() +

# Define x-axis breaks
scale_x_continuous(breaks = scales::pretty_breaks(n = 3)) +

# Labels
labs(x = "Change in Calf Strength (N)",
     title = "Athlete-Specific Change Score vs. SWC",
     subtitle = "Shaded Region: Athlete-Specific SWC | Error Bars: Change Score ± TE") +

# Use facet_wrap to show each athlete individually
facet_wrap(~Athlete, scales = "free") +

# Theme adjustments
theme(
  panel.grid.minor.x = element_blank(),
  panel.grid.major.y = element_blank(),
  axis.title.x = element_text(face = "bold"),
  axis.title.y = element_blank(),
  strip.text = element_text(face = "bold", size = 12)  # Format facet titles
)
```



Model Statistic

This approach has also most recently been highlighted by Harry et al. and is based on work by Bates et al. (1992). The model statistic method may loosely be considered as a single subject dependednt t-test, in which the observed difference between the sessions is compared with a proba- bilistic critical difference (Bates, 1992; Bates, 2016). Compared to our SEM/MDC method from earlier, this method allows for more athlete-specificity by generating athlete-specific mean-squared error terms, while the earlier method is pooled across athletes.

Concept

- **Like a paired t-test for individual athletes**
- **Compares observed difference to probabilistic critical difference**
- **This method does not require population-level comparisons—ideal for single-subject analysis**

```
df_model_stat <- df_long %>%
  group_by(Athlete, TimePoint) %>%
  summarise(Mean_Calf_Strength = mean(Calf_Strength), .groups = "drop") %>%
  pivot_wider(names_from = TimePoint, values_from = Mean_Calf_Strength,
              names_prefix = "TimePoint_") %>%
  mutate(Mean_Diff = TimePoint_2 - TimePoint_1)

#
df_model_stat <- df_long %>%
  group_by(Athlete) %>%
  summarise(SS = sum((Calf_Strength - mean(Calf_Strength))^2),  # Sum of Squares
            df = n() - 1,  # Degrees of freedom
            MSE = SS / df,  # Mean Squared Error
            SEM = sqrt(MSE), .groups = "drop") %>%
  right_join(df_model_stat, by = "Athlete")  # Merge with mean differences

#
df_model_stat <- df_model_stat %>%
  mutate(Model_Stat_Category = case_when(
    abs(Mean_Diff) > SEM ~ "True Difference",
    TRUE ~ "Trivial Difference"
  ))

#print data frame

print_df_model_stat <- df_model_stat %>%
  mutate(
    Mean_Diff = as.numeric(Mean_Diff),
    SEM = as.numeric(SEM),
  ) %>%
  mutate(across(where(is.numeric), round, 2)) %>%
  select(Athlete, Mean_Diff, SEM, Model_Stat_Category)

head(print_df_model_stat)
```

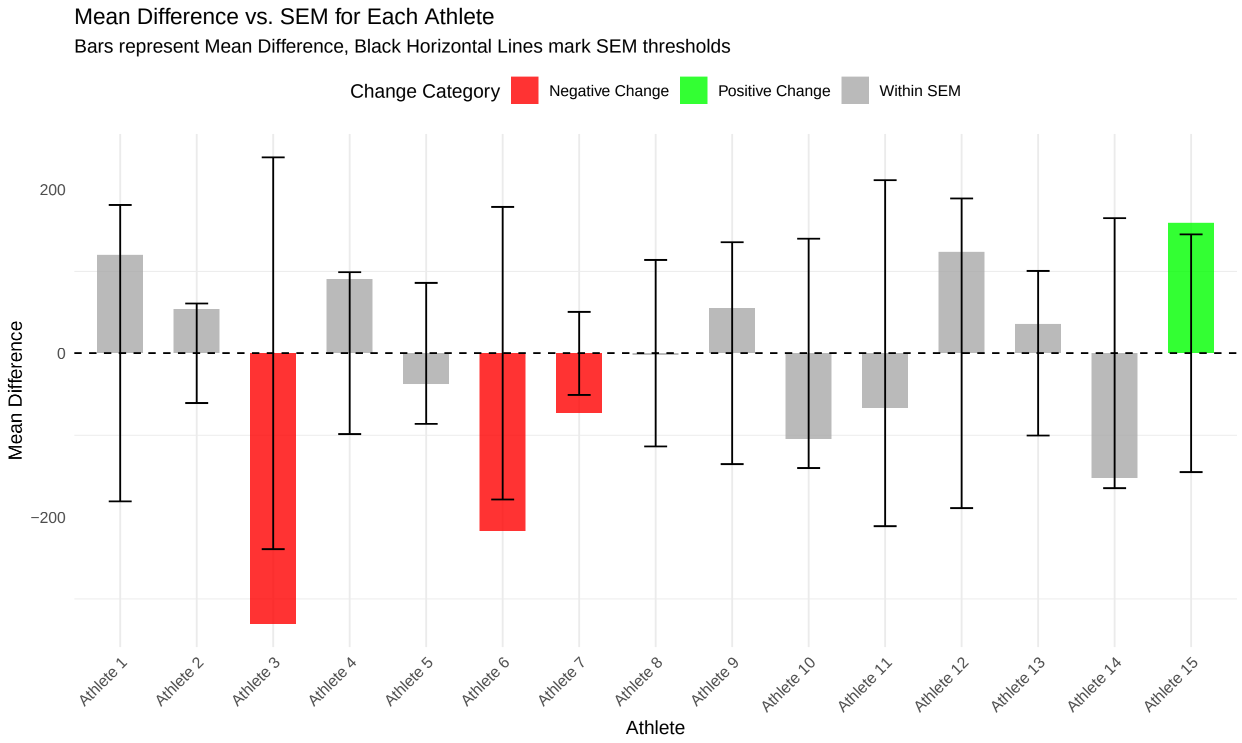
```
## # A tibble: 6 x 4
##   Athlete   Mean_Diff   SEM Model_Stat_Category
##   <fct>         <dbl> <dbl> <chr>
## 1 Athlete 1     120.  181.  Trivial Difference
## 2 Athlete 2      53.5  60.8 Trivial Difference
## 3 Athlete 3    -330.  239.  True Difference
## 4 Athlete 4      90.0  98.9 Trivial Difference
## 5 Athlete 5     -37.4  86.0 Trivial Difference
## 6 Athlete 6    -217.  179.  True Difference
```

Plot to highlight results

We can observer that athletes 3, 5, and 7 experienced potentially “meaningful” reductions in calf strength between the two timepoints, while athlete 15 experienced a potentially “meaningful” increase in calf strength.

```
#plot
# Categorize Mean Difference vs. SEM for color mapping
df_model_stat <- df_model_stat %>%
  mutate(Significance = case_when(
    Mean_Diff > SEM ~ "Positive Change",   # Green for significant positive change
    Mean_Diff < -SEM ~ "Negative Change", # Red for significant negative change
    TRUE ~ "Within SEM"                   # Gray for trivial changes
  ))

# Create the visualization
ggplot(df_model_stat, aes(y = Athlete, x = Mean_Diff, fill = Significance)) +
```

```r
  # Bars for Mean Difference
  geom_col(width = 0.6, alpha = 0.8) +

  # Horizontal line to represent SEM threshold
  geom_errorbarh(aes(xmin = -SEM, xmax = SEM), height = 0.3, color = "black", linewidth = 0.5) +

  # Define color mapping for bars
  scale_fill_manual(values = c(
    "Positive Change" = "green",  # Green for positive meaningful change
    "Negative Change" = "red",    # Red for negative meaningful change
    "Within SEM" = "darkgray"         # Gray for trivial changes
  )) +

  # Add a vertical line at zero for reference
  geom_vline(xintercept = 0, linetype = "dashed", color = "black") +

  # Labels and theme
  labs(
    title = "Mean Difference vs. SEM for Each Athlete",
    subtitle = "Bars represent Mean Difference, Black Horizontal Lines mark SEM thresholds",
    x = "Mean Difference",
    y = "Athlete",
    fill = "Change Category"
  ) +

  # Flip coordinates for better readability
  coord_flip() +

  # Minimal theme with no legend and angled x-axis labels
  theme_minimal() +
  theme(
    legend.position = "top",
    panel.grid.major.y = element_blank(),  # Remove horizontal grid lines for clarity
    axis.text.x = element_text(angle = 45, hjust = 1)  # Angles x-axis labels at 45 degrees
  )
```



Comparison of Different Methods for Assessing Meaningful Change

1. Standard Error of Measurement (SEM) & Minimal Detectable Change (MDC)

Main Idea: Uses measurement error to determine whether a change is real.

Threshold: - SEM represents the expected trial-to-trial variability in measurements. - MDC is a confidence-based threshold (MDC = SEM × 1.65 × squareroot 2) that a change must exceed to be considered meaningful.

Best For: Group-level comparisons where measurement error is a key consideration.

Limitations: May not account for individual differences, as the SEM is calculated for the entire group.

---

2. Coefficient of Variation (CV%) Method

Main Idea: Evaluates relative variability in performance by expressing the standard deviation as a percentage of the mean.

Threshold: - A performance change is considered meaningful if it exceeds the athlete's baseline CV%.

Best For: Tracking individual performance and assessing variability over time.

Limitations: - Can be sensitive to changes in mean values. - Less useful for comparing different populations with different baseline variability.

---

3. Smallest Worthwhile Change (SWC) Method

Main Idea: Uses 0.2 × SD (standard deviation) of baseline performance as a threshold to detect meaningful change.

Threshold: A change is meaningful if it exceeds 0.2 × SD of the athlete's pre-test measurements.

Best For: Quick assessments, especially in team sports or large datasets.

Limitations: Overly sensitive (many false positives).

---

4. Model Statistic Method

Main Idea: Compares the observed performance change to the athlete's own Mean Squared Error (MSE) from multiple trials.

Threshold: A change is considered meaningful if it exceeds the athlete's SEM, which is derived from MSE.

Best For: Single-subject monitoring in individualized athlete testing.

Limitations: - Requires multiple trials per athlete. - More computationally intensive compared to other methods.

References

- Bates BT, Dufek JS, Davis HP. The effect of trial size on statistical power. Med Sci Sports Exerc. 1992 Sep;24(9):1059–65.
- Bates BT, Dufek JS, James CR, Harry JR, Eggleston JD. The influence of experimental design on the detection of performance differences. Meas Phys Educ Exerc Sci. 2016 Oct;20(4):200–7.
- Harry JR, Hurwitz J, Agnew C, Bishop C. Statistical tests for sports science practitioners: Identifying performance gains in individual athletes. J Strength Cond Res [Internet]. 2023 Oct 25 [cited 2024 Apr 25]; Available from: https://repository.mdx.ac.uk/item/w20y4
- Weir JP. Quantifying test-retest reliability using the intraclass correlation coefficient and the SEM. J Strength Cond Res. 2005 Feb;19(1):231–40.