

vertical_jump.R

david

2024-10-06

```
# Install and load packages
packages <- c("dplyr", "tidyr", "lme4", "lmerTest", "emmeans", "ggplot2",
             "ggpubr", "pwr", "performance", "patchwork", "effects")

install_if_missing <- function(pkg) {
  if (!require(pkg, character.only = TRUE)) {
    install.packages(pkg, dependencies = TRUE)
    library(pkg, character.only = TRUE)
  }
}

lapply(packages, install_if_missing)

## Loading required package: dplyr
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
## Loading required package: tidyr
## Loading required package: lme4
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
##
##   expand, pack, unpack
## Loading required package: lmerTest
##
## Attaching package: 'lmerTest'
## The following object is masked from 'package:lme4':
##
##   lmer
```

```

## The following object is masked from 'package:stats':
##
##      step
## Loading required package: emmeans
## Welcome to emmeans.
## Caution: You lose important information if you filter this package's results.
## See '? untidy'
## Loading required package: ggplot2
## Loading required package: ggpubr
## Loading required package: pwr
## Loading required package: performance
## Loading required package: patchwork
## Loading required package: effects
## Loading required package: carData
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
##
## [[10]]
## NULL
##
## [[11]]
## NULL

```

```
# Load data
data_wide <- read.csv("vertical_jump_data_wide.csv")
```

```
# View data
head(data_wide)
```

```
##   sub_id age sex group dynamic_pre dynamic_post static_pre static_post
## 1     1  29  f    a           22           23           24           24
## 2     2  37  m    b           38           42           36           37
## 3     3  55  f    c           20           20           19           20
## 4     4  32  f    a           38           38           35           39
## 5     5  43  m    b           38           40           37           40
## 6     6  42  m    c           39           41           34           36
##   control_pre control_post
## 1           21           22
## 2           39           40
## 3           21           21
## 4           34           35
## 5           36           38
## 6           34           35
```

```
str(data_wide)
```

```
## 'data.frame':   11 obs. of  10 variables:
## $ sub_id      : int  1 2 3 4 5 6 7 8 9 10 ...
## $ age         : int  29 37 55 32 43 42 28 30 28 21 ...
## $ sex         : chr  "f" "m" "f" "f" ...
## $ group       : chr  "a" "b" "c" "a" ...
## $ dynamic_pre : int  22 38 20 38 38 39 36 48 26 31 ...
## $ dynamic_post: int  23 42 20 38 40 41 42 54 34 36 ...
## $ static_pre  : int  24 36 19 35 37 34 32 50 24 23 ...
## $ static_post : int  24 37 20 39 40 36 39 53 32 28 ...
## $ control_pre : int  21 39 21 34 36 34 34 53 23 30 ...
## $ control_post: int  22 40 21 35 38 35 37 55 25 33 ...
```

```
summary(data_wide)
```

```
##      sub_id      age      sex      group
## Min.   : 1.0   Min.   :21.0   Length:11   Length:11
## 1st Qu.: 3.5   1st Qu.:28.5   Class :character   Class :character
## Median : 6.0   Median :30.0   Mode  :character   Mode  :character
## Mean   : 6.0   Mean   :34.0
## 3rd Qu.: 8.5   3rd Qu.:39.5
## Max.   :11.0   Max.   :55.0
##   dynamic_pre dynamic_post static_pre static_post
## Min.   :20.00   Min.   :20.00   Min.   :19.00   Min.   :20.00
## 1st Qu.:28.50   1st Qu.:35.00   1st Qu.:24.00   1st Qu.:30.00
## Median :38.00   Median :40.00   Median :34.00   Median :37.00
## Mean   :34.45   Mean   :38.45   Mean   :32.73   Mean   :36.18
## 3rd Qu.:38.50   3rd Qu.:42.00   3rd Qu.:36.50   3rd Qu.:39.50
## Max.   :48.00   Max.   :54.00   Max.   :50.00   Max.   :53.00
##   control_pre control_post
## Min.   :21.00   Min.   :21.00
## 1st Qu.:26.50   1st Qu.:29.00
## Median :34.00   Median :35.00
```

```
## Mean :33.55 Mean :35.45
## 3rd Qu.:37.50 3rd Qu.:39.00
## Max. :53.00 Max. :55.00

n <- nrow(data_wide)
sex_count <- data_wide %>%
  count(sex)
print(sex_count)

## sex n
## 1 f 5
## 2 m 6

mean(data_wide$age)

## [1] 34

sd(data_wide$age)

## [1] 9.518403

data <- data_wide %>%
  pivot_longer(
    cols = c(dynamic_pre, dynamic_post, static_pre, static_post,
              control_pre, control_post),
    names_to = c("condition", ".value"),
    names_pattern = "(.*)_(.*)"
  ) %>%
  rename(pre_stretch = pre,
         post_stretch = post)

data$condition <- as.factor(data$condition)
data$condition <- factor(data$condition,
                        levels = c("static", "dynamic", "control"))

data <- data %>%
  mutate(jump_diff = post_stretch - pre_stretch)

# View data
head(data)

## # A tibble: 6 x 8
##   sub_id age sex group condition pre_stretch post_stretch jump_diff
##   <int> <int> <chr> <chr> <fct>         <int>         <int>         <int>
## 1     1    29 f    a    dynamic          22          23           1
## 2     1    29 f    a    static           24          24           0
## 3     1    29 f    a    control           21          22           1
## 4     2    37 m    b    dynamic          38          42           4
## 5     2    37 m    b    static           36          37           1
## 6     2    37 m    b    control           39          40           1

str(data)

## tibble [33 x 8] (S3: tbl_df/tbl/data.frame)
## $ sub_id      : int [1:33] 1 1 1 2 2 2 3 3 3 4 ...
## $ age         : int [1:33] 29 29 29 37 37 37 55 55 55 32 ...
## $ sex         : chr [1:33] "f" "f" "f" "m" ...
## $ group       : chr [1:33] "a" "a" "a" "b" ...
```

```
## $ condition : Factor w/ 3 levels "static","dynamic",...: 2 1 3 2 1 3 2 1 3 2 ...
## $ pre_stretch : int [1:33] 22 24 21 38 36 39 20 19 21 38 ...
## $ post_stretch: int [1:33] 23 24 22 42 37 40 20 20 21 38 ...
## $ jump_diff : int [1:33] 1 0 1 4 1 1 0 1 0 0 ...
```

```
summary(data)
```

```
##      sub_id      age      sex      group      condition
## Min.   : 1   Min.   :21   Length:33   Length:33   static :11
## 1st Qu.: 3   1st Qu.:28   Class :character   Class :character   dynamic:11
## Median : 6   Median :30   Mode  :character   Mode  :character   control:11
## Mean   : 6   Mean   :34
## 3rd Qu.: 9   3rd Qu.:42
## Max.   :11   Max.   :55
## pre_stretch post_stretch jump_diff
## Min.   :19.00   Min.   :20.0   Min.   : 0.000
## 1st Qu.:24.00   1st Qu.:32.0   1st Qu.: 1.000
## Median :34.00   Median :37.0   Median : 2.000
## Mean   :33.58   Mean   :36.7   Mean   : 3.121
## 3rd Qu.:38.00   3rd Qu.:41.0   3rd Qu.: 5.000
## Max.   :53.00   Max.   :55.0   Max.   :10.000
```

```
# Linear mixed-effects model
```

```
lmer_model <- lmer(jump_diff ~ condition + (1|sub_id), data = data)
```

```
summary(lmer_model)
```

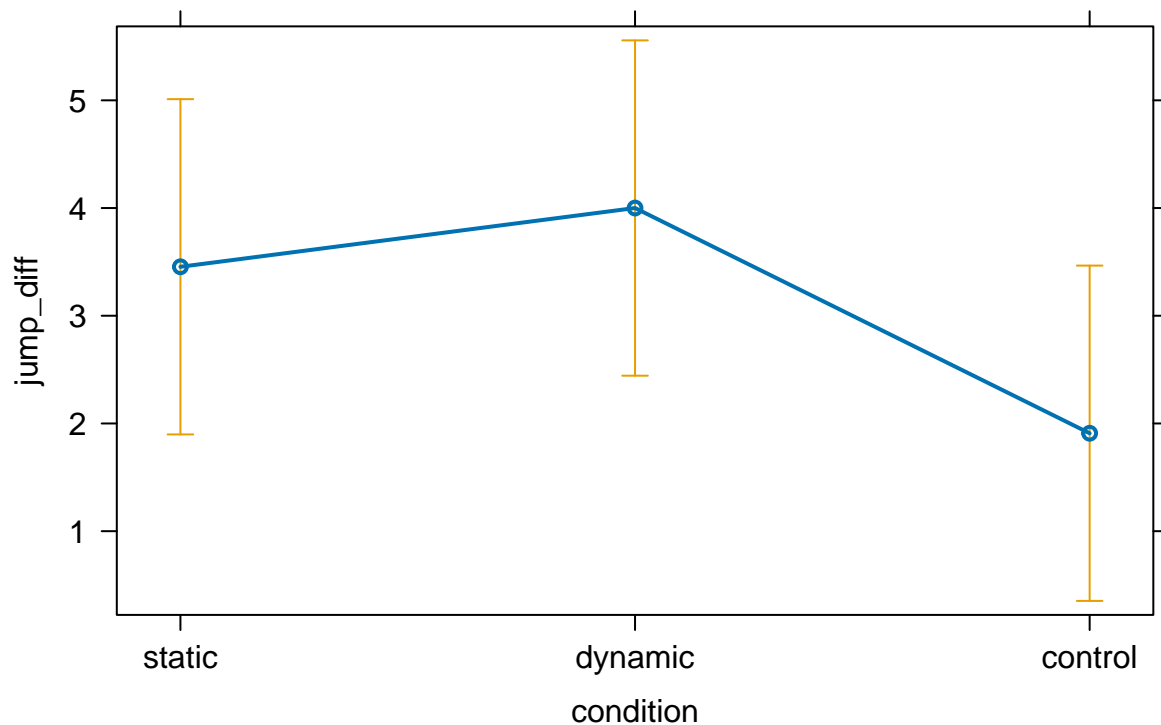
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: jump_diff ~ condition + (1 | sub_id)
##      Data: data
##
## REML criterion at convergence: 139
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.69936 -0.52954  0.07747  0.42722  2.06133
##
## Random effects:
##      Groups      Name      Variance Std.Dev.
## sub_id      (Intercept) 3.582    1.893
## Residual                2.806    1.675
## Number of obs: 33, groups: sub_id, 11
##
## Fixed effects:
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)    3.4545    0.7620 18.4183   4.533 0.000244 ***
## conditiondynamic 0.5455    0.7143 20.0000   0.764 0.453994
## conditioncontrol -1.5455    0.7143 20.0000  -2.164 0.042772 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) cndtnd
## condtdynmc -0.469
## condtnctrl -0.469  0.500
```

```
effects_model <- allEffects(lmer_model)
print(effects_model)
```

```
## model: jump_diff ~ condition
##
## condition effect
## condition
## static dynamic control
## 3.454545 4.000000 1.909091
```

```
plot(effects_model)
```

condition effect plot



```
lmer_null <- lmer(jump_diff ~ (1|sub_id), data = data)
summary(lmer_null)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: jump_diff ~ (1 | sub_id)
## Data: data
##
## REML criterion at convergence: 149.3
##
## Scaled residuals:
## Min 1Q Median 3Q Max
## -1.6618 -0.5698 -0.2330 0.5180 2.3568
##
## Random effects:
## Groups Name Variance Std.Dev.
## sub_id (Intercept) 3.275 1.810
```

```

## Residual          3.727    1.931
## Number of obs: 33, groups:  sub_id, 11
##
## Fixed effects:
##           Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   3.1212    0.6408 10.0000   4.871 0.000651 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# ANOVA of model and null
anova(lmer_model, lmer_null)

## refitting model(s) with ML (instead of REML)

## Data: data
## Models:
## lmer_null: jump_diff ~ (1 | sub_id)
## lmer_model: jump_diff ~ condition + (1 | sub_id)
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## lmer_null      3 156.22 160.71 -75.109   150.22
## lmer_model      5 151.88 159.36 -70.938   141.88 8.3425  2    0.01543 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# post hoc Estimated marginal means
emmeans_results <- emmeans(lmer_model, pairwise ~ condition)
emmeans_results

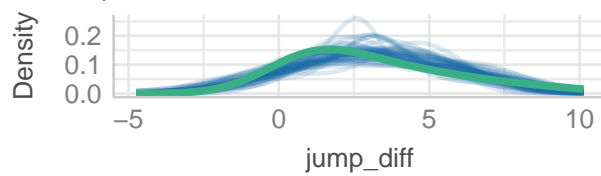
## $emmeans
##   condition emmean    SE   df lower.CL upper.CL
##   static      3.45 0.762 18.4    1.856    5.05
##   dynamic      4.00 0.762 18.4    2.402    5.60
##   control      1.91 0.762 18.4    0.311    3.51
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $contrasts
##   contrast      estimate    SE df t.ratio p.value
##   static - dynamic   -0.545 0.714 20   -0.764  0.7290
##   static - control    1.545 0.714 20    2.164  0.1024
##   dynamic - control    2.091 0.714 20    2.927  0.0217
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates

model_performance <- performance::check_model(lmer_model)
print(model_performance)

```

Posterior Predictive Check

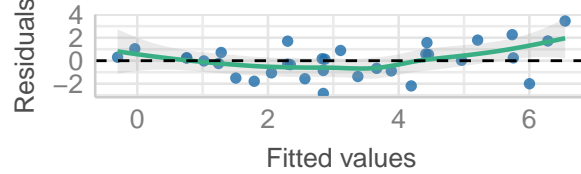
Model-predicted lines should resemble observed data



— Observed data — Model-predicted data

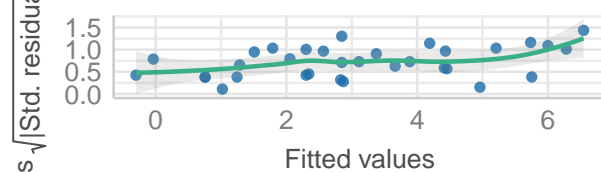
Linearity

Reference line should be flat and horizontal



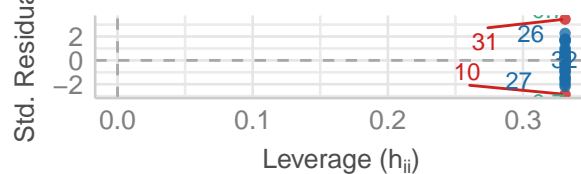
Homogeneity of Variance

Reference line should be flat and horizontal



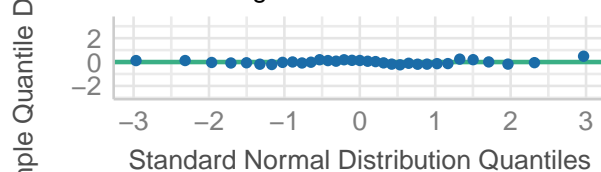
Influential Observations

Points should be inside the contour lines



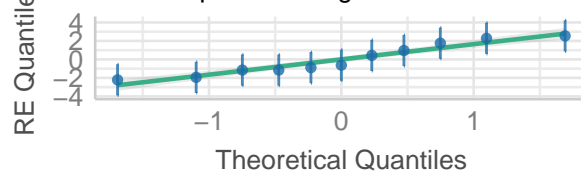
Normality of Residuals

Points should fall along the line



Normality of Random Effects (sub_id)

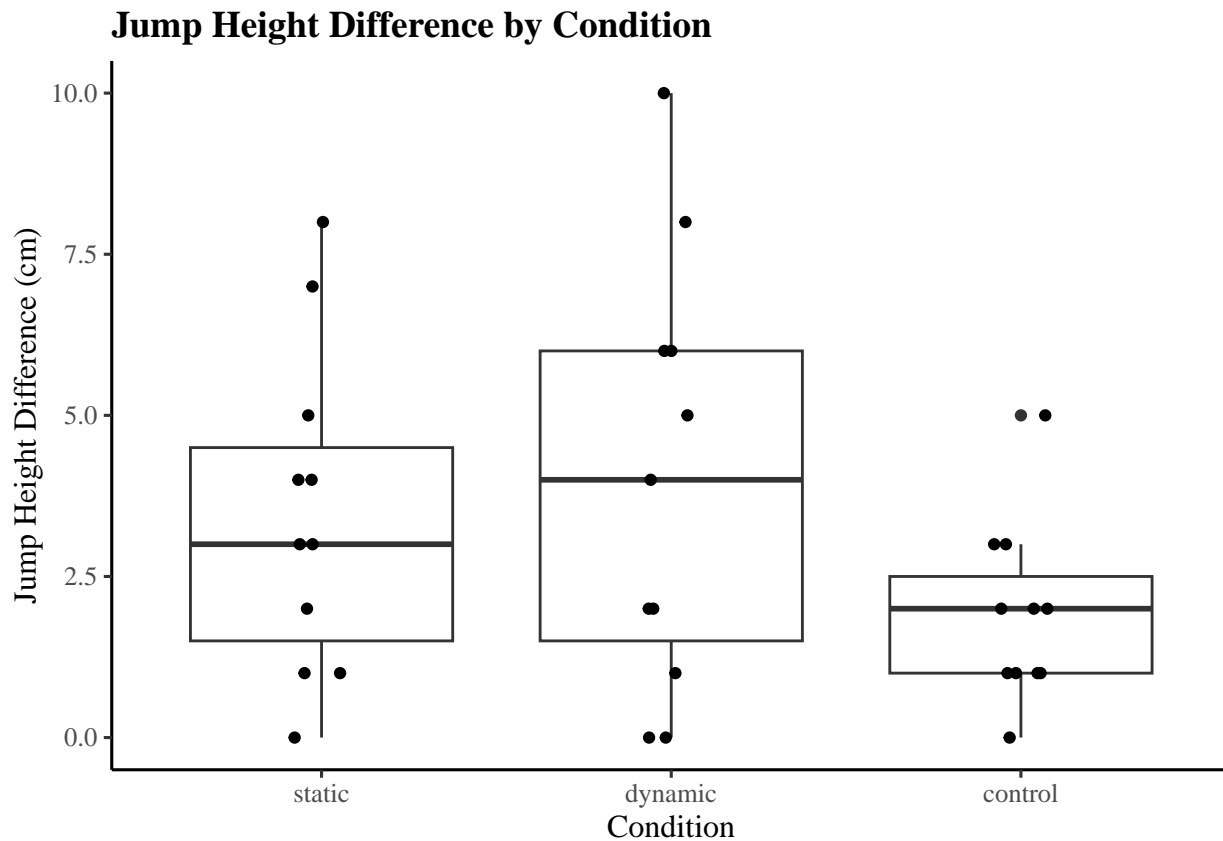
Points should be plotted along the line



```
# emmeans_results <- emmeans(lmer_model, ~ condition)
# pairwise_comparisons <- pairs(emmeans_results)
# pairwise_comparisons

# Box plot
# stat_compare_means(jump_diff ~ condition, data = data, method = "t.test")

ggplot(data, aes(x = condition, y = jump_diff, fill = condition)) +
  geom_boxplot() +
  scale_fill_manual(values = c("white", "white", "white")) +
  geom_point(position = position_jitterdodge()) +
  labs(title = "Jump Height Difference by Condition",
       x = "Condition",
       y = "Jump Height Difference (cm)") +
  theme_bw(base_size = 12, base_family = "Times") +
  theme(plot.title = element_text(size = 14, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10),
        panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(color = "black")) +
  guides(fill = "none")
```

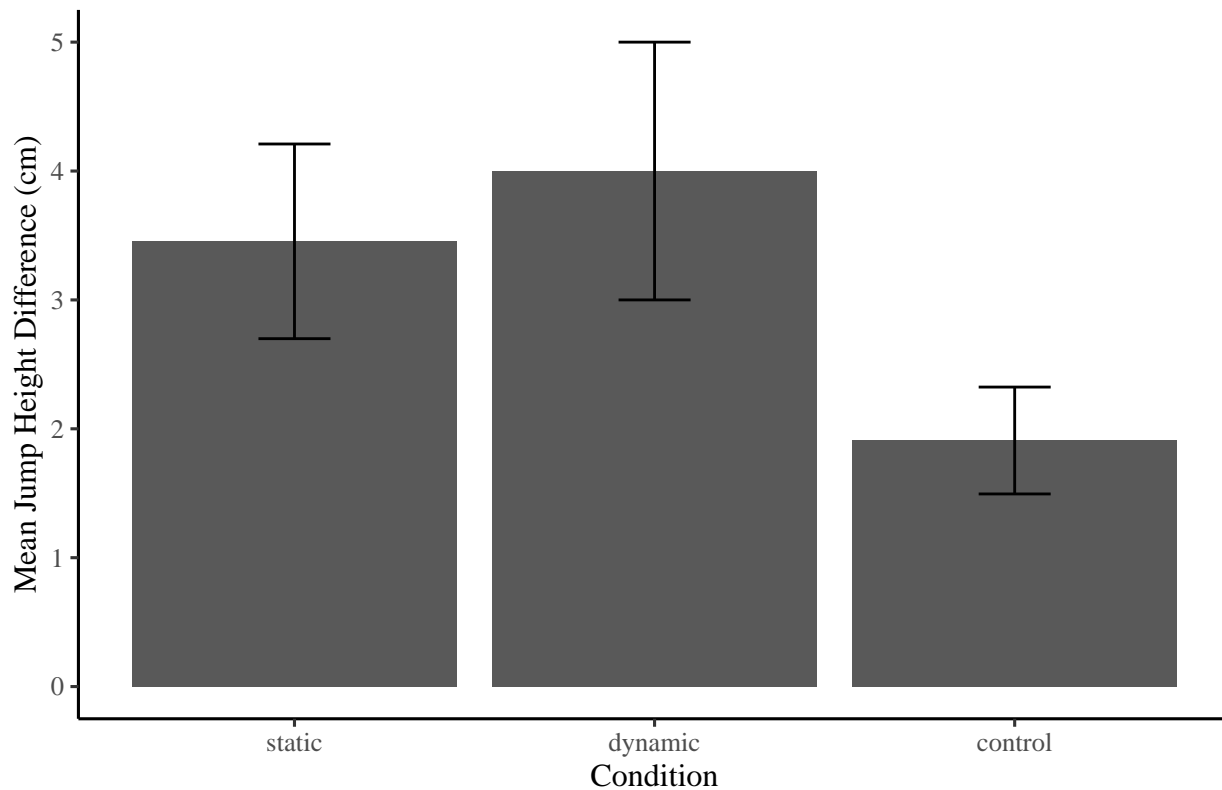
```
ggsave("boxplot.pdf", width = 6, height = 4, units = "in", dpi = 300)
```

```
# Bar chart with error bars
```

```
summary_data <- data %>%
  group_by(condition) %>%
  summarise(mean_diff = mean(jump_diff),
            se_diff = sd(jump_diff) / sqrt(n()))
```

```
ggplot(summary_data, aes(x = condition, y = mean_diff)) +
  geom_bar(stat = "identity") +
  geom_errorbar(aes(
    ymin = mean_diff - se_diff,
    ymax = mean_diff + se_diff), width = 0.2) +
  labs(title = "Bar Chart with Error Bars of Jump Height Differences",
       x = "Condition",
       y = "Mean Jump Height Difference (cm)") +
  theme_bw(base_size = 12, base_family = "Times") +
  theme(plot.title = element_text(size = 14, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10),
        panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(color = "black"))
```

Bar Chart with Error Bars of Jump Height Differences



```
ggsave("barchart.pdf", width = 6, height = 4, units = "in", dpi = 300)
```

```
interaction_data <- data %>%
  gather(key = "time",
         value = "jump_height", pre_stretch, post_stretch)
```

```
interaction_data
```

```
## # A tibble: 66 x 8
##   sub_id age sex group condition jump_diff time jump_height
##   <int> <int> <chr> <chr> <fct>      <int> <chr>      <int>
## 1     1     29 f   a   dynamic      1 pre_stretch      22
## 2     1     29 f   a   static        0 pre_stretch      24
## 3     1     29 f   a   control       1 pre_stretch      21
## 4     2     37 m   b   dynamic      4 pre_stretch      38
## 5     2     37 m   b   static        1 pre_stretch      36
## 6     2     37 m   b   control       1 pre_stretch      39
## 7     3     55 f   c   dynamic      0 pre_stretch      20
## 8     3     55 f   c   static        1 pre_stretch      19
## 9     3     55 f   c   control       0 pre_stretch      21
## 10    4     32 f   a   dynamic      0 pre_stretch      38
## # i 56 more rows
```

```
summary(interaction_data)
```

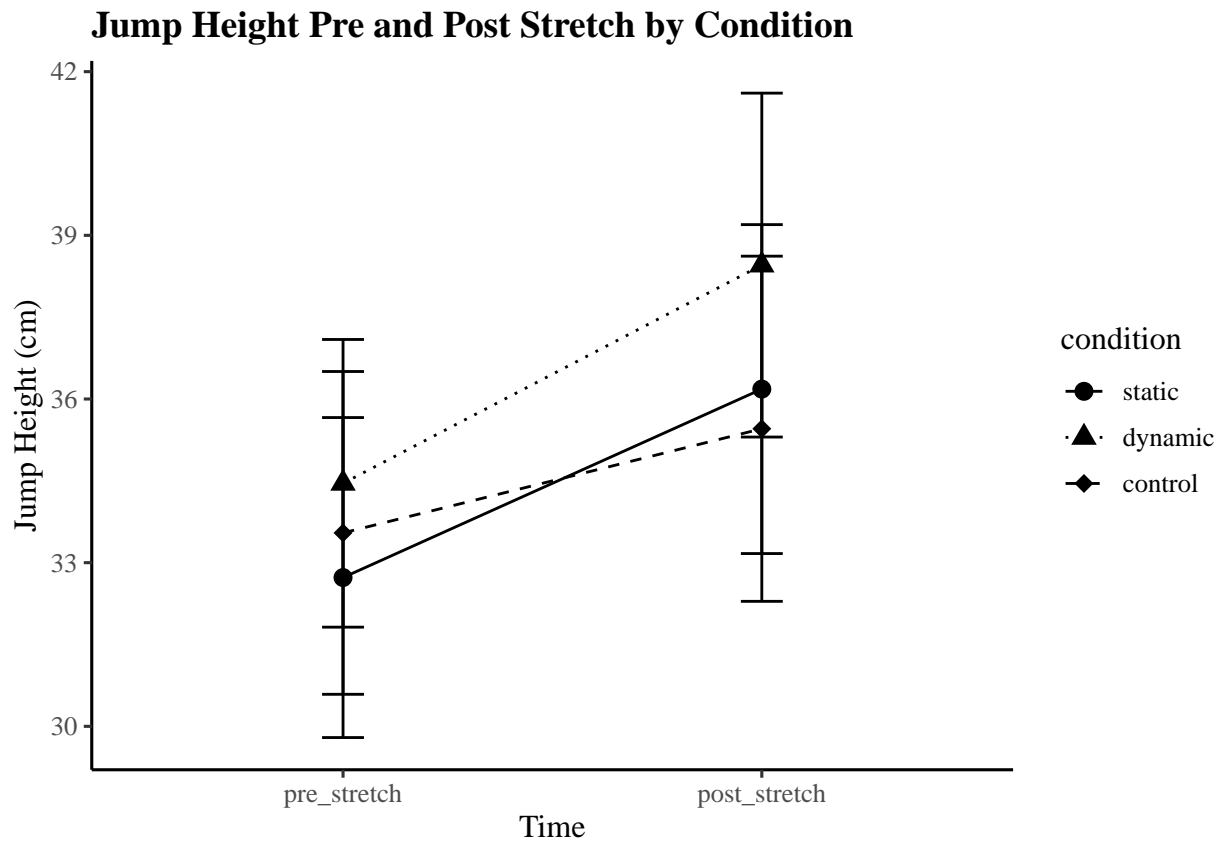
```
##      sub_id      age      sex      group      condition
## Min.   : 1   Min.   :21   Length:66   Length:66   static :22
## 1st Qu.: 3   1st Qu.:28   Class :character   Class :character   dynamic:22
```

```
## Median : 6   Median :30   Mode :character   Mode :character   control:22
## Mean   : 6   Mean   :34
## 3rd Qu.: 9   3rd Qu.:42
## Max.   :11   Max.   :55
##      jump_diff      time      jump_height
## Min.    : 0.000   Length:66   Min.      :19.00
## 1st Qu.: 1.000   Class :character 1st Qu.:26.50
## Median : 2.000   Mode  :character Median :36.00
## Mean    : 3.121                      Mean    :35.14
## 3rd Qu.: 5.000                      3rd Qu.:40.00
## Max.    :10.000                     Max.    :55.00
```

```
# Interaction plot
```

```
interaction_data$time <- factor(interaction_data$time,
                                levels = c("pre_stretch", "post_stretch"))

ggplot(interaction_data, aes(
  x = time, y = jump_height, color = condition, group = condition)) +
  geom_line(aes(
    linetype = condition),
    stat = "summary", fun = mean, color = "black") +
  geom_point(aes(
    shape = condition),
    stat = "summary", fun = mean, size = 3, color = "black") +
  geom_errorbar(
    stat = "summary", fun.data = mean_se, width = 0.1, color = "black") +
  scale_linetype_manual(values = c("solid", "dotted", "dashed")) +
  scale_shape_manual(values = c(16, 17, 18)) +
  labs(title = "Jump Height Pre and Post Stretch by Condition",
       x = "Time",
       y = "Jump Height (cm)") +
  theme_bw(base_size = 12, base_family = "Times") +
  theme(plot.title = element_text(size = 14, face = "bold"),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10),
        panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(color = "black")
  )
```



```
ggsave("interaction.pdf", width = 6, height = 4, units = "in", dpi = 300)
```

```
# Define the means and standard deviations for each condition
row_before_means <- rowMeans(data_wide[, c("dynamic_pre", "static_pre",
                                             "control_pre")])
```

```
mean_before <- mean(row_before_means)
mean_before
```

```
## [1] 33.57576
```

```
mean_after_static <- mean(data_wide$static_post)
mean_after_static
```

```
## [1] 36.18182
```

```
mean_after_dynamic <- mean(data_wide$dynamic_post)
mean_after_dynamic
```

```
## [1] 38.45455
```

```
mean_after_rest <- mean(data_wide$control_post)
mean_after_rest
```

```
## [1] 35.45455
```

```
combined_scores <- c(data_wide$static_post,
                      data_wide$dynamic_post, data_wide$control_post)
sd_before <- sd(combined_scores)
sd_before
```

```
## [1] 10.07312
```

```

sd_after_static <- sd(data_wide$control_post)
sd_after_static

## [1] 10.49156

sd_after_dynamic <- sd(data_wide$dynamic_post)
sd_after_dynamic

## [1] 10.45336

sd_after_rest <- sd(data_wide$control_post)
sd_after_rest

## [1] 10.49156

# Calculate the effect sizes
effect_size_static <- (mean_before - mean_after_static) / sd_before
effect_size_dynamic <- (mean_before - mean_after_dynamic) / sd_before
effect_size_rest <- (mean_before - mean_after_rest) / sd_before

# Calculate the power for each condition
power_static <- pwr.t.test(d = effect_size_static,
                           n = n, type = "paired",
                           alternative = "two.sided")$power
power_dynamic <- pwr.t.test(d = effect_size_dynamic,
                             n = n, type = "paired",
                             alternative = "two.sided")$power
power_rest <- pwr.t.test(d = effect_size_rest,
                          n = n, type = "paired",
                          alternative = "two.sided")$power

# Print the results
cat("Power for static stretching:", power_static, "\n")

## Power for static stretching: 0.121735
cat("Power for dynamic stretching:", power_dynamic, "\n")

## Power for dynamic stretching: 0.3066805
cat("Power for rest:", power_rest, "\n")

## Power for rest: 0.08680512

# Define the effect size
row_after_means <- rowMeans(data_wide[, c("dynamic_post",
                                           "static_post", "control_post")])
mean_after <- mean(row_after_means)

# Calculate the effect size (Cohen's d)
effect_size <- (mean_before - mean_after) / sd_before
effect_size

## [1] -0.3098555

# Calculate the required sample size for a power of 0.8
required_sample_size <- pwr.t.test(d = effect_size,
                                   power = 0.8, type = "paired",
                                   alternative = "two.sided")$n

```

```
# Print the result
cat("Minimum number of participants needed:",
    ceiling(required_sample_size), "\n")

## Minimum number of participants needed: 84
write.csv(data_wide, "data_wide_output.csv")
write.csv(data, "data_output.csv")
```