Hands-on Workshop: Exploring TMS Data

Welcome

This short tutorial will guide you through exploring Transcranial Magnetic Stimulation (TMS) study data using R. You don't need any prior coding or statistics experience, we'll go step by step.

Setup

Before we start, we need a few tools (called packages). These are small add-ons that make R more powerful.

Step 1 — Install the required packages

Run this code once on your computer to install what we'll use today.

```
# Install the required packages if not already installed
install.packages("dplyr")
install.packages("ggplot2")
install.packages("tableone")
install.packages("lme4")
```

Step 2 — Load the packages

You'll need to load these every time you open a new R session.

```
# Load the required packages
library(dplyr)
library(ggplot2)
library(tableone)
library(lme4)
```

Load and peek at the data

Let's start by reading our dataset into R.

What does this mean?

- `read.csv()` tells R to open a spreadsheet file that's in CSV format.
- The result is stored in something called a data frame, a bit like an Excel sheet.

```
# Path to your CSV file
path <- "data.csv"

# Read the CSV
data <- read.csv(path)
```

Take a quick look at the first few rows using `head()`, so you can confirm the data looks right.

```
# Peek at the first few rows
head(data)
```

```
##   subject_id age gender medication baseline_severity coil_intensity treatment
## 1          1  33 female          0          14.27806       74.30402    active
## 2          1  33 female          0          14.27806       74.30402    active
## 3          1  33 female          0          14.27806       74.30402    active
## 4          2  37 female          0          13.97849       70.46629    active
## 5          2  37 female          0          13.97849       70.46629    active
## 6          2  37 female          0          13.97849       70.46629    active
##   site session_count MEP_amplitude  EEG_theta  EEG_alpha  EEG_beta dropout_flag
## 1    B            28      2.167524 -0.5158105  0.8567855  0.628965            0
## 2    B            28      2.167524 -0.5158105  0.8567855  0.628965            0
## 3    B            28      2.167524 -0.5158105  0.8567855  0.628965            0
## 4    C            20      1.681755  0.4457241 -1.1037521 -1.262195            0
## 5    C            20      1.681755  0.4457241 -1.1037521 -1.262195            0
## 6    C            20      1.681755  0.4457241 -1.1037521 -1.262195            0
##       time site_effect rand_intercept delta effect_age effect_med
## 1      pre   0.5252107      -1.781594     0      -1.65          0
## 2     post   0.5252107      -1.781594    -6      -1.65          0
## 3 followup   0.5252107      -1.781594    -5      -1.65          0
## 4      pre  -0.3631982      -5.131139     0      -1.85          0
## 5     post  -0.3631982      -5.131139    -6      -1.85          0
## 6 followup  -0.3631982      -5.131139    -5      -1.85          0
##   effect_sessions effect_mep effect_eeg_theta      noise symptom_score
## 1          0.0000   0.000000        0.0000000 -3.3620355      8.009644
## 2         -0.6112  -2.167524        0.5158105  1.3835774      4.492344
## 3         -0.6112  -2.167524        0.5158105  4.5724284      8.681195
## 4          0.0000   0.000000        0.0000000  1.3034049      7.937556
## 5          0.0288  -1.681755       -0.4457241  0.5760011     -0.888527
## 6          0.0288  -1.681755       -0.4457241 -1.9687294     -2.433257
##   symptom_baseline delta_from_baseline
## 1         8.009644           0.0000000
## 2         8.009644           3.5173002
## 3         8.009644          -0.6715508
## 4         7.937556           0.0000000
## 5         7.937556           8.8260830
## 6         7.937556          10.3708134
```

Then, see the structure (column names and data types) using `str()`. This is useful to check that, for example, numbers are numeric and not text. This command shows:

- Each column name
- The type of data (numbers, text, etc.)
- A few example values from each column

Think of this like "What kind of information do I have in each column?"

```
str(data)
```

```
## 'data.frame':    450 obs. of  27 variables:
##  $ subject_id         : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ age                : int  33 33 33 37 37 37 59 59 59 41 ...
##  $ gender             : chr  "female" "female" "female" "female" ...
##  $ medication         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ baseline_severity  : num  14.3 14.3 14.3 14 14 ...
##  $ coil_intensity     : num  74.3 74.3 74.3 70.5 70.5 ...
##  $ treatment          : chr  "active" "active" "active" "active" ...
##  $ site               : chr  "B" "B" "B" "C" ...
##  $ session_count      : int  28 28 28 20 20 20 19 19 19 22 ...
##  $ MEP_amplitude      : num  2.17 2.17 2.17 1.68 1.68 ...
##  $ EEG_theta          : num  -0.516 -0.516 -0.516 0.446 0.446 ...
##  $ EEG_alpha          : num  0.857 0.857 0.857 -1.104 -1.104 ...
##  $ EEG_beta           : num  0.629 0.629 0.629 -1.262 -1.262 ...
##  $ dropout_flag       : int  0 0 0 0 0 0 1 1 1 0 ...
##  $ time               : chr  "pre" "post" "followup" "pre" ...
##  $ site_effect        : num  0.525 0.525 0.525 -0.363 -0.363 ...
##  $ rand_intercept     : num  -1.78 -1.78 -1.78 -5.13 -5.13 ...
##  $ delta              : int  0 -6 -5 0 -6 -5 0 -2 -1 0 ...
##  $ effect_age         : num  -1.65 -1.65 -1.65 -1.85 -1.85 -1.85 -2.95 -2.95 -2.95 -2.05 ...
##  $ effect_med         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ effect_sessions    : num  0 -0.6112 -0.6112 0 0.0288 ...
##  $ effect_mep         : num  0 -2.17 -2.17 0 -1.68 ...
##  $ effect_eeg_theta   : num  0 0.516 0.516 0 -0.446 ...
##  $ noise              : num  -3.362 1.384 4.572 1.303 0.576 ...
##  $ symptom_score      : num  8.01 4.492 8.681 7.938 -0.889 ...
##  $ symptom_baseline   : num  8.01 8.01 8.01 7.94 7.94 ...
##  $ delta_from_baseline: num  0 3.517 -0.672 0 8.826 ...
```

Finally, `summary()` provides summary statistics for each column. This gives quick numerical summaries:

- For numbers: minimum, mean, median, maximum
- For text or categories: how many of each type

At this point, you should have a good sense of what's inside your dataset.

```
# Quick summary of each column
summary(data)
```

```
##    subject_id         age           gender            medication
##  Min.   :  1.0   Min.   :12.00   Length:450         Min.   :0.0
##  1st Qu.: 38.0   1st Qu.:32.00   Class :character   1st Qu.:0.0
##  Median : 75.5   Median :39.00   Mode  :character   Median :0.0
##  Mean   : 75.5   Mean   :39.67                      Mean   :0.4
##  3rd Qu.:113.0   3rd Qu.:47.00                      3rd Qu.:1.0
##  Max.   :150.0   Max.   :66.00                      Max.   :1.0
##
##  baseline_severity coil_intensity   treatment            site
##  Min.   :-1.145    Min.   :31.90   Length:450         Length:450
##  1st Qu.:14.278    1st Qu.:54.43   Class :character   Class :character
##  Median :20.074    Median :60.90   Mode  :character   Mode  :character
##  Mean   :19.632    Mean   :60.64
##  3rd Qu.:25.279    3rd Qu.:67.41
```

```
##  Max.   :40.572    Max.   :84.30
##
##  session_count   MEP_amplitude        EEG_theta          EEG_alpha
##  Min.   :10.00   Min.   :-0.04048   Min.   :-2.38319   Min.   :-2.313736
##  1st Qu.:18.00   1st Qu.: 0.84058   1st Qu.:-0.76843   1st Qu.:-0.701804
##  Median :21.00   Median : 1.24759   Median :-0.11979   Median :-0.005609
##  Mean   :20.36   Mean   : 1.24166   Mean   :-0.07554   Mean   : 0.019734
##  3rd Qu.:23.00   3rd Qu.: 1.60023   3rd Qu.: 0.55262   3rd Qu.: 0.681830
##  Max.   :31.00   Max.   : 2.61846   Max.   : 3.28690   Max.   : 3.390371
##
##     EEG_beta         dropout_flag      time            site_effect
##  Min.   :-2.69533   Min.   :0.00   Length:450         Min.   :-0.3632
##  1st Qu.:-0.56611   1st Qu.:0.00   Class :character   1st Qu.:-0.3632
##  Median : 0.24571   Median :0.00   Mode  :character   Median :-0.2312
##  Mean   : 0.07721   Mean   :0.26                      Mean   :-0.0450
##  3rd Qu.: 0.72468   3rd Qu.:1.00                      3rd Qu.: 0.5252
##  Max.   : 2.28412   Max.   :1.00                      Max.   : 0.5252
##
##  rand_intercept         delta          effect_age       effect_med
##  Min.   :-9.14358   Min.   :-6.000   Min.   :-3.300   Min.   :0.0
##  1st Qu.:-1.78159   1st Qu.:-5.000   1st Qu.:-2.350   1st Qu.:0.0
##  Median :-0.06264   Median :-2.000   Median :-1.950   Median :0.0
##  Mean   : 0.16717   Mean   :-2.404   Mean   :-1.983   Mean   :0.6
##  3rd Qu.: 2.70863   3rd Qu.: 0.000   3rd Qu.:-1.600   3rd Qu.:1.5
##  Max.   : 9.87155   Max.   : 0.000   Max.   :-0.600   Max.   :1.5
##
##  effect_sessions     effect_mep      effect_eeg_theta       noise
##  Min.   :-0.8512   Min.   :-2.5176   Min.   :-3.28690   Min.   :-7.88798
##  1st Qu.:-0.0512   1st Qu.:-0.8600   1st Qu.: 0.00000   1st Qu.:-1.95443
##  Median : 0.0000   Median : 0.0000   Median : 0.00000   Median :-0.10225
##  Mean   : 0.0000   Mean   :-0.4472   Mean   : 0.06823   Mean   :-0.04063
##  3rd Qu.: 0.1088   3rd Qu.: 0.0000   3rd Qu.: 0.00000   3rd Qu.: 1.90796
##  Max.   : 0.8288   Max.   : 0.0000   Max.   : 2.23456   Max.   : 8.44825
##
##  symptom_score    symptom_baseline delta_from_baseline
##  Min.   :-9.751   Min.   :-3.36    Min.   :-8.4326
##  1st Qu.: 8.612   1st Qu.:12.00    1st Qu.: 0.0000
##  Median :15.300   Median :18.98    Median : 0.3445
##  Mean   :15.289   Mean   :18.67    Mean   : 3.0684
##  3rd Qu.:22.225   3rd Qu.:25.14    3rd Qu.: 6.0255
##  Max.   :41.800   Max.   :41.80    Max.   :21.1646
##  NA's   :39                        NA's   :39
```

Exploratory analysis

Now the fun part, visualizing what's happening in the data.

Before plotting, let's make sure the `time` variable appears in the correct order (pre > post > follow-up):

```
data <- data %>%
  mutate(time = factor(time, levels = c("pre", "post", "followup")))
```

Plot 1 - Symptom change over time by treatment

In TMS research, we often want to know: Do patients receiving active TMS improve more than those in the sham (placebo) group?

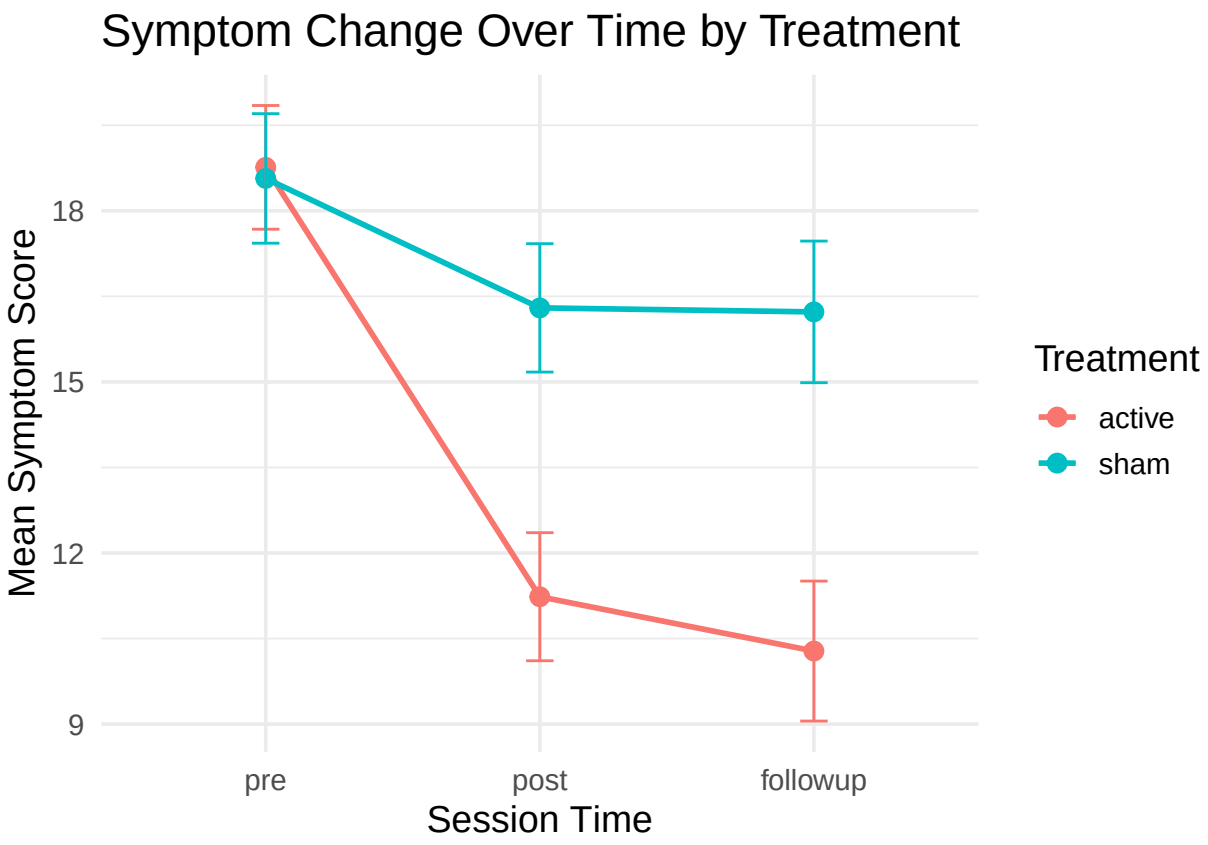
This plot shows average symptom scores at each stage of the study:

- **Pre**: before any treatment
- **Post**: immediately after treatment
- **Follow-up**: some time later

We'll add:

- **Dots** for mean symptom scores
- **Lines** connecting them over time
- **Error bars** showing uncertainty (standard error)

```
ggplot(data, aes(x = time, y = symptom_score, color = treatment, group = treatment)) +
  stat_summary(fun = mean, geom = "line", linewidth = 1) +
  stat_summary(fun = mean, geom = "point", size = 3) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.1) +
  labs(
    title = "Symptom Change Over Time by Treatment",
    x = "Session Time",
    y = "Mean Symptom Score",
    color = "Treatment"
  ) +
  theme_minimal(base_size = 14)
```



How to read this

- The y-axis shows average symptom severity.
- The x-axis shows time: pre → post → follow-up.
- Two lines: one for each treatment type (active vs sham).
- If the active line goes down more steeply, patients improved more with active TMS.
- If both lines look similar, the sham effect may be similar (placebo or no effect).

Clinically speaking

This is your first visual check: is there an apparent treatment benefit? Don't jump to conclusions yet, but you'll see whether the trend looks promising.

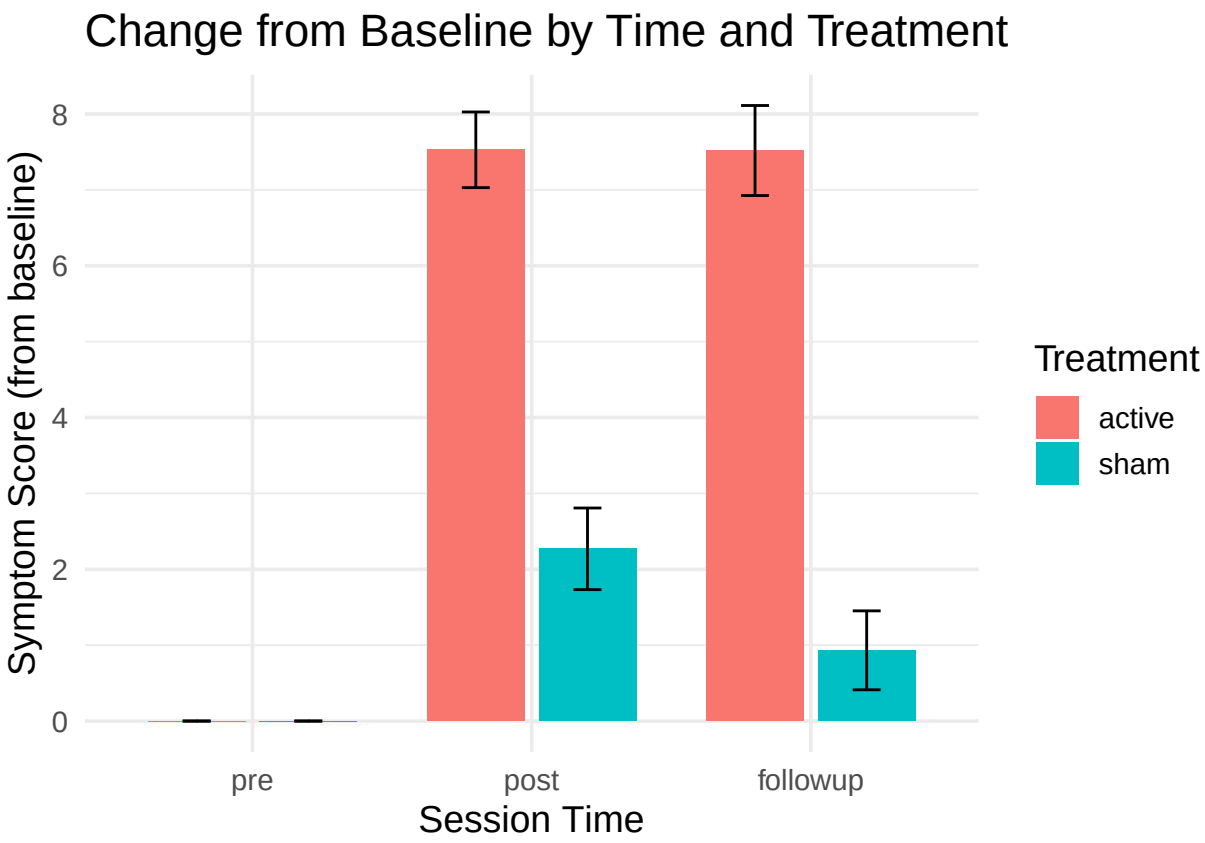
Plot 2 - Change from baseline

Absolute scores are useful, but clinicians often care more about change from baseline: How much did the patient improve compared to where they started?

In the data, this is already computed in the column `delta_from_baseline` (baseline - follow-up). A high value means improvement (symptoms decreased).

We'll plot these changes for each group and time point.

```
ggplot(data, aes(x = time, y = delta_from_baseline, fill = treatment)) +
  stat_summary(fun = mean, geom = "bar", position = position_dodge(0.8), width = 0.7) +
  stat_summary(fun.data = mean_se, geom = "errorbar",
               position = position_dodge(0.8), width = 0.2) +
  labs(
    title = "Change from Baseline by Time and Treatment",
    x = "Session Time",
    y = "Symptom Score (from baseline)",
    fill = "Treatment"
  ) +
  theme_minimal(base_size = 14)
```



How to read this

- Each bar shows the average change from baseline.
- High values indicate improvement.
- Compare active vs sham at post and follow-up.
- Look at the height difference: larger drops mean more improvement.

Clinically speaking

This plot gives a quick sense of treatment effect size, how much better the active group got relative to the sham group.

Your turn

Now that you've made two fundamental plots, try adding or changing things to dig deeper. Below are a few extra plots to explore, each demonstrating a different concept.

- Check whether brain response (MEP amplitude) is related to symptom improvement.
- Compare EEG signals between groups.

Table one

A common clinical research step is to create a Table One summarizing baseline characteristics of your groups. This helps ensure groups are comparable before treatment. Let's create one using the `tableone` package.

```r
# Define variables for Table One
vars <- c("age", "gender", "symptom_score", "delta_from_baseline")
catVars <- c("gender", "comorbidities") # Categorical variables
strataVar <- "treatment" # Grouping variable

# Create Table One
tableOne <- CreateTableOne(vars = vars, data = data, factorVars = catVars, strata = strataVar)
print(tableOne)
```

```
##                                Stratified by treatment
##                                 active        sham          p      test
##   n                               237           213
##   age (mean (SD))               40.13 (11.97) 39.15 (10.67)  0.366
##   gender = male (%)               126 (53.2)    129 (60.6)   0.137
##   symptom_score (mean (SD))     13.68 (10.44) 17.12 (9.35)   0.001
##   delta_from_baseline (mean (SD))  4.81 (5.10)   1.08 (3.46)  <0.001
```

Now, it is your turn! Modify the variables in `vars` and `catVars` to include other baseline characteristics relevant to your study. Run the code to see how the groups compare before treatment.

Regression model

Simple linear regression

First, we'll fit a simple linear regression model to see how treatment affects symptom scores at follow-up, controlling for baseline scores.

```r
data_followup <- data %>% filter(time == "followup")
model <- lm(symptom_score ~ treatment + symptom_baseline, data = data_followup)
summary(model)
```

```
##
## Call:
## lm(formula = symptom_score ~ treatment + symptom_baseline, data = data_followup)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9312  -2.8261   0.1856   2.9218   8.0640
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -5.5539     0.9226  -6.020 2.44e-08 ***
## treatmentsham      6.5159     0.7873   8.277 3.66e-13 ***
## symptom_baseline   0.8896     0.0425  20.932  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.124 on 108 degrees of freedom
##   (39 observations deleted due to missingness)
## Multiple R-squared:  0.821,  Adjusted R-squared:  0.8177
## F-statistic: 247.6 on 2 and 108 DF,  p-value: < 2.2e-16
```

Now, also add more predictors like age, gender and more to see if they influence outcomes.

Mixed effects model

In this section, we'll fit a mixed effects model to account for repeated measures over time within subjects, which is common in clinical trials where we have multiple observations per patient (at pre, post, and follow-up).

```
model_mixed <- lmer(symptom_score ~ treatment * time + (1 | subject_id), data = data)
summary(model_mixed)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: symptom_score ~ treatment * time + (1 | subject_id)
##    Data: data
##
## REML criterion at convergence: 2567.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.96817 -0.58003 -0.03979  0.59277  2.36810
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  subject_id (Intercept) 82.726   9.095
##  Residual                9.786   3.128
## Number of obs: 411, groups:  subject_id, 150
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                  18.7618     1.0821  17.338
## treatmentsham                -0.1942     1.5729  -0.123
## timepost                     -7.5280     0.4977 -15.124
## timefollowup                 -7.5994     0.5491 -13.839
## treatmentsham:timepost        5.2583     0.7235   7.268
## treatmentsham:timefollowup    6.6932     0.8139   8.224
##
## Correlation of Fixed Effects:
##               (Intr) trtmnt timpst tmfllw trtmntshm:tmp
## treatmntshm   -0.688
## timepost      -0.230  0.158
## timefollowp   -0.208  0.143  0.453
## trtmntshm:tmp  0.158 -0.230 -0.688 -0.312
## trtmntshm:tmf  0.141 -0.204 -0.306 -0.675  0.444
```