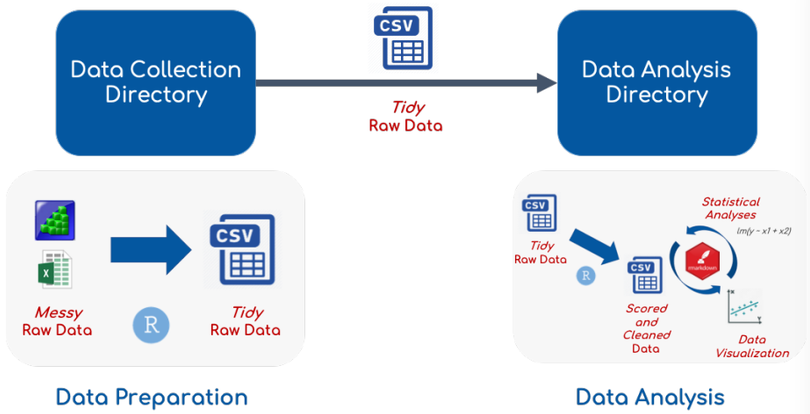
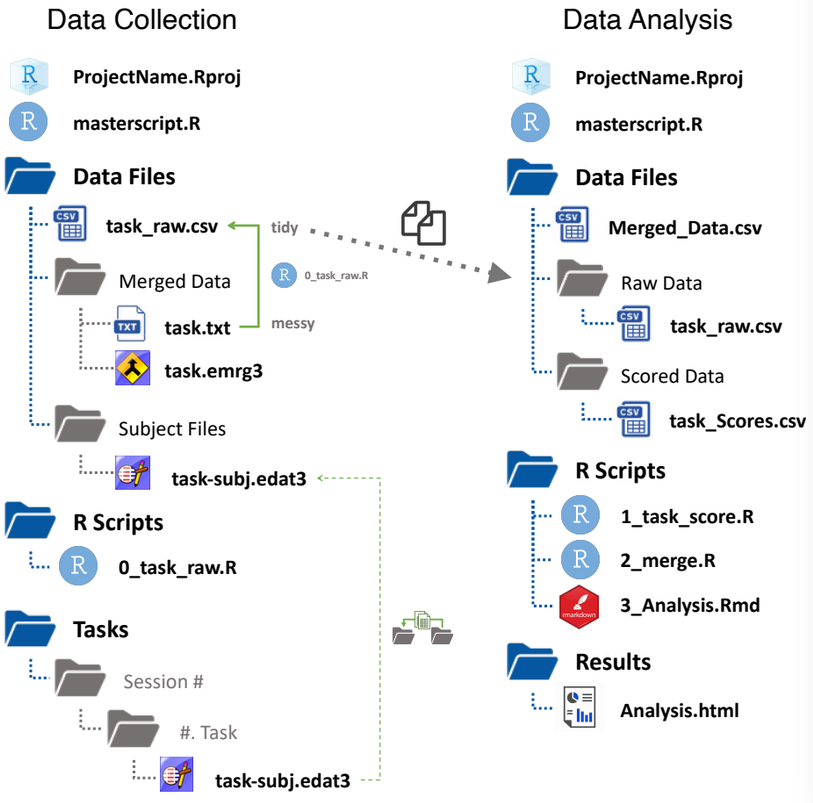
EngleLab : : useRguide



Custom Packages to Install

```
devtools::install_github("EngleLab/englelab")
devtools::install_github("dr-JT/workflow")
devtools::install_github("dr-JT/datawrangling")
devtools::install_github("dr-JT/semoutput")
```

Project Organization



Data Preparation

Create Data Collection Directory

1. Navigate to **File** -> **New Project...** -> **New Directory**
2. Select the template: **Research Study**
3. Change Repository Type: **data collection**
4. Specify the **# of Sessions**

```
workflow::template(rawscript = TRUE)

englelab::get_script(type = "raw",
            wmc = TRUE, gf = TRUE,
            antisaccade = TRUE,
            visualarrays = TRUE,
            sact = TRUE, flankerDL = TRUE,
            stroopDL = TRUE)
```

Common Functions: messy to tidy

Filter on real and practice trials only

```
rename(TrialProc = `Procedure[Trial]`)
filter(TrialProc == "TrialProc" | TrialProc == "PracProc")
```

Change labels of values in columns

```
mutate(TrialProc = case_when(TrialProc == "TrialProc" ~ "real",
                             TrialProc == "PracProc" ~ "practice",
                             TRUE ~ as.character(NA)))
```

Select value at first or last occurrence in column

```
group_by(Subject)
mutate(AdminTime = last(AdminTime)) # Use first() for first occurrence
```

Select a subset of columns

```
select(Subject, TrialProc, Trial, Condition, Accuracy, RT, Response,
       CorrectResponse, AdminTime, SessionDate, SessionTime)
```

Data Analysis

Create Data Analysis Directory

1. Navigate to **File** -> **New Project...** -> **New Directory**
2. Select the template: **Research Study**
3. Change Repository Type: **data analysis**

```
workflow::template(scorescript = TRUE)
workflow::template(mergescript = TRUE)

englelab::get_script(type = "score",
            wmc = TRUE, gf = TRUE,
            antisaccade = TRUE,
            visualarrays = TRUE,
            sact = TRUE, flankerDL = TRUE,
            stroopDL = TRUE)
```

library(datawrangling)

Import multiple files and merge

```
files_join(path = import_dir,
           pattern = "Scores",
           id = "Subject",)
```

Create composite factor

```
composite(variables = c("Score1",
                        "Score2",
                        "Score3"),
          name = "comp_name",
          type = c("mean", "sum", "fa"),
          standardize = c(TRUE, FALSE),
          missing.allowed = 1)
```

Common Functions: tidy to scored

Calculate aggregate score

```
group_by(Subject, Condition)
summarise(Accuracy.mean = mean(Accuracy, na.rm = TRUE))
```

Transform data frame to wider format - spread rows across columns

```
pivot_wider(id_cols = "Subject",
            names_from = "Condition",
            values_from = "Accuracy.mean")
```

Remove problematic subjects

```
acc_criteria <- -3.5

data_remove <- data_scores %>%
mutate(Accuracy.mean_z = scale(Accuracy.mean, center = TRUE, scale = TRUE) %>%
filter(Accuracy.mean_z <= acc_criteria)

data_scores <- filter(data_scores, !(Subject %in% data_remove$Subject))
```

Remove outliers

```
outlier_criteria <- 3.0

data_outliers <- data_scores %>%
mutate(Task_Score_z = scale(Task_Score, center = TRUE, scale = TRUE) %>%
filter(Task_Score_z >= outlier_criteria | Task_Score_z <= -1*outlier_criteria)

data_scores <- filter(data_scores, !(Subject %in% data_outliers$Subject))
```

Calculate split-half reliability

```
group_by(Subject, Condition)
mutate(Index = row_number(), Split = ifelse(Index %% 2, "odd", "even"))
group_by(Subject, Condition, Split)
. calculate task score here
summarise(r = cor(Task_Score_even, Task_Score_odd))
mutate(r = (2 * r) / (1 + r))
```