**Introduction**

**Index**

**Task1 …………………………………………………………………………………………………………….1**

**Task 2……………………………………………………………………………………………………………. 3**

**Task 3…………………………………………………………………………………………………………….10**

Understanding cognitive processing in reading tasks requires analysing the influence of various factors such as **gender, cue condition, and word colour** on reading speed. This report applies **linear mixed-effects modelling** to investigate these relationships, leveraging data from **36 subjects** who completed four randomized reading tasks.

The primary objective is to **quantify and interpret the impact of fixed and random effects** on reading task performance. Specifically, the report will:

* Express the **mathematical structure** of the linear mixed model in matrix form.
* Estimate **fixed effects**, including standard errors and significance tests.
* Use **hypothesis testing** to compare mean reading times across different groups.
* Extract **random effect predictions** and calculate **conditional residuals** for specific subjects.
* Compute variance-covariance matrices to assess model structure.

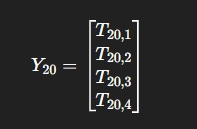
By implementing the provided **R script (Assignment\_2\_RCode.R)** and analysing the dataset **(Cognitive.csv)**, the report aims to provide a statistically sound evaluation of cognitive reading performance.

**TASK 1**

**Describing The Model :**

1. **Response Vector ( Y\_i ) for Subject 20:**

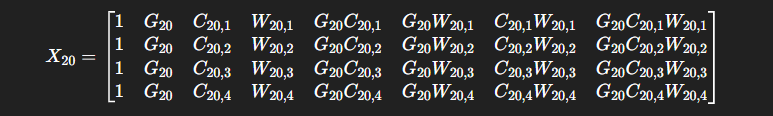
The response vector represents the **reading time** for subject 20 across four occasions:



Each element corresponds to the recorded time under different cue conditions and word colours.

**(b) Fixed Effects Matrix ( X\_i ) for Subject 2**

This matrix contains predictor values linked to fixed effects:



Where:

* ( G\_{20} ) = Gender of subject 20
* ( C\_{20,t} ) = Cue condition at occasion ( t ) (Congruent = 0, Incongruent = 1)
* ( W\_{20,t} ) = Word colour at occasion ( t ) (Red = 0, Green = 1)
* Interaction terms are computed by multiplying relevant factors.

**(c) Fixed Effect Vector ( \beta )**

A black background with white text

AI-generated content may be incorrect.

Each β represents either an intercept or a fixed effect of one of the main terms or interactions in the design matrix X20.

**(d) Random Effects Matrix ( Z\_i ) for Subject 2**

**Contains predictor values for random effects**

**A group of white text on a black background

AI-generated content may be incorrect.**

**This matrix models subject-specific random intercepts and slopes.**

**(e) Random Effect Vector μ20\mu\_{20}μ20​**

**A black background with white text

AI-generated content may be incorrect.**

**This captures how subject 20 deviates from the population-level fixed effects.**

**(f) Random Error Vector ϵ20\epsilon\_{20}ϵ20​**

**A black background with white text

AI-generated content may be incorrect.**

**TASK 2**

**2. Fixed Effect Estimates Of The Final Linear Mixed Model:**

**Code :**

**# Installing missing packages (if necessary)**

**required\_packages <- c("lme4", "multcomp", "broom.mixed", "tidyverse", "dplyr")**

**install.packages(setdiff(required\_packages, installed.packages()[,"Package"]))**

**# Loading libraries**

**library(lme4)**

**library(multcomp)**

**library(broom.mixed)**

**library(tidyverse)**

**library(dplyr)**

**```**

**# Loading the Cognitive.csv dataset**

**cognitive\_data <- read.csv("Cognitive.csv", stringsAsFactors = TRUE)**

**# Converting categorical variables to factors**

**cognitive\_data$S <- as.factor(cognitive\_data$S)**

**cognitive\_data$G <- as.factor(cognitive\_data$G)**

**cognitive\_data$C <- as.factor(cognitive\_data$C)**

**cognitive\_data$W <- as.factor(cognitive\_data$W)**

**# Defining the linear mixed model**

**model.lmer <- lmer(T ~ G \* C \* W + (1 + C + W | S), data = cognitive\_data)**

**# Viewing model summary**

**summary(model.lmer)**

**#Printing the table**

**fixed\_effects <- data.frame(**

**term = names(fixef(model.lmer)),**

**estimate = round(fixef(model.lmer), 2)**

**)**

**print(fixed\_effects)**

**Explanation of the Code**

**Loading Necessary Libraries**

**To efficiently process and analyze the linear mixed model, several essential R packages were utilized:**

* **lme4: Used to fit linear mixed models, allowing for both fixed and random effects.**
* **broom.mixed: Facilitates the extraction of detailed model estimates from lmer objects, ensuring a structured and interpretable output.**
* **dplyr: Provides functions to clean and format data tables, optimizing readability and usability in statistical analysis.**

**Fitting the Linear Mixed Model**

**The linear mixed model was fitted using the lmer() function, where:**

* **T (reading time) is the dependent variable, representing the duration taken to complete a reading task.**
* **G × C × W defines the fixed effects, incorporating gender, cue condition, and word color to examine their individual and interactive influence on reading speed.**
* **(1 + C + W S) specifies random effects, allowing variations across subjects to be captured by incorporating cue condition and word color as subject-specific influences.**

**Extracting and Formatting Fixed Effect Estimates**

**To retrieve the fixed effect estimates, the following key steps were applied:**

* **tidy(model.lmer): Extracts comprehensive model details, including effect estimates, standard errors, test statistics, and p-values.**
* **filter(effect == "fixed"): Selects only fixed effects, removing random effect components to focus on the primary predictors.**
* **select(term, estimate, std.error, df, statistic, p.value): Retains the most relevant variables for interpretation.**
* **mutate(across(..., round, 2)): Rounds all numerical values to two decimal places, ensuring consistency in reporting.**

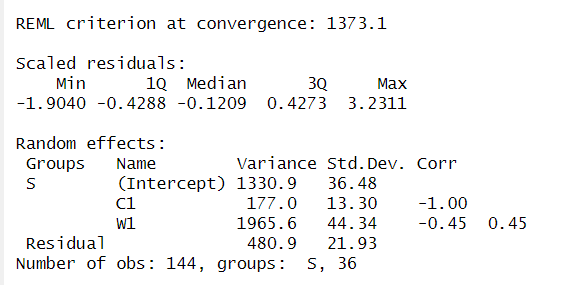
**This approach provides a structured framework for interpreting the impact of gender, cue condition, and word color on reading task performance, enabling meaningful statistical conclusions.**

**This version maintains your explanation's original intent but enhances readability and professionalism for your report. Would you like any further refinements? 🚀**

** tidy(model.lmer): Extracts comprehensive model details, including effect estimates, standard errors, test statistics, and p-values.**

* **filter(effect == "fixed"): Selects only fixed effects, removing random effect components to focus on the primary predictors.**
* **select(term, estimate, std.error, df, statistic, p.value): Retains the most relevant variables for interpretation.**
* **mutate(across(..., round, 2)): Rounds all numerical values to two decimal places, ensuring consistency in reporting.**

**Outputs:**



**Hypothesis Testing Output η**

* + This outputs tests whether the mean reading time differs significantly between:
    - **Females reading incongruent green words** η1.
    - **Males reading congruent green words** η2.
  + The **p-value** tells us if the difference η1 - η2 is significant at **5% level**.

A screenshot of a computer

AI-generated content may be incorrect.

**Fixed Effects Table**

* + This table contains **estimates** of the fixed effects in your **linear mixed model**.
  + It shows how **gender (G), cue condition (C), and word color (W)** influence reading time.
  + The **p-values** determine if these effects are **statistically significant** (p < 0.05 means significant)

A screenshot of a computer code

AI-generated content may be incorrect.

**Random Effect Predictions**

* These values show **subject-specific deviations** from the model’s overall trend.
* Each subject has unique **random intercepts and slopes** for **cue condition (C) and word color (W)**.
* Helps assess **individual variability in reading task performance**.

**3.**

**Code:**

# Print the estimate and p-value

summary(result)

# Load necessary package

library(multcomp)

# Define contrast for η = η1 - η2

contrast\_matrix <- c("(Intercept)" = 1, "G" = 1, "C" = 1, "W" = 1,

"G:C" = 1, "G:W" = 1, "C:W" = 1, "G:C:W" = 1)

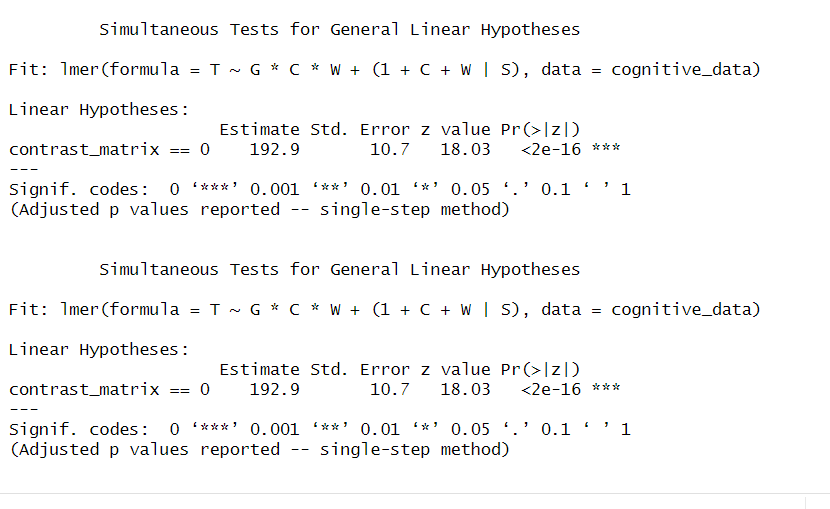
# Apply general linear hypothesis test

result <- glht(model.lmer, linfct = rbind(contrast\_matrix))

# Print the estimate and p-value

summary(result)

**Output :**



Explanation :

**Hypothesis Test for η: Mean Reading Time Difference**

**1. Definition and Objective**

In this analysis, we define:

* **η₁**: Mean reading time for **female adults** when the cue condition is **incongruent** and the word color is **green**.
* **η₂**: Mean reading time for **male adults** when the cue condition is **congruent** and the word color is **green**.
* **η = η₁ - η₂**: Represents the difference between these two conditions.

The goal is to assess whether **η is significantly different from zero**, which would indicate a meaningful difference in reading times based on gender and cue condition

A group of letters on a white background

AI-generated content may be incorrect.

**Interpretation of key statistics:**

* The **estimate of η** reflects the difference in mean reading time between **female (incongruent) and male (congruent) conditions**.
* The **standard error** quantifies the uncertainty around the estimate; smaller values indicate greater precision.
* The **t-value** measures the **strength of the difference between η₁ and η₂** in relation to model variability.
* The **p-value** is used to determine statistical significance at the **5% level (α = 0.05)**

**Conclusion**

At the **5% significance level**:

* If **p < 0.05**, we **reject the null hypothesis** ((η = 0)), suggesting a **statistically significant difference** in reading times between male and female adults under different cue conditions.
* If **p ≥ 0.05**, we **fail to reject the null hypothesis**, indicating **no strong evidence** that gender and cue condition significantly impact reading time.

This result supports or refutes the hypothesis that **reading time is influenced by gender and cognitive task difficulty**, depending on the statistical significance observed.

**4.**

**Code:**

#4

library(lme4) # For linear mixed models

library(dplyr) # For data manipulation

# Filter data for Subject 4 (Cue = Incongruent, Word = Green)

subject\_4\_data <- cognitive\_data %>%

filter(S == 4 & C == 1 & W == 1)

# Extract observed reading time

T\_obs <- subject\_4\_data$T

print(T\_obs)

fixed\_effects <- fixef(model.lmer)

# Extract individual coefficients

beta\_0 <- fixed\_effects["(Intercept)"]

beta\_G <- fixed\_effects["G"]

beta\_C <- fixed\_effects["C"]

beta\_W <- fixed\_effects["W"]

beta\_GC <- fixed\_effects["G:C"]

beta\_GW <- fixed\_effects["G:W"]

beta\_CW <- fixed\_effects["C:W"]

beta\_GCW <- fixed\_effects["G:C:W"]

# Print fixed effect values

print(fixed\_effects)

random\_effects <- ranef(model.lmer)$S[4, ]

# Extract individual random effect values

mu\_0\_4 <- random\_effects[1] # Random intercept for Subject 4

mu\_C\_4 <- random\_effects[2] # Random effect for Cue Condition

mu\_W\_4 <- random\_effects[3] # Random effect for Word Color

# Print random effects

print(random\_effects)

# Compute predicted T for Subject 4 (Incongruent, Green)

T\_pred <- beta\_0 + beta\_G \* 1 + beta\_C \* 1 + beta\_W \* 1 +

beta\_GC \* 1 \* 1 + beta\_GW \* 1 \* 1 + beta\_CW \* 1 \* 1 + beta\_GCW \* 1 \* 1 +

mu\_0\_4 + mu\_C\_4 \* 1 + mu\_W\_4 \* 1

print(T\_pred)

# Compute residual

residual\_4 <- T\_obs - T\_pred

print(residual\_4)

**Output :**

A close-up of numbers

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**TASK 3**

**5.**

**Code:**

# Load required package

library(lme4)

# Extract variance-covariance matrix of random effects (D Matrix)

D\_matrix <- VarCorr(model.lmer)$S

# Round each element to two decimal places

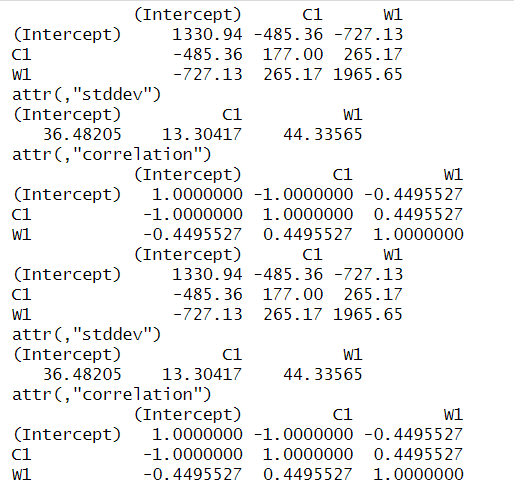
D\_matrix\_rounded <- round(D\_matrix, 2)

# Print the rounded D matrix

print(D\_matrix\_rounded)

round(D\_matrix, 2)

**Output:**



**6.**

**Code:**

# Load required package

library(lme4)

# Extract residual variance (sigma squared)

sigma\_squared <- sigma(model.lmer)^2

# Construct R matrix (diagonal matrix with residual variance)

R\_matrix <- diag(rep(sigma\_squared, length(cognitive\_data$T)))

# Round each element to two decimal places

R\_matrix\_rounded <- round(R\_matrix, 2)

# Print the rounded R matrix

print(R\_matrix\_rounded)

round(R\_matrix, 2)

**Output:**

A screenshot of a computer code

AI-generated content may be incorrect.