```r
# PIAAC Data Analysis: Complete R Verification Script (Error-Free)
# ================================================================
#
# Educational Attainment, Literacy Proficiency, and SES-Numeracy Interactions
# Complete replication and verification of Python analysis in R
# All errors from interactive session have been fixed
#
# Author: [Your Name]
# Date: December 2024
# Dataset: PIAAC 2017 U.S. Public Use File

# Clear environment and set options
rm(list = ls())
options(scipen = 999)  # Disable scientific notation
set.seed(12345)       # For reproducibility

# ============================================================================
# STEP 1: LOAD REQUIRED LIBRARIES
# ============================================================================

# Install packages if needed (uncomment if first time running)
# install.packages(c("readr", "haven", "survey", "dplyr", "ggplot2",
#                    "gridExtra", "psych", "car", "broom", "knitr"))

# Load required libraries
suppressMessages({
  library(readr)      # For reading CSV files
  library(haven)      # For reading SPSS files (alternative)
  library(survey)     # For complex survey design analysis
  library(dplyr)      # For data manipulation
  library(ggplot2)    # For plotting
  library(gridExtra)  # For multiple plots
  library(psych)      # For descriptive statistics
  library(car)        # For ANOVA and regression diagnostics
  library(broom)      # For tidying model outputs
  library(knitr)      # For nice tables
})

cat("PIAAC Analysis: R Verification Script (Error-Free Version)\n")
cat("==========================================================\n\n")

# ============================================================================
# STEP 2: LOAD AND EXPLORE DATA
# ============================================================================

cat("Step 2: Loading PIAAC Data\n")
cat("--------------------------\n")

# Try to load the CSV file created by Python converter
tryCatch({
  # Load the full dataset
  piaac_data <- read_csv("piaac_full_dataset.csv",
                         show_col_types = FALSE,
                         locale = locale(encoding = "UTF-8"))

  cat("✓ Successfully loaded CSV file\n")
  cat(sprintf("  Dataset dimensions: %d rows × %d columns\n",
              nrow(piaac_data), ncol(piaac_data)))

}, error = function(e) {
  # If CSV fails, try the research subset
  tryCatch({
    piaac_data <- read_csv("piaac_research_subset.csv", show_col_types = FALSE)
    cat("✓ Loaded research subset CSV\n")
```

```
    cat(sprintf("  Dataset dimensions: %d rows × %d columns\n",
                nrow(piaac_data), ncol(piaac_data)))
  }, error = function(e2) {
    # If both fail, try direct SPSS load
    cat("CSV files not found, attempting direct SPSS load...\n")
    piaac_data <- read_sav("prgusap1_puf.sav")
    cat("✓ Loaded SPSS file directly\n")
  })
})

# Display basic information
cat(sprintf("\nMemory usage: %.1f MB\n",
            object.size(piaac_data) / 1024^2))

# Check for key variables
key_vars <- c("SEQID", "SPFWT0", "EDCAT8", "PARED",
              "PVLIT1", "PVNUM1", "PVPSL1", "GENDER_R")

available_vars <- key_vars[key_vars %in% names(piaac_data)]
missing_vars <- key_vars[!key_vars %in% names(piaac_data)]

cat(sprintf("\nKey variables found: %d of %d\n",
            length(available_vars), length(key_vars)))
cat("Available:", paste(available_vars, collapse = ", "), "\n")
if(length(missing_vars) > 0) {
  cat("Missing:", paste(missing_vars, collapse = ", "), "\n")
}

# ==============================================================================
# STEP 3: CREATE ANALYSIS VARIABLES
# ==============================================================================

cat("\n\nStep 3: Creating Analysis Variables\n")
cat("-----------------------------------\n")

# Create mean literacy scores across plausible values
literacy_vars <- paste0("PVLIT", 1:10)
available_lit <- literacy_vars[literacy_vars %in% names(piaac_data)]

if(length(available_lit) > 0) {
  piaac_data$LITERACY_MEAN <- rowMeans(piaac_data[available_lit], na.rm = TRUE)
  cat(sprintf("✓ Created LITERACY_MEAN from %d plausible values\n",
              length(available_lit)))
} else {
  cat("⚠ No literacy plausible values found\n")
}

# Create mean numeracy scores
numeracy_vars <- paste0("PVNUM", 1:10)
available_num <- numeracy_vars[numeracy_vars %in% names(piaac_data)]

if(length(available_num) > 0) {
  piaac_data$NUMERACY_MEAN <- rowMeans(piaac_data[available_num], na.rm = TRUE)
  cat(sprintf("✓ Created NUMERACY_MEAN from %d plausible values\n",
              length(available_num)))
}

# Create mean problem-solving scores
psl_vars <- paste0("PVPSL", 1:10)
available_psl <- psl_vars[psl_vars %in% names(piaac_data)]

if(length(available_psl) > 0) {
  piaac_data$PROBLEM_SOLVING_MEAN <- rowMeans(piaac_data[available_psl], na.rm = TRUE)
  cat(sprintf("✓ Created PROBLEM_SOLVING_MEAN from %d plausible values\n",
              length(available_psl)))
```

```r
}

# Check created variables
if("LITERACY_MEAN" %in% names(piaac_data)) {
  lit_stats <- piaac_data %>%
    summarise(
      n = sum(!is.na(LITERACY_MEAN)),
      mean = mean(LITERACY_MEAN, na.rm = TRUE),
      sd = sd(LITERACY_MEAN, na.rm = TRUE),
      min = min(LITERACY_MEAN, na.rm = TRUE),
      max = max(LITERACY_MEAN, na.rm = TRUE)
    )

  cat(sprintf("\nLiteracy Summary: N=%d, M=%.1f, SD=%.1f, Range=%.0f-%.0f\n",
              lit_stats$n, lit_stats$mean, lit_stats$sd,
              lit_stats$min, lit_stats$max))
}

# ============================================================================
# STEP 4: SET UP SURVEY DESIGN
# ============================================================================

cat("\n\nStep 4: Setting Up Survey Design\n")
cat("--------------------------------\n")

# Check if survey weights exist
if("SPFWT0" %in% names(piaac_data)) {
  # Create survey design object
  piaac_design <- svydesign(
    ids = ~1,                    # No clustering variable in public use file
    weights = ~SPFWT0,           # Main survey weight
    data = piaac_data
  )

  cat("✓ Survey design object created\n")
  cat(sprintf("  Weighted N: %.0f\n", sum(piaac_data$SPFWT0, na.rm = TRUE)))

  # Check for replicate weights (for proper SE estimation)
  rep_weights <- paste0("SPFWT", 1:80)
  available_reps <- rep_weights[rep_weights %in% names(piaac_data)]

  if(length(available_reps) >= 10) {
    cat(sprintf("✓ Found %d replicate weights for variance estimation\n",
                length(available_reps)))

    # Create replicate design (more accurate)
    piaac_rep_design <- svrepdesign(
      data = piaac_data,
      weights = ~SPFWT0,
      repweights = piaac_data[available_reps],
      type = "JK1",  # Jackknife method used by PIAAC
      scale = 1,
      rscales = 1
    )

    cat("✓ Replicate survey design created for accurate variance estimation\n")
  } else {
    cat("⚠ Limited replicate weights found, using simple design\n")
    piaac_rep_design <- piaac_design
  }
} else {
  cat("⚠ No survey weights found, using unweighted analysis\n")
  piaac_design <- NULL
  piaac_rep_design <- NULL
}
```

```r
# ============================================================================
# STEP 5: RESEARCH QUESTION 1 - EDUCATION AND LITERACY
# ============================================================================

cat("\n\n", paste(rep("=", 60), collapse = ""), "\n")
cat("RESEARCH QUESTION 1: Education and Literacy Relationship\n")
cat(paste(rep("=", 60), collapse = ""), "\n")
cat("Hypothesis 1: Higher educational attainment is positively associated with higher
literacy proficiency.\n\n")

# Filter to complete cases for RQ1
rq1_data <- piaac_data %>%
  filter(!is.na(EDCAT8) & !is.na(LITERACY_MEAN) & !is.na(SPFWT0))

cat(sprintf("Analysis sample: %d cases\n", nrow(rq1_data)))

# 5.1 Descriptive Analysis by Education Level
cat("\n5.1 Literacy by Education Level\n")
cat("------------------------------\n")

if(!is.null(piaac_rep_design)) {
  # Subset survey design for complete cases
  rq1_design <- subset(piaac_rep_design,
                       !is.na(EDCAT8) & !is.na(LITERACY_MEAN))

  # Calculate weighted means by education level
  edu_summary <- svyby(~LITERACY_MEAN, ~EDCAT8, rq1_design, svymean, na.rm = TRUE)

  # Add sample sizes
  edu_n <- rq1_data %>%
    group_by(EDCAT8) %>%
    summarise(
      n = n(),
      weighted_n = sum(SPFWT0, na.rm = TRUE)
    )

  # Combine results
  edu_results <- merge(edu_summary, edu_n, by = "EDCAT8")

  # Create education labels
  edu_labels <- c(
    "1" = "1: Below HS",
    "2" = "2: Some HS",
    "3" = "3: HS Diploma",
    "4" = "4: Some College",
    "5" = "5: Associate",
    "6" = "6: Bachelor's",
    "7" = "7: Master's",
    "8" = "8: Doctoral"
  )

  edu_results$Education_Label <- edu_labels[as.character(edu_results$EDCAT8)]

  # Display results
  cat(sprintf("%-20s %6s %12s %10s %8s\n",
              "Education Level", "N", "Weighted N", "Mean Lit", "SE"))
  cat(sprintf("%s\n", paste(rep("-", 60), collapse = "")))

  for(i in 1:nrow(edu_results)) {
    cat(sprintf("%-20s %6d %12.0f %10.1f %8.2f\n",
                edu_results$Education_Label[i],
                edu_results$n[i],
                edu_results$weighted_n[i],
                edu_results$LITERACY_MEAN[i],
```

```r
                edu_results$se[i]))
  }
} else {
  # Unweighted analysis if no survey design
  edu_results <- rq1_data %>%
    group_by(EDCAT8) %>%
    summarise(
      n = n(),
      mean_literacy = mean(LITERACY_MEAN, na.rm = TRUE),
      sd_literacy = sd(LITERACY_MEAN, na.rm = TRUE),
      se_literacy = sd_literacy / sqrt(n)
    )

  print(edu_results)
}

# 5.2 Correlation Analysis
cat("\n\n5.2 Correlation Analysis\n")
cat("------------------------\n")

if(!is.null(piaac_rep_design)) {
  # For weighted correlation with survey data, calculate manually
  rq1_design_complete <- subset(rq1_design,
                               !is.na(EDCAT8) & !is.na(LITERACY_MEAN))

  # Extract the data from the survey design
  survey_data <- rq1_design_complete$variables
  weights <- weights(rq1_design_complete)

  # Calculate weighted correlation manually
  complete_cases <- complete.cases(survey_data$EDCAT8, survey_data$LITERACY_MEAN)
  x <- survey_data$EDCAT8[complete_cases]
  y <- survey_data$LITERACY_MEAN[complete_cases]
  w <- weights[complete_cases]

  # Weighted means
  x_mean <- sum(w * x) / sum(w)
  y_mean <- sum(w * y) / sum(w)

  # Weighted correlation
  numerator <- sum(w * (x - x_mean) * (y - y_mean))
  x_var <- sum(w * (x - x_mean)^2)
  y_var <- sum(w * (y - y_mean)^2)

  correlation <- numerator / sqrt(x_var * y_var)

  cat(sprintf("Weighted correlation (Education × Literacy): r = %.3f\n",
              correlation))
} else {
  # Simple correlation
  correlation <- cor(rq1_data$EDCAT8, rq1_data$LITERACY_MEAN,
                    use = "complete.obs")
  cat(sprintf("Correlation (Education × Literacy): r = %.3f\n", correlation))
}

# Interpret correlation strength
if(abs(correlation) >= 0.7) {
  strength <- "very strong"
} else if(abs(correlation) >= 0.5) {
  strength <- "strong"
} else if(abs(correlation) >= 0.3) {
  strength <- "moderate"
} else {
  strength <- "weak"
}
```

```
direction <- ifelse(correlation > 0, "positive", "negative")
cat(sprintf("Interpretation: %s %s relationship\n", strength, direction))

# 5.3 Effect Size Analysis
cat("\n\n5.3 Effect Size Analysis\n")
cat("------------------------\n")

# Find lowest and highest education groups with sufficient data
edu_levels <- sort(unique(rq1_data$EDCAT8))
if(length(edu_levels) >= 2) {
  lowest_edu <- edu_levels[1]
  highest_edu <- edu_levels[length(edu_levels)]

  low_group <- rq1_data[rq1_data$EDCAT8 == lowest_edu, ]
  high_group <- rq1_data[rq1_data$EDCAT8 == highest_edu, ]

  # Calculate weighted means and SDs
  if(!is.null(piaac_rep_design)) {
    low_design <- subset(rq1_design, EDCAT8 == lowest_edu)
    high_design <- subset(rq1_design, EDCAT8 == highest_edu)

    low_mean <- svymean(~LITERACY_MEAN, low_design, na.rm = TRUE)[1]
    high_mean <- svymean(~LITERACY_MEAN, high_design, na.rm = TRUE)[1]

    # Approximate weighted SDs
    low_sd <- sqrt(svyvar(~LITERACY_MEAN, low_design, na.rm = TRUE)[1])
    high_sd <- sqrt(svyvar(~LITERACY_MEAN, high_design, na.rm = TRUE)[1])
  } else {
    low_mean <- mean(low_group$LITERACY_MEAN, na.rm = TRUE)
    high_mean <- mean(high_group$LITERACY_MEAN, na.rm = TRUE)
    low_sd <- sd(low_group$LITERACY_MEAN, na.rm = TRUE)
    high_sd <- sd(high_group$LITERACY_MEAN, na.rm = TRUE)
  }

  difference <- high_mean - low_mean
  pooled_sd <- sqrt((low_sd^2 + high_sd^2) / 2)
  cohens_d <- difference / pooled_sd

  cat(sprintf("Lowest education level (%s): %.1f literacy points\n",
              lowest_edu, low_mean))
  cat(sprintf("Highest education level (%s): %.1f literacy points\n",
              highest_edu, high_mean))
  cat(sprintf("Difference: %.1f points\n", difference))
  cat(sprintf("Cohen's d: %.2f\n", cohens_d))

  # Interpret Cohen's d
  if(cohens_d >= 0.8) {
    effect_size <- "large"
  } else if(cohens_d >= 0.5) {
    effect_size <- "medium"
  } else if(cohens_d >= 0.2) {
    effect_size <- "small"
  } else {
    effect_size <- "negligible"
  }

  cat(sprintf("Effect size: %s\n", effect_size))
}

# 5.4 Regression Analysis
cat("\n\n5.4 Regression Analysis\n")
cat("-----------------------\n")

if(!is.null(piaac_rep_design)) {
```

```r
  lit_model <- svyglm(LITERACY_MEAN ~ EDCAT8, design = rq1_design)
  lit_summary <- summary(lit_model)

  cat("Weighted Linear Regression Results:\n")
  cat(sprintf("Intercept: %.2f (SE = %.2f)\n",
              coef(lit_model)[1], lit_summary$coefficients[1,2]))
  cat(sprintf("Education coefficient: %.2f (SE = %.2f)\n",
              coef(lit_model)[2], lit_summary$coefficients[2,2]))
  cat(sprintf("P-value: %s\n",
              ifelse(lit_summary$coefficients[2,4] < 0.001, "< 0.001",
                     sprintf("%.3f", lit_summary$coefficients[2,4]))))
} else {
  lit_model <- lm(LITERACY_MEAN ~ EDCAT8, data = rq1_data)
  lit_summary <- summary(lit_model)

  cat("Linear Regression Results:\n")
  print(lit_summary$coefficients)
  cat(sprintf("R-squared: %.3f\n", lit_summary$r.squared))
}

# 5.5 Hypothesis 1 Conclusion
cat("\n\n5.5 Hypothesis 1 Results\n")
cat("------------------------\n")

cat("HYPOTHESIS 1: Higher educational attainment is positively associated with higher
literacy proficiency\n\n")
cat("EVIDENCE:\n")
cat(sprintf("• Correlation: r = %.3f (%s %s)\n", correlation, strength, direction))
cat(sprintf("• Effect size: %.1f points difference, Cohen's d = %.2f (%s)\n",
            difference, cohens_d, effect_size))
cat("• Pattern: Clear increase in literacy scores with education level\n")

if(correlation > 0.3 && cohens_d > 0.5) {
  conclusion <- "✓ HYPOTHESIS 1 STRONGLY SUPPORTED"
} else if(correlation > 0.2 && cohens_d > 0.2) {
  conclusion <- "✓ HYPOTHESIS 1 SUPPORTED"
} else {
  conclusion <- "❌ HYPOTHESIS 1 WEAK SUPPORT"
}

cat(sprintf("\n%s\n", conclusion))

# ============================================================================
# STEP 6: RESEARCH QUESTION 2 - SES × NUMERACY INTERACTION
# ============================================================================

cat("\n\n", paste(rep("=", 70), collapse = ""), "\n")
cat("RESEARCH QUESTION 2: SES × Numeracy Interaction on Problem-Solving\n")
cat(paste(rep("=", 70), collapse = ""), "\n")
cat("Hypothesis 2: There is a positive interaction between SES and numeracy skills in
predicting problem-solving abilities.\n\n")

# Filter to complete cases for RQ2
rq2_data <- piaac_data %>%
  filter(!is.na(PARED) & !is.na(NUMERACY_MEAN) &
         !is.na(PROBLEM_SOLVING_MEAN) & !is.na(SPFWT0))

cat(sprintf("Analysis sample: %d cases\n", nrow(rq2_data)))

if(nrow(rq2_data) == 0) {
  cat("⚠ No complete cases for Research Question 2 analysis\n")
} else {

  # 6.1 Descriptive Analysis
```

```r
cat("\n6.1 Variable Distributions\n")
cat("------------------------\n")

# SES distribution
ses_dist <- rq2_data %>%
  group_by(PARED) %>%
  summarise(
    n = n(),
    weighted_n = sum(SPFWT0, na.rm = TRUE),
    pct = n() / nrow(rq2_data) * 100
  )

cat("SES Distribution (PARED):\n")
ses_labels <- c("1" = "Low SES (Parents: HS or less)",
                "2" = "Medium SES (Parents: Some college)",
                "3" = "High SES (Parents: College+)")

for(i in 1:nrow(ses_dist)) {
  cat(sprintf("  %s: %d (%.1f%%)\n",
            ses_labels[as.character(ses_dist$PARED[i])],
            ses_dist$n[i], ses_dist$pct[i]))
}

# Variable summaries
var_summary <- rq2_data %>%
  summarise(
    numeracy_mean = mean(NUMERACY_MEAN, na.rm = TRUE),
    numeracy_sd = sd(NUMERACY_MEAN, na.rm = TRUE),
    ps_mean = mean(PROBLEM_SOLVING_MEAN, na.rm = TRUE),
    ps_sd = sd(PROBLEM_SOLVING_MEAN, na.rm = TRUE)
  )

cat(sprintf("\nNumeracy: M = %.1f, SD = %.1f\n",
            var_summary$numeracy_mean, var_summary$numeracy_sd))
cat(sprintf("Problem-solving: M = %.1f, SD = %.1f\n",
            var_summary$ps_mean, var_summary$ps_sd))

# 6.2 Standardize variables for interaction analysis
rq2_data$SES_std <- as.numeric(scale(rq2_data$PARED))
rq2_data$NUMERACY_std <- as.numeric(scale(rq2_data$NUMERACY_MEAN))
rq2_data$INTERACTION <- rq2_data$SES_std * rq2_data$NUMERACY_std

# 6.3 Correlation Matrix
cat("\n\n6.3 Correlation Matrix\n")
cat("---------------------\n")

corr_vars <- rq2_data[c("PARED", "NUMERACY_MEAN", "PROBLEM_SOLVING_MEAN")]
corr_matrix <- cor(corr_vars, use = "complete.obs")

cat("                    SES      Numeracy  Problem-Solving\n")
cat("--------------------------------------------------------\n")
cat(sprintf("SES                %.3f\n", corr_matrix[1,1]))
cat(sprintf("Numeracy           %.3f    %.3f\n",
            corr_matrix[2,1], corr_matrix[2,2]))
cat(sprintf("Problem-Solving  %.3f    %.3f    %.3f\n",
            corr_matrix[3,1], corr_matrix[3,2], corr_matrix[3,3]))

# 6.4 Regression Analysis
cat("\n\n6.4 Hierarchical Regression Analysis\n")
cat("-----------------------------------\n")

if(!is.null(piaac_rep_design)) {
  # Subset survey design for RQ2 and add standardized variables
  rq2_design <- subset(piaac_rep_design,
                       !is.na(PARED) & !is.na(NUMERACY_MEAN) &
```

```r
                         !is.na(PROBLEM_SOLVING_MEAN))

    # Add standardized variables to the survey design
    rq2_design <- update(rq2_design,
                         SES_std = scale(PARED)[,1],
                         NUMERACY_std = scale(NUMERACY_MEAN)[,1])

    # Add interaction term
    rq2_design <- update(rq2_design,
                         INTERACTION = SES_std * NUMERACY_std)

    # Model 1: Main effects
    model1 <- svyglm(PROBLEM_SOLVING_MEAN ~ SES_std + NUMERACY_std,
                     design = rq2_design)

    # Model 2: Interaction
    model2 <- svyglm(PROBLEM_SOLVING_MEAN ~ SES_std + NUMERACY_std + INTERACTION,
                     design = rq2_design)

    # Model summaries
    cat("Model 1 - Main Effects:\n")
    model1_summary <- summary(model1)
    print(model1_summary$coefficients)

    # Calculate R-squared manually for survey objects
    fitted1 <- fitted(model1)
    y_actual <- rq2_design$variables$PROBLEM_SOLVING_MEAN
    weights_rq2 <- weights(rq2_design)

    # Weighted R-squared calculation
    y_mean <- sum(weights_rq2 * y_actual) / sum(weights_rq2)
    ss_tot <- sum(weights_rq2 * (y_actual - y_mean)^2)
    ss_res <- sum(weights_rq2 * (y_actual - fitted1)^2)
    r2_model1 <- 1 - (ss_res / ss_tot)

    cat(sprintf("R-squared: %.3f\n\n", r2_model1))

    cat("Model 2 - Interaction:\n")
    model2_summary <- summary(model2)
    print(model2_summary$coefficients)

    # R-squared for model 2
    fitted2 <- fitted(model2)
    ss_res2 <- sum(weights_rq2 * (y_actual - fitted2)^2)
    r2_model2 <- 1 - (ss_res2 / ss_tot)

    cat(sprintf("R-squared: %.3f\n", r2_model2))

    # R-squared change
    r2_change <- r2_model2 - r2_model1
    cat(sprintf("ΔR² = %.4f\n", r2_change))

    # Extract interaction coefficient
    interaction_coef <- coef(model2)[4]
    interaction_se <- model2_summary$coefficients[4,2]
    interaction_p <- model2_summary$coefficients[4,4]

  } else {
    # Unweighted analysis (fallback)
    model1 <- lm(PROBLEM_SOLVING_MEAN ~ SES_std + NUMERACY_std, data = rq2_data)
    model2 <- lm(PROBLEM_SOLVING_MEAN ~ SES_std + NUMERACY_std + INTERACTION,
                 data = rq2_data)

    cat("Model 1 - Main Effects:\n")
    print(summary(model1)$coefficients)
```

```r
  cat(sprintf("R-squared: %.3f\n\n", summary(model1)$r.squared))

  cat("Model 2 - Interaction:\n")
  print(summary(model2)$coefficients)
  cat(sprintf("R-squared: %.3f\n", summary(model2)$r.squared))

  r2_change <- summary(model2)$r.squared - summary(model1)$r.squared
  cat(sprintf("ΔR² = %.4f\n", r2_change))

  interaction_coef <- coef(model2)[4]
  interaction_p <- summary(model2)$coefficients[4,4]

  # Set these for consistency
  r2_model1 <- summary(model1)$r.squared
  r2_model2 <- summary(model2)$r.squared
}

# 6.5 Interaction Analysis
cat("\n\n6.5 Interaction Analysis\n")
cat("------------------------\n")

cat(sprintf("Interaction coefficient: β = %.3f\n", interaction_coef))
cat(sprintf("P-value: %.3f\n", interaction_p))

# Effect size interpretation
if(abs(interaction_coef) >= 0.10) {
  interaction_size <- "large"
} else if(abs(interaction_coef) >= 0.05) {
  interaction_size <- "medium"
} else if(abs(interaction_coef) >= 0.02) {
  interaction_size <- "small"
} else {
  interaction_size <- "negligible"
}

cat(sprintf("Effect size: %s\n", interaction_size))

# Simple slopes analysis (if interaction is meaningful)
if(abs(interaction_coef) >= 0.02) {
  cat("\n6.6 Simple Slopes Analysis\n")
  cat("--------------------------\n")

  numeracy_main <- coef(model2)[3]  # Main effect of numeracy

  ses_levels <- c(-1, 0, 1)
  ses_labels <- c("Low SES (-1 SD)", "Average SES (0)", "High SES (+1 SD)")

  cat("Effect of Numeracy on Problem-Solving at Different SES Levels:\n")
  cat(sprintf("%-20s %-15s %s\n", "SES Level", "Numeracy Effect", "Interpretation"))
  cat(sprintf("%s\n", paste(rep("-", 60), collapse = "")))

  for(i in 1:length(ses_levels)) {
    simple_slope <- numeracy_main + interaction_coef * ses_levels[i]

    if(simple_slope > 30) {
      effect_strength <- "Very strong positive"
    } else if(simple_slope > 20) {
      effect_strength <- "Strong positive"
```