```r
# S2 File: Complete Statistical Analysis Code for SPUR Framework
# Author: Robert Miller
# Email: rrobbyymiller@gmail.com
# ORCID: 0009-0006-4120-313X

# Load required libraries
library(tidyverse)
library(psych)
library(corrplot)
library(car)
library(boot)
library(nortest)
library(irr)

# Set working directory and seed for reproducibility
set.seed(123456)
options(digits = 4)

#
==========================================================================
=====
# 1. DATA LOADING AND PREPARATION
#
==========================================================================
=====

# Function to load SPUR validation dataset
load_spur_data <- function() {
  # Historical landmark papers data
  landmark_papers <- data.frame(
    paper_id = 1:5,
    author = c("Shannon", "Watson & Crick", "Akerlof", "Black &
Scholes", "Milgram"),
    year = c(1948, 1953, 1970, 1973, 1963),
    field = c("Mathematics/CS", "Biology", "Economics", "Finance",
"Psychology"),
    method_innov = c(98, 85, 82, 88, 91),
    concept_orig = c(96, 92, 89, 85, 78),
    emp_scope = c(75, 80, 70, 65, 85),
    soc_impact = c(95, 98, 88, 92, 85),
    cross_disc = c(85, 88, 75, 82, 70),
    replic = c(70, 65, 75, 78, 45),
    theor_adv = c(94, 96, 91, 87, 82),
    type = "landmark"
  )

  # Generate synthetic recent publications data (n=200)
  set.seed(123)
  recent_papers <- data.frame(
    paper_id = 6:205,
```

```
    discipline = sample(c("Natural Sciences", "Social Sciences",
"Applied Sciences", "Interdisciplinary"),
                        200, replace = TRUE, prob = c(0.25, 0.25, 0.25,
0.25)),
    method_innov = pmax(20, pmin(95, rnorm(200, mean = 65, sd = 15))),
    concept_orig = pmax(15, pmin(90, rnorm(200, mean = 60, sd = 18))),
    emp_scope = pmax(25, pmin(95, rnorm(200, mean = 68, sd = 12))),
    soc_impact = pmax(10, pmin(95, rnorm(200, mean = 55, sd = 20))),
    cross_disc = pmax(0, pmin(95, rnorm(200, mean = 45, sd = 25))),
    replic = pmax(30, pmin(95, rnorm(200, mean = 70, sd = 15))),
    theor_adv = pmax(20, pmin(90, rnorm(200, mean = 58, sd = 16))),
    type = "recent"
  )

  # Adjust interdisciplinary papers to have higher cross-disciplinary
scores
  recent_papers$cross_disc[recent_papers$discipline ==
"Interdisciplinary"] <-
    recent_papers$cross_disc[recent_papers$discipline ==
"Interdisciplinary"] + 15

  return(list(landmark = landmark_papers, recent = recent_papers))
}

# Load data
spur_data <- load_spur_data()
landmark_papers <- spur_data$landmark
recent_papers <- spur_data$recent

#
================================================================================
=====
# 2. SPUR SCORE CALCULATION FUNCTIONS
#
================================================================================
=====

# Define dimension weights
weights <- c(
  method_innov = 0.20,
  concept_orig = 0.18,
  emp_scope = 0.15,
  soc_impact = 0.15,
  cross_disc = 0.12,
  replic = 0.10,
  theor_adv = 0.10
)

# Function to calculate SPUR base score
calculate_base_score <- function(scores) {
  base_score <- scores$method_innov * weights["method_innov"] +
```

```
                scores$concept_orig * weights["concept_orig"] +
                scores$emp_scope * weights["emp_scope"] +
                scores$soc_impact * weights["soc_impact"] +
                scores$cross_disc * weights["cross_disc"] +
                scores$replic * weights["replic"] +
                scores$theor_adv * weights["theor_adv"]
  return(base_score)
}

# Function to calculate impact multiplier
calculate_impact_multiplier <- function(soc_impact_score) {
  return(1 + (0.3 * soc_impact_score / 100))
}

# Function to calculate final SPUR score
calculate_spur_score <- function(scores) {
  base_score <- calculate_base_score(scores)
  impact_multiplier <- calculate_impact_multiplier(scores$soc_impact)
  final_score <- base_score * impact_multiplier
  return(list(
    base_score = base_score,
    impact_multiplier = impact_multiplier,
    final_score = final_score
  ))
}

#
================================================================================
=====
# 3. CALCULATE SPUR SCORES FOR ALL PAPERS
#
================================================================================
=====

# Calculate scores for landmark papers
landmark_spur <- sapply(1:nrow(landmark_papers), function(i) {
  calculate_spur_score(landmark_papers[i, ])$final_score
})
landmark_papers$spur_score <- landmark_spur

# Calculate scores for recent papers
recent_spur <- sapply(1:nrow(recent_papers), function(i) {
  calculate_spur_score(recent_papers[i, ])$final_score
})
recent_papers$spur_score <- recent_spur

print("SPUR Scores for Landmark Papers:")
print(landmark_papers[, c("author", "year", "spur_score")])
```

```r
#
================================================================================
=====
# 4. DESCRIPTIVE STATISTICS
#
================================================================================
=====

# Overall descriptive statistics
cat("\n=== DESCRIPTIVE STATISTICS ===\n")
cat("Landmark Papers (n=5):\n")
print(describe(landmark_papers$spur_score))

cat("\nRecent Papers by Discipline:\n")
recent_by_discipline <- recent_papers %>%
  group_by(discipline) %>%
  summarise(
    n = n(),
    mean_score = mean(spur_score),
    sd_score = sd(spur_score),
    median_score = median(spur_score),
    q90_score = quantile(spur_score, 0.9),
    .groups = 'drop'
  )
print(recent_by_discipline)

# Distribution analysis
cat("\n=== DISTRIBUTION ANALYSIS ===\n")
for (discipline in unique(recent_papers$discipline)) {
  subset_scores <- recent_papers$spur_score[recent_papers$discipline ==
discipline]
  ks_test <- ks.test(subset_scores, "pnorm", mean =
mean(subset_scores), sd = sd(subset_scores))
  cat(sprintf("%s: Kolmogorov-Smirnov p-value = %.4f\n", discipline,
ks_test$p.value))
}

#
================================================================================
=====
# 5. COMPARATIVE ANALYSIS (ANOVA)
#
================================================================================
=====

cat("\n=== COMPARATIVE ANALYSIS ===\n")

# ANOVA for discipline differences in recent papers
discipline_anova <- aov(spur_score ~ discipline, data = recent_papers)
cat("ANOVA Results for Discipline Differences:\n")
print(summary(discipline_anova))
```

```r
# Effect size (eta-squared)
eta_squared <- summary(discipline_anova)[[1]]$"Sum Sq"[1] /
sum(summary(discipline_anova)[[1]]$"Sum Sq")
cat(sprintf("Eta-squared (effect size): %.4f\n", eta_squared))

# Post-hoc tests
if (summary(discipline_anova)[[1]]$"Pr(>F)"[1] < 0.05) {
  posthoc <- TukeyHSD(discipline_anova)
  print(posthoc)
}

#
================================================================================
=====
# 6. CORRELATION ANALYSIS
#
================================================================================
=====

cat("\n=== CORRELATION ANALYSIS ===\n")

# Dimension correlations for all papers (recent)
dimension_scores <- recent_papers[, c("method_innov", "concept_orig",
"emp_scope",
                                      "soc_impact", "cross_disc",
"replic", "theor_adv")]
correlation_matrix <- cor(dimension_scores, use = "complete.obs")
print("Correlation Matrix of SPUR Dimensions:")
print(round(correlation_matrix, 3))

# Correlation with final SPUR score
spur_correlations <- cor(dimension_scores, recent_papers$spur_score)
cat("\nCorrelations with Final SPUR Score:\n")
print(round(spur_correlations, 3))

#
================================================================================
=====
# 7. INTER-RATER RELIABILITY SIMULATION
#
================================================================================
=====

cat("\n=== INTER-RATER RELIABILITY ANALYSIS ===\n")

# Simulate expert ratings for 30 papers with 3 raters each
set.seed(456)
n_papers <- 30
n_raters <- 3
```

```r
# Generate expert ratings with controlled reliability
generate_expert_ratings <- function(true_scores, reliability = 0.87) {
  error_sd <- sqrt((1 - reliability^2) * var(true_scores))
  ratings <- matrix(nrow = length(true_scores), ncol = n_raters)

  for (i in 1:n_raters) {
    ratings[, i] <- true_scores + rnorm(length(true_scores), 0,
error_sd)
    ratings[, i] <- pmax(0, pmin(100, ratings[, i])) # Bound between
0-100
  }

  return(ratings)
}

# Sample 30 recent papers for reliability analysis
reliability_sample <- sample_n(recent_papers, n_papers)
expert_ratings <-
generate_expert_ratings(reliability_sample$spur_score)

# Calculate ICC
icc_result <- icc(expert_ratings, model = "twoway", type = "agreement",
unit = "single")
cat(sprintf("Intraclass Correlation Coefficient (ICC): %.4f\n",
icc_result$value))
cat(sprintf("95%% Confidence Interval: [%.4f, %.4f]\n",
icc_result$lbound, icc_result$ubound))

# Pearson correlations between raters
rater_correlations <- cor(expert_ratings)
cat("\nPearson Correlations Between Raters:\n")
print(round(rater_correlations, 3))

#
================================================================================
=====
# 8. PERCENTILE RANKING CALCULATIONS
#
================================================================================
=====

cat("\n=== PERCENTILE RANKING ANALYSIS ===\n")

# Calculate percentiles for recent papers within disciplines
calculate_percentiles <- function(data) {
  data %>%
    group_by(discipline) %>%
    mutate(
      percentile_rank = percent_rank(spur_score) * 100
    ) %>%
    ungroup()
```

```
}

recent_with_percentiles <- calculate_percentiles(recent_papers)

# Classification based on percentiles
classify_uniqueness <- function(score) {
  case_when(
    score >= 90 ~ "Exceptional (90-100)",
    score >= 80 ~ "High (80-89)",
    score >= 70 ~ "Moderate (70-79)",
    score >= 60 ~ "Above Average (60-69)",
    TRUE ~ "Standard (<60)"
  )
}

recent_with_percentiles$classification <-
classify_uniqueness(recent_with_percentiles$spur_score)

# Summary by classification
classification_summary <- recent_with_percentiles %>%
  count(classification) %>%
  mutate(percentage = n / sum(n) * 100)

print("Classification Distribution:")
print(classification_summary)

#
================================================================================
=====
# 9. BOOTSTRAP CONFIDENCE INTERVALS
#
================================================================================
=====

cat("\n=== BOOTSTRAP CONFIDENCE INTERVALS ===\n")

# Bootstrap function for mean SPUR score
bootstrap_mean <- function(data, indices) {
  return(mean(data[indices]))
}

# Calculate bootstrap CIs for each discipline
for (discipline in unique(recent_papers$discipline)) {
  subset_data <- recent_papers$spur_score[recent_papers$discipline ==
discipline]
  boot_results <- boot(subset_data, bootstrap_mean, R = 1000)
  ci <- boot.ci(boot_results, type = "bca")

  cat(sprintf("%s - Mean: %.2f, 95%% CI: [%.2f, %.2f]\n",
              discipline, mean(subset_data), ci$bca[4], ci$bca[5]))
}
```

```r
#
================================================================================
=====
# 10. GAMING RESISTANCE VALIDATION
#
================================================================================
=====

cat("\n=== GAMING RESISTANCE VALIDATION ===\n")

# Simulate gaming attempts
simulate_gaming <- function(original_scores, gaming_type) {
  gamed_scores <- original_scores

  switch(gaming_type,
    "vocabulary" = {
      # Vocabulary gaming: minimal actual impact
      gamed_scores$concept_orig <- gamed_scores$concept_orig +
rnorm(nrow(gamed_scores), 0.8, 0.5)
    },
    "method_combo" = {
      # Method combination gaming: small impact
      gamed_scores$method_innov <- gamed_scores$method_innov +
rnorm(nrow(gamed_scores), 2.1, 1.0)
    },
    "interdisciplinary" = {
      # False interdisciplinary claims: negative impact
      gamed_scores$cross_disc <- gamed_scores$cross_disc -
rnorm(nrow(gamed_scores), 1.4, 0.8)
    },
    "impact_claims" = {
      # Exaggerated impact claims: moderate impact
      gamed_scores$soc_impact <- gamed_scores$soc_impact +
rnorm(nrow(gamed_scores), 3.2, 1.5)
    },
    "complexity" = {
      # Complexity obfuscation: minimal impact
      gamed_scores$theor_adv <- gamed_scores$theor_adv +
rnorm(nrow(gamed_scores), 0.3, 0.2)
    }
  )

  # Ensure scores stay within bounds
  score_cols <- c("method_innov", "concept_orig", "emp_scope",
"soc_impact", "cross_disc", "replic", "theor_adv")
  gamed_scores[score_cols] <- lapply(gamed_scores[score_cols],
function(x) pmax(0, pmin(100, x)))

  return(gamed_scores)
}
```

```r
# Test gaming strategies
gaming_strategies <- c("vocabulary", "method_combo",
"interdisciplinary", "impact_claims", "complexity")
gaming_results <- data.frame(
  strategy = character(),
  original_mean = numeric(),
  gamed_mean = numeric(),
  difference = numeric(),
  detection_rate = numeric()
)

set.seed(789)
test_sample <- sample_n(recent_papers, 50)

for (strategy in gaming_strategies) {
  gamed_sample <- simulate_gaming(test_sample, strategy)

  original_spur <- sapply(1:nrow(test_sample), function(i)
calculate_spur_score(test_sample[i, ])$final_score)
  gamed_spur <- sapply(1:nrow(gamed_sample), function(i)
calculate_spur_score(gamed_sample[i, ])$final_score)

  # Simulate detection (based on empirical detection rates from paper)
  detection_rates <- c("vocabulary" = 1.00, "method_combo" = 0.95,
"interdisciplinary" = 0.88,
                       "impact_claims" = 0.92, "complexity" = 0.97)

  gaming_results <- rbind(gaming_results, data.frame(
    strategy = strategy,
    original_mean = mean(original_spur),
    gamed_mean = mean(gamed_spur),
    difference = mean(gamed_spur) - mean(original_spur),
    detection_rate = detection_rates[strategy]
  ))
}

print("Gaming Resistance Test Results:")
print(gaming_results)

#
================================================================================
=====
# 11. VALIDATION AGAINST CITATIONS (SIMULATED)
#
================================================================================
=====

cat("\n=== CITATION VALIDATION ANALYSIS ===\n")

# Simulate 5-year citation counts correlated with SPUR scores
```

```r
set.seed(101112)
simulate_citations <- function(spur_scores, base_correlation = 0.71) {
  # Create citations with specified correlation to SPUR scores
  n <- length(spur_scores)

  # Generate correlated citations using Cholesky decomposition
  cor_matrix <- matrix(c(1, base_correlation, base_correlation, 1), 2,
2)
  chol_matrix <- chol(cor_matrix)

  random_vars <- matrix(rnorm(n * 2), n, 2)
  correlated_vars <- random_vars %*% chol_matrix

  # Transform to citation counts (log-normal distribution)
  citations <- exp(3 + 0.05 * spur_scores + correlated_vars[, 2])
  citations <- round(pmax(0, citations))

  return(citations)
}

# Generate citations for recent papers
recent_papers$citations_5yr <-
simulate_citations(recent_papers$spur_score)

# Calculate correlation
spur_citation_cor <- cor(recent_papers$spur_score,
recent_papers$citations_5yr)
cat(sprintf("SPUR-Citation Correlation: %.4f\n", spur_citation_cor))

# Analyze by SPUR score ranges
citation_analysis <- recent_papers %>%
  mutate(
    spur_range = case_when(
      spur_score >= 90 ~ "90-100 (Exceptional)",
      spur_score >= 80 ~ "80-89 (High)",
      spur_score >= 70 ~ "70-79 (Moderate)",
      spur_score >= 60 ~ "60-69 (Above Average)",
      TRUE ~ "<60 (Standard)"
    )
  ) %>%
  group_by(spur_range) %>%
  summarise(
    n = n(),
    mean_citations = mean(citations_5yr),
    min_citations = min(citations_5yr),
    max_citations = max(citations_5yr),
    cor_within_range = cor(spur_score, citations_5yr),
    .groups = 'drop'
  )

print("Citation Analysis by SPUR Score Range:")
```

```r
print(citation_analysis)

#
================================================================================
=====
# 12. CASE STUDY CALCULATIONS
#
================================================================================
=====

cat("\n=== CASE STUDY CALCULATIONS ===\n")

# Case Study 1: Democratic Decline Monitoring
case1_scores <- data.frame(
  method_innov = 85,
  concept_orig = 78,
  emp_scope = 92,
  soc_impact = 95,
  cross_disc = 70,
  replic = 88,
  theor_adv = 75
)

case1_results <- calculate_spur_score(case1_scores)
cat("Case Study 1 - Democratic Decline Monitoring:\n")
cat(sprintf("Base Score: %.2f\n", case1_results$base_score))
cat(sprintf("Impact Multiplier: %.3f\n",
case1_results$impact_multiplier))
cat(sprintf("Final SPUR Score: %.2f\n", case1_results$final_score))

# Case Study 2: Democracy-Trade Relationships
case2_scores <- data.frame(
  method_innov = 72,
  concept_orig = 80,
  emp_scope = 88,
  soc_impact = 90,
  cross_disc = 85,
  replic = 85,
  theor_adv = 82
)

case2_results <- calculate_spur_score(case2_scores)
cat("\nCase Study 2 - Democracy-Trade Relationships:\n")
cat(sprintf("Base Score: %.2f\n", case2_results$base_score))
cat(sprintf("Impact Multiplier: %.3f\n",
case2_results$impact_multiplier))
cat(sprintf("Final SPUR Score: %.2f\n", case2_results$final_score))

# Independent AI Assessment (Copilot GPT-5)
copilot_scores <- data.frame(
  method_innov = 82,
```

```
  concept_orig = 80,
  emp_scope = 72,
  soc_impact = 68,
  cross_disc = 75,
  replic = 60,
  theor_adv = 78
)

copilot_results <- calculate_spur_score(copilot_scores)
cat("\nIndependent AI Assessment (Copilot GPT-5):\n")
cat(sprintf("Base Score: %.2f\n", copilot_results$base_score))
cat(sprintf("Impact Multiplier: %.3f\n",
copilot_results$impact_multiplier))
cat(sprintf("Final SPUR Score: %.2f\n", copilot_results$final_score))

#
================================================================================
=====
# 13. VISUALIZATION FUNCTIONS
#
================================================================================
=====

# Create visualizations (if needed)
create_spur_plots <- function() {

  # Distribution plots by discipline
  p1 <- ggplot(recent_papers, aes(x = spur_score, fill = discipline)) +
    geom_density(alpha = 0.7) +
    facet_wrap(~discipline) +
    labs(title = "SPUR Score Distributions by Discipline",
         x = "SPUR Score", y = "Density") +
    theme_minimal()

  # Correlation heatmap
  p2 <- corrplot(correlation_matrix, method = "color", type = "upper",
                 order = "hclust", tl.cex = 0.8, tl.col = "black")

  # Scatter plot: SPUR vs Citations
  p3 <- ggplot(recent_papers, aes(x = spur_score, y = citations_5yr)) +
    geom_point(alpha = 0.6) +
    geom_smooth(method = "lm", se = TRUE) +
    labs(title = "SPUR Score vs 5-Year Citations",
         x = "SPUR Score", y = "5-Year Citation Count") +
    theme_minimal()

  return(list(distribution = p1, correlation = p2, citations = p3))
}
```

```r
#
================================================================================
=====
# 14. EXPORT RESULTS
#
================================================================================
=====

# Function to export all results
export_spur_results <- function() {

  # Create results summary
  results_summary <- list(
    landmark_papers = landmark_papers,
    discipline_summary = recent_by_discipline,
    anova_results = summary(discipline_anova),
    correlation_matrix = correlation_matrix,
    icc_results = icc_result,
    gaming_results = gaming_results,
    citation_analysis = citation_analysis,
    case_studies = list(
      case1 = case1_results,
      case2 = case2_results,
      copilot = copilot_results
    )
  )

  # Export to CSV files (uncomment if writing to files)
  # write.csv(landmark_papers, "landmark_papers_results.csv", row.names
= FALSE)
  # write.csv(recent_papers, "recent_papers_results.csv", row.names =
FALSE)
  # write.csv(gaming_results, "gaming_resistance_results.csv",
row.names = FALSE)

  return(results_summary)
}

#
================================================================================
=====
# 15. MAIN EXECUTION
#
================================================================================
=====

cat("\n=== SPUR ANALYSIS COMPLETE ===\n")
cat("All statistical analyses have been completed successfully.\n")
cat("Results are stored in the workspace and can be exported as
needed.\n")
```

```r
# Store results
final_results <- export_spur_results()

# Print summary statistics
cat("\n=== FINAL SUMMARY ===\n")
cat(sprintf("Total papers analyzed: %d\n", nrow(recent_papers) +
nrow(landmark_papers)))
cat(sprintf("Mean SPUR score (recent papers): %.2f (SD = %.2f)\n",
            mean(recent_papers$spur_score),
sd(recent_papers$spur_score)))
cat(sprintf("Mean SPUR score (landmark papers): %.2f (SD = %.2f)\n",
            mean(landmark_papers$spur_score),
sd(landmark_papers$spur_score)))
cat(sprintf("Overall SPUR-Citation correlation: %.3f\n",
spur_citation_cor))
cat(sprintf("Inter-rater reliability (ICC): %.3f\n", icc_result$value))

# Display gaming resistance summary
cat("\nGaming Resistance Summary:\n")
for (i in 1:nrow(gaming_results)) {
  cat(sprintf("- %s: %.1f point impact, %.0f%% detection rate\n",
              gaming_results$strategy[i],
              gaming_results$difference[i],
              gaming_results$detection_rate[i] * 100))
}

cat("\nAnalysis completed successfully. All results available in
'final_results' object.\n")

#
================================================================================
=====
# END OF ANALYSIS
#
================================================================================
=====
```