CIRF Dataset Preparation Framework for Statistical Analysis

Executive Summary

This framework transforms your case study data into analysis-ready format for testing the three multiplicative effects. The preparation process converts qualitative case information into quantitative variables suitable for regression analysis, interaction modeling, and cross-validation.

---

1. Current Dataset Structure Assessment

1.1 What You Have (Based on Your Case Studies)

Primary Variables Available:

```
Case Information:
- Case Name/ID
- Geographic Region
- Cultural Sector
- Time Period
- Success/Failure Classification

CIRF Component Scores:
- Operational Pillars (4): Economic Value (0/1), Cultural Integrity (0/1),
  Adaptability (0/1), Social Empowerment (0/1)
- Community Control Filters (5): Community Benefit (0/1), Cultural Protection (0/1),
  Community Relevance (0/1), Sustainable Development (0/1), Dignity & Empowerment (0/1)
- Resilience Capacities (4): Protective (0/1), Adaptive (0/1),
  Transformative (0/1), Generative (0/1)
- Total CIRF Score (0-13)
```

1.2 What You Need for Statistical Analysis

Required Data Structure:

```
Rows: 180+ cases (observations)
Columns: 25-30 variables (components + interactions + controls)
Format: CSV/Excel with numerical codes
Missing Data: <5% per variable for robust analysis
```

---

2. Step-by-Step Data Preparation Process

Step 1: Create Master Dataset Template

```
r
```

```r
# Create empty dataframe with required structure
cirf_data <- data.frame(
  # Case Identifiers
  Case_ID = character(),
  Case_Name = character(),

  # Geographic and Sectoral Controls
  Region = character(),
  Country = character(),
  Sector = character(),

  # Temporal Variables
  Start_Year = numeric(),
  End_Year = numeric(),
  Years_Operating = numeric(),

  # CIRF Component Scores (0/1 binary)
  Economic_Value = numeric(),
  Cultural_Integrity = numeric(),
  Adaptability = numeric(),
  Social_Empowerment = numeric(),

  # Community Control Filters (0/1 binary)
  Community_Benefit = numeric(),
  Cultural_Protection = numeric(),
  Community_Relevance = numeric(),
  Sustainable_Development = numeric(),
  Dignity_Empowerment = numeric(),

  # Resilience Capacities (0/1 binary)
  Protective_Capacity = numeric(),
  Adaptive_Capacity = numeric(),
  Transformative_Capacity = numeric(),
  Generative_Capacity = numeric(),

  # Derived Scores
  Operational_Pillars_Total = numeric(),    # Sum of 4 components (0-4)
  Community_Control_Total = numeric(),      # Sum of 5 components (0-5)
  Resilience_Capacity_Total = numeric(),    # Sum of 4 components (0-4)
  CIRF_Total_Score = numeric(),             # Sum of all components (0-13)

  # Success Classification
  Success_Binary = numeric(),               # 1 if score ≥7, 0 if score <7
  Success_Category = character(),           # "High", "Medium", "Low"
```

```
  stringsAsFactors = FALSE
)
```

Step 2: Populate Data from Case Studies

Data Entry Template (Excel/Google Sheets):

Case_ID	Case_Name	Region	Country	Sector	Start_Year	End_Year	EV	CI	AD	SE	CB	CI
001	Mi'kmaq Clearwater	North America	Canada	Seafood	2020	2024	1	1	1	1	1	1
002	Caribbean Cruise Tourism	Caribbean	Multiple	Tourism	2010	2024	1	0	1	0	0	0

Column Key:

- EV = Economic Value, CI = Cultural Integrity, AD = Adaptability, SE = Social Empowerment
- CB = Community Benefit, CP = Cultural Protection, CR = Community Relevance, SD = Sustainable Development, DE = Dignity & Empowerment
- PC = Protective Capacity, AC = Adaptive Capacity, TC = Transformative Capacity, GC = Generative Capacity

Step 3: Data Validation and Quality Control

```
r
```

```r
# Data validation functions
validate_cirf_data <- function(df) {

  # Check for missing values
  missing_summary <- df %>%
    summarise_all(~sum(is.na(.))) %>%
    pivot_longer(everything(), names_to = "Variable", values_to = "Missing_Count") %>%
    mutate(Missing_Percent = Missing_Count / nrow(df) * 100)

  # Check binary coding (should be 0 or 1 only)
  binary_vars <- c("Economic_Value", "Cultural_Integrity", "Adaptability", "Social_Empowerment",
                   "Community_Benefit", "Cultural_Protection", "Community_Relevance",
                   "Sustainable_Development", "Dignity_Empowerment",
                   "Protective_Capacity", "Adaptive_Capacity", "Transformative_Capacity", "Generative_Capacity")

  binary_check <- df[binary_vars] %>%
    summarise_all(~all(. %in% c(0, 1, NA))) %>%
    pivot_longer(everything(), names_to = "Variable", values_to = "Valid_Binary")

  # Check CIRF score calculations
  df$Calculated_Score <- rowSums(df[binary_vars], na.rm = TRUE)
  score_discrepancies <- sum(df$CIRF_Total_Score != df$Calculated_Score, na.rm = TRUE)

  # Return validation results
  list(
    missing_data = missing_summary,
    binary_validation = binary_check,
    score_discrepancies = score_discrepancies,
    total_cases = nrow(df),
    complete_cases = sum(complete.cases(df))
  )
}

# Run validation
validation_results <- validate_cirf_data(cirf_data)
print(validation_results)
```

Step 4: Create Interaction Variables

```r
```

```r
# Function to create all required interaction variables
create_interaction_variables <- function(df) {

  # Multiplicative Effect 1: Economic Control Multiplier
  df$Community_Control_Score <- df$Community_Benefit + df$Cultural_Protection +
                                df$Community_Relevance + df$Sustainable_Development +
                                df$Dignity_Empowerment

  # Normalize Community Control to 0-1 scale for exponential calculation
  df$Community_Control_Normalized <- df$Community_Control_Score / 5

  # Economic Value × Community Control^2.3
  df$EV_CC_Multiplier <- df$Economic_Value * (df$Community_Control_Normalized^2.3)

  # Multiplicative Effect 2: Innovation Balance
  df$CI_AD_Difference <- abs(df$Cultural_Integrity - df$Adaptability)
  df$CI_AD_Balance <- 1 - df$CI_AD_Difference  # Higher when CI ≈ AD
  df$Innovation_Index <- df$Cultural_Integrity * df$Adaptability * df$CI_AD_Balance

  # Multiplicative Effect 3: Capacity Compound
  df$Resilience_Score <- df$Protective_Capacity + df$Adaptive_Capacity +
                         df$Transformative_Capacity + df$Generative_Capacity

  # Learning factor (logarithmic time effect)
  df$Learning_Factor <- log(df$Years_Operating + 1)

  # SE × RC × Learning compound effect
  df$SE_RC_Compound <- df$Social_Empowerment * df$Resilience_Score * df$Learning_Factor

  # Additional useful interactions
  df$EV_CI_Balance <- df$Economic_Value * df$Cultural_Integrity  # Economic-Cultural synergy
  df$CC_SE_Empowerment <- df$Community_Control_Score * df$Social_Empowerment  # Control-Empowerment

  return(df)
}

# Apply interaction variable creation
cirf_data <- create_interaction_variables(cirf_data)
```

Step 5: Create Control Variables

```r
```

```r
# Geographic regional coding
create_regional_variables <- function(df) {

  # Create regional dummy variables
  df$North_America <- ifelse(df$Region == "North America", 1, 0)
  df$Europe <- ifelse(df$Region == "Europe", 1, 0)
  df$Asia <- ifelse(df$Region == "Asia", 1, 0)
  df$Africa <- ifelse(df$Region == "Africa", 1, 0)
  df$Latin_America <- ifelse(df$Region == "Latin America", 1, 0)
  df$Oceania <- ifelse(df$Region == "Oceania", 1, 0)
  df$Middle_East <- ifelse(df$Region == "Middle East", 1, 0)

  # Create sectoral dummy variables
  df$Traditional_Crafts <- ifelse(df$Sector == "Traditional Crafts", 1, 0)
  df$Cultural_Tourism <- ifelse(df$Sector == "Cultural Tourism", 1, 0)
  df$Performing_Arts <- ifelse(df$Sector == "Performing Arts", 1, 0)
  df$Food_Heritage <- ifelse(df$Sector == "Food & Heritage", 1, 0)
  df$Digital_Media <- ifelse(df$Sector == "Digital Media", 1, 0)
  df$Heritage_Sites <- ifelse(df$Sector == "Heritage Sites", 1, 0)

  # Economic development level (based on country income classification)
  df$Development_Level <- case_when(
    df$Country %in% c("USA", "Canada", "Germany", "France", "UK", "Japan", "Australia") ~ "High Income",
    df$Country %in% c("China", "Brazil", "Mexico", "Turkey", "Thailand") ~ "Upper Middle Income",
    df$Country %in% c("India", "Indonesia", "Philippines", "Vietnam", "Morocco") ~ "Lower Middle Income",
    df$Country %in% c("Bangladesh", "Nepal", "Cambodia", "Rwanda") ~ "Low Income",
    TRUE ~ "Unknown"
  )

  # Indigenous/community-based classification
  df$Indigenous_Led <- ifelse(grepl("Indigenous|Aboriginal|First Nations|Inuit|Maori|Native",
                                    df$Case_Name, ignore.case = TRUE), 1, 0)

  return(df)
}

# Apply control variable creation
cirf_data <- create_regional_variables(cirf_data)
```

---

3. Data Preparation Checklist

Pre-Analysis Quality Checks

```r
```

```r
# Comprehensive data quality assessment
quality_check <- function(df) {

  cat("=== CIRF Dataset Quality Assessment ===\n\n")

  # Basic descriptive statistics
  cat("1. DATASET OVERVIEW\n")
  cat("Total cases:", nrow(df), "\n")
  cat("Total variables:", ncol(df), "\n")
  cat("Complete cases:", sum(complete.cases(df)), "\n")
  cat("Completion rate:", round(sum(complete.cases(df))/nrow(df)*100, 1), "%\n\n")

  # CIRF score distribution
  cat("2. CIRF SCORE DISTRIBUTION\n")
  print(table(df$CIRF_Total_Score))
  cat("Mean CIRF Score:", round(mean(df$CIRF_Total_Score, na.rm=TRUE), 2), "\n")
  cat("SD CIRF Score:", round(sd(df$CIRF_Total_Score, na.rm=TRUE), 2), "\n\n")

  # Success rate by threshold
  cat("3. SUCCESS RATES\n")
  success_7plus <- sum(df$CIRF_Total_Score >= 7, na.rm=TRUE)
  cat("Cases scoring 7+/13:", success_7plus, "(", round(success_7plus/nrow(df)*100,1), "%)\n")
  failure_below7 <- sum(df$CIRF_Total_Score < 7, na.rm=TRUE)
  cat("Cases scoring <7/13:", failure_below7, "(", round(failure_below7/nrow(df)*100,1), "%)\n\n")

  # Regional distribution
  cat("4. REGIONAL DISTRIBUTION\n")
  print(table(df$Region))
  cat("\n")

  # Sectoral distribution
  cat("5. SECTORAL DISTRIBUTION\n")
  print(table(df$Sector))
  cat("\n")

  # Key interaction variable summary
  cat("6. INTERACTION VARIABLES SUMMARY\n")
  interaction_vars <- c("EV_CC_Multiplier", "Innovation_Index", "SE_RC_Compound")
  print(df[interaction_vars] %>% summary())

  # Correlation matrix for key variables
  cat("\n7. KEY VARIABLE CORRELATIONS\n")
  key_vars <- c("CIRF_Total_Score", "EV_CC_Multiplier", "Innovation_Index", "SE_RC_Compound")
  cor_matrix <- cor(df[key_vars], use = "complete.obs")
  print(round(cor_matrix, 3))
```

```
}

# Run quality check
quality_check(cirf_data)
```

Missing Data Strategy

r

```
# Handle missing data systematically
handle_missing_data <- function(df) {

  # For binary CIRF components: Missing = 0 (conservative assumption)
  binary_vars <- c("Economic_Value", "Cultural_Integrity", "Adaptability", "Social_Empowerment",
                   "Community_Benefit", "Cultural_Protection", "Community_Relevance",
                   "Sustainable_Development", "Dignity_Empowerment",
                   "Protective_Capacity", "Adaptive_Capacity", "Transformative_Capacity", "Generative_Capacity")

  df[binary_vars] <- lapply(df[binary_vars], function(x) ifelse(is.na(x), 0, x))

  # For Years_Operating: Use median imputation
  if(sum(is.na(df$Years_Operating)) > 0) {
    median_years <- median(df$Years_Operating, na.rm = TRUE)
    df$Years_Operating[is.na(df$Years_Operating)] <- median_years
  }

  # Recalculate derived variables after imputation
  df <- create_interaction_variables(df)

  return(df)
}

# Apply missing data handling
cirf_data <- handle_missing_data(cirf_data)
```

---

4. Sample Data Population Template

Template for Your Case Studies

Based on your case study format, here's how to systematically extract data:

Example: Mi'kmaq Clearwater Seafoods (12/13 CIRF Score)

Case_ID: 001
Case_Name: Mi'kmaq Clearwater Seafoods Partnership
Region: North America
Country: Canada
Sector: Seafood/Indigenous Enterprise
Start_Year: 2020
End_Year: 2024
Years_Operating: 4

Component Scores (from your analysis):
Economic_Value: 1 (✓ Economic Value Creation present)
Cultural_Integrity: 1 (✓ Cultural Integrity maintained)
Adaptability: 1 (✓ Adaptability demonstrated)
Social_Empowerment: 1 (✓ Social Empowerment achieved)

Community_Benefit: 1 (✓ Community Benefit clear)
Cultural_Protection: 1 (✓ Cultural Protection active)
Community_Relevance: 1 (✓ Community Relevance high)
Sustainable_Development: 1 (✓ Sustainable Development achieved)
Dignity_Empowerment: 1 (✓ Dignity & Empowerment strong)

Protective_Capacity: 1 (✓ Protective capacity demonstrated)
Adaptive_Capacity: 1 (✓ Adaptive capacity shown)
Transformative_Capacity: 1 (✓ Transformative capacity evident)
Generative_Capacity: 0 (✗ Generative capacity not documented)

CIRF_Total_Score: 12
Success_Binary: 1 (score ≥ 7)
Success_Category: High

Batch Processing Template

r

```r
# Function to convert your case study text to data rows
process_case_study <- function(case_text, case_id) {

  # Extract CIRF scores using text parsing
  # This assumes your case studies follow consistent format

  operational_pattern <- "Operational Pillars \\((\\d)/4\\)"
  community_pattern <- "Community Control Filters \\((\\d)/5\\)"
  resilience_pattern <- "Resilience Capacities \\((\\d)/4\\)"

  operational_score <- as.numeric(str_extract(case_text, operational_pattern, group = 1))
  community_score <- as.numeric(str_extract(case_text, community_pattern, group = 1))
  resilience_score <- as.numeric(str_extract(case_text, resilience_pattern, group = 1))

  # Convert to individual component scores (you'll need to customize this based on your format)
  # This is a simplified example - you'll need more sophisticated parsing

  case_row <- data.frame(
    Case_ID = case_id,
    Case_Name = extract_case_name(case_text),
    Region = extract_region(case_text),
    # ... other fields
    CIRF_Total_Score = operational_score + community_score + resilience_score
  )

  return(case_row)
}
```

---

5. Final Dataset Structure Verification

Required Final Structure

```r
r
```

```r
# Verify final dataset has all required variables for analysis
required_vars <- c(
  # Identifiers
  "Case_ID", "Case_Name", "Region", "Sector",

  # Time variables
  "Years_Operating", "Learning_Factor",

  # Core CIRF components
  "Economic_Value", "Cultural_Integrity", "Adaptability", "Social_Empowerment",
  "Community_Control_Score", "Resilience_Score", "CIRF_Total_Score",

  # Key interaction variables
  "EV_CC_Multiplier", "Innovation_Index", "SE_RC_Compound",

  # Control variables
  "North_America", "Europe", "Asia", "Africa", "Traditional_Crafts", "Cultural_Tourism",

  # Outcome variables
  "Success_Binary"
)

# Check all variables present
missing_vars <- setdiff(required_vars, names(cirf_data))
if(length(missing_vars) > 0) {
  cat("Missing required variables:", paste(missing_vars, collapse = ", "), "\n")
} else {
  cat("All required variables present ✓\n")
}

# Final dataset summary
cat("Final dataset ready for analysis:\n")
cat("Cases:", nrow(cirf_data), "\n")
cat("Variables:", ncol(cirf_data), "\n")
cat("Ready for statistical modeling ✓\n")
```

---

6. Implementation Timeline

Week 1: Data Structure Setup

- [ ] Create master dataset template
- [ ] Define all variable coding schemes
- [ ] Set up validation functions

Week 2: Data Population

- [ ] Extract data from 50 highest-scoring cases
- [ ] Extract data from 50 lowest-scoring cases
- [ ] Extract data from remaining cases
- [ ] Run quality validation checks

Week 3: Variable Creation

- [ ] Create all interaction variables
- [ ] Generate control variables
- [ ] Handle missing data
- [ ] Final validation and cleaning

Week 4: Analysis Preparation

- [ ] Descriptive statistics
- [ ] Variable distribution checks
- [ ] Correlation analysis
- [ ] Dataset ready for statistical modeling

Immediate Action Items

Start Today:

1. Create the Excel/CSV template with column headers
2. Begin with your 10 highest-scoring success cases (12-13/13)
3. Code your 10 lowest-scoring failure cases (0-2/13)
4. This gives you 20 cases to test the basic multiplicative effect hypotheses

Priority Cases for Initial Analysis:

- **High Multiplier**: Mi'kmaq Clearwater, Nollywood, Trinidad Carnival
- **Low Multiplier**: Caribbean Cruise Tourism, Failed Heritage Sites
- **Boundary Zone**: Italian Fashion Industry, Japanese Anime Industry

With these 20 cases properly coded, you can run preliminary statistical tests to confirm the multiplicative effects exist before coding the full 180+ dataset.

Would you like me to help you create the specific Excel template or walk through coding the first few cases?