**DEMATEL Sensitivity Analysis Shiny App - Setup Guide**

This guide will help you set up and run the DEMATEL Sensitivity Analysis Shiny application.

**Directory Structure**

Create the following directory structure:

shiny\_dematel\_app/

├── app.R # Main Shiny application

├── R/

│ ├── dematel\_spectral.R # Your existing spectral analysis code

│ ├── sensitivity-core.R # Your existing sensitivity core code

│ ├── sensitivity-methods.R # Your existing sensitivity methods code

│ └── ui\_components.R # UI helper functions (new)

├── www/

│ └── styles.css # Custom CSS styling (new)

└── example\_data/

└── A\_example.csv # Example dataset (new)

**Step-by-Step Installation**

**1. Install Required R Packages**

Run the following R code to install all necessary packages:

# Core Shiny packages

install.packages(c(

"shiny",

"shinydashboard",

"shinyWidgets"

))

# Data manipulation and visualization

install.packages(c(

"DT",

"ggplot2",

"plotly",

"viridis",

"reshape2",

"gridExtra"

))

# Additional utilities

install.packages(c(

"tools",

"utils"

))

**2. Create Directory Structure**

# Create the main directory

dir.create("shiny\_dematel\_app", showWarnings = FALSE)

# Create subdirectories

dir.create("shiny\_dematel\_app/R", showWarnings = FALSE)

dir.create("shiny\_dematel\_app/www", showWarnings = FALSE)

dir.create("shiny\_dematel\_app/example\_data", showWarnings = FALSE)

**3. Copy Your Existing R Files**

Copy your existing R files to the R/ directory:

* dematel\_spectral.R
* sensitivity-core.R
* sensitivity-methods.R
* sensitivity-integration.R (optional)
* sensitivity-visualization.R (optional)

**4. Create New Files**

Create the new files as provided:

* app.R (main application)
* R/ui\_components.R (UI helpers)
* www/styles.css (styling)
* example\_data/A\_example.csv (example data)

**5. Verify Installation**

Run this verification script to check if everything is properly installed:

# Set working directory to your app folder

setwd("shiny\_dematel\_app")

# Check if all required packages are installed

required\_packages <- c("shiny", "shinydashboard", "shinyWidgets",

"DT", "ggplot2", "plotly", "viridis",

"reshape2", "gridExtra")

missing\_packages <- required\_packages[!sapply(required\_packages, require, character.only = TRUE, quietly = TRUE)]

if (length(missing\_packages) > 0) {

cat("Missing packages:", paste(missing\_packages, collapse = ", "), "\n")

cat("Please install them using: install.packages(c(", paste0("'", missing\_packages, "'", collapse = ", "), "))\n")

} else {

cat("✅ All required packages are installed!\n")

}

# Check if all files exist

required\_files <- c(

"app.R",

"R/dematel\_spectral.R",

"R/sensitivity-core.R",

"R/sensitivity-methods.R",

"R/ui\_components.R",

"www/styles.css",

"example\_data/A\_example.csv"

)

missing\_files <- required\_files[!file.exists(required\_files)]

if (length(missing\_files) > 0) {

cat("Missing files:\n")

cat(paste("-", missing\_files), sep = "\n")

} else {

cat("✅ All required files are present!\n")

}

**Running the Application**

**Local Development**

# Navigate to app directory

setwd("shiny\_dematel\_app")

# Run the app

shiny::runApp()

# Or run with specific options

shiny::runApp(port = 3838, host = "0.0.0.0", launch.browser = TRUE)

**Production Deployment**

**Option 1: RStudio Connect**

# Install rsconnect if not already installed

install.packages("rsconnect")

# Configure your RStudio Connect account (one time setup)

rsconnect::setAccountInfo(name='your-account',

token='your-token',

secret='your-secret')

# Deploy the app

rsconnect::deployApp(appDir = "shiny\_dematel\_app",

appName = "dematel-sensitivity-analysis")

**Option 2: Shiny Server**

1. Install Shiny Server on your server
2. Copy the app directory to /srv/shiny-server/
3. Access via http://your-server-ip:3838/shiny\_dematel\_app/

**Option 3: shinyapps.io**

# Install rsconnect

install.packages("rsconnect")

# Configure shinyapps.io account

rsconnect::setAccountInfo(name='your-account',

token='your-token',

secret='your-secret')

# Deploy to shinyapps.io

rsconnect::deployApp(appDir = "shiny\_dematel\_app")

**Troubleshooting**

**Common Issues**

**1. Package Loading Errors**

# If you get namespace errors, try:

detach("package:shiny", unload = TRUE)

library(shiny)

# Or restart R session and try again

.rs.restartR()

**2. File Path Issues**

Make sure your working directory is set correctly:

# Check current directory

getwd()

# Set to app directory if needed

setwd("path/to/shiny\_dematel\_app")

**3. Memory Issues with Large Matrices**

If you encounter memory issues with large matrices:

# Increase memory limit (Windows)

memory.limit(size = 8192) # 8GB

# Monitor memory usage

gc()

object.size(your\_matrix)

**4. Plot Rendering Issues**

If plots don't render correctly:

# Clear graphics device

dev.off()

# Check if required graphics packages are installed

install.packages(c("png", "jpeg", "Cairo"))

**Performance Optimization**

**For Large Matrices (>100×100):**

1. Use numerical method with larger epsilon (0.05-0.1)
2. Limit visualization options
3. Consider server deployment with more RAM
4. Implement progress bars for long computations

**For Better Responsiveness:**

# Add these options to your app.R

options(shiny.maxRequestSize = 30\*1024^2) # 30MB file uploads

options(shiny.usecairo = TRUE) # Better plot rendering

**Customization Guide**

**Adding New Analysis Methods**

1. **Create the R function** in R/sensitivity-methods.R:

your\_new\_method <- function(obj, ...) {

# Your implementation

return(results)

}

1. **Add UI controls** in app.R:

radioButtons("method\_choice",

choices = list("Numerical" = "numerical",

"Analytical" = "analytical",

"Your Method" = "your\_method"))

1. **Update server logic** in app.R:

if (input$method\_choice == "your\_method") {

sens\_obj <- your\_new\_method(sens\_obj, ...)

}

**Customizing Appearance**

1. **Modify CSS** in www/styles.css
2. **Change color schemes** by updating the gradient variables
3. **Add custom themes** by creating new CSS classes

**Adding New Visualizations**

1. **Create plot function** in R/sensitivity-visualization.R
2. **Add UI output** in the appropriate tab
3. **Add server render function**

**Data Format Requirements**

**Input Matrix Requirements:**

* **Format**: CSV file
* **Structure**: Square matrix (n×n)
* **Values**: Numeric only
* **Diagonal**: Should be zero (no self-influence)
* **Headers**: Optional, can be included or excluded
* **Encoding**: UTF-8 recommended

**Example Valid Matrix:**

0,3,2,1

2,0,3,2

1,2,0,3

2,1,2,0

**Example with Headers:**

Factor1,Factor2,Factor3,Factor4

0,3,2,1

2,0,3,2

1,2,0,3

2,1,2,0

**Advanced Configuration**

**Custom Factor Names**

# In your CSV file with headers, or

# Use the factor names input field in the app

factor\_names <- c("Leadership", "Communication", "Innovation", "Quality")

**Computation Parameters**

# Numerical method settings

epsilon\_values <- c(0.001, 0.01, 0.1) # Accuracy vs speed tradeoff

# Critical threshold options

threshold\_percentiles <- seq(50, 99, by = 5) # 50%, 55%, ..., 99%

**Memory Management**

# For large matrices, implement batching

process\_in\_batches <- function(matrix, batch\_size = 100) {

# Split large computations into smaller chunks

}

**Security Considerations**

**For Production Deployment:**

1. **File Upload Limits**:

options(shiny.maxRequestSize = 30\*1024^2) # Limit to 30MB

1. **Input Validation**:

# Validate matrix dimensions

if (nrow(matrix) > 1000) {

stop("Matrix too large for this server")

}

1. **Error Handling**:

tryCatch({

# Sensitive operations

}, error = function(e) {

showNotification("An error occurred", type = "error")

# Log error securely without exposing details

})

**Getting Help**

**Resources:**

* **Shiny Documentation**: https://shiny.rstudio.com/
* **DEMATEL Method**: Original academic papers
* **R Help**: ?function\_name for specific functions

**Support:**

* Check error messages in R console
* Use browser developer tools for JavaScript errors
* Enable Shiny debugging: options(shiny.trace = TRUE)

**Community:**

* RStudio Community: https://community.rstudio.com/
* Stack Overflow: Tag questions with [r] [shiny] [dematel]

**Next Steps**

After successful setup:

1. **Test with example data** to verify functionality
2. **Upload your own data** to perform real analysis
3. **Customize the interface** to match your needs
4. **Deploy to production** for broader use
5. **Add advanced features** based on user feedback

**Version History**

* **v1.0**: Initial release with basic functionality
* **v1.1**: Added intervention analysis
* **v1.2**: Enhanced visualizations and UI improvements
* **v1.3**: Performance optimizations and bug fixes

Remember to keep your R packages updated for the best performance and security!