Graphs of 2spa

Jimmy Zhang

2022-10-02

```
# Loading packages
library(lavaan)
```

```
## This is lavaan 0.6-12
## lavaan is FREE software! Please report any bugs.
```

```
library(stats)
devtools::load_all()
```

```
## i Loading R2spa
```

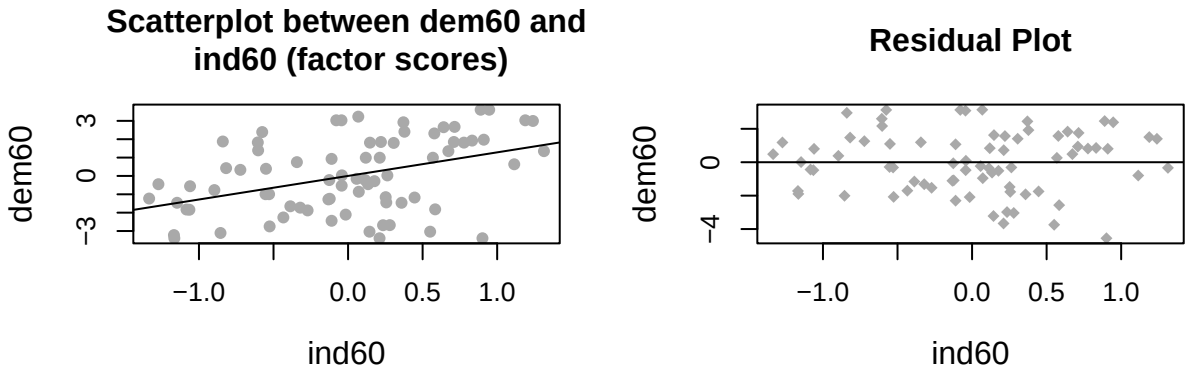
Function

```
tspa_plot <- function (tspa_fit) {

  fit_data <- parameterestimates(tspa_fit)
  # latent_estimates <- lavPredict(tspa_fit)
  latent_scores <- lavInspect(tspa_fit, what = "data")

  if (is.list(latent_scores)) {
    latent_names <- colnames(latent_scores[[1]])
    df_latent_scores <- lapply(latent_scores, data.frame)
    latent_dv <- c(t(na.omit(fit_data[1:(nrow(fit_data)/length(latent_scores)), ][which(fit_data$op == 
    latent_iv <- c(t(na.omit(fit_data[1:(nrow(fit_data)/length(latent_scores)), ][which(fit_data$op == 
    latent_model <- list()

    for (g in seq(length(df_latent_scores))) {
      for (i in seq(length(latent_dv))) {
        par(mfrow=c(2,2))
        latent_model[[g]] <- lm(as.numeric(t(df_latent_scores[[g]][paste0("fs_", latent_dv[i])])) ~
                                as.numeric(t(df_latent_scores[[g]][paste0("fs_", latent_iv[i])])),
                              data = df_latent_scores[[g]])

        plot(latent_scores[[g]][ ,paste0("fs_", latent_iv[i])],
             latent_scores[[g]][ ,paste0("fs_", latent_dv[i])],
             ylab = paste(latent_dv[i]),
             xlab = paste(latent_iv[i]),
             main = paste0("Scatterplot between ", latent_dv[i], " and\n ",
```

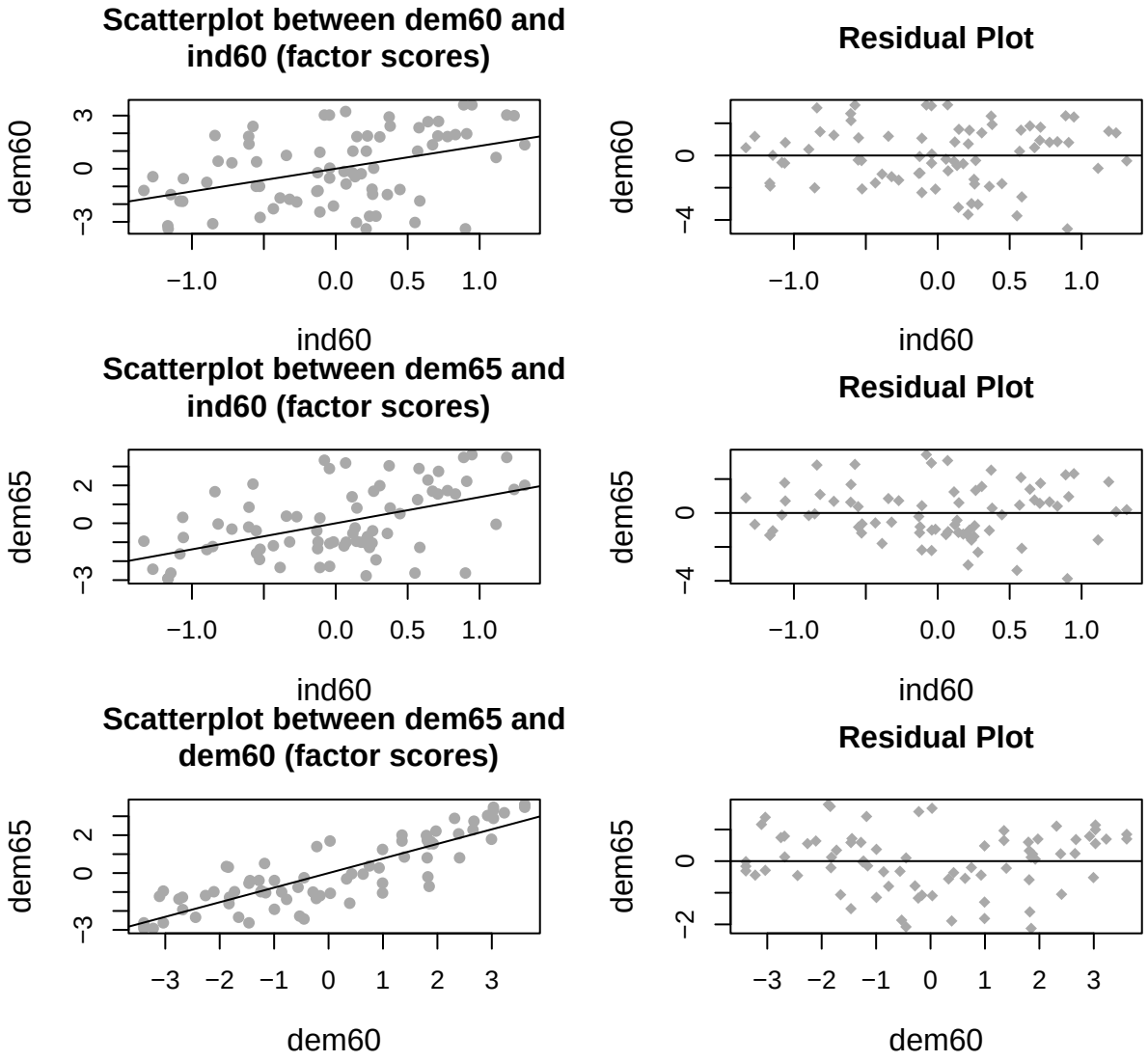
```
                          latent_iv[i], " (factor scores; group ", g, ")"),
            pch = 16,
            ps = 1.25,
            col = "darkgray",
            cex.lab = 1.2,
            cex.axis = 1)
      abline(latent_model[[g]])

      df_latent_scores[[g]]$residuals <-  latent_model[[g]]$residuals
      plot(latent_scores[[g]][ ,paste0("fs_", latent_iv[i])], df_latent_scores[[g]]$residuals,
            ylab = latent_dv[i],
            xlab = latent_iv[i],
            main = paste0("Residual Plot (group ", g, ")"),
            pch = 18,
            ps = 1.25,
            col = "darkgray",
            cex.lab = 1.2,
            cex.axis = 1)
      abline(0, 0)
    }
  }

} else {
  latent_names <- colnames(latent_scores)
  df_latent_scores <- data.frame(latent_scores)
  latent_dv <- c(t(fit_data[which(fit_data$op == "~"),]["lhs"]))
  latent_iv <- c(t(fit_data[which(fit_data$op == "~"),]["rhs"]))

  for (i in seq(length(latent_dv))) {
    par(mfrow=c(2,2))
    latent_model <- lm(as.numeric(t(df_latent_scores[paste0("fs_", latent_dv[i])])) ~
                         as.numeric(t(df_latent_scores[paste0("fs_", latent_iv[i])])),
                       data = df_latent_scores)
    # df_latent_estimates$fitted_values <- predict(latent_model)

    plot(latent_scores[ ,paste0("fs_", latent_iv[i])],
         latent_scores[ ,paste0("fs_", latent_dv[i])],
         ylab = paste(latent_dv[i]),
         xlab = paste(latent_iv[i]),
         main = paste("Scatterplot between", latent_dv[i], "and\n",
                      latent_iv[i], "(factor scores)"),
         pch = 16,
         ps = 1.25,
         col = "darkgray",
         cex.lab = 1.2,
         cex.axis = 1)
    abline(latent_model)
    # lines(as.numeric(t(df_latent_estimates[latent_iv[i]])),
    #       df_latent_estimates$fitted_values, lwd=2)

    df_latent_scores$residuals <-  latent_model$residuals
    plot(latent_scores[ ,paste0("fs_", latent_iv[i])], df_latent_scores$residuals,
         ylab = latent_dv[i],
```

```
            xlab = latent_iv[i],
            main = "Residual Plot",
            pch = 18,
            ps = 1.25,
            col = "darkgray",
            cex.lab = 1.2,
            cex.axis = 1)
      abline(0, 0)
    }

  }

}
```

Example 1: Single group, single predictor



Example 2: Single group, multiple predictors



Example 3: Multiple groups, multiple predictors