Partial Least Squares Path Modelling & VIF – Key Commands

Required Package

plspm

Installing plspm

```
install.packages("devtools")
library(devtools)
install.packages("git2r")
library(git2r)
library(gitcreds)
install.packages("githubinstall")      ← possibly, this command will not work & is not required
library(githubinstall)                 ← possibly, this command will not work & is not required
install_github("gastonstat/plspm")
```

Get going

```
require(plspm)
```

Key indicators of the structural model (outer model)

Loadings:

- Correlation of an item (indicator) with its factor (latent variable)
- Threshold: 0.5 (relaxed), 0.6 (moderate) or 0.7 (strict)

Cronbach's alpha:

- Average inter-item correlation within one factor (latent variable)
- Threshold: 0.6 (relaxed) or 0.7 (strict)

Dillon Goldstein's rho:

- Similar indicator like Cronbach's alpha – but without the assumption of similar/equal loadings
- Threshold: 0.7

Average variance extracted (AVE):

- Indicators' (items') variance that can be explained by the latent variable in proportion to the total variance within the latent variable (factor)
- Threshold: 0.5

Partial Least Squares Path Modeling (package: plspm)

(1) PLSPM – Basic Application

Building the inner model

```
latent_variable_A = c(0, 0, 0)
latent_variable_B = c(0, 0, 0)
latent_variable_C = c(1, 1, 0)
```

→ "1" = dependent from column

```
pls = rbind(latent_variable_A, latent_variable_B, latent_variable_C)
colnames(pls) = rownames(pls)

pls
innerplot(pls)
```

Building the outer model

```
pls_blocks = list(c("variableA1", "variableA2", "variableA3"), c("variableB1", "variableB2", "variableB3"), c("variableC1", "variableC2", "variableC3"))
pls_modes = rep("A", n)
```

→ Please note: n = number of latent variables

Running plspm & results

```
pls_results = plspm(data, pls, pls_blocks, pls_modes)

pls_results

pls_results$outer_model
plot(pls_results, what = "loadings", arr.width = 0.1)
pls_results$unidim
pls_results$crossloadings

pls_results$inner_model
pls_results$path_coefs
plot(pls_results)
pls_results$inner_summary
```

Bootstrap validation

```
pls_results_val = plspm(data, pls, pls_blocks, modes = pls_modes, boot.val = TRUE, br = 5000)
pls_results_val$boot
```

(2) PLSPM – Group Comparison

1st Perform basic PLSPM

2nd Perform group comparison

```
pls_group_1 = dataset[dataset$variable == "attribute_1", ]
pls_group_1_results = plspm(pls_group_1, pls, pls_blocks, modes = pls_modes)
pls_group_2 = dataset[dataset$variable == "attribute_2", ]
pls_group_2_results = plspm(pls_group_2, pls, pls_blocks, modes = pls_modes)

dataset$variable <- as.factor(dataset$variable)
```

→ Please note: this step is required as the grouping variable must be a factor

```
group_comparison_results = plspm.groups(pls_results, dataset$variable, method = "bootstrap")
group_comparison_results
```

(3) PLSPM – Moderator Variable

1st Perform basic PLSPM

2nd Calculate product indicators

Definitions:

variable_1 = indicator 1 of influenced variable

moderator_1 = indicator 1 of moderator variable

inter_1 = indicator 1 of interaction/moderating effect

(for categorical moderators (2 groups):

```
        moderator_1 = rep(0, number of rows)
        moderator_1 [grouping_variable == “attribute”] = 1
        dataset$moderator_1 = moderator_1
    )

dataset$inter_1 = dataset$moderator_1 * dataset$variable_1
dataset$inter_2 = dataset$moderator_1 * dataset $variable_2
…
dataset$inter_3 = dataset$moderator_2 * dataset $variable_1
dataset$inter_4 = dataset$moderator_2 * dataset $variable_2
…
```

3rd Perform PLSPM with interacting variable (inter)

(4) Variance Inflation Factors – adequacy of formatively measured constructs

Required Package

```
faraway
install.packages(“faraway”)
```

Get going

```
require(faraway)
```

Calclating variance inflation factors

```
dataset_new <- cbind(dataset$variable1, dataset$variable2,…)
colnames(dataset_new) <- c(“name1”, “name2”,…)

vif(dataset_new)
```

→ please note: thresholds are at 5.0 (strict) or 10.0 (relaxed)