Evaluation of Reflective Measurement Models

Contents

J. F. Hair Jr. et al., *Partial Least Squares Structural Equation Modeling (PLS-SEM) Using R*, Classroom Companion: Business,
https://doi.org/10.1007/978-3-030-80519-7_4

Learning Objectives

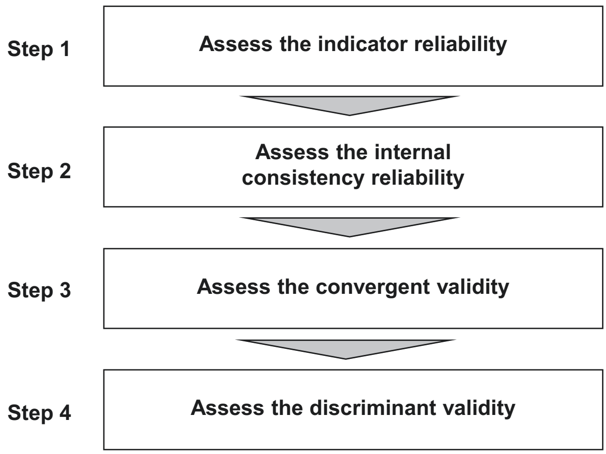
After reading this chapter, you should understand:

1. The concept of indicator reliability
2. The different metrics for assessing internal consistency reliability
3. How to interpret the average variance extracted (AVE) as a measure of convergent validity
4. How to evaluate discriminant validity using the HTMT criterion
5. How to use SEMinR to assess reflectively measured constructs in the corporate reputation example

4.1 Introduction

This chapter describes how to evaluate the quality of reflective measurement models estimated by PLS-SEM, both in in terms of reliability and validity. Assessing reflective measurement models includes evaluating the reliability of measures, on both an indicator level (indicator reliability) and a construct level (internal consistency reliability). Validity assessment focuses on each measure's convergent validity using the average variance extracted (AVE). Moreover, the heterotrait–monotrait (HTMT) ratio of correlations allows to assess a reflectively measured construct's discriminant validity in comparison with other construct measures in the same model. ◘ Figure 4.1 illustrates the reflective measurement model evaluation process. In the following sections, we address each criterion for the evaluation of reflective measurement models and offer rules of thumb for their use. In the second part of this chapter, we explain how to apply the metrics to our corporate reputation example using SEMinR.

◘ **Fig. 4.1** Reflective measurement model assessment procedure. (Source: authors' own figure)



4.2 Indicator Reliability

The first step in reflective measurement model assessment involves examining how much of each indicator's variance is explained by its construct, which is indicative of **indicator reliability**. To compute an indicator's explained variance, we need to square the indicator loading, which is the bivariate correlation between indicator and construct. As such, the indicator reliability indicates the **communality** of an indicator. **Indicator loadings** above 0.708 are recommended, since they indicate that the construct explains more than 50 percent of the indicator's variance, thus providing acceptable indicator reliability.

Researchers frequently obtain weaker indicator loadings (< 0.708) for their measurement models in social science studies, especially when newly developed scales are used (Hulland, 1999). Rather than automatically eliminating indicators when their loading is below 0.70, researchers should carefully examine the effects of indicator removal on other reliability and validity measures. Generally, indicators with loadings between 0.40 and 0.708 should be considered for removal only when deleting the indicator leads to an increase in the internal consistency reliability or convergent validity (discussed in the next sections) above the suggested threshold value. Another consideration in the decision of whether to delete an indicator is the extent to which its removal affects **content validity**, which refers to the extent to which a measure represents all facets of a given construct. As a consequence, indicators with weaker loadings are sometimes retained. Indicators with very low loadings (below 0.40) should, however, always be eliminated from the measurement model (Hair, Hult, Ringle, & Sarstedt, 2022).

4.3 Internal Consistency Reliability

The second step in reflective measurement model assessment involves examining **internal consistency reliability**. Internal consistency reliability is the extent to which indicators measuring the same construct are associated with each other. One of the primary measures used in PLS-SEM is Jöreskog's (1971) **composite reliability rho$_c$**. Higher values indicate higher levels of reliability. For example, reliability values between 0.60 and 0.70 are considered "acceptable in exploratory research," whereas values between 0.70 and 0.90 range from "satisfactory to good." Values above 0.90 (and definitely above 0.95) are problematic, since they indicate that the indicators are redundant, thereby reducing construct validity (Diamantopoulos, Sarstedt, Fuchs, Wilczynski, & Kaiser, 2012). Reliability values of 0.95 and above also suggest the possibility of undesirable response patterns (e.g., straight-lining), thereby triggering inflated correlations among the error terms of the indicators.

Cronbach's alpha is another measure of internal consistency reliability, which assumes the same thresholds as the composite reliability (rho$_c$). A major limitation of Cronbach's alpha, however, is that it assumes all indicator loadings are the same in the population (also referred to as tau-equivalence). The violation of this assumption manifests itself in lower reliability values than those produced by rho_c. Nevertheless, researchers have shown that even in the absence of tau-equivalence, Cronbach's alpha is an acceptable lower-bound approximation of the true internal consistency reliability (Trizano-Hermosilla & Alvarado, 2016).

While Cronbach's alpha is rather conservative, the composite reliability rho_c may be too liberal, and the construct's true reliability is typically viewed as within these two extreme values. As an alternative and building on Dijkstra (2010), subsequent research has proposed the exact (or consistent) **reliability coefficient rho_A** (Dijkstra, 2014; Dijkstra & Henseler, 2015). The reliability coefficient rho_A usually lies between the conservative Cronbach's alpha and the liberal composite reliability and is therefore considered and acceptable compromise between these two measures.

4.4 Convergent Validity

The third step is to assess (the) **convergent validity** of each construct. Convergent validity is the extent to which the construct converges in order to explain the variance of its indicators. The metric used for evaluating a construct's convergent validity is the **average variance extracted (AVE)** for all indicators on each construct. The AVE is defined as the grand mean value of the squared loadings of the indicators associated with the construct (i.e., the sum of the squared loadings divided by the number of indicators). Therefore, the AVE is equivalent to the **communality** of a construct. The minimum acceptable AVE is 0.50 – an AVE of 0.50 or higher indicates the construct explains 50 percent or more of the indicators' variance that make up the construct (Hair et al., 2022).

4.5 Discriminant Validity

The fourth step is to assess **discriminant validity**. This metric measures the extent to which a construct is empirically distinct from other constructs in the structural model. Fornell and Larcker (1981) proposed the traditional metric and suggested that each construct's AVE (squared variance within) should be compared to the squared inter-construct correlation (as a measure of shared variance between constructs) of that same construct and all other reflectively measured constructs in the structural model – the shared variance between all model constructs should not be larger than their AVEs. Recent research indicates, however, that this metric is not suitable for discriminant validity assessment. For example, Henseler, Ringle, and Sarstedt (2015) show that the Fornell–Larcker criterion (i.e., FL in SEMinR) does not perform well, particularly when the indicator loadings on a construct differ only slightly (e.g., all the indicator loadings are between 0.65 and 0.85). Hence, in empirical applications, the Fornell–Larcker criterion often fails to reliably identify discriminant validity problems (Radomir & Moisescu, 2019) and should therefore be avoided. Nonetheless, we include this criterion in our discussion, as many researchers are familiar with it.

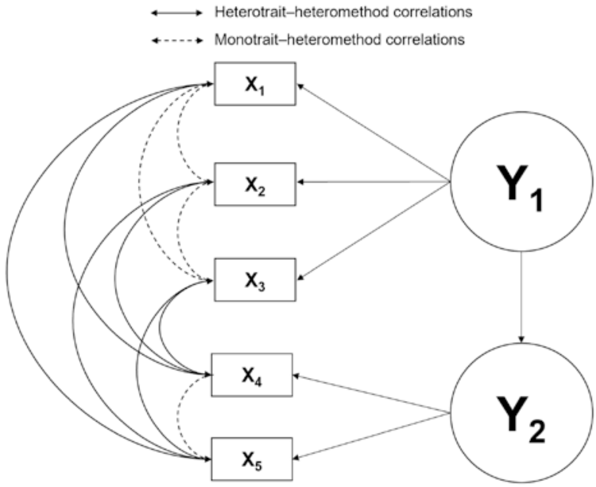


Fig. 4.2 Discriminant validity assessment using the HTMT. (Source: authors' own figure)

As a better alternative, we recommend the **heterotrait–monotrait ratio (HTMT)** of correlations (Henseler et al., 2015) to assess discriminant validity. The HTMT is defined as the mean value of the indicator correlations across constructs (i.e., the **heterotrait–heteromethod correlations**) relative to the (geometric) mean of the average correlations for the indicators measuring the same construct (i.e., the **monotrait–heteromethod correlations**). Figure 4.2 illustrates this concept. The arrows connecting indicators of different constructs represent the heterotrait–heteromethod correlations, which should be as small as possible. On the contrary, the monotrait–heteromethod correlations – represented by the dashed arrows – represent the correlations among indicators measuring the same concept, which should be as high as possible.

Discriminant validity problems are present when HTMT values are high. Henseler et al. (2015) propose a threshold value of 0.90 for structural models with constructs that are conceptually very similar, such as cognitive satisfaction, affective satisfaction, and loyalty. In such a setting, an HTMT value above 0.90 would suggest that discriminant validity is not present. But when constructs are conceptually more distinct, a lower, more conservative, threshold value is suggested, such as 0.85 (Henseler et al., 2015).

In addition, bootstrap confidence intervals can be used to test if the HTMT is significantly different from 1.0 (Henseler et al., 2015) or a lower threshold value, such as 0.9 or 0.85, which should be defined based on the study context (Franke & Sarstedt, 2019). To do so, we need to assess whether the upper bound of the 95% confidence interval (assuming a significance level of 5%) is lower than 0.90 or 0.85. Hence, we have to consider a 95% one-sided bootstrap confidence interval, whose upper boundary is identical to the one produced when computing a 90% two-sided bootstrap confidence interval. To obtain the bootstrap confidence intervals, in line with Aguirre-Urreta and Rönkkö (2018), researchers should generally use the percentile method. In addition, researchers should always use 10,000 bootstrap

4

Table 4.1 Summary of the criteria and rules of thumb for their use

Criterion	Metrics and thresholds
Reflective indicator loadings	≥ 0.708
Internal consistency reliability	Cronbach's alpha is the lower bound, and the composite reliability rho_c is the upper bound for internal consistency reliability. The reliability coefficient rho_A usually lies between these bounds and may serve as a good representation of a construct's internal consistency reliability Minimum 0.70 (or 0.60 in exploratory research) Maximum of 0.95 to avoid indicator redundancy, which would compromise content validity Recommended 0.80 to 0.90
Convergent validity	AVE ≥ 0.50
Discriminant validity	For conceptually similar constructs, HTMT <0.90 For conceptually different constructs, HTMT <0.85 Test if the HTMT is significantly lower than the threshold value

Source: authors' own table

samples (Streukens & Leroi-Werelds, 2016). See ► Chap. 5 for details on bootstrapping and confidence intervals.

Table 4.1 summarizes all the metrics that need to be applied when assessing reflective measurement models.

4.6 Case Study Illustration: Reflective Measurement Models

We continue analyzing the simple corporate reputation PLS path model introduced in the previous chapter. In ► Chap. 3, we explained and demonstrated how to load the data, create the structural model and measurement model objects, and estimate the PLS path model using the SEMinR syntax. In the following, we discuss how to evaluate reflective measurement models, using the simple corporate reputation model (► Fig. 3.2 in ► Chap. 3) as an example.

Recall that to specify and estimate the model, we must first load the data and specify the measurement model and structural model. The model is then estimated by using the `estimate_pls()` command, and the output is assigned to an object. In our case study, we name this object `corp_rep_pls_model`. Once the PLS path model has been estimated, we can access the reports and analysis results by running the `summary()` function. To be able to view different parts of the analysis in greater detail, we suggest assigning the output to a newly created object that we call `summary_corp_rep` in our example (Fig. 4.3).

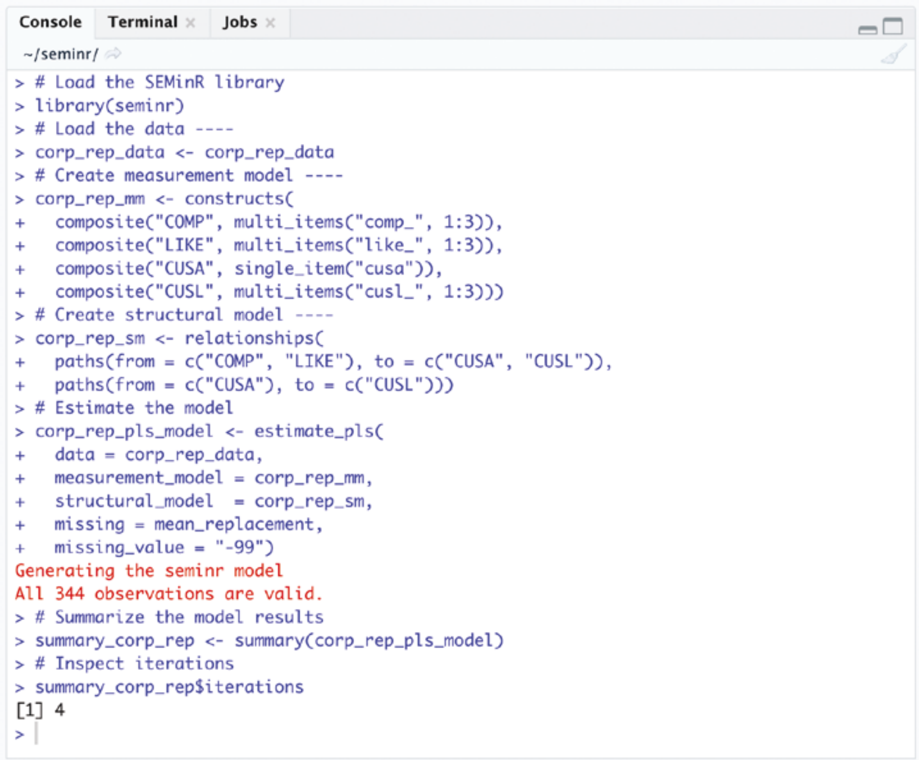


Fig. 4.3 Recap on loading data, specifying and summarizing the model, and inspecting iterations. (Source: authors' screenshot from RStudio)

```
# Load the SEMinR library
library(seminr)

# Load the data
corp_rep_data <- corp_rep_data

# Create measurement model
corp_rep_mm <- constructs(
  composite(“COMP”, multi_items(“comp_”, 1:3)),
  composite(“LIKE”, multi_items(“like_”, 1:3)),
  composite(“CUSA”, single_item(“cusa”)),
  composite(“CUSL”, multi_items(“cusl_”, 1:3)))

# Create structural model
corp_rep_sm <- relationships(
  paths(from = c(“COMP”, “LIKE”), to = c(“CUSA”, “CUSL”)),
  paths(from = c(“CUSA”), to = c(“CUSL”)))

# Estimating the model
corp_rep_pls_model <- estimate_pls(
```

```
    data = corp_rep_data,
    measurement_model = corp_rep_mm,
    structural_model = corp_rep_sm,
    missing = mean_replacement,
    missing_value = "-99")

  # Summarize the model results
  summary_corp_rep <- summary(corp_rep_pls_model)
```

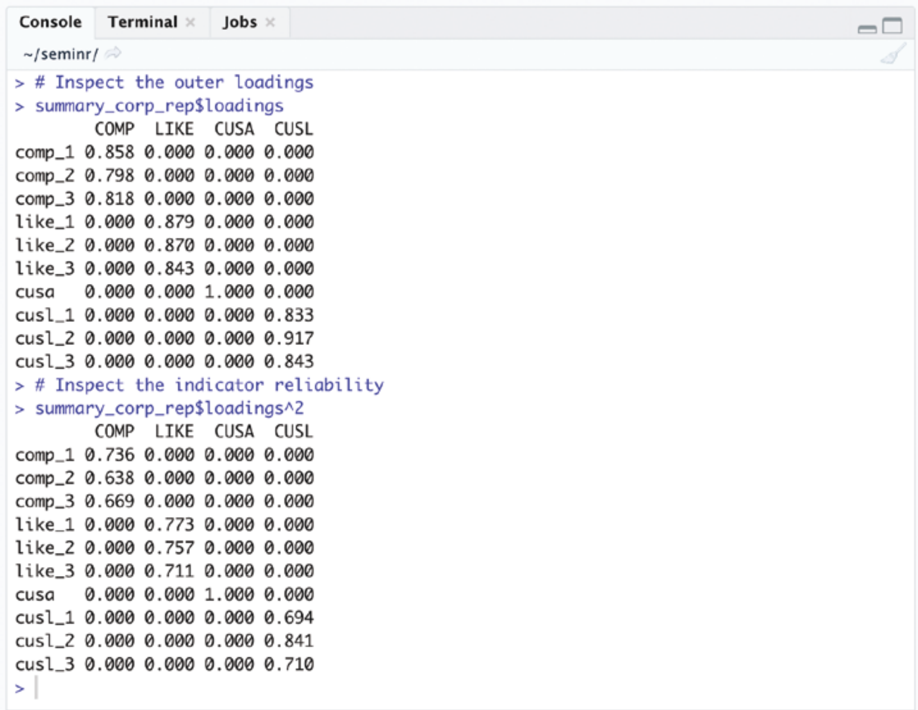

Note that the results are not automatically shown but can be extracted as needed from the **summary_corp_rep** object. For a reminder on what is returned from the **summary()** function applied to a SEMinR model and stored in the **summary_corp_rep** object, refer to ▶ Table 3.5. Before analyzing the results, we advise to first check if the algorithm converged (i.e., the stop criterion of the algorithm was reached and not the maximum number of iterations – see ▶ Table 3.4 for setting these arguments in the **estimate_pls()** function). To do so, it is necessary to inspect the **iterations** element within the **summary_corp_rep** object by using the **$** operator.

```
  # Iterations to converge
  summary_corp_rep$iterations
```

The upper part of ◘ Fig. 4.3 shows the code for loading the model, estimating the object **corp_rep_pls_model**, and summarizing the model to the **summary_corp_rep** object. The lower part of the figure shows the number of **iterations** that the PLS-SEM algorithm needed to converge. This number should be lower than the maximum number of iterations (e.g., 300). The bottom of ◘ Fig. 4.3 indicates that the algorithm converged after iteration 4.

If the PLS-SEM algorithm does not converge in fewer than 300 iterations, which is the default setting in most PLS-SEM software, the algorithm could not find a stable solution. This kind of situation almost never occurs. But if it does occur, there are two possible causes: (1) The selected stop criterion is set at a very small level (e.g., 1.0E-10 as opposed to the standard of 1.0E-7), so that small changes in the coefficients of the measurement models prevent the PLS-SEM algorithm from stopping, or (2) there are problems with the data and it needs to be checked carefully. For example, data problems may occur if the sample size is too small or if the responses to an indicator include many identical values (i.e., the same data points, which results in insufficient variability, error message is singular matrix).

In the following, we inspect the **summary_corp_rep** object to obtain statistics relevant for assessing the construct measures' internal consistency reliability, convergent validity, and discriminant validity. The simple corporate reputation model contains three constructs with reflective measurement models (i.e., *COMP*, *CUSL*, and *LIKE*) as well as a single-item construct (*CUSA*). For the reflective measure-



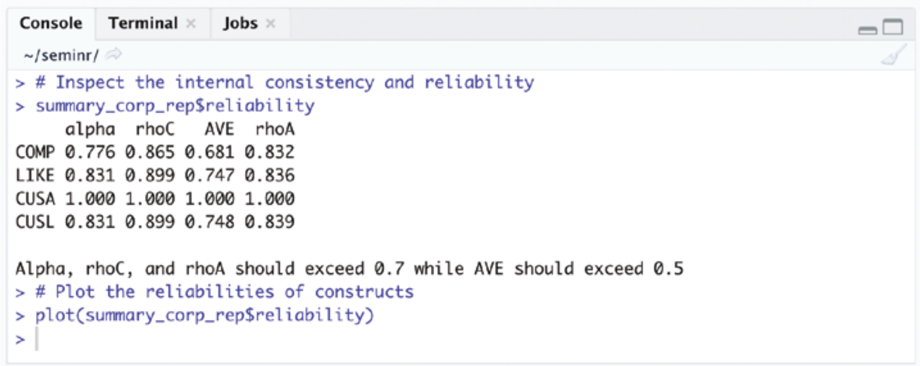
```
Console  Terminal ×  Jobs ×
~/seminr/
> # Inspect the outer loadings
> summary_corp_rep$loadings
        COMP  LIKE  CUSA  CUSL
comp_1 0.858 0.000 0.000 0.000
comp_2 0.798 0.000 0.000 0.000
comp_3 0.818 0.000 0.000 0.000
like_1 0.000 0.879 0.000 0.000
like_2 0.000 0.870 0.000 0.000
like_3 0.000 0.843 0.000 0.000
cusa   0.000 0.000 1.000 0.000
cusl_1 0.000 0.000 0.000 0.833
cusl_2 0.000 0.000 0.000 0.917
cusl_3 0.000 0.000 0.000 0.843
> # Inspect the indicator reliability
> summary_corp_rep$loadings^2
        COMP  LIKE  CUSA  CUSL
comp_1 0.736 0.000 0.000 0.000
comp_2 0.638 0.000 0.000 0.000
comp_3 0.669 0.000 0.000 0.000
like_1 0.000 0.773 0.000 0.000
like_2 0.000 0.757 0.000 0.000
like_3 0.000 0.711 0.000 0.000
cusa   0.000 0.000 1.000 0.000
cusl_1 0.000 0.000 0.000 0.694
cusl_2 0.000 0.000 0.000 0.841
cusl_3 0.000 0.000 0.000 0.710
>
```

Fig. 4.4 Indicator loadings and indicator reliability. (Source: authors' screenshot from RStudio)

ment model, we need to estimate the relationships between the reflectively measured constructs and their indicators (i.e., loadings). Figure 4.4 displays the results for the indicator loadings, which can be found by using the **$** operator when inspecting the **summary_corp_rep** object. The calculation of indicator reliability (Fig. 4.4) can be automated by squaring the values in the indicator loading table by using the ^ operator to square all values (i.e., ^2):

```
# Inspect the indicator loadings
summary_corp_rep$loadings
# Inspect the indicator reliability
summary_corp_rep$loadings^2
```

All indicator loadings of the reflectively measured constructs *COMP*, *CUSL*, and *LIKE* are well above the threshold value of **0.708** (Hair, Risher, Sarstedt, & Ringle, 2019), which suggests sufficient levels of indicator reliability. The indicator *comp_2* (loading, **0.798**) has the smallest indicator-explained variance with a value of **0.638** (= 0.798^2), while the indicator *cusl_2* (loading, **0.917**) has the highest explained variance, with a value of **0.841** (= 0.917^2) – both values are well above the threshold value of **0.5**.



```
Console   Terminal ×   Jobs ×
~/seminr/
> # Inspect the internal consistency and reliability
> summary_corp_rep$reliability
     alpha  rhoC   AVE  rhoA
COMP 0.776 0.865 0.681 0.832
LIKE 0.831 0.899 0.747 0.836
CUSA 1.000 1.000 1.000 1.000
CUSL 0.831 0.899 0.748 0.839

Alpha, rhoC, and rhoA should exceed 0.7 while AVE should exceed 0.5
> # Plot the reliabilities of constructs
> plot(summary_corp_rep$reliability)
>
```

◘ **Fig. 4.5** Construct reliability and convergent validity table. (Source: authors' screenshot from RStudio)

To evaluate the composite reliability of the construct measures, once again inspect the **`summary_corp_rep`** object by using **`$reliability`**:

```
# Inspect the composite reliability
summary_corp_rep$reliability
```

The internal consistency reliability values are displayed in a matrix format (◘ Fig. 4.5). With rho_A values of **0.832** (*COMP*), **0.839** (*CUSL*), and **0.836** (*LIKE*), all three reflectively measured constructs have high levels of internal consistency reliability. Similarly, the results for Cronbach's alpha (**0.776** for *COMP*, **0.831** for *CUSL*, and **0.831** for *LIKE*) and the composite reliability rho_c (**0.865** for *COMP*, **0.899** for *CUSL*, and **0.899** for *LIKE*) are above the 0.70 threshold (Hair et al., 2019), indicating that all construct measures are reliable. Note that the internal consistency reliability values of *CUSA* (**1.000**) must not be interpreted as an indication of perfect reliability – since *CUSA* is measured with a single item and its internal consistency reliability is by definition 1.

The results can also be visualized using a bar chart, requested by the **`plot()`** function on the **`summary_corp_rep$reliability`** object. This plot visualizes the reliability in terms of Cronbach's alpha, rho_A, and rho_C for all constructs. Note that the plots will be outputted to the plots panel window in RStudio (◘ Fig. 4.6):

```
# Plot the reliabilities of constructs
plot(summary_corp_rep$reliability)
```

The horizontal dashed blue line indicates the common minimum threshold level for the three reliability measures (i.e., 0.70). As indicated in ◘ Fig. 4.6, all Cronbach's alpha, rho_A, and rho_C values exceed the threshold.

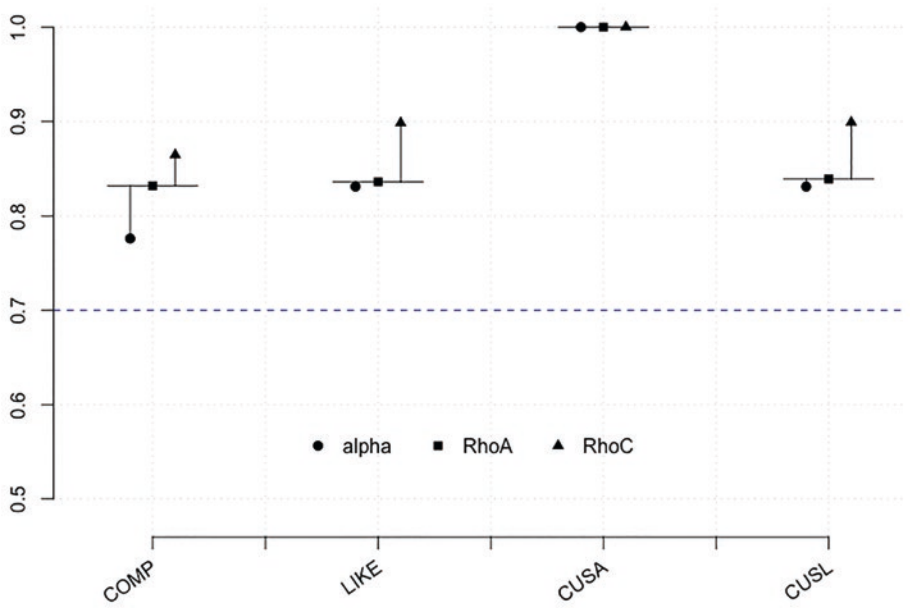


Fig. 4.6 Reliability charts. (Source: authors' screenshot from R)

Convergent validity assessment is based on the average variance extracted (AVE) values (Hair et al., 2019), which can also be accessed by **`summary_corp_rep$reliability`**. Figure 4.5 shows the AVE values along with the internal consistency reliability values. In this example, the AVE values of *COMP* (**0.681**), *CUSL* (**0.748**), and *LIKE* (**0.747**) are well above the required minimum level of 0.50 (Hair et al., 2019). Thus, the measures of the three reflectively measured constructs have high levels of convergent validity.

Finally, SEMinR offers several approaches to assess whether the construct measures empirically demonstrate discriminant validity. According to the Fornell–Larcker criterion (Fornell & Larcker, 1981), the square root of the AVE of each construct should be higher than the construct's highest correlation with any other construct in the model (this notion is identical to comparing the AVE with the squared correlations between the constructs). These results can be outputted by inspecting the **`summary_corp_rep`** object and **`validity`** element for the **`fl_criteria`**:

```
# Table of the FL criteria
summary_corp_rep$validity$fl_criteria
```

Figure 4.7 shows the results of the Fornell–Larcker criterion assessment with the square root of the reflectively measured constructs' AVE on the diagonal and

```
Console   Terminal ×   Jobs ×
~/seminr/
> # Table of the FL criteria
> summary_corp_rep$validity$fl_criteria
      COMP  LIKE  CUSA  CUSL
COMP 0.825     .     .     .
LIKE 0.645 0.864     .     .
CUSA 0.436 0.528 1.000     .
CUSL 0.450 0.615 0.689 0.865

FL Criteria table reports square root of AVE on the diagonal and construct correlations on
 the lower triangle.
>
```

Fig. 4.7 Fornell–Larcker criterion table. (Source: authors' screenshot from RStudio)

```
Console   Terminal ×   Jobs ×
~/seminr/
> # HTMT Ratio
> summary_corp_rep$validity$htmt
      COMP  LIKE  CUSA CUSL
COMP     .     .     .    .
LIKE 0.780     .     .    .
CUSA 0.465 0.577     .    .
CUSL 0.532 0.737 0.755    .
>
```

Fig. 4.8 HTMT result table. (Source: authors' screenshot from RStudio)

the correlations between the constructs in the off-diagonal position. For example, the reflectively measured construct *COMP* has a value of **0.825** for the square root of its AVE, which needs to be compared with all correlation values in the column of *COMP* (i.e., **0.645**, **0.436**, and **0.450**). Note that for CUSA, the comparison makes no sense, as the AVE of a single-item construct is **1.000** by design. Overall, the square roots of the AVEs for the reflectively measured constructs *COMP* (**0.825**), *CUSL* (**0.865**), and *LIKE* (**0.864**) are all higher than the correlations of these constructs with other latent variables in the PLS path model.

Note that while frequently used in the past, the Fornell–Larcker criterion does not allow for reliably detecting discriminant validity issues. Specifically, in light of the Fornell–Larcker criterion's poor performance in detecting discriminant validity problems (Franke & Sarstedt, 2019; Henseler et al., 2015), any violation indicated by the criterion should be considered a severe issue. The primary criterion for discriminant validity assessment is the HTMT criterion, which can be accessed by inspecting the **`summary_corp_rep()`** object and **`validity`** element for the **`$htmt`**.

```
# HTMT criterion
summary_corp_rep$validity$htmt
```

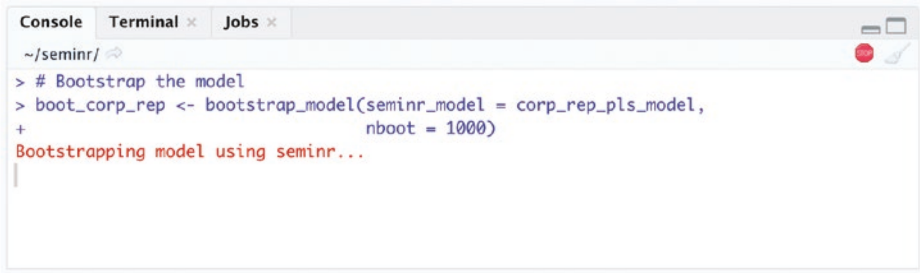


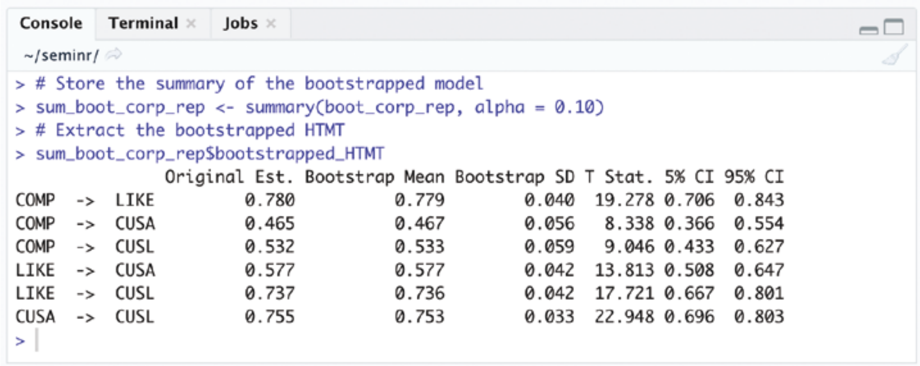
Fig. 4.9 Bootstrapping processing. (Source: authors' screenshot from RStudio)

Figure 4.8 shows the HTMT values for all pairs of constructs in a matrix format. As can be seen, all HTMT values are clearly lower than the more conservative threshold value of 0.85 (Henseler et al., 2015), even for *CUSA* and *CUSL*, which, from a conceptual viewpoint, are very similar. Recall that the threshold value for conceptually similar constructs, such as *CUSA* and *CUSL* or *COMP* and *LIKE*, is 0.90.

In addition to examining the HTMT values, researchers should test whether the HTMT values are significantly different from 1 or a lower threshold, such as 0.9 or even 0.85. This analysis requires computing bootstrap confidence intervals obtained by running the bootstrapping procedure. To do so, use the **bootstrap_model()** function and assign the output to an object, such as **boot_corp_rep**. Then, run the **summary()** function on the **boot_corp_rep** object and assign it to another object, such as **sum_boot_corp_rep**. In doing so, we need to set the significance level from 0.05 (default setting) to 0.10 using the ***alpha*** argument. In this way, we obtain 90% two-sided bootstrap confidence intervals for the HTMT values, which is equivalent to running a one-tailed test at 5%.

```
# Bootstrap the model
boot_corp_rep <- bootstrap_model(seminr_model = corp_rep_pls_
model, nboot = 1000)
sum_boot_corp_rep <- summary(boot_corp_rep, alpha = 0.10)
```

► Chapter 5 includes a more detailed introduction to the bootstrapping procedure and the argument settings. Bootstrapping should take a few seconds, since it is a processing-intensive operation. As the bootstrap computation is being performed, a red **STOP** indicator should show in the top-right corner of the console (Fig. 4.9). This indicator will automatically disappear when computation is complete, and the console will display "**SEMinR Model successfully bootstrapped**."



```
Console  Terminal ×  Jobs ×
~/seminr/
> # Store the summary of the bootstrapped model
> sum_boot_corp_rep <- summary(boot_corp_rep, alpha = 0.10)
> # Extract the bootstrapped HTMT
> sum_boot_corp_rep$bootstrapped_HTMT
               Original Est. Bootstrap Mean Bootstrap SD T Stat. 5% CI 95% CI
COMP  ->  LIKE         0.780          0.779        0.040  19.278 0.706  0.843
COMP  ->  CUSA         0.465          0.467        0.056   8.338 0.366  0.554
COMP  ->  CUSL         0.532          0.533        0.059   9.046 0.433  0.627
LIKE  ->  CUSA         0.577          0.577        0.042  13.813 0.508  0.647
LIKE  ->  CUSL         0.737          0.736        0.042  17.721 0.667  0.801
CUSA  ->  CUSL         0.755          0.753        0.033  22.948 0.696  0.803
>
```

Fig. 4.10 Bootstrapped results and confidence intervals for HTMT. (Source: authors' screenshot from RStudio)

After running bootstrapping, access the bootstrapping confidence intervals of the HTMT by inspecting the **`$bootstrapped_HTMT`** of the **`sum_boot_corp_rep`** variable:

```
# Extract the bootstrapped HTMT
sum_boot_corp_rep$bootstrapped_HTMT
```

The output in Fig. 4.10 displays the original ratio estimates (column: **Original Est.)**, bootstrapped mean ratio estimates (column: **Bootstrap Mean**), bootstrap standard deviation (column: **Bootstrap SD**), bootstrap *t*- statistic (column: **T Stat.**), and 90% confidence interval (columns: **5% CI** and **95% CI**, respectively) as produced by the percentile method. Note that the results in Fig. 4.10 might differ slightly from your results due to the random nature of the bootstrapping procedure. The differences in the overall bootstrapping results should be marginal if you use a sufficiently large number of bootstrap subsamples (e.g., 10,000). The columns labeled **5% CI** and **95% CI** show the lower and upper boundaries of the 90% confidence interval (percentile method). As can be seen, the confidence intervals' upper boundaries, in our example, are always lower than the threshold value of 0.90. For example, the lower and upper boundaries of the confidence interval of HTMT for the relationship between *COMP* and *CUSA* are **0.366** and **0.554**, respectively (again, your values might look slightly different because bootstrapping is a random process). To summarize, the bootstrap confidence interval results of the HTMT criterion clearly demonstrate the discriminant validity of the constructs and should be favored above the inferior Fornell–Larcker criterion.

Summary
The goal of reflective measurement model assessment is to ensure the reliability and validity of the construct measures and therefore provides support for the suitability of their inclusion in the path model. The key criteria include indicator reliability, internal consistency reliability (Cronbach's alpha, reliability rho_A, and composite reliability rho_C), convergent validity, and discriminant validity. Convergent validity implies that a construct includes more than 50% of the indicator's variance and is being evaluated using the AVE statistic. Another fundamental element of validity assessment concerns establishing discriminant validity, which ensures that each construct is empirically unique and captures a phenomenon not represented by other constructs in a statistical model. While the Fornell–Larcker criterion has long been the primary criterion for discriminant validity assessment, more recent research highlights that the HTMT criterion should be the preferred choice. Researchers using the HTMT should use bootstrapping to derive confidence intervals that allow assessing whether the values significantly differ from a specific threshold. Reflective measurement models are appropriate for further PLS-SEM analyses if they meet all these requirements.

Exercise

In this exercise, we once again call upon the influencer model and dataset described in the exercise section of ► Chap. 3. The data is called **`influencer_data`** and consists of 222 observations of 28 variables. The influencer model is illustrated in ► Fig. 3.10, and the indicators are described in ► Tables 3.9 and 3.10.

1. Load the influencer data, reproduce the influencer model in SEMinR syntax, and estimate the model.
2. Focus your attention on the three reflectively measured constructs product liking (*PL*), perceived quality (*PQ*), and purchase intention (*PI*). Evaluate the construct measures' reliability and validity as follows:
   (a) Do all three constructs meet the criteria for indicator reliability?
   (b) Do all three constructs meet the criteria for internal consistency reliability?
   (c) Do these three constructs display sufficient convergent validity?
   (d) Do these three constructs display sufficient discriminant validity?

References

Aguirre-Urreta, M. I., & Rönkkö, M. (2018). Statistical inference with PLSc using bootstrap confidence intervals. *MIS Quarterly, 42*(3), 1001–1020.

Diamantopoulos, A., Sarstedt, M., Fuchs, C., Wilczynski, P., & Kaiser, S. (2012). Guidelines for choosing between multi-item and single-item scales for construct measurement: A predictive validity perspective. *Journal of the Academy of Marketing Science, 40*(3), 434–449.

Dijkstra, T. K. (2010). Latent variables and indices: Herman Wold's basic design and partial least squares. In V. Esposito Vinzi, W. W. Chin, J. Henseler, & H. Wang (Eds.), *Handbook of partial least squares: Concepts, methods and applications Springer Handbooks of Computational Statistics Series* (Vol. II, pp. 23–46). Berlin: Springer.

Dijkstra, T. K. (2014). PLS' Janus face–response to professor Rigdon's 'rethinking partial least squares modeling: In praise of simple methods. *Long Range Planning, 47*(3), 146–153.
Dijkstra, T. K., & Henseler, J. (2015). Consistent partial least squares path modeling. *MIS Quarterly, 39*(2), 297–316.
Fornell, C., & Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research, 18*(1), 39–50.
Franke, G., & Sarstedt, M. (2019). Heuristics versus statistics in discriminant validity testing: A comparison of four procedures. *Internet Research, 29*(3), 430–447.
Hair, J. F., Hult, T., Ringle, C. M., & Sarstedt, M. (2022). *A primer on partial least squares structural equation modeling (PLS-SEM)* (3rd ed.). Thousand Oaks: Sage.
Henseler, J., Ringle, C. M., & Sarstedt, M. (2015). A new criterion for assessing discriminant validity in variance-based structural equation modeling. *Journal of the Academy of Marketing Science, 43*(1), 115–135.
Hair, J. F., Risher, J. J., Sarstedt, M., & Ringle, C. M. (2019). When to use and how to report the results of PLS-SEM. *European Business Review, 31*(1), 2–24.
Hulland, J. (1999). Use of partial least squares (PLS) in strategic management research: A review of four recent studies. *Strategic Management Journal, 20*(2), 195–204.
Jöreskog, K. G. (1971). Simultaneous factor analysis in several populations. *Psychometrika, 36*(4), 409–426.
Radomir, L., & Moisescu, O. I. (2019). Discriminant validity of the customer-based corporate reputation scale: Some causes for concern. *Journal of Product & Brand Management, 29*(4), 457–469.
Streukens, S., & Leroi-Werelds, S. (2016). Bootstrapping and PLS-SEM: A step-by-step guide to get more out of your bootstrapping results. *European Management Journal, 34*(6), 618–632.
Trizano-Hermosilla, I., & Alvarado, J. M. (2016). Best alternatives to Cronbach's alpha reliability in realistic conditions: Congeneric and asymmetrical measurements. *Frontiers in Psychology, 7*, 769.

Suggested Reading

Hair, J. F., Hult, T., Ringle, C. M., & Sarstedt, M. (2022). *A primer on partial least squares structural equation modeling (PLS-SEM)* (3rd ed.). Thousand Oaks: Sage.
Hair, J. F., Risher, J. J., Sarstedt, M., & Ringle, C. M. (2019). When to use and how to report the results of PLS-SEM. *European Business Review, 31*(1), 2–24.
Hair, J. F., Sarstedt, M., & Ringle, C. M. (2019). Rethinking some of the rethinking of partial least squares. *European Journal of Marketing, 53*(4), 566–584.
Sarstedt, M., Hair, J. F., & Ringle, C. M. (2021). Partial least squares structural equation modeling. In C. Homburg, M. Klarmann, & A. Vomberg (Eds.), *Handbook of Market Research*. Cham: Springer.