FactorFlow: A Visual Analytics Workspace with Large Language Model–Assisted Interpretation for Factor Analysis

Justin Philip Tuazon[1] Joemari Olea[2] Richelle Ann Juayong[1]

[1]Department of Computer Science, University of the Philippines Diliman
[2]Department of Educational Psychology, University of Texas at Austin

July 2026

Abstract

In exploratory factor analysis (EFA), one typically aims to extract and describe a small number of *factors* (i.e., latent variables) based on the relationships among numerous *manifest variables*. In practice, performing EFA entails examining different factor models (and rotations) to identify the underlying latent structure. Now, the primary criterion for evaluating a factor model is interpretability. That is, the preferred model is the one that yields a meaningful, coherent, and theoretically defensible factor structure. However, gauging a model's interpretability is not a trivial task, as it is subjective and often requires keeping track of large amounts of information simultaneously. Because of this, researchers typically employ various *visualizations* to interpret models and determine the "best" one. Hence, we introduce *FactorFlow*, a visual analytics workspace for performing EFA end-to-end. Using FactorFlow, one can fit and rotate factor models, perform model diagnostics, and more. The main component of the tool is a dashboard with a comprehensive set of interactive visualizations, where a user can easily dissect a factor model and even compare two models side-by-side at the same time. Moreover, several large language models are integrated with FactorFlow, enabling the user to generate and assess automated factor interpretations written in natural language. Ultimately, FactorFlow can enable the researcher to efficiently and effectively perform EFA. Finally, we conducted a usability study to identify strengths and weaknesses, and capture feedback.

Keywords: factor analysis, factor model, interpretability, visualization, large language model

Contents

The latest version of FactorFlow, V3.7.2, is hosted here: https://factorflow-efa.streamlit.app/.

1 Background

1.1 Factor Analysis

In many cases, researchers and practitioners usually study variables that are *directly* observable or measurable. Examples of such kind of variables include the height of a person in meters, the speed of an animal in meters per second, and the response of a customer to a Likert-type item (e.g., "Strongly Agree"). As mentioned in Mulaik (2010), such variables are called *manifest variables*. However, there are also cases where the variables of interest are *not* directly measurable. For instance, how would you measure the anxiety of a student when they are about to take an exam? How would you measure the "voting ideology" of citizens in a country? In contrast, such variables are referred to as *latent variables*, also as mentioned in Mulaik (2010). As opposed to manifest variables, latent variables are *abstract* concepts. While we can use a ruler to quickly measure length, we do not have a tool with which we can directly measure "happiness", for instance. Clearly, we need "special" methods to deal with latent variables.

On this note, there are several approaches that we can take to study latent variables or phenomena. One such method is *factor analysis*. Usually, factor analysis is used when the manifest variables and the factors are both numerical and continuous, but it can also be applied in other cases. Notably, it is often applicable to ordinal data (e.g., Liket-type data), as described in Robitzsch (2020). One can also use specialized correlation matrices or procedures to apply factor analysis to ordinal data (Gorusch, 1983). Given these capabilities, the method is widely used and has been utilized in numerous fields. Indeed, it has become one of the most popular methods in psychometrics, the field where the method originated (Mulaik, 2010). However, beyond psychometrics, factor analysis has also been used in several other fields. For example, Yilmaz et al. (2024) leveraged EFA to identify the factors that constitute sustainable manure utilization in farming. Meanwhile, Ledesma et al. (2021) provided a review of the application of EFA in transportation research (e.g., to study driver behavior) and established guidelines for using such method in the field.

Now, in factor analysis, the latent variables are also called as *factors*. Essentially, factor analysis aims to identify and characterize the factors (i.e., the latent structure) based on the observed relationships among the manifest variables (i.e., their correlations), as discussed in Mulaik (2010). Usually, the number of factors is much smaller than the number of manifest variables. Moreover, using factor analysis, one can "measure" the latent variables through estimating *factor scores* (Gorusch, 1983). Here, we focus on *exploratory factor analysis* (EFA), which is primarily a way to come up with hypotheses about the latent structure. As its name implies, EFA involves "exploring" the data and identifying the factors. This is in contrast to *confirmatory factor analysis* (CFA), which is used to *test* the generated hypotheses and establish the factors, as discussed in Gorusch (1983).

At this point, we will briefly go over some foundational concepts in EFA. As described in Johnson and Wichern (2007), the common-factor model $\mathscr{F}$, which we estimate using factor analysis, is given by

$$\underline{\boldsymbol{X}}_{M\times 1} = \underline{\boldsymbol{\mu}}_{M\times 1} + \underline{\boldsymbol{L}}_{M\times T}\underline{\boldsymbol{F}}_{T\times 1} + \underline{\boldsymbol{\varepsilon}}_{M\times 1},$$

where M is the number of manifest variables, $T\,(\ll M)$ is the number of factors,

$$\underline{\boldsymbol{X}} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_M \end{pmatrix},\ \underline{\boldsymbol{\mu}} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_M \end{pmatrix},\ \underline{\boldsymbol{L}} = \begin{pmatrix} l_{1,1} & \dots & l_{1,T} \\ \vdots & \ddots & \vdots \\ l_{M,1} & \dots & l_{M,T} \end{pmatrix},\ \underline{\boldsymbol{F}} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_M \end{pmatrix},\ \text{and } \underline{\boldsymbol{\varepsilon}} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_M \end{pmatrix}$$

Here, $X_1, \dots, X_M$ are the manifest variables while $F_1, F_2, \dots, F_T$ are the factors. Moreover, $\underline{\boldsymbol{L}}$ is called the *loading matrix*, with $l_{i,j}$ as the loading of X_i on F_j. Note that the loading is the covariance (correlation, if the manifest variable was standardized) of the manifest variable with the factor. In line with this, the loading (or factor loading) is a measure of how much the factor generalizes to the manifest variable (in other words, how much the factor "explains" the manifest variable), as mentioned in Gorusch (1983). Finally, $\underline{\boldsymbol{\mu}}$ is the mean vector of $\underline{\boldsymbol{X}}$ and $\varepsilon_1, \dots, \varepsilon_M$ are the error terms. There additional assumptions imposed on the variables, which also depend on the class of models considered (e.g., orthogonal factor models). We will not discuss these anymore but details of such can be found in Mulaik (2010) and Gorusch (1983).

As mentioned earlier, the primary goal of EFA is to determine the latent structure. As such, one needs to *interpret* (i.e., assign a meaning to) the factors, which is not a trivial task. As discussed by Johnson and Wichern (2007), the success of factor analysis depends on a "WOW" criterion, where the factor model is "good" if it is substantially meaningful. On the same note, as Lara et al. (1992) stated, the one that is more interpretable between two statistically valid factor models is preferred because it provides greater substantive utility and a more meaningful insight into the underlying structure of the data. Clearly, this is a difficult task since interpretability cannot be easily quantified and interpretation itself can be subjective.

Operationally, factors are typically interpreted by examining the loading matrix $\underline{\boldsymbol{L}}$. When the absolute value of the loading $l_{i,j}$ is large, the manifest variable X_i is considered to be strongly associated with factor F_j. Thus, the substantive meaning of X_i can be used to inform the interpretation of F_j. For example, suppose that all manifest variables come from Likert-type items in a questionnaire (an example questionnaire is shown in Table 1). To interpret a factor, one looks at the collection of questions that correspond to high-loading manifest variables for that factor. If the questions all reflect anxiety-related content, then the factor may be interpreted as an "anxiety score" or "anxiety component". In contrast, if the high-loading items reflect several unrelated constructs, then the factor may lack a coherent substantive interpretation.

Finally, note that the factor model $\mathscr{F}$ is not *identifiable*, as it is rotationally indeterminate (i.e., rotating the loading matrix yields and equally valid model with the same model-implied covariance matrix). This adds another challenge to the identification and interpretation of the factors, as the researcher needs to identify an orientation of the loadings that "make sense". Overall, the process of identification and interpretation depends heavily on the researcher's judgment and how they "process" the information (e.g., loadings).

Table 1: Example Questionnaire with Likert-type Items

Statement	**1** Strongly Disagree	**2** Disagree	**3** Neutral	**4** Agree	**5** Strongly Agree
I feel energized when I start my day.	☐	☐	☐	☐	☐
I find it easy to concentrate on tasks.	☐	☐	☐	☐	☐

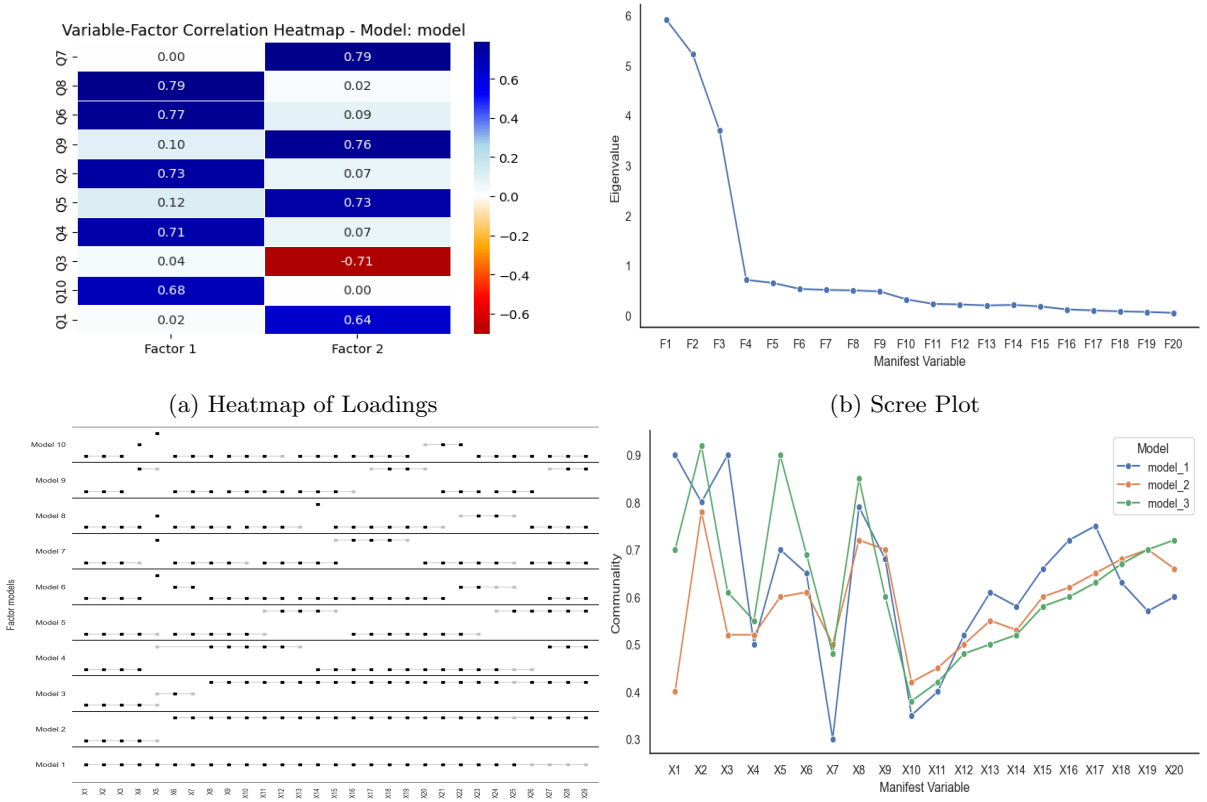
1.2 Visualizations in Exploratory Factor Analysis

Now, making (or attempting to make) interpretations based on the loading matrix can be difficult, as it requires identifying patterns from a large number of loadings and looking at the loading matrix at different "perspectives" (e.g., in terms of magnitude, in terms of trends). For example, if there are 56 manifest variables and 7 factors, the number of values (loadings) one has to look at simultaneously is $56 \times 7 = 392$. This becomes even more complicated when the researcher needs to compare multiple candidate models (which is almost always the case). As such, *visualizations* are critical when performing EFA and several types are commonly used in factor analysis.

Perhaps the most important and common one is the *heatmap*, an example of which is shown in Figure 1(a). The figure shows a heatmap of the loading matrix but heatmaps can also be constructed for other "parts" of the factor model. For instance, it is common to construct heatmaps for the sample correlation matrix, the communalities, and the sampling adequacies. In the case of heatmaps for loadings, one can "dichotomize" the loadings such that a loading is colored if its absolute value meets or exceeds a threshold, and is uncolored otherwise. In any case, the heatmap makes it obvious which values are large and which are not. In the case of the loading matrix, this makes it easier to identify which manifest variables should be considered when interpreting the factor.

Another type of visualization commonly used in EFA is the *scree plot* proposed by Cattell (1966), which provides a heuristic for identifying the ideal number of factors in the model. An example of a scree plot is shown in Figure 1(b). This plot shows the eigenvalues of the (sample) correlation matrix, ordered from largest to smallest. Using the scree plot, one can use several heuristics for deciding on the optimal number of factors. For instance, one can look at the number of eigenvalues that exceed a certain threshold, as in the case of the eigenvalues-greater-than-one rule from Kaiser (1960, 1961), or look at the "elbow" of the plot based on the criterion from Cattell (1966), both of which are described in Mulaik (2010).

Furthermore, Klinke and Wagner (2008) illustrated the *factor model plot* and the *communality plot*. In the factor model plot, the x-axis tick labels refer to the manifest variables and the y-axis tick labels refer to the different factor models. In a given row or factor model, the manifest variable is marked with a square if its loading is large enough and squares are connected with segments if they are explained by the same factor. Thus, the factor model plot can make it easy to compare different factor models (or different rotations). In Figure 1(c), variables $X6$, $X7$, $X8$, $X9$, $X10$, and $X11$ are explained by the same factor in factor model 5. Meanwhile, the communality plot, an example of which is Figure 1(d), shows which manifest variables have low or high communalities, where different lines (distinguished perhaps by color or line style) correspond to different factor models. The communality of a manifest variable is the amount or portion of the variance of the manifest variable that is explained by the factor model. Ideally, a manifest variable should have a high communality.



(a) Heatmap of Loadings

(b) Scree Plot

(c) Factor Model Plot

(d) Communality Plot

Figure 1: Common Visualizations in Factor Analysis

Then, given the exploratory nature of EFA and the numerous visualizations employed, it is obvious that a *dashboard* can be useful for performing EFA. In fact, Lu and Wang (2024) developed *FAVis*, a visual analytics dashboard for exploratory factor analysis in psychological research. A screenshot of FAVis is shown in Figure 2.

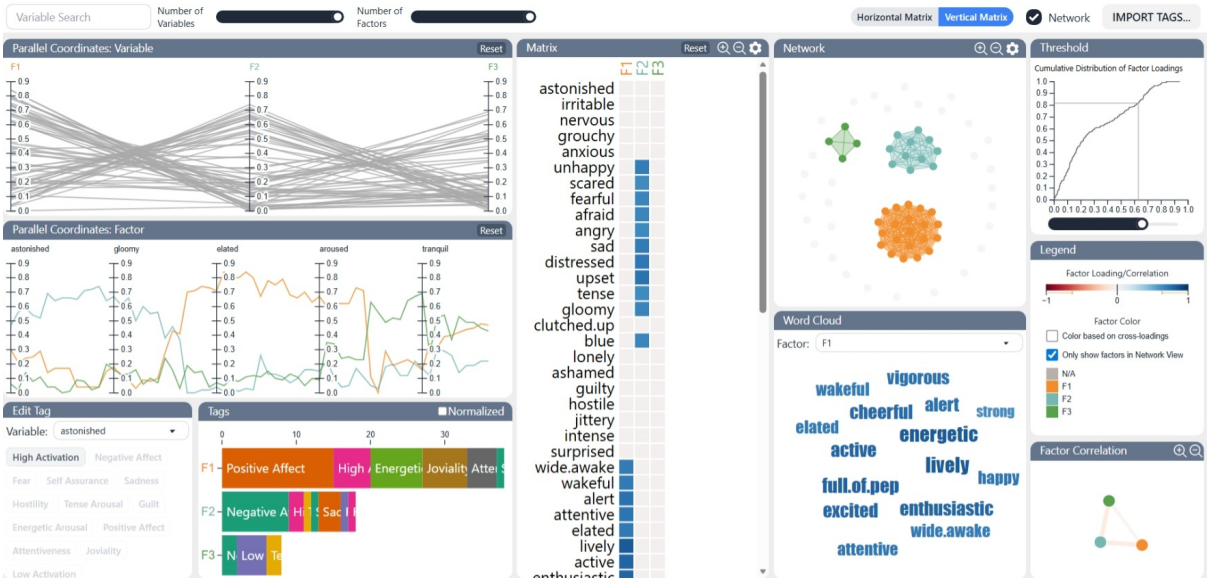


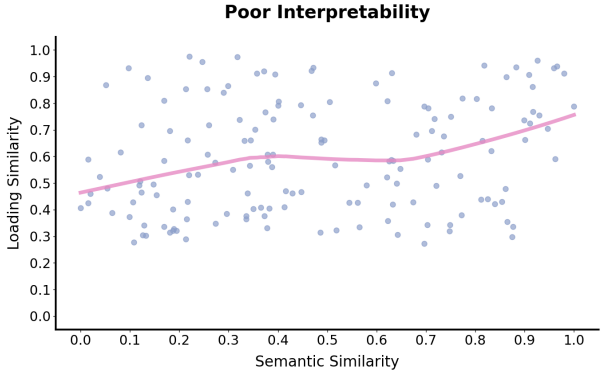
Figure 2: FAVis by Lu and Wang (2024)

Lu and Wang (2024) included *parallel coordinates plots* for the loadings, one for comparing loadings across manifest variables and another for comparing loadings across factors. Using such plots, it is easy to spot manifest

variables (or factors) with similar loadings (i.e., those that have parallel or coincident lines are similar), even at a relative scale (e.g., when lines are not necessarily coincident but are parallel). They also added a *network graph*, which shows the *cross-loadings* (i.e., manifest variables with large loadings on more than one factor[1].). A stacked bar chart is also available to help relate a theory to the model (i.e., adding *a priori* tags to the manifest variables and looking at how much of each tag does each factor have). A key feature highlighted by Lu and Wang (2024) is the capability to change and select the optimal threshold for binarizing the loading matrix. This interactivity is certainly important for performing EFA as one is able to quickly interpret the model as the threshold changes and ultimately find a configuration that is ideal.

Finally, there are also nontraditional or new visualizations for EFA. One example is the *interpretability plot* proposed by Tuazon et al. (2025). The interpretability plot directly visualizes the interpretability (at least, a definition of it) of a factor model by showing how much the loadings align with the semantics of the questions (or in general, with *a priori* information or expectations). Examples of the interpretability plot are shown in Figure 3. It is a scatter plot that shows the relationship between the *loading similarity* and the *semantic similarity* (or prior similarity). Ideally, the overall trend should be monotonically increasing because such indicates that the loadings "agree" with the meanings of the questions (in the case of a questionnaire), or with the *a priori* information or expectations (when using the more general prior similarities), which then corresponds to a meaningful and semantically coherent factor structure. Essentially, in the case of questionnaires, the model is interpretable if the "clustering" of manifest variables based on the loading matrix is consistent with the "clustering" of items based on semantic similarities, yielding semantically coherent factors.

(a) Poor Interpretability



(b) Good Interpretability

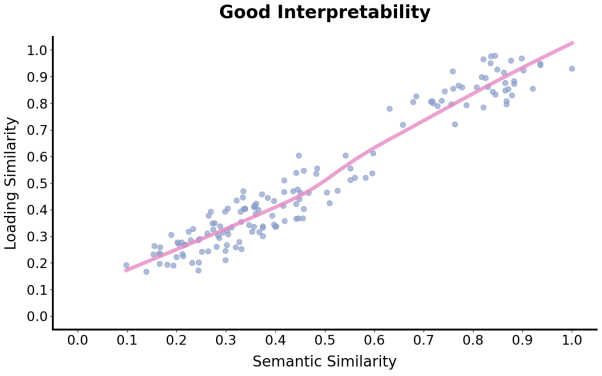


Figure 3: Interpretability Plots

1.3 FactorFlow

Indeed, visualizations are important when performing EFA and due to the interactive nature of EFA, dashboards can be beneficial for EFA researchers. While FAVis is an excellent dashboard for EFA, we explore several areas where improvements can be made:

1. As noted by Lu and Wang (2024), FAVis does not support visualizing and directly evaluating multiple factor models (or loading matrices) at the same time. Because of the exploratory nature of EFA, it is often the case that the researcher needs to compare different rotations and models to find the "ideal" one.
2. Instead of focusing exclusively on providing visual analytics for the last step of interpreting the factor model, we aim to cover the end-to-end workflow for EFA. For example, we consider visualizations and figures for diagnostics (e.g., evaluating communalities, deciding on the number of factors, basic statistics), and a mechanism for fitting factor models with different configurations.
3. The set of visualizations available in the dashboard can potentially be expanded. For instance, additional toggles or filters can be included, and even new visualizations (such as the interpretability plot, discussed earlier) can be added.

With the rise in the popularity and availability of powerful large language models, it is also worthwhile to explore how large language models can be integrated in the EFA workflow. For instance, it can possibly be used

[1] *Sparsity*, which partially refers to the absence of cross-loadings, is often taken as a proxy for interpretability, described in Browne (2001). A manifest variable should have a high loading for only one factor (i.e., no cross-loadings), which translates to the network graph as non-overlapping networks.

to generate automated factor interpretations, making the process of interpreting a model much easier. Thus, this paper introduces *FactorFlow*, which aims to address the opportunities described previously and more.

Figure 4: FactorFlow Logo

Using FactorFlow, researchers can perform EFA end-to-end, from performing diagnostics to fitting factor models and downloading final results. FactorFlow is a visual analytics workspace with large language model-assisted interpretation for exploratory factor analysis, with the following key features:

1. Supports core exploratory factor analysis, implementing classical estimation methods, rotations, and visualizations
2. Provides implementations of pairwise target rotation and interpretability plots from Tuazon et al. (2025) for going beyond the classical methods
3. Integrates large language models to provide automated language-model-assisted interpretation and semantic guidance
4. Enables deep analysis workflows, with multiple tabs available for basic statistics, hypothesis testing, diagnostics, model comparisons, and exploratory deep dives

The general goal of FactoFlow is to enable users to perform EFA end-to-end. On the other hand, the specific goals of FactorFlow are as follows, some of which were adopted and modified from Lu and Wang (2024):

1. **G1 - Enable users to fit various factors models**. A user should be able to fit a factor model with customizable specifications based on their dataset. For example, factor models can differ in terms of estimation method, rotation method, number of factors, included manifest variables, and so on.
2. **G2 - Support various correlation types**. Pearson correlation may not be the best type of correlation for ordinal data. Thus, there should also be an option to use polychoric correlation for ordinal-level variables.
3. **G3 - Enable users to perform model diagnostics and explore raw data**. A user should be able to asses the goodness-of-fit of an estimated factor model, as well as the adequacy of the sample data for factor analysis. Likewise, general summary information about the raw data can be useful for tangent hypotheses.
4. **G4 - Effectively present general information about fitted models**. Factor correlations, factor scores, and other information (e.g., correlation matrix) are important considerations when examining a factor model.
5. **G5 - Effectively present useful information for factor interpretation**. Associations between manifest variables and factors, which are reflected in the loading matrix, are the primary figures used for interpreting a factor model.
6. **G6 - Enable users to select a good threshold for binarizing factor loadings**. Sufficiently small loadings do not affect a factor's interpretation but can possibly make the process more difficult. Thus, binarizing (i.e., setting sufficiently small loadings to 0) can be useful but there must be a balance between interpretability and model fit.
7. **G7 - Provide information about the sparsity of a factor model**. Sparsity in the loading matrix is sometimes used as a proxy for interpretability in factor models.
8. **G8 - Provide a way to encode an existing theory or *a priori* information**. An existing theory can be used to attempt interpreting factors in a factor model, where one can gauge how much of each hypothesized construct is reflected in each factor. However, beyond classical rotation methods, *a priori* information about the loading matrix can also be used to influence the loadings themselves.
9. **G9 - Assist the user with interpreting factor models through generated natural language interpretations**. A large language model can be used to interpret a factor model based on the loading matrix and the associated statements.
10. **G10 - Provide a mechanism to compare multiple models**. EFA generally involves comparing multiple candidate factor models. Thus, there should be a way to easily compare two models side-by-side.
11. **G11 - Provide a comprehensive set of visualizations for assessing a factor model**. A factor model can be visualized in terms of the loadings, loading comparisons across manifest variables or factors, agreement with *a priori* information or theory, sparsity, and goodness-of-fit.

2 Methodology

2.1 Technology

The source code for FactorFlow was primarily written in `Python 3`. FactorFlow is a *Streamlit* application and thus, uses the `streamlit` package. Streamlit is an open-source Python framework for building data applications (Streamlit, n.d.). Using Streamlit, one can quickly build a web application from their data script. For the interactive visualizations inside the application, the primary visualization library used was `plotly`.

Now, for most calculations related to factor analysis, the `interpretablefa` package was used, which extends the `factor_analyzer` package. Several other scientific packages such as `numpy`, `pandas`, `scipy`, and `statsmodels` were also utilized. Custom `JavaScript` was also written inside the Streamlit application to handle features related to natural language processing. The `TensorFlow.js`[2] was used to load the *universal sentence encoder* from Cer et al. (2018) while the Groq application programming interface (API) was leveraged using the `groq` Python package for generating automated interpretations.

The application's source code can be found on GitHub here: https://github.com/jptuazon/factorflow. As of writing, FactorFlow is hosted on the free tier of Streamlit cloud and can be accessed here: https://factorflow-efa.streamlit.app/.

2.2 Usability Survey

An usability study was also conducted to gather feedback about the application. Similar to Lu and Wang (2024), the survey was informal in the sense that it had only 5 participants, who were all selected via convenience sampling. Still, the survey was extremely useful as its primary goal was to gather feedback and not conduct statistical inference.

The survey consisted of several sections. One gathered information about the respondent's demographics, which also ensured that the respondent had experience with EFA. Then, other sections covered the tool usage experience, the app's outputs and results, and the general usability. Most questions were Likert-type items, where a larger number indicates a "better" response (from 1 to 5). However, several open-ended questions were also included to gather qualitative data. Note that the version evaluated by the testers was `3.5.7`.

Before each respondent answered the questionnaire, they were provided an orientation, where the project context was introduced and a quick demonstration of the app was done. The application contains a guide or a tutorial, which includes sample datasets. The respondents were tasked to try out the application by following the guide (and possibly exploring further) and asynchronously complete the questionnaire after.

3 Design, Features, and Usage

3.1 General Components

The application has several parts. First, it has a ***Menu* sidebar**. There are four different **panels** in the sidebar:

1. **NLP Models**. This shows the statuses of the NLP models used in the app. You can also configure the large language model here.
2. **Data**. This is where you can upload your datasets. You can also examine your dataset in this panel by clicking the Stats button. High-level information is also displayed here.
3. **Factor Models**. You can fit new factor models using this panel. It also shows the factor models that you have fitted and saved so far. You can examine each saved models by clicking View. High-level information is also displayed here.
4. **Settings**. You can configure the application to match your preferences. You can change things like the color palettes used and the chart styles.

Moreover, it has several pages:

1. **Overview**. This is the home page, where you can find general information about the app.

[2]The universal sentence encoder can also be directly imported via `Python` but due to limited resources in the cloud, the computations were offloaded to the front-end via `JavaScript`.

2. **Diagnostics**. This is where you can examine diagnostics for factor analysis. Here, you can determine how many factors to use or which manifest variables have low communalities, and so on.
3. **Dashboard**. In this tab, you can see multiple figures and visualizations for assessing the factor model. You can also compare two factor models at the same time here.
4. **About**. You can find development details for the app here.
5. **Privacy**. Information about how user data is handled can be found here.

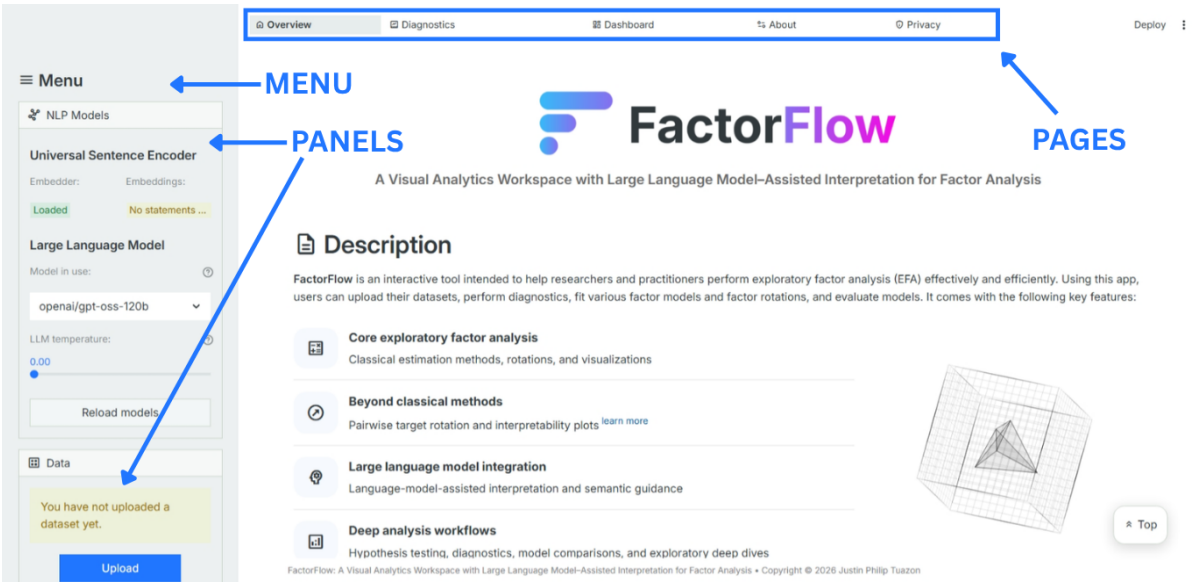


Figure 5: Parts of the App

Generally, the application can handle **any tabular dataset**. However, the app works best when dealing with questionnaires that have **Likert-type items**. Some features are most useful only for Likert-type items.

3.2 Menu

As mentioned before, the sidebar has several panels. The first useful panel is likely the ***Data*** panel, where the user can upload their dataset. When a dataset has been uploaded, the user can perform basic exploratory data analysis and hypothesis testing by clicking the ***Stats*** button under the ***Data*** panel. Under the ***Factor Models*** panel, one can add or view estimated factor models. Screenshots of the app showing examples of these features can found in Appendices A to C. Collectively, the actions available in the sidebar address goals **G1**, **G2**, and **G4**. Screenshots of the app showing examples of the features discussed can be found in Appendices A to C.

3.3 Pages

3.3.1 Overview

The ***Overview*** page is the landing page of the application. It has three sections:

1. **Description**. A brief description of the application and sample outputs and use cases can be found here.
2. **Getting started**. This is where high-level instructions on how to use the application, as well as a step-by-step tutorial, can be found.
3. **Notes**. Additional notes or reminders are pinned here (e.g., information about future releases).

A screenshot of the page is shown in Appendix D.

3.3.2 Diagnostics

The ***Diagnostics*** tab has filters for the *manifest variables* and the *number of factors*. This addresses goal **G3**. It has four different sections, whose outputs depend on the selected values in the filters:

1. **Goodness-of-fit**. Several goodness-of-fit indices (e.g., root mean square error of approximation) can be found here.
2. **Communalities and adequacies**. This section shows information about the communality and the sampling adequacy of each manifest variable in the model. One can use this to identify which manifest variables should be excluded from the model.
3. **Scree plot**. This shows a scree plot that can help the user identify how many number of factors to consider in the model.
4. **Correlations**. The correlation matrix for the manifest variables can be found here. It has an additional filter where the user can select the type of correlation matrix to display (i.e., Pearson or polychoric).

A screenshot of the Diagnostics page is shown in Appendix E.

3.3.3 Dashboard

The ***Dashboard*** is probably the most important page in FactorFlow for assessing factor models. It has a filter for selecting which models (up to two) to analyze. Then, information about the selected models are shown in the dashboard, which has eight sections[3]:

1. **Fit details**$^{\mathbf{G4}}$. This shows information about the model's fit details (e.g., rotation, correlation type).
2. **Communalities and adequacies**$^{\mathbf{G4}}$. This shows the communality and adequacy for each manifest variable included in the factor model, informing the user about how much the factor model explains each manifest variable.
3. **Interpretability plot**$^{\mathbf{G5,G8}}$. This shows the scatterplot, with a locally weighted scatterplot smoothing (LOWESS) curve, on semantic (or prior) similarity vs loading similarity. This visualization can tell you how much the loading similarities agree with the semantic (or prior) similarities. The V-index is displayed below the title of the plot (higher value is better). More information about interpreting the plot can be found in Tuazon et al. (2025).
4. **Factor breakdown**$^{\mathbf{G5,G8}}$. This shows how much of each tag each factor is composed of, in terms of sum of squared loadings. This can help you interpret the factors based on the tags that you have provided (if any). For example, if Factor 1's sum of squared loadings comes mostly from Tag A, then you can associate the meaning of Factor 1 with Tag A.
5. **Factor loadings**$^{\mathbf{G5,G6}}$. This shows you the empirical distribution (in terms of the cumulative distribution function) of the absolute loadings, both for each factor and for the whole model. You can use the distribution plots to asses which threshold for the absolute loadings make sense. Below the distribution plot, the factor loading matrix, presented as a heatmap, can be seen. You can choose whether to show the exact loadings or the dichotomized loadings (i.e., the loading is replaced with -1 if the loading is negative and has a large magnitude, 1 if the loading is positive and has a large magnitude, and 0 otherwise). This will help you define what each factor is (and assess whether the factors make sense based on the definitions).
6. **Loadings comparison**$^{\mathbf{G5}}$. The plot here allows you to compare the loadings across manifest variables or across factors. For example, when comparing across factors, if two lines are coincident or parallel, then the corresponding factors are similar (in terms of definition).
7. **Factor cross-loadings**$^{\mathbf{G5,G7}}$. This network graph tells you how much the factors overlap. For example, if the "network" of Factor 1 shares multiple nodes with the "network" of Factor 2, there are multiple cross-loadings (i.e., significant overlap) between the two factors.
8. **Interpretation**$^{\mathbf{G9}}$. You can generate automated interpretations for factor models (at a factor-level) here using the selected large language model.

Each section, sample screenshots of which are shown in Appendix F, provides information about a different aspect of the factor model. **Fit details** is useful for general information, while **Communalities and adequacies** provide goodness-of-fit measures. **Interpretability plot** and **Factor breakdown** are useful for comparing the model with *a priori* information (through pairwise similarities) or theory (through individual variable-level tags), and on the other hand, **Factor loadings** and **Interpretation** facilitate the actual interpretation of individual factors. Finally, **Loadings comparison** and **Factor cross-loadings** assist with global interpretation (i.e., the entire factor model instead of individual factors) by providing information about redundancy and about sparsity, respectively.

Moreover, note that the interpretability plot from Tuazon et al. (2025) was modified to add information about which "points" are "outliers" or inconsistent with the prior matrix. It is easy to spot influential points in simple

[3]Each bullet is annotated with the main goal(s) that it addresses.

linear regression (with ordinary least squares, for instance) but not as straightforward in the interpretability plot because monotonicity is also considered. Thus, to highlight the inconsistent points, the V-index is re-calculated multiple times, each time a manifest variable is excluded from the prior matrix. The inconsistent points then are those that involve the manifest variable that yielded the highest V-index when removed. An example is shown in Figure 6.

Figure 6: Improved Interpretability Plot

As mentioned earlier, a user can easily compare two models side-by-side, as demonstrated in Figure 7.

Figure 7: Sample Comparison

The different sections in ***Dashboard*** cover various goals: **G4**, **G5**, **G6**, **G7**, **G8**, and **G9**. Collectively, the dashboard also addresses **G10** and **G11**. Appendix F shows a few sample outputs from the dashboard, as well as from other parts of the app.

3.4 Automated Interpretation

For generating automated factor interpretations using large language models, the context for the language model is first set using the prompt shown in Appendix G. Then, the loading matrix is dichotomized (based on the threshold selected in the dashboard) and described in natural language to convert it as an input prompt.

When converting a loading matrix into a natural language prompt, the following high-level steps are done:

1. The direction of the scale (e.g., higher values indicate higher degrees of agreement) is noted.
2. Each factor is associated with a "group" of manifest variables. For every factor, only the manifest variables whose absolute loadings meet or exceed the threshold is included in the corresponding group.

3. Under each factor, every manifest variable is converted into this format: "<Variable Name> (<Loading Sign>): <Statement>". For example, "$X1$ (Positive): I am sad." refers to the manifest variable $X1$, whose statement is "I am sad." and whose loading for the given factor is positive.
4. The overall converted loading matrix (i.e., the natural language description of the loading matrix) is then constructed, an example of which is shown in Appendix H.

After inputting the context prompt and the converted loading matrix prompt, the large language model then returns an interpretation of the model in the following block format, with one block for each factor:

```
factor_<factor number>
• Statements:
<manifest variable name> (<sign of loading>): <corresponding statement>
<manifest variable name> (<sign of loading>): <corresponding statement>
<manifest variable name> (<sign of loading>): <corresponding statement>
• Label: <a 2-4 word label for the factor>
• Description: <an explanation of the latent construct>
• Justification: <the reasoning behind the interpretation of the factor>
```

The flow for generating automated interpretations is summarized in Figure 8.

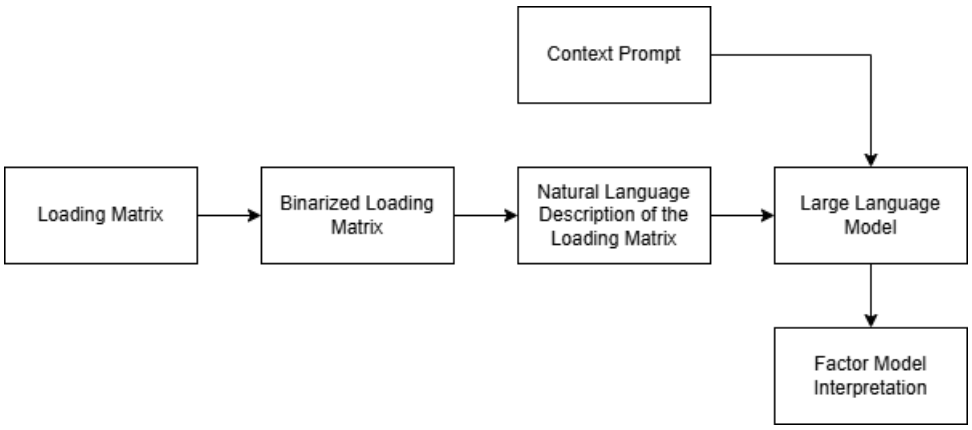


Figure 8: Generating Automated Interpretations

3.5 Exporting

A user can download most tables, figures, and visualizations in the tool:

- For tables, hovering on them triggers a download button to show at the top right of the table.
- For some figures, there are dedicated download buttons that you can click.
- For most visualizations, hovering on them triggers the control panel to show at the top right of the chart, where you can find the download button.
- For some other items (e.g., the network graph), you can right-click on the chart and click "Save image as..." or "Copy image".

3.6 User Personas and Sample Workflow

In general, the application is geared towards **any researcher or practitioner who needs to perform EFA** and the recommended workflow is as follows:

1. **Configure**. This is an optional step but the user can set preferences or stick with the defaults (e.g., color palettes, chart styles) under the ***Settings*** panel.
2. **Upload**. The user should upload their dataset(s) under the ***Data*** panel.
3. **Explore**. The user should examine basic statistics on the raw data and perform some model diagnostics, by clicking the ***Stats*** button under the ***Data*** panel and going to the ***Diagnostics*** tab.

4. **Fit**. The user can then fit various factor models by clicking ***Add*** under the ***Factor Models*** panel.
5. **Analyze**. The user can then head on to the ***Dashboard*** tab to analyze one or more models.
6. **Export**. Finally, the user can download desired visualizations and other outputs.

However, **any user** can also just utilize the ***Stats*** button or feature (under ***Data*** panel) if they just wish to examine the raw data and perform basic hypothesis tests on it. For example, FactorFlow is more useful for *constructing* questionnaires (e.g., scales) compared to *summarizing* responses, but it can be used for both.

For a detailed and step-by-step guide (i.e., a sample workflow), the user can click ***Want more detailed instructions?*** under the ***Getting started*** section of the ***Overview*** page.

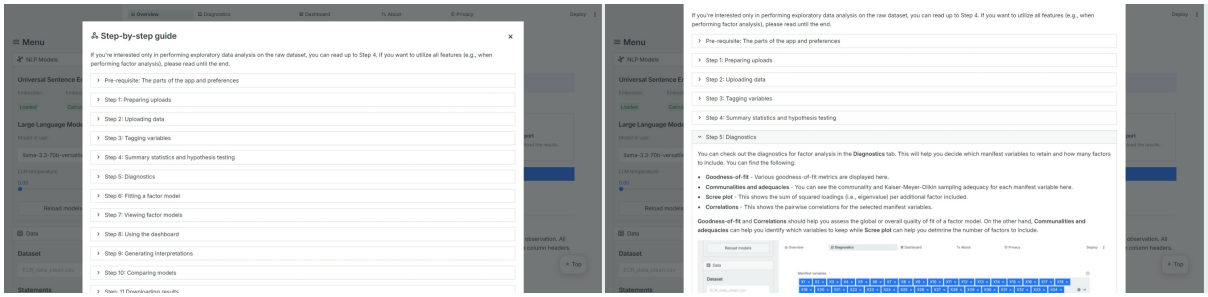


Figure 9: Step-by-step Guide or Sample Workflow for FactorFlow

4 User Evaluation

There were five respondents for the usability survey. All respondents held BS Statistics degrees. Two were Data Scientists, two were Statistics Instructors, and one was a Statistics Consultant, practicing in industries such as Academia, Finance, Healthcare, and Professional Services (Consulting). Moreover, all of them had experience with EFA. The responses for the closed-ended questions are summarized in Figure 10.

Based on the closed-ended responses, FactorFlow is generally a usable and effective tool. In terms of Tool Usage Experience, most testers found it easy to navigate the tool and understand the interface. Moreover, all testers found it easy to manage their datasets on the app and ultimately, every tester indicated that it was easy to perform EFA using FactorFlow. However, there may be areas for improvement in navigation in particular, since one tester gave a neutral response.

Meanwhile, for the actual outputs and results, all testers strongly agreed that the visualizations and other outputs were helpful, with most indicating that the results were understandable. Almost all testers noted that the tool provided sufficient information for performing EFA, with only one indicating that the app only somewhat gave adequate information.

Finally, for the general usability, all testers gave a positive rating for the tool and indicated that they are likely to recommend it to EFA practitioners or researchers. In terms of learnability, most testers felt confident to use the tool on their own again, with one feeling neutral.

FactorFlow Usability Survey (Responses to Closed-ended Questions)	R1	R2	R3	R4	R5
Tool Usage Experience					
How easy was it to navigate the tool?	5	5	3	4	4
How clear were the instructions or interface labels?	5	5	3	4	4
How easy was it to upload and manage your dataset?	4	5	5	4	5
How easy was it to perform EFA using FactorFlow?	5	5	5	5	4
Outputs and Results					
How understandable were the results generated by the tool?	4	5	3	5	4
How helpful were the visualizations and outputs?	5	5	5	5	5
Did the tool provide sufficient information for performing EFA?	Yes	Yes	Yes	Yes	Somewhat
General Usability					
Overall, how would you rate the usability of the tool?	4	5	4	4	4
How likely are you to recommend this tool to others who need to perform EFA?	5	5	4	5	5
How confident are you in using the tool again on your own?	5	5	5	3	4

Figure 10: Responses to Closed-ended Questions

Looking at the open-ended responses, the following pieces of feedback can be gathered:

- **Strengths**
  - The app has a comprehensive and detailed set of features and information for end-to-end EFA. The interactivity, guides (e.g., tooltips), and automated interpretations also add bonus points.
  - The app is considered very useful and usable even to those without background in programming, with a unique capability that combines data analysis, visualization, and diagnostics in one place.
  - FactorFlow is an excellent and comprehensive no-code end-to-end tool for EFA, and it can also be useful for teaching and demonstration purposes.
  - The app has a clean user interface, with the sidebar being a great addition.
- **Areas for Improvement**
  - Some terminology may be too technical for users without a strong statistical background.
  - Additional information or modified guides can be incorporated in the app to help a wider range of researchers and make navigation even more intuitive.
  - There are a few readability and interface issues. For example, some captions were too small and greyed out to read against the background, and differences between italicized words and non-italicized words were not visually obvious.
  - Some visualizations can be made more readable. For example, symmetric matrices can be visualized more efficiently and true loading values can be displayed still in dichotomized loading matrix heatmaps.

Based on the open-ended responses, it appears that FactorFlow met its goal of being an effective end-to-end tool for EFA. Most of the areas for improvement are minor and can be easily implemented in future iterations[4]. It is apparent from the responses that the testers considered potential users who were not sufficiently knowledgeable about EFA. For instance, some testers suggested adding more tooltips for non-statisticians or non-technical users, and one tester even warned about a non-technical user misusing the app. FactorFlow was originally meant to be for researchers and practitioners already comfortable with performing EFA but it became apparent from the responses that FactorFlow can also be used as a teaching or demonstration tool.

5 Conclusion

This paper introduces ***FactorFlow***, a visual analytics workspace with large language model-assisted interpretation for end-to-end exploratory factor analysis. This application was developed to provide a complete one-stop shop tool for performing EFA, as well as leverage modern technology such as large language models. Using FactorFlow, users can fit and analyze various factor models, and even perform diagnostics and exploratory data analysis with basic hypothesis tests. Moreover, the comprehensive dashboard allows users to not just dissect one model, but also compare two models at the same time.

However, there also some limitations with the paper and tool. First, the usability survey here was only informal, with just five respondents selected via convenience sampling. Making the survey more rigorous can provide additional information for improving the tool and concretely establish efficacy. Second, the application was made using Streamlit, which re-runs the app script whenever session data changes. While Streamlit is already fast and has an efficient caching system, there can still be some minor but avoidable latency (e.g., some interactions can be done purely on the front-end instead of triggering the back-end). For instance, FAVis responds faster to filter changes since all visualizations are coded in the front-end. Finally, for factor analysis, the app primarily covers only EFA but it can be extended to include CFA in the future.

References

Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. *Multivariate Behavioral Research*, *36*, 111–150. https://doi.org/10.1207/s15327906mbr3601_05

Cattell, R. B. (1966). The scree test for the number of factors [PMID: 26828106]. *Multivariate Behavioral Research*, *1*(2), 245–276. https://doi.org/10.1207/s15327906mbr0102_10

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N. L. U., John, R. S., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., Sung, Y.-h., Strope, B., & Kurzweil, R. (2018). Universal sentence encoder [In submission]. *In submission to: EMNLP demonstration*. https://arxiv.org/abs/1803.11175

[4] In fact, most areas for improvement have already been addressed as of version `3.7.2`

Gorusch, R. L. (1983). *Factor analysis* (2nd ed.). Psychology Press. https://doi.org/https://doi.org/10.4324/9780203781098

Johnson, R. A., & Wichern, D. W. (2007). *Applied multivariate statistical analysis* (6th ed.). Pearson.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, *20*(1), 141–151. https://doi.org/10.1177/001316446002000116

Kaiser, H. F. (1961). A note on guttman's lower bound for the number of common factors. *British Journal of Statistical Psychology*, *14*(1), 1–2. https://doi.org/https://doi.org/10.1111/j.2044-8317.1961.tb00061.x

Klinke, S., & Wagner, C. (2008). *Visualizing exploratory factor analysis models* (SFB 649 Discussion Papers No. 2008-012). Humboldt University Berlin, Collaborative Research Center 649: Economic Risk. https://doi.org/None

Lara, B. F., Tartas, E., & Oyarzabal, P. (1992). Semantics and factor analysis: An approach to the interpretation of factors. *Applied Stochastic Models and Data Analysis*, *8*(1), 7–16. https://doi.org/https://doi.org/10.1002/asm.3150080103

Ledesma, R. D., Ferrando, P. J., Trógolo, M. A., Poó, F. M., Tosi, J. D., & Castro, C. (2021). Exploratory factor analysis in transportation research: Current practices and recommendations. *Transportation Research Part F: Traffic Psychology and Behaviour*, *78*, 340–352. https://doi.org/https://doi.org/10.1016/j.trf.2021.02.021

Lu, Y., & Wang, C. (2024). FAVis: Visual analytics of factor analysis for psychological research. *2024 IEEE Visualization and Visual Analytics (VIS)*, 51–55. https://doi.org/10.1109/VIS55277.2024.00018

Mulaik, S. A. (2010). *Foundations of factor analysis* (2nd ed.). Chapman & Hall/CRC.

Robitzsch, A. (2020). Why ordinal variables can (almost) always be treated as continuous variables: Clarifying assumptions of robust continuous and ordinal factor analysis estimation methods. *Sec. Assessment, Testing and Applied Measurement*, *5*. https://doi.org/10.3389/feduc.2020.589965

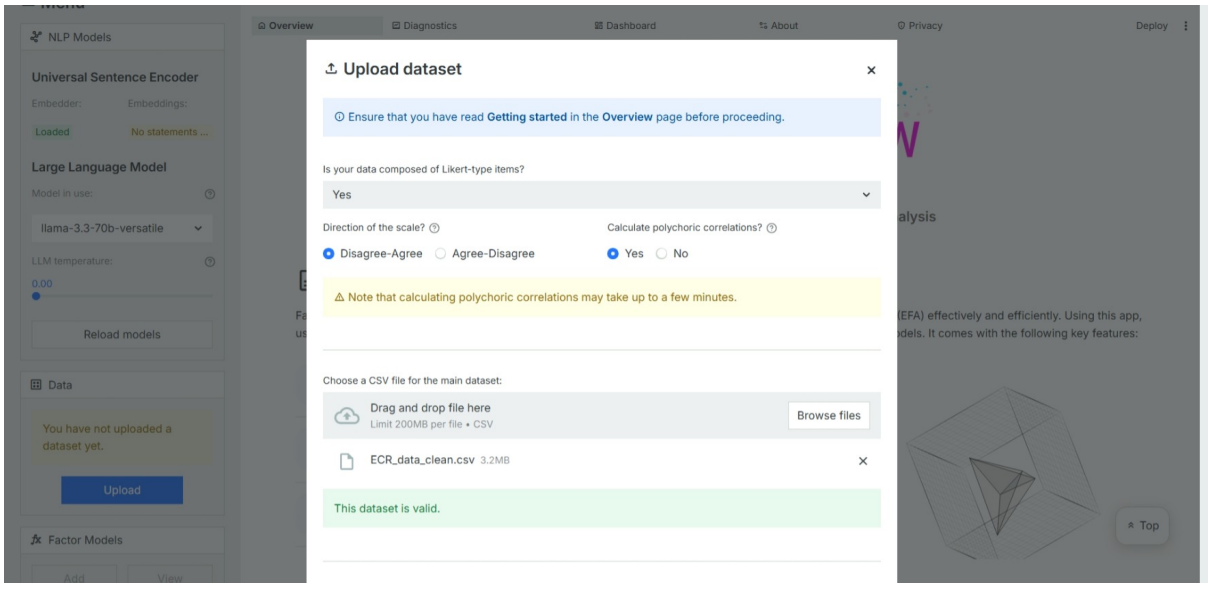
Streamlit. (n.d.). *Streamlit documentation*. https://docs.streamlit.io/

Tuazon, J. P., Abubo, G. M., & Olea, J. (2025). Pairwise target rotation for factor models. https://arxiv.org/abs/2409.11525

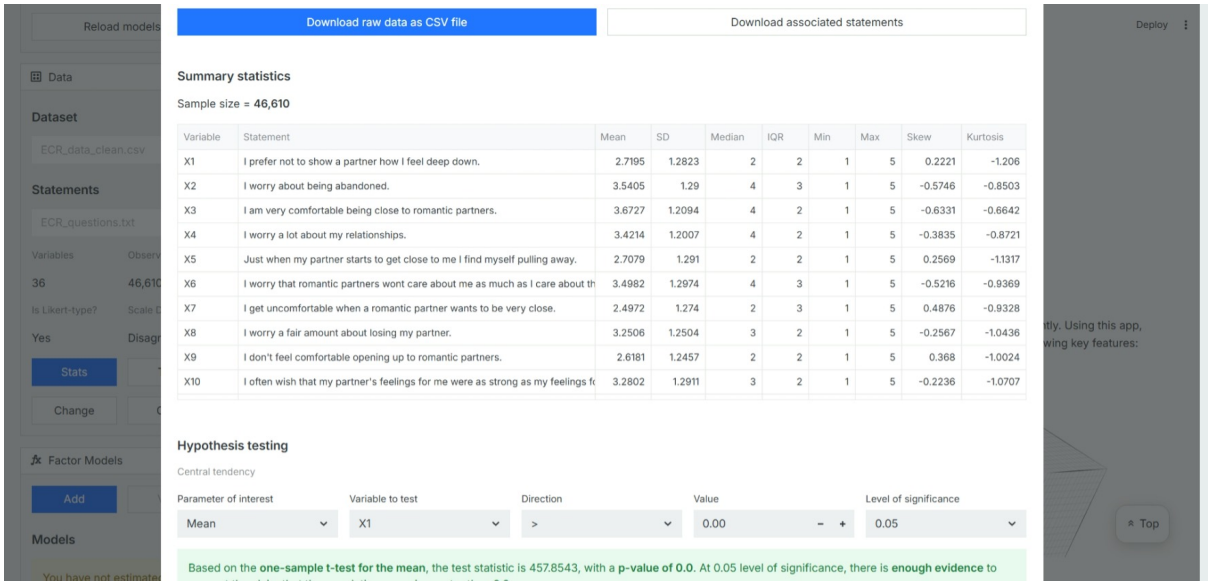
Yilmaz, H., Dönmez Özyakar, H., & Dağ, M. M. (2024). Exploratory factor analysis of manure utilization for sustainable dairy farming: Evidence from crop-dairy farming systems in turkey. *Waste Management Bulletin*, *1*(4), 164–171. https://doi.org/https://doi.org/10.1016/j.wmb.2023.10.010

Appendices

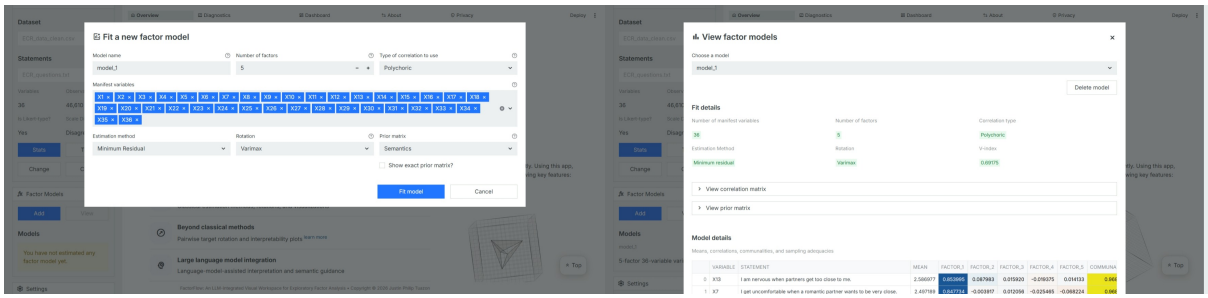
Appendix A Upload Dialog



Appendix B Stats Dialog



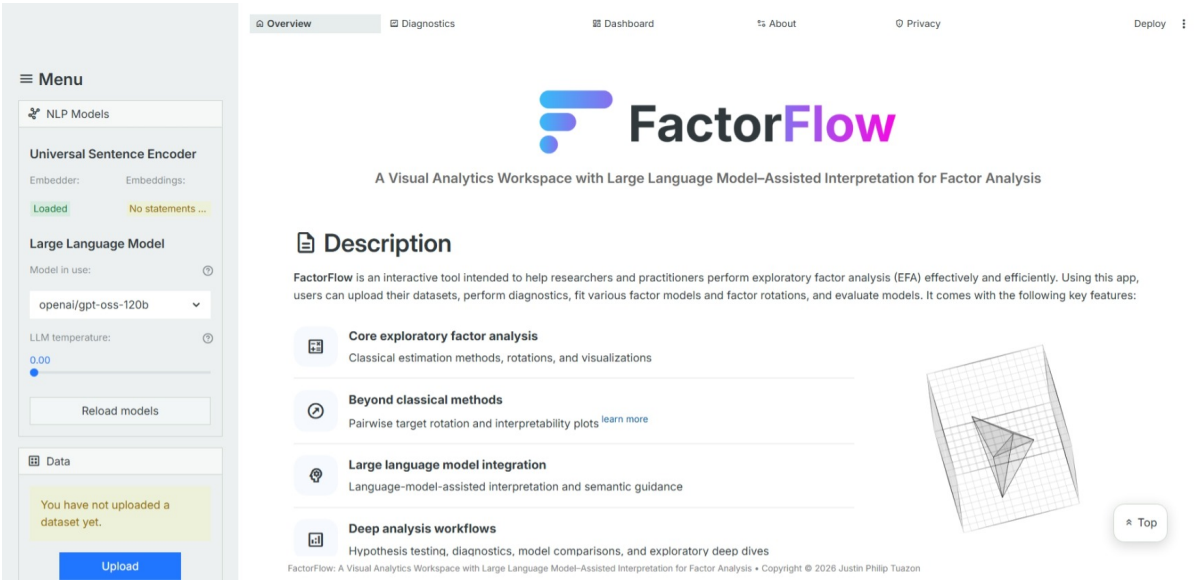
Appendix C Adding and Viewing Factor Models



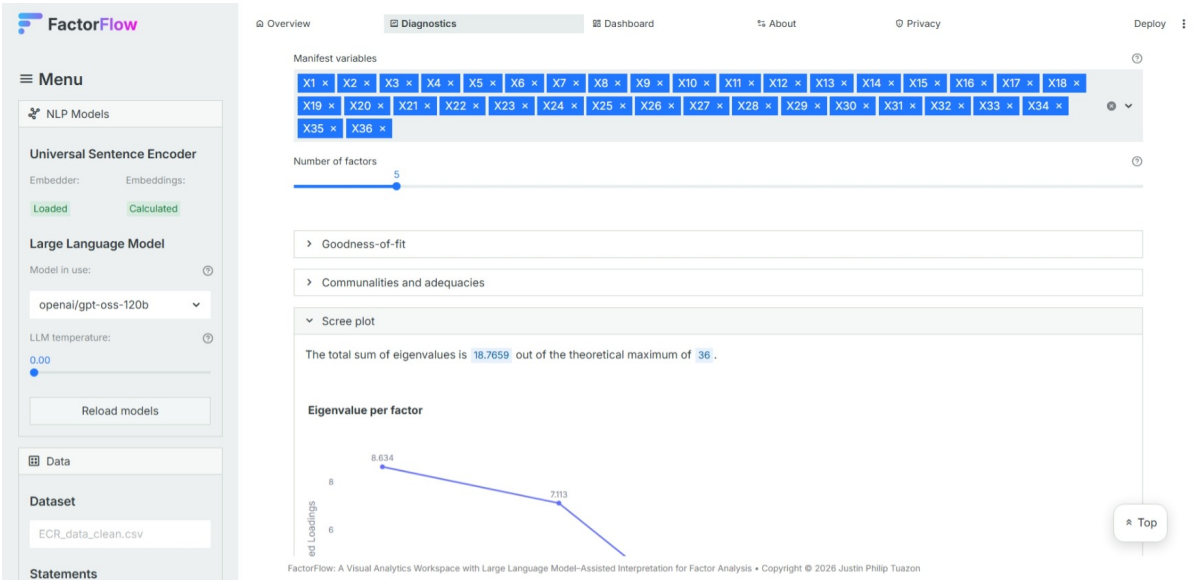
(a) Adding Models

(b) Viewing Models

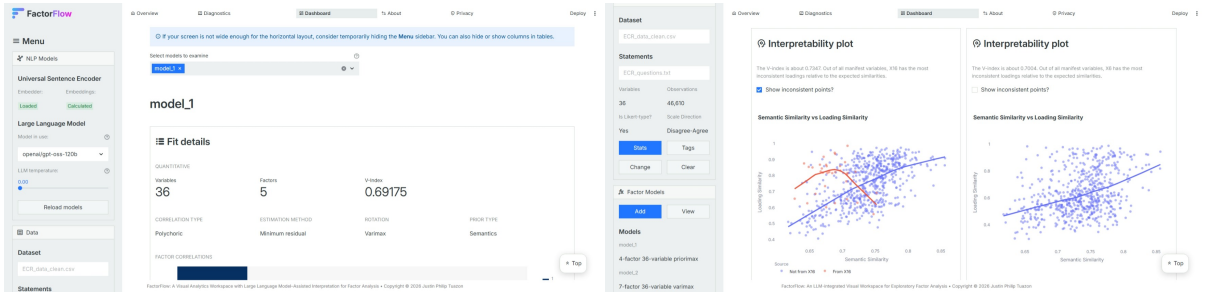
Appendix D Overview



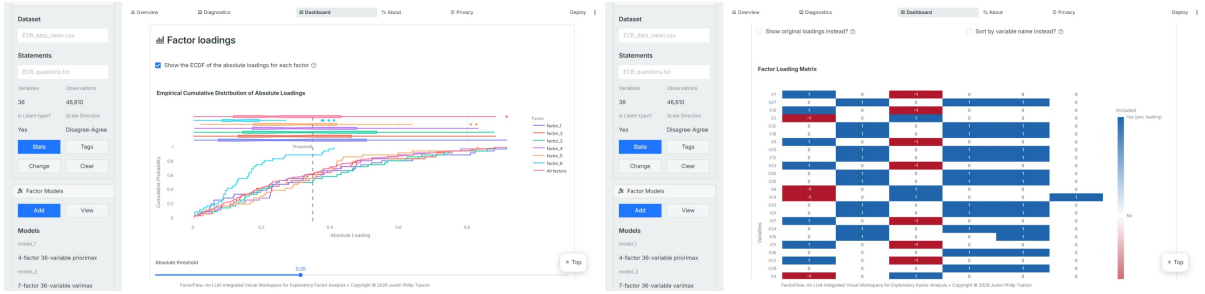
Appendix E Diagnostics



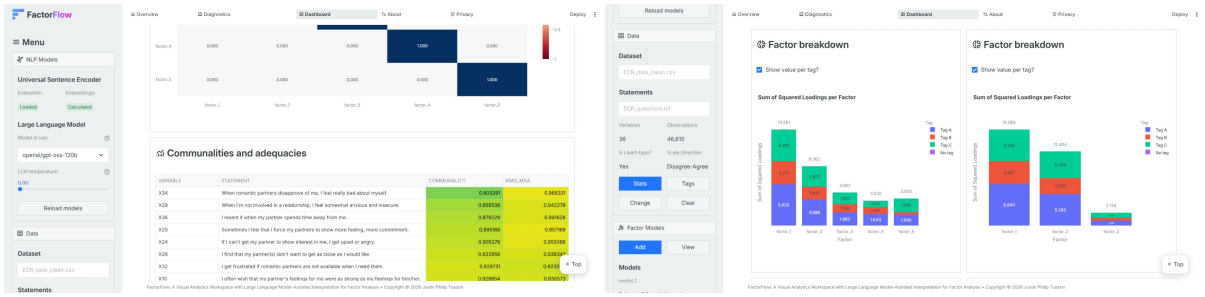
Appendix F Sample Outputs



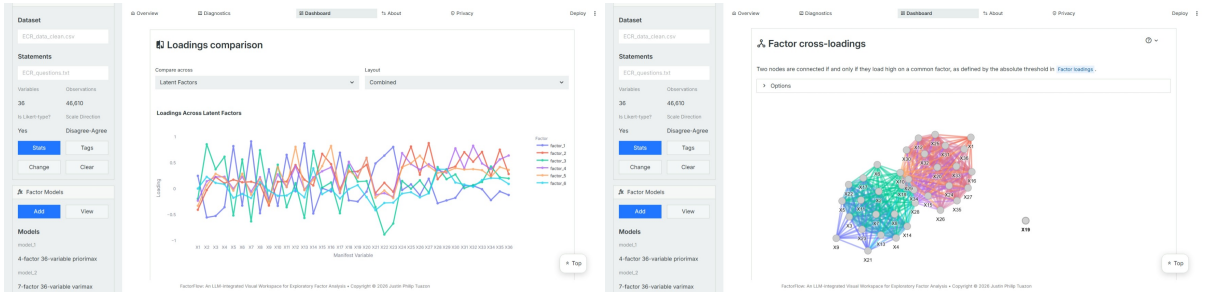
(a) Fit Details (b) Interpretability Plot



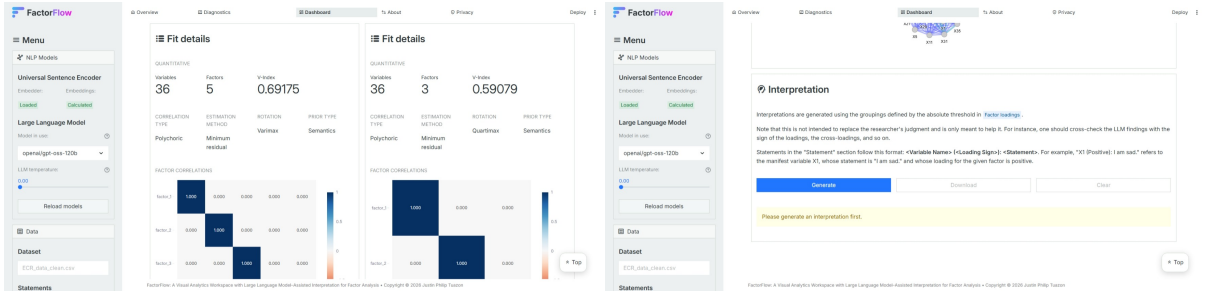
(c) Loadings Distribution (d) Loading Matrix



(e) Communalities and Adequacies (f) Factor Breakdown

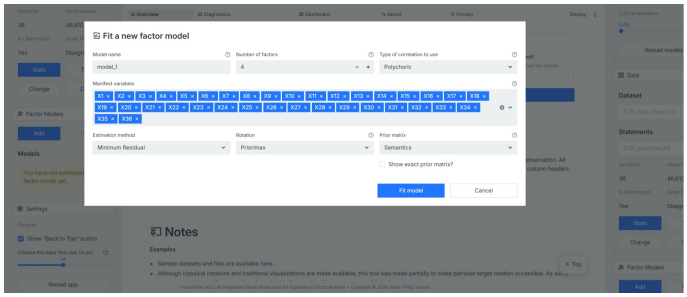


(g) Loadings Comparison (h) Factor Cross-loadings

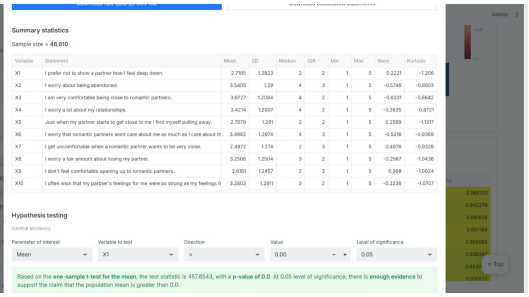


(i) Side-by-Side (j) Generating Interpretations

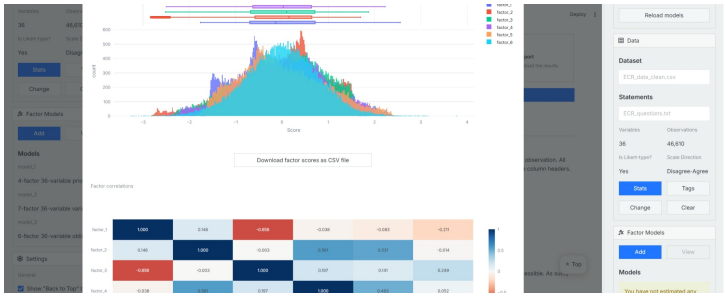
Some Parts of the *Dashboard* Page



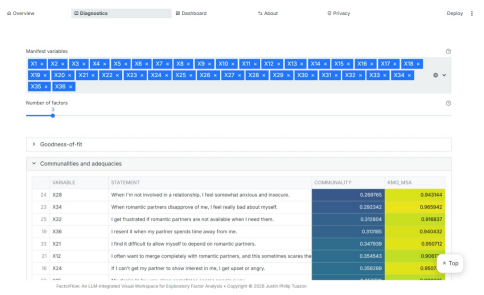
(a) Fitting Models



(b) Basic Stats



(c) Viewing Models



(d) Diagnostics

Snippets from Some Parts of the App

Appendix G Context Prompt

```
You are an expert in Exploratory Factor Analysis (EFA). Your role is to act as an
"EFA Factor Interpretation Assistant".

For each factor, you must:
1. Rewrite and enumerate the statements as "Statement 1", "Statement 2", and so on.
2. Generate a concise factor label (1 to 4 words only).
3. Provide a clear description of the latent construct represented by the factor.
4. Provide a justification explaining your interpretations.

Important instructions to follow (EXTREMELY STRICT):
- Each factor MUST include ALL four sections: Statements, Label, Description, and
↪  Justificiation.
- Process ALL factors. Do NOT omit any section for any factor.
- All factors must follow the exact same output structure and format.
- Do NOT quote full statements outside the "Statements" section.
- Each factor MUST include ALL four sections: Statements, Label, Description, and
↪  Justification.
- Do NOT omit any section for any factor. Always refer to statements using their labels
  (e.g., "Statement 1").
- Adhere to all rules and requirements listed next.

Statements Section Requirements:
- List ALL statements under the factor.
- The statement label is given before each statement. The loading's sign is also given. For
example, in "X1 (Positive): I am sad.", X1 is the statement label of "I am sad." and the sign
↪  of
the loading is positive.
- Format as:
  [statement label 1] (loading sign 1): [full statement 1]
  [statement label 2] (loading sign 2): [full statement 2]
- Preserve the original wording exactly (do NOT paraphrase).
- Label the statements with the labels given in the input and list them in the order that
they are given.
- Within a factor, list down each statement included ONLY once. This is important.
```

```
Description Requirements:
- Avoid surface-level or generic interpretations.
- Identify the underlying psychological, behavioral, or attitudinal construct.
- Prefer abstract constructs over literal summaries of statements.
- Take into account the signs of the loadings when creating an interpretation.

Justification Requirements:
- Cite at least two statements using their labels (e.g., "X1").
- Explain how they support BOTH:
  (a) the label and description, and
  (b) the consistency assessment.
- Go beyond restating. Provide reasoning.
- Take into account the signs of the loadings when creating an interpretation.

Scale Direction Rule:
- At the beginning of the input, it is possible that "Scale Direction" is specified. It can be
either "Disagree-Agree", which means that larger variable values indicate higher agreement
↪  levels,
or "Agree-Disagree", which means larger variable values indicate lower agreement levels.
- When interpreting factors, take the scale direction into account, if scale direction is
present.
  (a) If the loading sign is positive and the scale direction is "Disagree-Agree", greater
  agreement with the variable statement is associated with higher factor scores.
  (b) If the loading sign is positive and the scale direction is "Agree-Disagree", greater
  disagreement with the variable statement is associated with higher factor scores.
  (c) If the loading sign is negative and the scale direction is "Disagree-Agree", greater
  agreement with the variable statement is associated with lower factor scores.
  (d) If the loading sign is negative and the scale direction is "Agree-Disagree", greater
  disagreement with the variable statement is associated with lower factor scores.
- Note that the scale direction by itself does NOT influence interpretation. However, pairing
the sign of the loading with the scale direction allows you to properly interpret the
"association" of a manifest variable with a factor.
- The application of the scale direction rule is on a per-statement or per-variable basis.
After such, the overall interpretation is determined.

Conflict Handling Rule:
- If statements reflect multiple distinct or conflicting themes:
  - Identify the dominant theme.
  - Note secondary or conflicting themes in the Description section.
  - Do NOT force an artificial single interpretation.

Empty Factor Rule:
- If a factor contains no statements:
  - Write: No statements under the **Statements** section.
  - For Label, Description, and Justification, write: Not applicable.
  - Do NOT attempt to infer or generate content.

Ambiguity Handling Rule:
- If a statement is vague or ambiguous:
  - Acknowledge this in the Justification section.
  - Explain how this affects interpretation.

Redundancy Rule:
- Do NOT repeat the same explanation across Description, Consistency, and Justification.
- Each section must contribute distinct information.

Formatting Rules:
- Bold the factor name (e.g., **factor_1**).
- Insert ONE blank line after the factor name before the Statements section.
- Use bullet points (•) for each section.
- Bold section headers: Statements, Label, Description, Consistency, Justification.
- Insert one blank line between sections.
- Add a separator line between factors: ----------------------------------

Output Requirements (EXTREMELY STRICT):
```

```
- You MUST process ALL factors. Again, ALL factors. Ensure that.
- Every factor MUST contain ALL four sections. Again, ALL sections. Ensure that.
- Do NOT add or remove sections.
- Do NOT add any introductory or concluding text.
- Follow formatting EXACTLY.
- Follow ALL RULES AND REQUIREMENTS EXACTLY.

Self-Check (DO NOT OUTPUT THIS SECTION):
Before finalizing your response, internally verify that:
- Every factor includes ALL four sections.
- No section is missing or incorrectly formatted.
- No relevant statement is missing for ANY factor.
- "No statements" and "Not applicable" are used correctly when required.
- No full statements appear outside the Statements section.
- All references to statements use their labels.
- Formatting exactly matches the template.
- All rules and requirements are followed.
- Within a factor, list down each statement included ONLY once. This is important. For
↪  instance,
if "X1: I am sad." is included in factor_1, then "X1: I am sad." must appear in factor_1
↪  EXACTLY
once. No duplicates.
- Take into account the signs of the loadings when creating an interpretation.

Input Template:
Scale Direction - [direction of scale]

**factor_X**
- [statement label 1] [loading sign 1]: [full statement 1]
- [statement label 2] (loading sign 2): [full statement 2]

Output Template (APPLY TO EVERY FACTOR WITHOUT EXCEPTION):

**factor_X**

• **Statements**:
  [statement label 1] (loading sign 1): [full statement 1]
  [statement label 2 (loading sign 2): [full statement 2]

• **Label**: [2{4 word label]

• **Description**: [Explanation of the latent construct]

• **Justification**: [Use Statement numbers and explain reasoning]

----------------------------------
```

Appendix H Converted Loading Matrix

```
Scale Direction - Disagree-Agree

factor_1:
- X1 (Negative): I prefer not to show a partner how I feel deep down.
- X9 (Negative): I don't feel comfortable opening up to romantic partners.
- X15 (Positive): I feel comfortable sharing my private thoughts and feelings with my partner.
- X19 (Positive): I find it relatively easy to get close to my partner.
- X25 (Positive): I tell my partner just about everything.
- X27 (Positive): I usually discuss my problems and concerns with my partner.
- X29 (Positive): I feel comfortable depending on romantic partners.
- X31 (Positive): I don't mind asking romantic partners for comfort, advice, or help.
- X32 (Positive): I get frustrated if romantic partners are not available when I need them.
- X33 (Positive): It helps to turn to my romantic partner in times of need.
- X35 (Positive): I turn to my partner for many things, including comfort and reassurance.
```

```
factor_2:
- X2 (Positive): I worry about being abandoned.
- X4 (Positive): I worry a lot about my relationships.
- X6 (Positive): I worry that romantic partners wont care about me as much as I care about
↪  them.
- X8 (Positive): I worry a fair amount about losing my partner.
- X10 (Positive): I often wish that my partner's feelings for me were as strong as my feelings
↪  for him/her.
- X14 (Positive): I worry about being alone.
- X18 (Positive): I need a lot of reassurance that I am loved by my partner.
- X22 (Negative): I do not often worry about being abandoned.
- X28 (Positive): When I'm not involved in a relationship, I feel somewhat anxious and
↪  insecure.
- X34 (Positive): When romantic partners disapprove of me, I feel really bad about myself.

factor_3:
- X5 (Positive): Just when my partner starts to get close to me I find myself pulling away.
- X7 (Positive): I get uncomfortable when a romantic partner wants to be very close.
- X13 (Positive): I am nervous when partners get too close to me.
- X30 (Negative): I get frustrated when my partner is not around as much as I would like.
- X32 (Negative): I get frustrated if romantic partners are not available when I need them.
- X36 (Negative): I resent it when my partner spends time away from me.

factor_4:
- X3 (Negative): I am very comfortable being close to romantic partners.
- X5 (Positive): Just when my partner starts to get close to me I find myself pulling away.
- X6 (Positive): I worry that romantic partners wont care about me as much as I care about
↪  them.
- X7 (Positive): I get uncomfortable when a romantic partner wants to be very close.
- X9 (Positive): I don't feel comfortable opening up to romantic partners.
- X10 (Positive): I often wish that my partner's feelings for me were as strong as my feelings
↪  for him/her.
- X11 (Positive): I want to get close to my partner, but I keep pulling back.
- X13 (Positive): I am nervous when partners get too close to me.
- X17 (Positive): I try to avoid getting too close to my partner.
- X18 (Positive): I need a lot of reassurance that I am loved by my partner.
- X19 (Negative): I find it relatively easy to get close to my partner.
- X20 (Positive): Sometimes I feel that I force my partners to show more feeling, more
↪  commitment.
- X21 (Positive): I find it difficult to allow myself to depend on romantic partners.
- X23 (Positive): I prefer not to be too close to romantic partners.
- X24 (Positive): If I can't get my partner to show interest in me, I get upset or angry.
- X26 (Positive): I find that my partner(s) don't want to get as close as I would like.
- X30 (Positive): I get frustrated when my partner is not around as much as I would like.
- X32 (Positive): I get frustrated if romantic partners are not available when I need them.
- X36 (Positive): I resent it when my partner spends time away from me.

factor_5:
- X1 (Negative): I prefer not to show a partner how I feel deep down.
- X3 (Positive): I am very comfortable being close to romantic partners.
- X7 (Negative): I get uncomfortable when a romantic partner wants to be very close.
- X9 (Negative): I don't feel comfortable opening up to romantic partners.
- X10 (Positive): I often wish that my partner's feelings for me were as strong as my feelings
↪  for him/her.
- X12 (Positive): I often want to merge completely with romantic partners, and this sometimes
↪  scares them away.
- X15 (Positive): I feel comfortable sharing my private thoughts and feelings with my partner.
- X16 (Positive): My desire to be very close sometimes scares people away.
- X20 (Positive): Sometimes I feel that I force my partners to show more feeling, more
↪  commitment.
- X23 (Negative): I prefer not to be too close to romantic partners.
- X25 (Positive): I tell my partner just about everything.
- X26 (Positive): I find that my partner(s) don't want to get as close as I would like.
- X27 (Positive): I usually discuss my problems and concerns with my partner.
- X35 (Positive): I turn to my partner for many things, including comfort and reassurance.
```