Austrian Journal of Statistics
MMMMMM YYYY, Volume VV, Issue II, 1–xx.
http://www.ajs.or.at/

Interactively Visualizing Multivariate Market Segmentation Using the R Package Lionfish

Matthias Medl
Institute of Statistics
BOKU University
Vienna

Dianne Cook
Econometrics and Business Statistics
Monash University
Melbourne

Ursula Laa
Institute of Statistics
BOKU University
Vienna

Abstract

Lorem ipsum Lorem ipsum

Keywords: interactive graphics, tourr, exploratory data analysis, R, python.

1. Introduction

Clustering algorithms are often used to find a smaller number of observations (the cluster means) that adequately summarize a much larger number of observations. For market segmentation, clustering can allow partitioning of observations into a small number of groups, by incorporating associations between the variables. Market segmentation supports targeted approaches to different groups of customers based on common traits. It provides a data-driven solution to partitioning customer data. Leisch, Dolnicar, and Grün (2018) provides an extensive overview of using clustering for market segmentation.

A difference between clustering analysis and partitioning is typically the nature of the data. With cluster analysis, we usually envision data that contains separated clusters, and a successful clustering result is one that divides the data based on these gaps. With partitioning, it is usual that there are no gaps in the data, but it is still useful to partition the data. Figure 1 illustrates how the k-means algorithm would partition a 2D data set into four groups depending on the correlation between the two features. When the correlation is high, the partitioning will be along the combination of features that produces the highest variance. With lower correlation, it will segment the bottom and top, and divide the middle into two parts in the opposite direction. When there is no association, the partitioning is radial like a windmill.

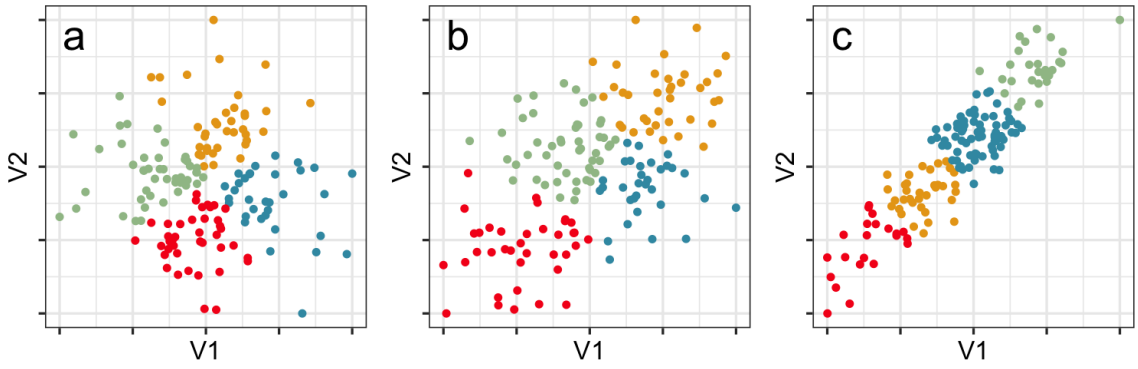

Figure 1: Examples of how k-means partitions 2D data with different association structure. If the correlation is high (c), the partitioning happens along the primary direction of the association.

One can see that the data is perfectly divided into four parts. However, if one were to plot the two features individually, this would not be obvious. Figure 2 shows histograms of the two features, V1, V2, with the colour matching the four partitions. From the histograms, we can see some differences in the partitions for the four different association structures, but they are all overlapping. The distinct border between the partitions can only be seen from the scatterplots of both features.

When there are more than two features, histograms of the individual features are commonly used to display the partitioning results. This means that the analyst likely cannot understand how the partitioning divides the data. All that they can observe is roughly how the individual features relate to the partition, which is useful but inadequate. Here is where using tour methods to view high dimensions can be helpful. A tour (Asimov (1985), and see Lee, Cook, da Silva, Laa, Spyrison, Wang, and Zhang (2022)) can be used to show scatterplots of combinations of features and thus provide views like that in Figure 1 where distinct differences between partitions can be observed. High dimensions are still tricky, and a combination of animations of the linear combinations, and interactive control (Cook and Buja 1997; Laa, Aumann, Cook, and Valencia 2023) over the combinations is important. A scatterplot of a combination of features can be considered to be a projection of the data, and thus like a shadow of a 3D object, some aspects of the data (object) can be obscured. Using slices of the projected data (Laa, Cook, and Valencia 2020) can be a useful addition to projections. This paper illustrates how to do this to better understand partitioning results for multivariate data.

This paper is organised as follows. Section 2 describes the software interface. The user workflow is explained in 3. The methods are illustrated in Section 4 using Austrian and Australian tourism data provided in Leisch *et al.* (2018). Sections 5 and 6 discuss the limitations and potential future developments.

2. Interactive interface for partitioning

The aim of this work is to build an interface that allows for an interactive exploration of partitions generated by clustering of multivariate market data. In some combinations of the features, one should be able to the separations as can be seen in the 2D example in Figure 1. The objectives for the interface are to enable:

1. Visualizing the partitions between clusters in combinations of features using tour methods, with both a grand tour and manual tour.

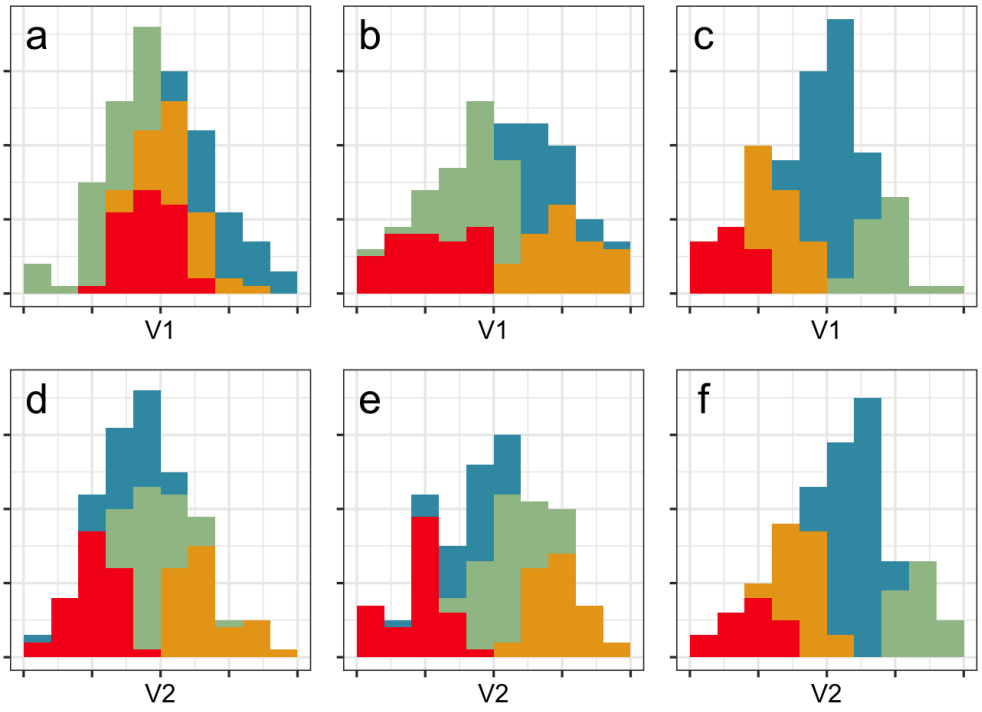

Figure 2: The four k-means partitions plotted as histograms of the individual features. While differences between groups can be seen, the clear separation cannot. This is important for understanding why high-dimensional visualisation methods are useful for summarising partitioning results.

2. Interactively brushing groups of points to refine the cluster solution, like done with spin-and-brush tools (**?**).

3. Linking multiple displays to focus on particular clusters, and simplify the problem to better understand the clustering solution.

4. Update displays based on user selections such as feature selection or cluster selection, or re-scaling.

While tour animations are best obtained within R using the tourr package (Wickham, Cook, Hofmann, and Buja 2011), it does not enable the interactivity required for example for a manual tour (Laa *et al.* 2023). Interactive graphics are available when using Javascript, as implemented in the detourr package (Hart and Wang 2023). This allows to replay a recorded tour path with interactive graphics, and can also be linked with additional displays, but lacks capabilities for manual tours.

To integrate user interactions with the capabilities of the tourr package an active communication with the interface is required. For example, we may wish to explore the local neighbourhood of a projection selected by the user with a local tour animation provided by tourr, or we may want to optimize a guided tour path using groups identified via brushing. Our solution is using Python for high-performance interactivity, through the packages `TKinter` (Lundh 1999), `CustomTKinter` (Schimansky 2024), and `matplotlib` (Hunter 2007), with integration to the tourr package (Wickham *et al.* 2011) via `reticulate` (Ushey, Allaire, and Tang 2024), a framework that facilitates seamless interoperability between Python and R.

The interface was implemented in the R package `lionfish` and offers a variety of linked interactive plot types, providing users with the flexibility to visualize their data from multiple perspectives. The ability to navigate through various projections of the displayed tours directly within a graphical user interface (GUI) enables users to explore different aspects of

the dataset. Furthermore, users can initiate new tours directly from the interface. The GUI also supports interactive feature selection, allowing users to specify which subset of features should be visualized in the plots. Once users have identified interesting views or settings, `lionfish` allows them to save the displayed projections, subsets, and plots. This functionality ensures that analysis states can be preserved for further examination or reporting, making the package particularly useful for iterative analysis where findings may need to be revisited or shared with collaborators.

With its high level of interactivity, performance, and ease of use, `lionfish` streamlines the exploration of complex datasets, offering a powerful tool for researchers working with high-dimensional data.

2.1. Overview of the graphical user interface (GUI)

The `lionfish` GUI can be launched using the function `interactive_tour()`. At a minimum, the user needs to provide both the dataset and the instructions for constructing the desired plots. The dataset must be supplied as a `data.table`, while the plotting instructions should be passed as a list containing the named elements `type` and `obj`. The `type` element specifies the type of display to generate, such as `scatter` for a scatterplot or `2d_tour` for a 2-dimensional tour. The `obj` element further defines the properties of the chosen display. For example, to create a 2-dimensional tour, the user must provide a `tour_history` object, which can be generated using the `tourr` package. For a scatterplot, the user needs to provide a vector of strings specifying the names of the features to be displayed. The user can optionally specify the feature names, the arrangement of the plots, predefined subsets of the data (e.g., cluster solutions), custom names for these subsets, the number of available subsets, and the size of each plot.

The GUI is divided into two main sections: a sidebar on the left, which contains a comprehensive set of interactive controls, and the display area on the right, where the selected plots are shown (see Figure 3).

At the top of the sidebar, users can select and deselect features using checkboxes (Figure 3C), controlling which features are displayed in the plots. Below this, each subset has its own checkbox to designate the active subset (Figure 3D). When data points are manually selected in the plots, they will be assigned to the active subset and colored accordingly. For scatterplots, data points can be selected by encircling them directly on the plot while holding down the left mouse button. For barplots, clicking on a specific bar selects the data represented by that bar. The colored boxes next to the subset names indicate the assigned colors for the data points. Clicking on these boxes adjusts the transparency of the points, which is helpful for highlighting and comparing subsets. Subsets can also be renamed using the provided text boxes. The `Reset original selection` button allows users to revert the subset selections to their initial state. All plots displayed in the GUI are linked, meaning that changes in one plot will affect the other plots. For instance, if observations are moved from one subset into another subset in one plot then the reassigment will also occur in the other plots.

The frame selection interface (Figure 3E) displays the frames of the currently shown tours and enables users to jump between frames of the tour objects. Below this, there are three additional interfaces (Figure 3F). The first can be used to adjust the number of bins in histograms—more bins result in higher resolution but slower display updates. The second allows users to animate tours, automatically shifting to the next projection after a specified amount of time. The third offers the ability to hide projection axes with a norm smaller than a chosen threshold, helping to reduce clutter. Users can also save and load projections and subsets using the respective buttons (Figure 3G). New tours can be started using the current settings via the interface at the bottom (Figure 3H), and users can select different metrics for some plot types e.g. heatmaps (Figure 3I).

One can move through the projections of a tour by pressing the arrow keys or by specifying

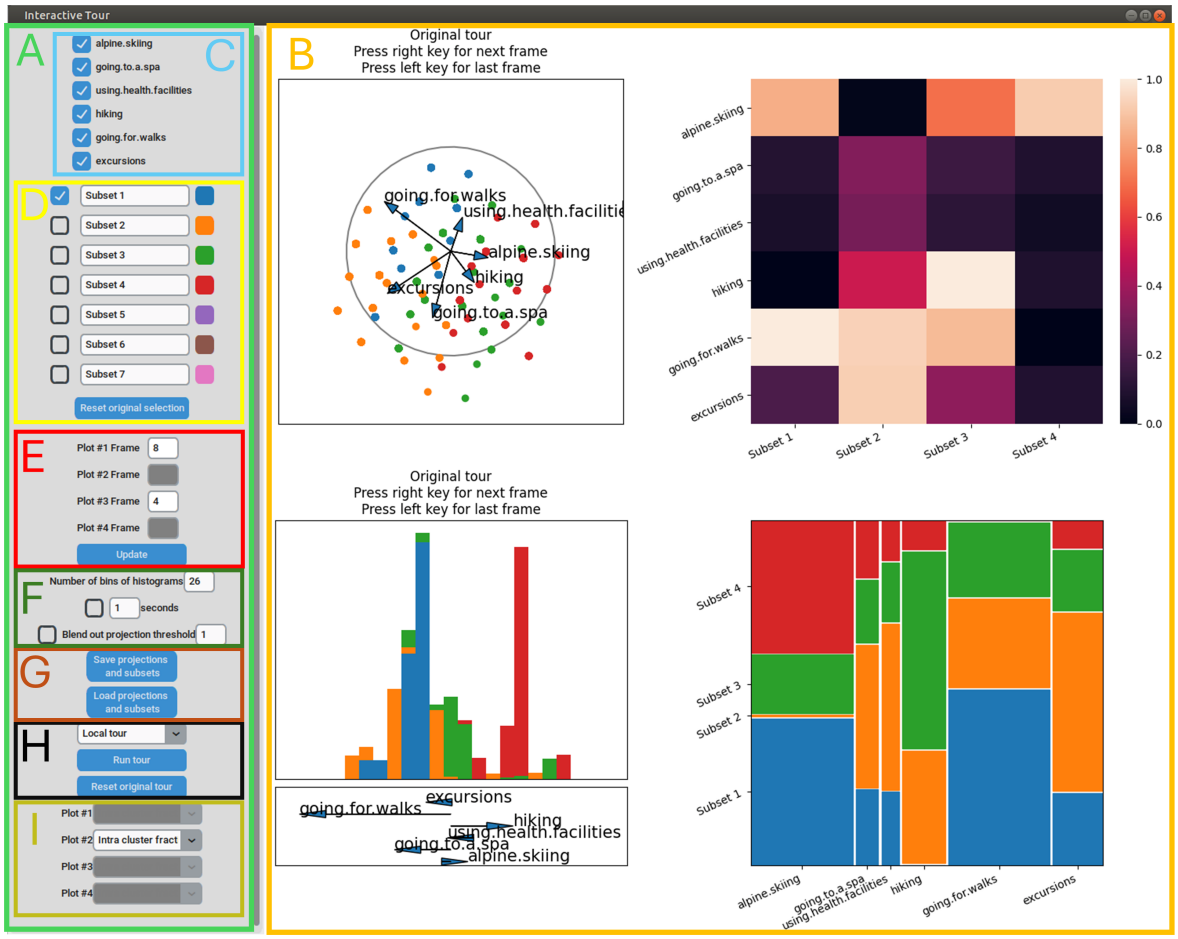

Figure 3: Overview of the GUI. A: Sidebar controls; B: Display area; C: Feature selection checkboxes; D: Subset selection with color indicators; E: Frame selection interface; F: Interfaces for adjusting the number of bins of histograms, animating tours and blending out projection axes with a low norm; G: Save and load buttons; H: Interface for starting new tours; I: Metric selection interface.

the index of the projection the be displayed and pressing the `Update frames` button. Users can animate the tours by toggling the `Animate` checkbox and specifying a time interval, so that the displays automatically move to the next projections after the specified time interval.

The `Save projections and subsets` button can be used to save the current state of the analysis. These states can be recovered by using pressing the `Load projections and subsets` button. Additionally, users can initiate new tours directly from the sidebar. The options for new tour paths are: a local tour around the currently shown projection and guided tours that search for projections based on the holes- or the linear discriminant analysis (LDA) index. The holes index is sensitive to projections with few points in the center of the projections and the LDA index aims to maximize the distance between the centers of the selectet subsets. XXX is this true? other options? Additionally, all tour displays allow users to perform a manual tour; by right-clicking and dragging the arrowheads of projection axes, the projection is recalculated accordingly. This allows for manual exploration of the data.

Mention additional interactive features in the displays - manual tour, anything else that might be relevant?

2.2. R/Python interface

The majority of `lionfish` was written in Python (Van Rossum and Drake 2009), while the R (R Core Team 2024) side of the package handles setting up the Python environment within

the interactive interface is being run, launching the interactive tour, and generating new tours when initiated through the interface. To incorporate the functionality of the `tourr` package without translating large portions of its code from R to Python, the `reticulate` package was used. This approach allows `lionfish` to automatically benefit from updates to `tourr`.

However, to reduce inefficiencies caused by cross-language communication and to simplify debugging, `tourr` functions were only accessed when necessary. Core functionalities, such as performing data transformations using projections, were implemented directly in Python. These transformations are based on well-established mathematical principles and are straightforward to replicate, ensuring they remain stable and efficient.

2.3. Structure of the Python code

The central component of the Python code is the `customTKinter` class `InteractiveTourInterface`. This class centrally stores attributes related to all plots, such as the dataset, sub-selections, feature selections, and other shared information. Plot-specific data is organized in dictionaries (the Python equivalent of named lists in R), including the display type, construction instructions, tour projections (only in case of tour displays), color schemes for the displayed data, and, where applicable, the selector and manual projection manipulation classes.

The selector classes handle the behavior when users manually select data points to move them to the active subset. After a selection is made, the selector class updates the centrally stored sub-selection attribute and ensures all other displays reflect these changes. The manual projection manipulation classes construct the arrows representing the projection axes in the displays and manage the manual adjustment of projections, and thus enable manual tours. Users can right-click and drag the arrowheads to modify the projections, after which the class orthonormalizes the projection axes and updates both the projection and the transformed data accordingly.

Maybe here you can say something about adding new display types, what would be needed for implementing them I am unsure if this can easily be generalized. Some plots are easy to implement (scatterplot), but others like the mosaic plot come with a lot of complications, since that required a deep understanding on how matplotlib works. Implementing new interactive features would be especially challenging.

2.4. Bit blit

The implementation of bit blitting was crucial to ensuring fast plot updates and providing a smooth user experience. With bit blit, the static elements of the display, such as the outer frames of the plots, are stored as a background image. When a plot is manipulated, only the affected plot is updated, and within that, only the interactive elements, such as the data points and projection axes in a 2-dimensional tour, are rendered on top of the background image.

In practice, this means that the background, without the interactive elements, must be captured either during initialization or after major updates. The entire plot is first rendered without the interactive elements, the background is then saved, and finally, the plot is redrawn with the interactive elements blended in. Since this process is relatively slow, full updates are only triggered during initialization or after significant changes, such as modifications to the set of active features.

3. Workflow with the lionfish package

Before launching the `lionfish` interface the user should have performed a partitioning of their choice, and provide the initial clustering solution used by the interface. We launch the interface to explore and potentially refine this solution.

3.1. Feature and cluster relationships

A first step is to assess which features are important for the cluster solution. The interface provides different capabilities that support feature selection: tour view and summary heatmaps.

The 2D tour view can be used to understand the sensitivity of the grouping to individual features. When using the manual tour to remove one feature at a time, the relationship between this variable and the partition between clusters can be examined. The ideal starting view is that obtained through a projection pursuit guided tour (Cook, Buja, Cabrera, and Hurley 1995) that was optimized for the separation between the labeled groups (Lee, Cook, Klinke, and Lumley 2005).

The summary heatmaps provide an overview of the cluster compositions relative to the features. Consider the matrix,

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1p} \\ c_{21} & c_{22} & \dots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \dots & c_{kp} \end{bmatrix}$$

where $c_{ij}, i = 1, ..., k$ (number of clusters); $j = 1, ..., p$ (number of features) are a summary of each feature in each cluster. Because our examples use binary features, the c_{ij} is the number of 1's of feature j in cluster i. There are several ways that these values can be normalized to examine different aspects: $f^o_{ij} = \frac{c_{ij}}{n}$, $f^c_{ij} = \frac{c_{ij}}{n_i}$ and $f^f_{ij} = \frac{c_{ij}}{n_j}$, where n_i, n_j are the row and column totals. The first, f^o is the overall fraction, where counts are normalised by the overall number of observations n. It gives a quick overview on the overall magnitude of the features.

The second, f^c is normalized relative to the size of each cluster and can be considered the distribution of features in each cluster. It is useful for comparing the composition of each cluster relative to the features. For example, for $p = 4$, if $f^c_{1j} = (0.9, 0.2, 0.1, 0.1)$ it suggests that high values (1's) of feature 1 distinguishes cluster 1, and that values of the other features were low. In the context of the later examples, this would mean cluster 1 contains tourists that especially engaged in activity 1, but not in activities 2, 3 and 4. This normalisation produces what is called the **intra-cluster fraction**. It corresponds to the cluster means. XXX why not larger values in the example, the maximum would be 1 so 0.4 still seems very moderate. I adjusted the example a bit.

The last, f^f is normalized relative to each feature, which can be considered to be the distribution of clusters on each feature. It is useful to examine how features are related to a cluster. Considering the last metric, for example, if $f^f_{i1} = (0.2, 0.7, 0, 0.1, 0)$ $(k = 5)$ would indicate that high values on feature 1 primarily are in cluster 2. In the context of the later examples, this would mean activity 1 is most commonly listed in cluster 2. This normalisation produces what is called the **intra-feature fraction**. Note that this must be interpreted with care in case of inbalanced cluster sizes.

In this example, while the intra-cluster fraction for cluster 1 and feature 1 is notably high at 0.9, the intra-feature fraction is comparatively low at 0.2. Applied to the subsequent analysis, this indicates that 90 % of tourists in cluster 1 participated in activity 1, yet only 20 % of all tourists who engaged in activity 1 were part of cluster 1. The majority of individuals participating in activity 1 were members of cluster 2. This discrepancy may be attributed to cluster 1 representing a relatively small group of tourists, characterized by a pronounced preference for activity 1 and a lack of interest in other activities.

Generally, if the features are ordered categories or numerical scores, decomposing the cluster-feature summary is still possible by using an analysis of variance breakdown into row and column effects.

3.2. Subset selection

The spin-and-brush approach suggests to cluster data manually when using a tour: we run a tour animation, stop when we see a group of points that are different from the rest of the distribution, brush them, and then continue. Different projections will enable the separation of different groups, and for well-separated clusters we will be able to recover a full cluster solutions in this manner.

A similar approach can be used to refine a partitioning solution. In a *visual analytics* approach we use interactive visualizations to integrate human judgement with statistical and machine learning models (**?**) to optimize knowledge extraction from data. Here this is in particular useful to integrate prior knowledge or business interests in a given cluster solution. In the interface we can keep the provided clustering, but separate out new subsets via manual selection, for example after we found a group of particular interest via a manual tour.

3.3. Reproducibility

Ensuring the reproducibility of data analysis is a fundamental principle in scientific research. It allows others to verify the validity of the findings and is key to the integrity of the scientific process. Reproducibility not only builds trust in the research outcomes but also enables the scientific community to build upon existing work. When analyses can be replicated, it can be validated whether the conclusions drawn from the data are robust and not dependent on the specific conditions or idiosyncrasies of the original analyst. Moreover, reproducible research can serve as a foundational building block for subsequent studies, fostering incremental advancements in knowledge.

One challenge in the context of interactive data analysis is that not all steps of the analysis are precisely documented in the form of code, especially when using graphical user interfaces (GUIs) where user-driven interactions might not leave a traceable history. This lack of documentation can hinder the ability of others to reproduce the analysis or to understand how specific results were obtained. To mitigate this challenge, it is essential to implement mechanisms that allow users to easily save and share intermediate snapshots of their analyses.

One measure to combat this is to make saving intermediate snapshots of the analysis easy and accessible. Specifically, the `Save projections and subsets` button enables users to take snapshots of their analysis, including visual representations, selected data, and parameter settings. Upon pressing this button, a file browser is triggered, allowing users to specify the destination for saving these snapshots. Each save operation generates multiple files:

- A `.png` file containing the currently displayed graphics,
- `.csv` files that capture the feature and subset selection as well as projections of the tours displayed at the time of the snapshot,
- two `.pkl` files that contain state features of the GUI, allowing for complete recovery of the snapshot.

These files provide dual utility. First, they allow users to fully recover the state of the analysis within the GUI. This can be achieved either by using the `Load projections and subsets` button, or by launching a new GUI instance with the `load_interactive_tour()` function. The latter approach, using `load_interactive_tour()`, has the added flexibility of only requiring the original dataset and the directory containing the saved files. This function also allows users to modify display settings, such as adjusting the size of the interactive plots or changing the arrangement of the display grid. In contrast, when loading the saved state directly from within the GUI, it is crucial that the active session was initiated with the same dataset and plot objects that were present at the time of saving. This ensures that the analysis environment is accurately replicated.

Second, the saved `.csv` files provide a way to inspect and further analyze the data outside of the original interface. This opens up opportunities for deeper analysis and extensions of the work.

This level of interactivity and documentation is crucial for reproducibility, as it ensures that even exploratory, interactive data analysis can be retraced and validated by others. Ultimately, these features facilitate a reproducible workflow that balances the flexibility of interactive exploration with the rigor of reproducible research.

4. Applications

The Austrian Vacation Activities Dolnicar and Leisch (2003) and the Australian Vacation Activities Cliff (2009) datasets are used to illustrate the methods.

4.1. Austrian Vacation Activities dataset

The Austrian Vacation Activities dataset comprises responses from 2,961 adult tourists who spent their holiday in Austria during the 1997/98 season. Participants were asked to evaluate the importance of 27 different activities during their vacation. The survey categorized responses based on four levels of importance: "totally important", "mostly important", "a bit important", and "not important". For analysis, the responses were binarized: a value of 1 was assigned if the activity was rated as "totally important", and a value of 0 if any of the other categories were selected. The survey was conducted by the Europäisches Tourismus Institut GmbH at the University of Trier and focused exclusively on tourists who did not stay in the country's capital cities. Figure 4 shows three projections from a grand tour of this data: when mostly two features contribute to a projection the data will look apparently clustered, due to the binary values, but when more features contribute to the projection it looks more continuous.

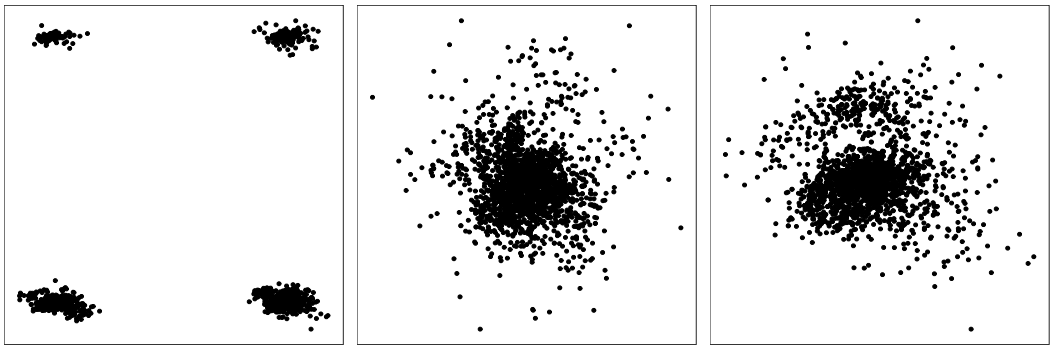

Figure 4: Three 2D projections from a grand tour on the Austrian Vacation Activities data. Because the 27 features are binary, you'll see some apparent clusters (left) but with this many binary features the data mostly looks continuous.

To gain further insight into the dataset a k-means clustering as described in Leisch *et al.* (2018) has been performed. Therefore, the function `stepcclust` of the R package `flexclust`(Leisch 2006) with $k = 6$ and $nrep = 20$ was used.

Feature selection

The original dataset contains p=27 features. Some of these features are more informative than others. We only want to keep the most informative ones. Additionally, interacting with the `lionfish` GUI becomes cumbersome when handling more than ~15 features, making

it necessary to reduce the dimensionality of the dataset. An effective and intuitive way to perform feature selection is by using the heatmap display within `lionfish`.

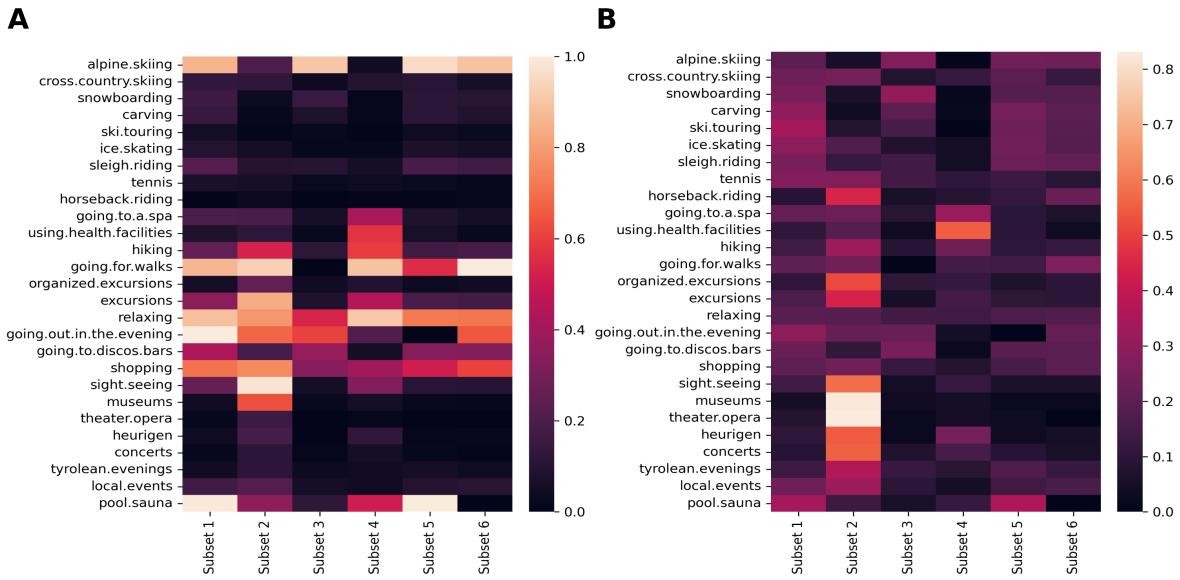

Figure 5: Traditional overview of clusters. Color represents (A) the intra-cluster fraction, and (B) the intra-feature fraction, with lighter indicating higher values. From A, we can see that cluster 3 tourists like alpine skiing, going out in the evening and going to discos and bars. They also like relaxing, shopping and sightseeing but these are popular among all tourists. From B, we can see the distribution of activities across clusters, e.g. using health facilities is popular in cluster 4 while going to a pool or sauna is popular in clusters 1 and 5.

In Figure 5A, where colors show normalised column counts, we can observe the general interests of tourists within each cluster. Some activities are high on almost all clusters (e.g. relaxing, shopping), and some are low on all clusters (e.g. ski touring and horseback riding), and ignoring these can be helpful when assessing the distribution of activities between clusters. When comparing clusters 5 and 6 we can see that alpine skiing is high in both, but cluster 5 tourists also like going to the pool or sauna, but cluster 6 tourists prefer going for walks.

In Figure 5B, we can determine whether for a particular feature by tourists in which clusters. For example, nearly all tourists who visited museums are grouped in cluster 2, and those who used health facilities are primarily attributed to cluster 4. Some activities, such as relaxing, are popular across all clusters.

These heatmaps can help with selecting features to focus on using the tour. Unpopular and universally popular activities can be removed. After performing the feature selection by unchecking the corresponding checkboxes in the GUI using this strategy, the following 12 activities remained: alpine skiing, going to a spa, using health facilities, hiking, going for walks, excursions, going out in the evening, going to discos/bars, shopping, sightseeing, museums, and pool/sauna.

We can now repeat the k-means clustering with `stepcclust` on the reduced dataset. Silhouette plots of both cluster solutions can be seen in Figure 6. By comparing both silhouette plots, we can see that the cluster solution with the reduced dataset results in a clustering of higher quality. Thus, we will continue with the analysis on the reduced dataset with the corresponding cluster solution. We can also see in Figure 6B that cluster 3 is of comparatively low quality. It is important to note that the silhouette scores were generally quite low, reflecting the lack of clearly separable clusters in the data. Consequently, there is no objectively correct or optimal solution. The primary aim of this analysis, therefore, is to gain insights into the data rather than to identify the optimal clustering configuration.

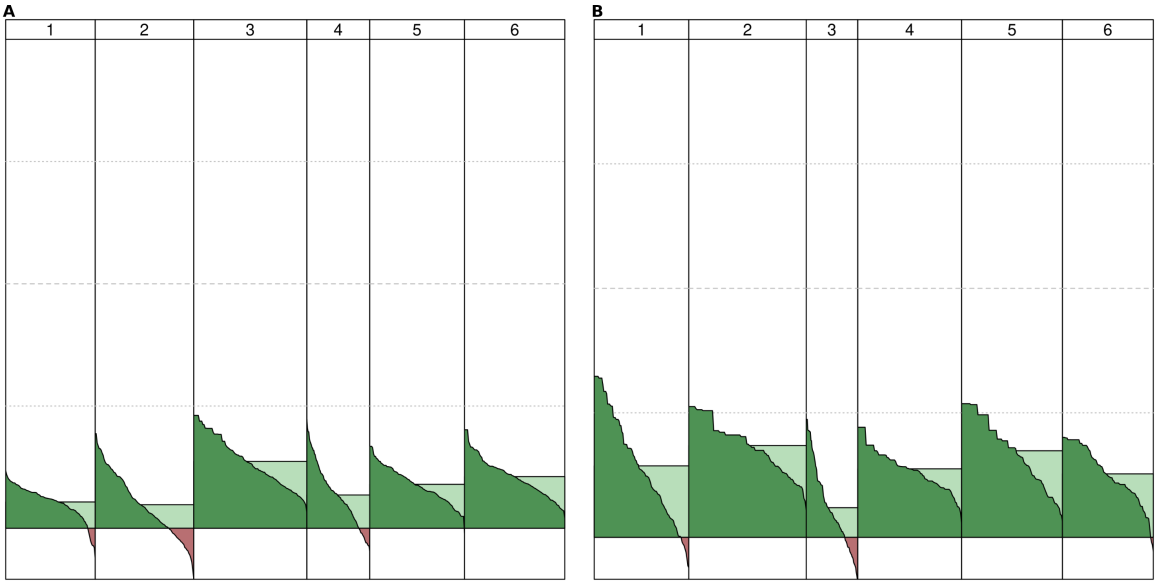

Figure 6: Comparison of silhouette plots of two k-means cluster solutions of the Austrian vacation activities dataset with k=6. (A) shows the silhouette plot of the k-means solution of the full dataset and (B) the silhouette plot of the k-means solution of the dataset after manual feature selection. We can see that the cluster solution with the reduced dataset achieved better silhouette scores and that clusters 1 and 3 contain observations with negative silhouette scores.

We can further explore the similarities between the clusters by initializing an `interactive_tour()` with a 2D tour based on the linear discriminant analysis (LDA) projection pursuit index. By navigating through the tour, we can observe various projections, and when a projection that separates the clusters is found, we highlight each cluster sequentially. The different highlighted clusters can be seen in Figure 6.

This process allows us to visually assess the separation and similarities between the clusters, providing insight into the structure of the dataset. By highlighting each cluster individually, we can evaluate their distinctiveness in different projections. The most influential features shown in Figure 7 are pool/sauna, alpine skiing, museums, and going to the spa. The projection roughly separates clusters 1 (blue), 2 (orange), and 3 (green) from each other and the other three clusters (red, violet and brown), which appear to be quite similar.

By manually manipulating the projection axes or initiating a local tour, we can gain further insight into the similarities between the different clusters. This interactive exploration allows for a more nuanced understanding of the relationships between clusters and the influence of key features on the separation of the data.

Redefining cluster assignments - learning more about museum goers

There are several reasons why we might want to manually modify a clustering solution. One is to capture observations that do not fit well within their assigned clusters. Another reason is to explore specific features in more detail. The advantage of manual cluster selection is that it preserves most of the current clustering structure, allowing us to adjust specific parts of the solution without starting from scratch. This approach is particularly useful when we already have a cluster solution that reveals interesting patterns in the data.

In Figure 6, we observed that clusters 1 and 3 contained data points that did not fit well into their respective clusters. To further investigate this, we can initialize an `interactive_tour()` with the following components:

- A 2D tour using the linear discriminant analysis projection pursuit index,

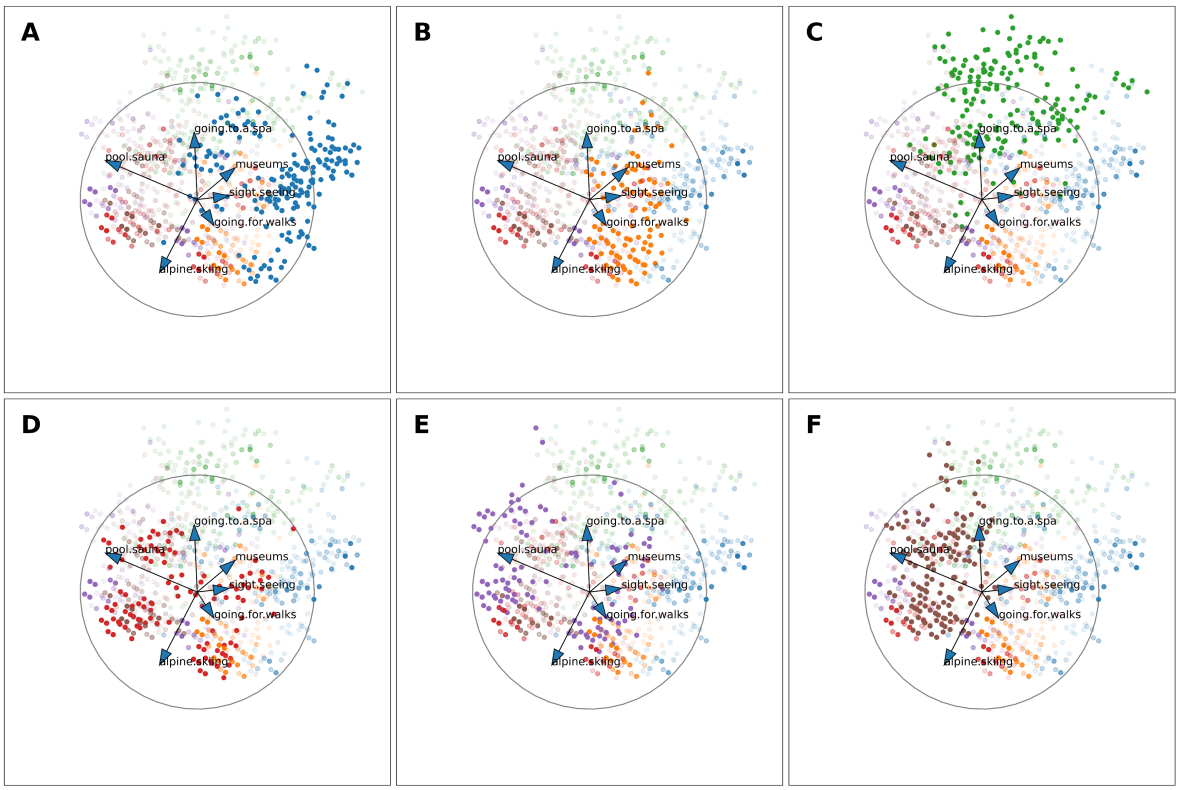

Figure 7: Display of a projection of the Austrian vacation activities dataset with the six clusters of the k-means cluster solution highlighted in different colors. Projection axes with a norm < 0.1 have been blend out to reduce cluttering. The colors indicate which clusters the highlighted observations belong to with cluster 1 being blue (A), cluster 2 being orange (B), cluster 3 being green (C), cluster 4 being red (D), cluster 5 being violet (E) and cluster 6 being brown (F). Some datapoints appear to be highlighted always, which occurs due to overlap of many datapoints in one spot. We can see that the projection separates some clusters well, however also that there is considerable overlap of clusters 4, 5 and 6.

- A heatmap showing the intra-cluster fraction,
- A 1D tour with the linear discriminant analysis projection pursuit index, and
- A mosaic plot.

This setup produces the display shown in Figure 8. It can also be seen that both clusters 1 and 3 contain tourists that didn't go alpine skiing and the main difference between them is that tourists in cluster 3 enjoyed going to the spa and health facilities as well as going to the pool, while the ones in cluster 1 didn't. Other than that, the clusters were mostly similar. Now we might be interested in the subset of tourists that enjoy going to museums. Therefore, we can adjust the projection axes so that these axes are elongated and point into different directions. We can see that there is indeed overlap between clusters 1 and 3, as shown in Figure 9. As a next step we can reassign the overlapping section to a new cluster - cluster 7 (pink). By selecting the checkbox for subset 7 and manually selecting the region of overlap, we can form a new cluster, which is visualized in Figure 10.

We can observe slight behavioral differences between tourists in clusters 1 and 7. Tourists in cluster 7 all enjoyed both museums and sightseeing, whereas most tourists in cluster 1 engaged in sightseeing but showed no interest in museums. Instead, participants in cluster 1 exhibited a greater preference for hiking. Despite this, tourists in both clusters generally shared similar interests. This insight could be valuable for enhancing museum marketing strategies. While clusters 1 and 7 have overlapping interests, it appears that current marketing efforts may

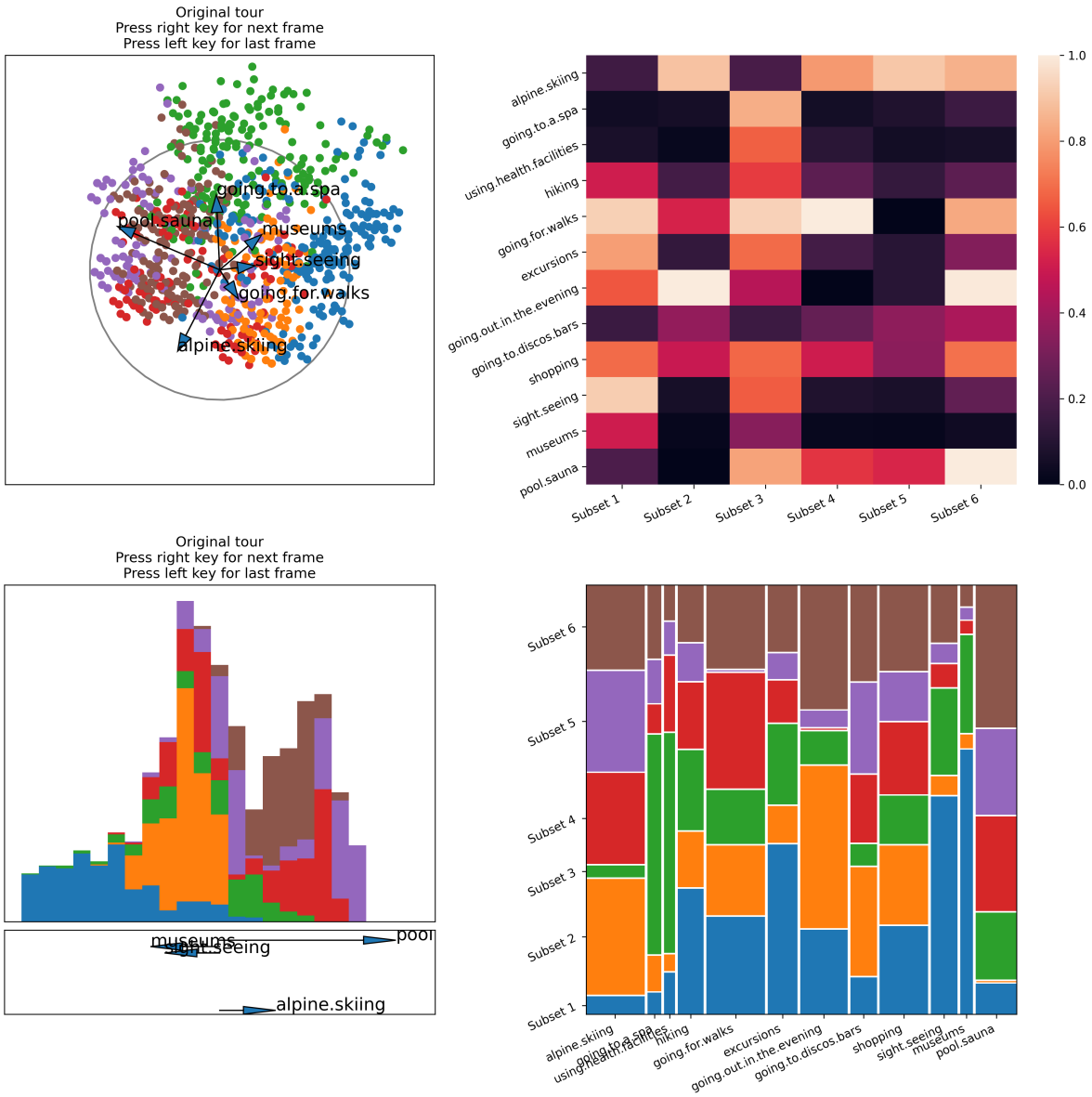

Figure 8: Interactive tour GUI loaded with multiple plots showing different aspects of the k-means solution of the Austrian vacation activities dataset. Projection axes with a norm $<$ 0.1 have been blend out to reduce cluttering. Top left: 2D tour with the linear discriminant analysis projection pursuit index. Top right: heatmap with the intra-cluster fraction. Bottom left: 1D tour with the linear discriminant analysis projection pursuit index. Bottom right: mosaic plot. Tourists in both clusters 1 and 3 didn't participate in skiing a lot, but tourists in cluster 3 were much more interested in going to the pool, spa and health facilities compared to cluster 1.

not effectively reach tourists in cluster 1. By increasing targeted marketing at hiking trails, popular excursion destinations, and shopping centers, it may be possible to attract more interest in museums from tourists in cluster 1.

4.2. Australian Vacation Activities dataset

The second dataset, the Australian Vacation Activities dataset, includes responses from 1,003 adult Australians who were surveyed through a permission-based internet panel. The survey was conducted in 2007. Participants were asked whether they engaged in 44 specific vacation activities during their most recent vacation within Australia. Similar to the Austrian dataset, responses were binarized: a value of 1 indicates that the participant took part in the activity,

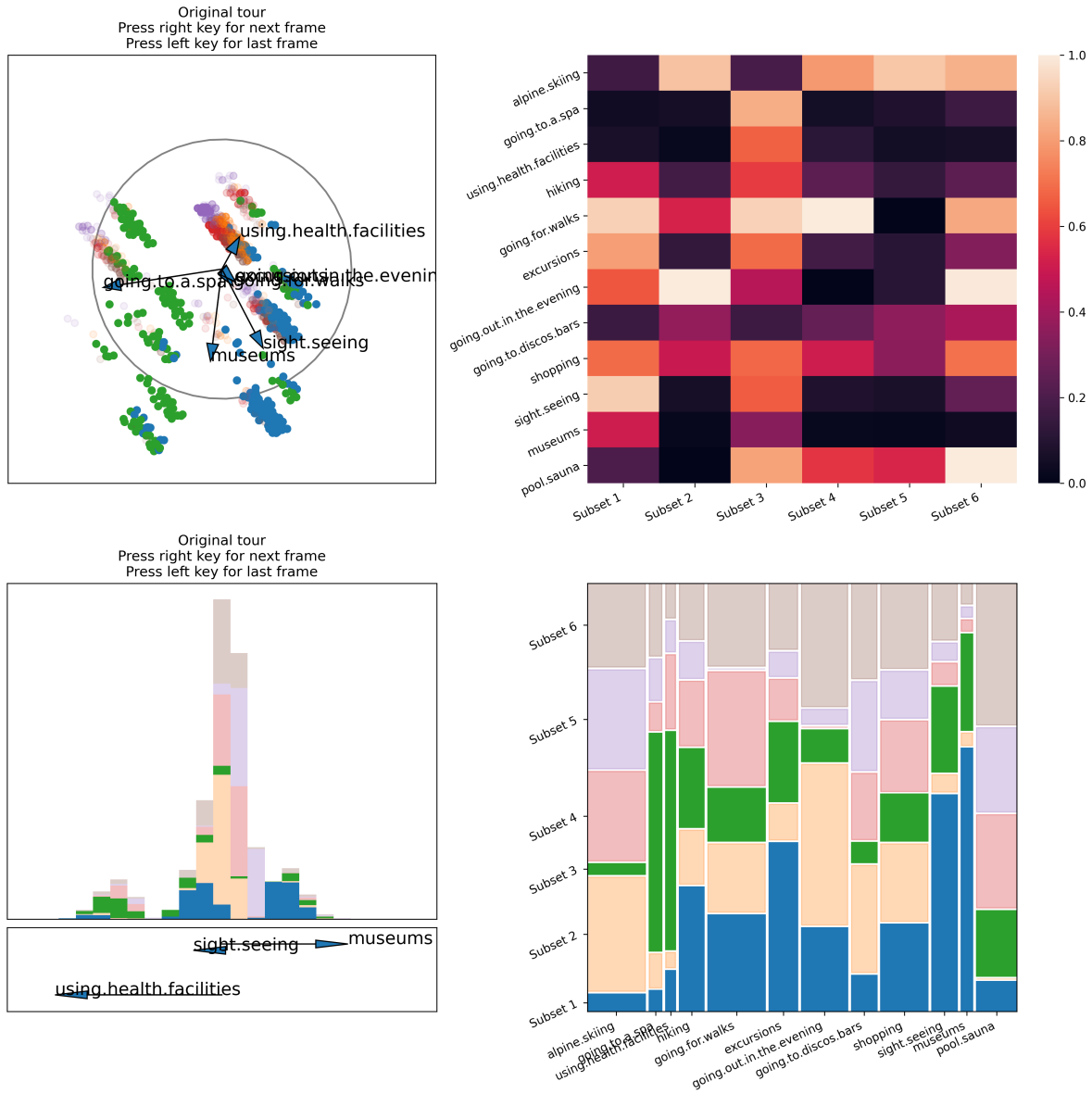

Figure 9: Interactive tour GUI loaded with multiple plots showing different aspects of the k-means solution of the Austrian vacation activities dataset, with manually adjusted projections. Projection axes with a norm < 0.1 have been blend out to reduce cluttering. Top left: 2D tour with the linear discriminant analysis projection pursuit index. Top right: heatmap with the intra-cluster fraction. Bottom left: 1D tour with the linear discriminant analysis projection pursuit index. Bottom right: mosaic plot. Changing the projection axes of "going to a spa", "museums", "sightseeing" and "using health facilities" reveals the preferences and overlap of clusters 1 and 3.

while a value of 0 signifies they did not. Surveys where participants claimed they partook in more than 40 activities or no activity at all were removed as they are considered faulty.

Feature selection

At first, hierarchical clustering using Ward's method (Murtagh and Legendre 2014) and the Jaccard index was applied to the features. The resulting dendrogram is shown in Figure 11. Based on this clustering, $k = 15$ clusters were identified, and generally, only one representative feature from each cluster was selected for further analysis. Clusters containing unpopular activities, such as "Adventure", which only had 42 participants, were discarded. The cluster containing popular features like "Beach", "Swimming", "ScenicWalks", "Markets", "Sightsee-

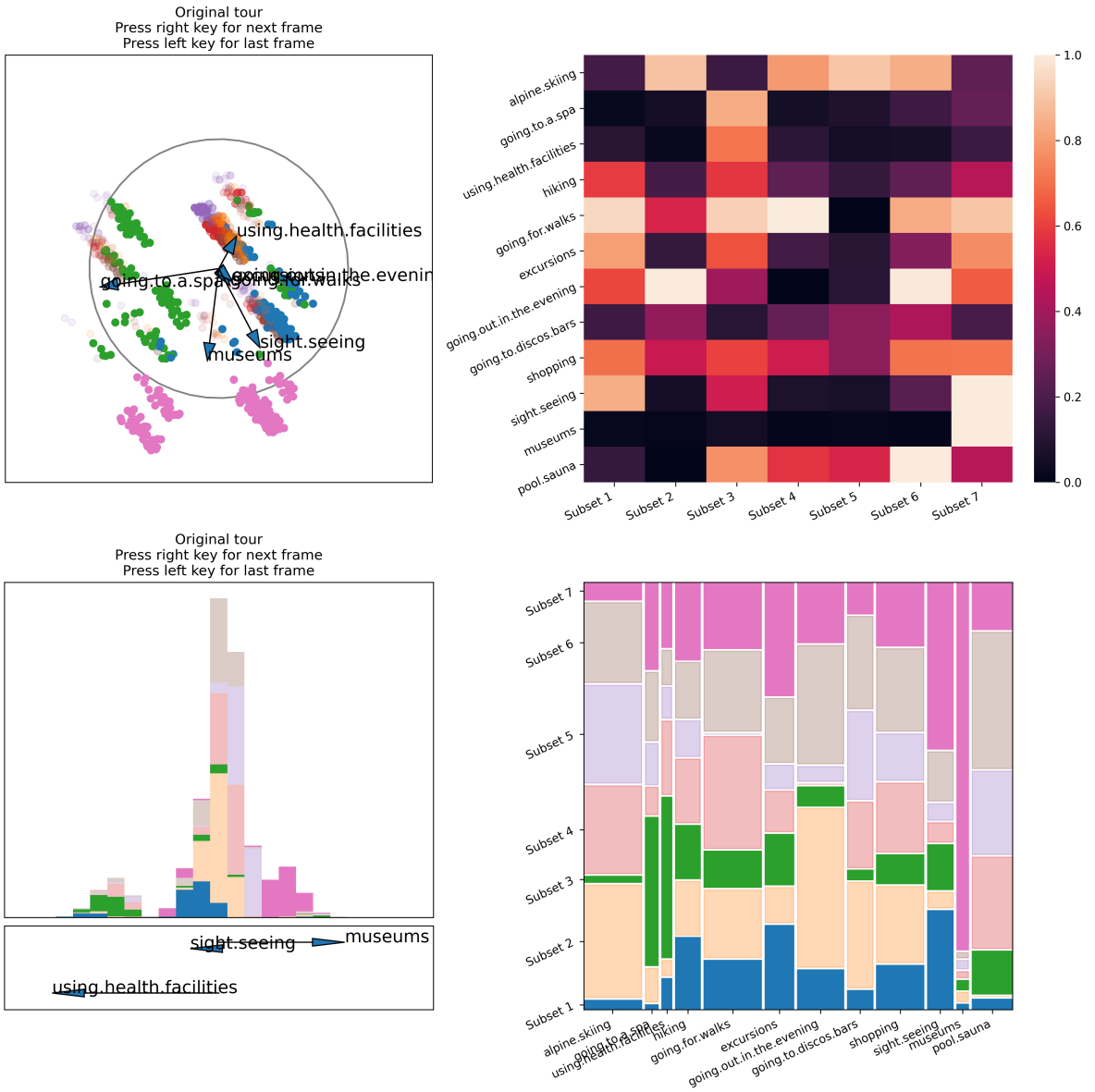

Figure 10: Interactive tour GUI loaded with multiple plots showing different aspects of the k-means solution of the Austrian vacation activities dataset, after new subsets have been selected. Projection axes with a norm < 0.1 have been blend out to reduce cluttering. Top left: 2D tour with the linear discriminant analysis projection pursuit index. Top right: heatmap with the intra-cluster fraction. Bottom left: 1D tour with the linear discriminant analysis projection pursuit index. Bottom right: mosaic plot. We can see that now almost all museum goers are in the new manually selected subset 7 and what their preferences are compared to the other clusters.

ing", "Friends", "Pubs", "BBQ", "Shopping", "Eating", "EatingHigh", "Movies", and "Relaxing" was treated differently. Multiple features from this cluster were retained to preserve as much information as possible. After feature selection, the observations were reclustered using k-means with $k = 15$.

Segmentation of solo travellers

In the heatmap displaying the intra-cluster fraction shown in Figure 12, we observe that subsets 1, 2, and 6 tend to prefer travelling alone. As a tourist agency, we might be interested in targeting solo travellers more effectively. However, subsets 2 and 6 also include individuals who enjoy spending time with their friends. To further explore the dataset, we can launch

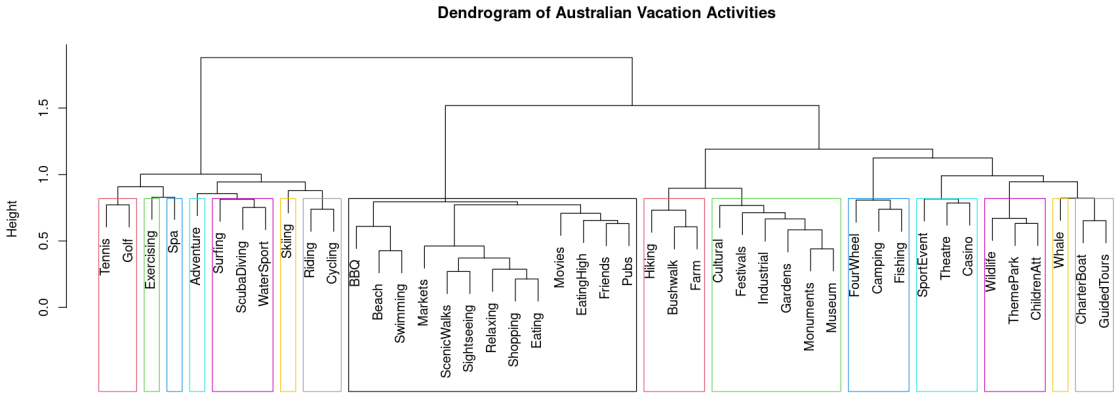

Figure 11: Dendogram of the features of the Australian Vacation Activities dataset using Ward's method with the Jaccard index. Features which were clustered together are marked by the colored boxes. We can see, which activities had similar patterns in tourist interest.

an `interactive_tour()` with the same configuration as in the previous example. To achieve better separation between the subsets, we can skip to the last frames of a 2 dimensional tour optimized for the LDA index.

Since we are particularly interested in tourists who did not spend time with friends, we can extend the projection axis "Friends" outward to separate the data based on that feature. Datapoints in the opposite direction of the "Friends" axes are the solo travellers we are interested in, which can be seen in Figure 13 on the left side of the top left plot. Subsequently, we can pull all other projection axes in one direction to separate data points based on their overall activity level. Tourists that fall in the direction of the axes generally engaged in more activities compared to those in the opposite direction of the projection axes. This separation is evident, as observations in subset 6 (brown) are located opposite the axes, and the heatmap shows that they did not engage in many activities. Similarly we can also see that subset 2 (orange), which contains quite active tourists, is shifted towards the direction of the axes.

Using this logic, we can segment the data points on the left into three segments: active tourists (more upward), moderately active tourists (center), and largely inactive tourists (bottom). The result of this segmentation can be seen in Figure 14. Analyzing the heatmap, we can identify several interesting patterns.

By comparing subset 3 (green), which contains active tourists who spent time with their friends, with subset 7 (pink), the active tourists who traveled alone, we notice that subset 7 showed less interest in visiting the casino, theatre, and chartering a boat.

We also observe distinctive differences in the interests of the subsets that traveled alone. Subset 7 was interested in relaxing, shopping, sightseeing, wildlife, going to the beach, visiting pubs, and exploring farms. Given that subset 7 showed a notable interest in going to pubs, we can assume they want to meet new people. This insight can be utilized in a marketing campaign that bundles the activities subset 7 is interested in, creating a package tailored for solo travellers. This way, they can meet other solo travellers while engaging in activities they enjoy.

Although subset 8 (grey) was generally less active, almost everyone still engaged in sightseeing. For them, the focus was on sightseeing, relaxing, shopping, and going to the beach, with much less interest in other activities. It can be assumed that they value their time alone. These tourists could potentially be targeted more effectively by offering sightseeing options with minimal interaction, such as using a phone app to provide information about interesting locations, rather than relying on a tour guide.

Finally, since subset 9 (gold) is notably inactive, one might infer that they prefer spending much of their time in their accommodation. Consequently, these individuals might be most interested in accommodations that offer well-equipped, comfortable living spaces with amenities that cater to relaxation and leisure.

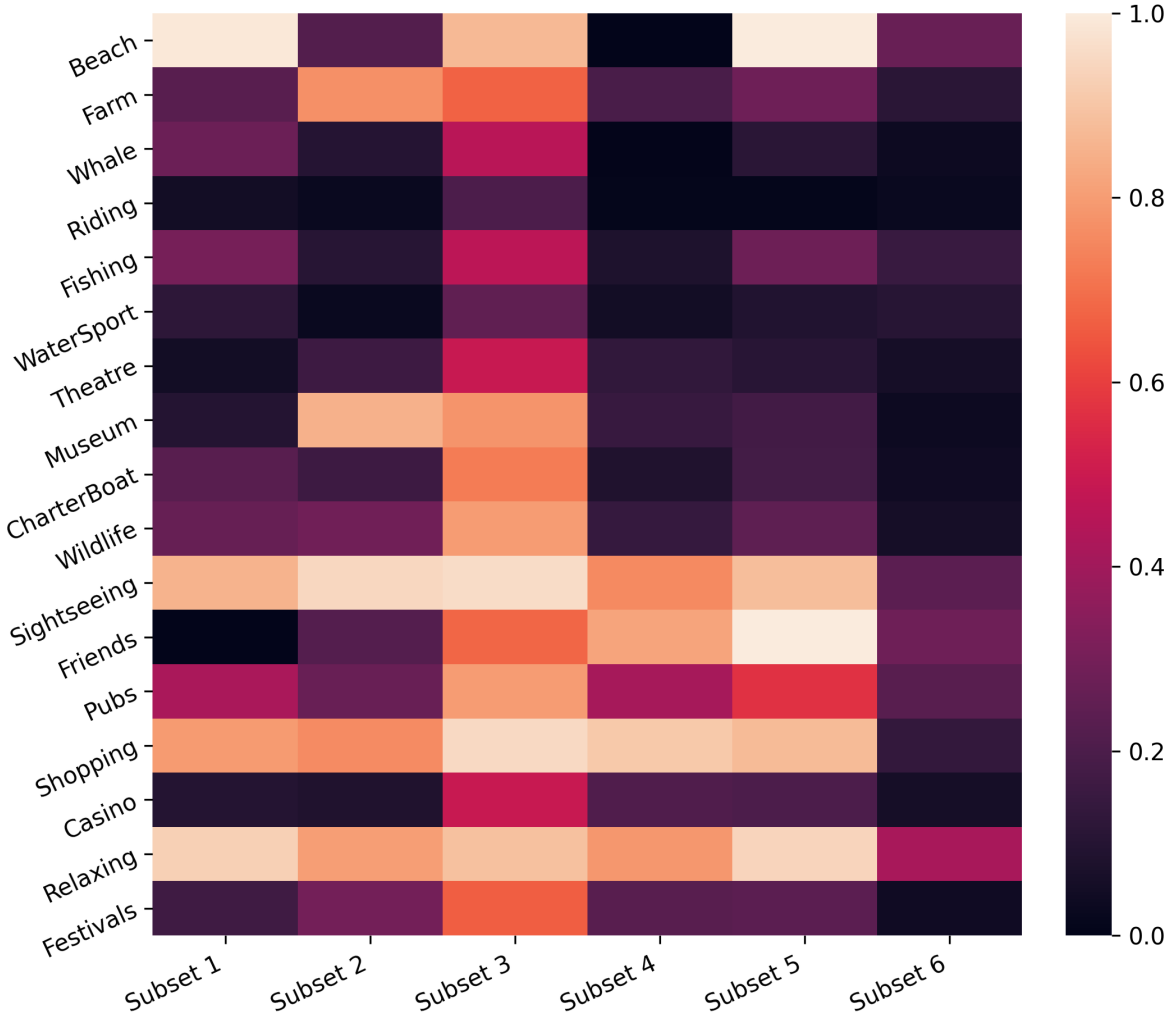

Figure 12: Heatmap of the intra cluster fraction (indicated by color) of the k-means solution with $k = 6$ and the reduced feature subset. We can observe that tourists in subsets 1, 2 and 6 all prefer to travel without their friends.

5. Discussion

One might question the necessity of manual exploratory data analysis, considering it is inherently subjective and relies heavily on intuition. However, in situations like those presented here, where there is no clearly defined or optimal clustering solution or feature selection, manual exploration becomes indispensable. While it is possible to optimize clustering metrics to improve separation between clusters, this alone may not yield conclusions that are useful for practical applications. The lack of clear boundaries and the overlapping nature of clusters in the datasets underscore the limitations of purely automated methods in capturing the complexity and nuance of real-world data.

In such cases, manual exploration allows analysts to interweave expert knowledge, intuition, and specific objectives with the initial clustering solution, which serves as the backbone of the analysis. This approach is particularly valuable when no single solution can be deemed

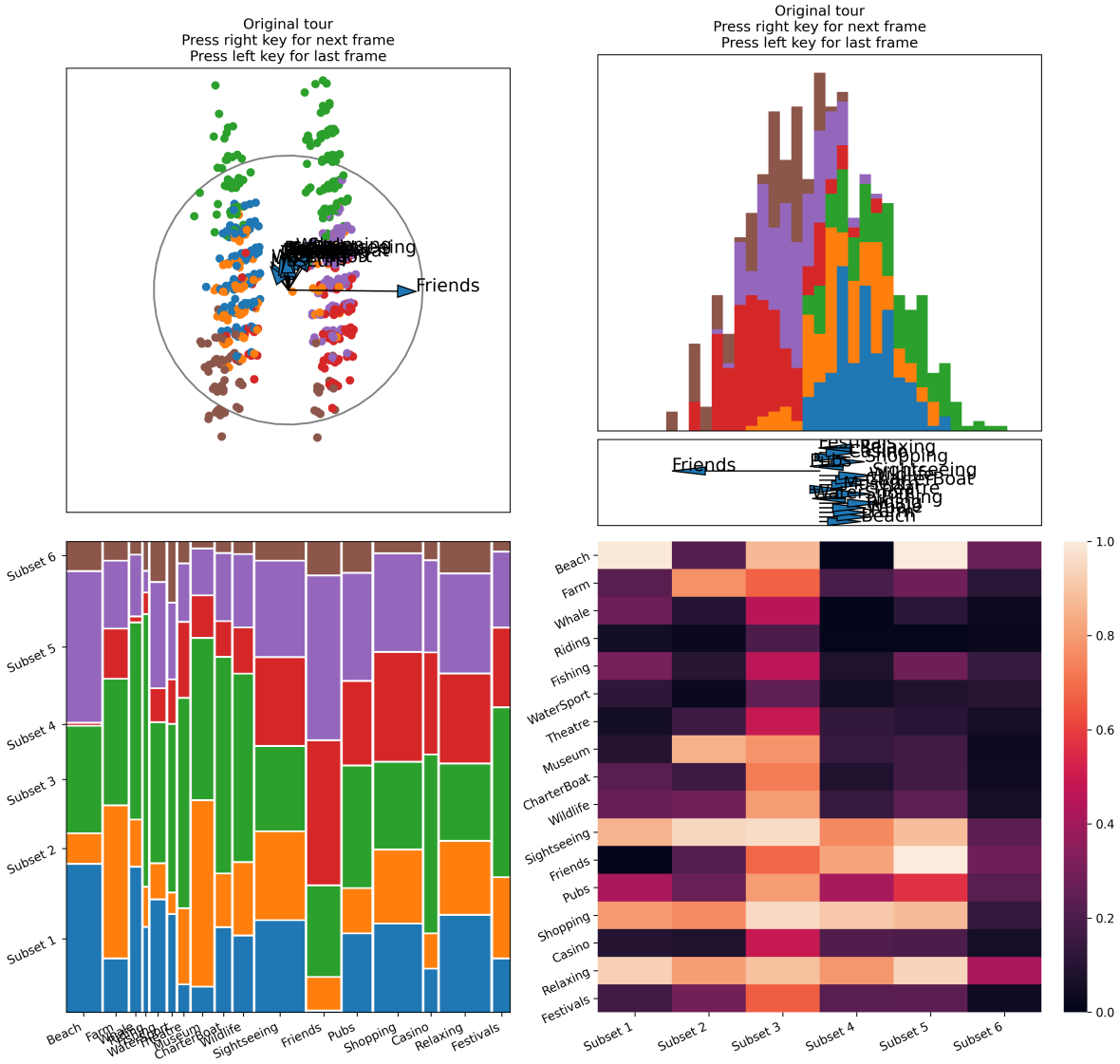

Figure 13: Interactive tour GUI loaded with multiple plots showing different aspects of the k-means solution of the Australian vacation activities dataset, with the projection axes of the feature "Friends" pointing into one direction and all the other ones into another one. Top left: 2D tour. Top right: 1D tour. Bottom left: Mosaic plot. Bottom right: Heatmap with the intra cluster fraction. We can see how the feature "Friends" and the general activity level (represented by the other projection axes pointing into one direction) separates the data.

"correct". For instance, in the Austrian dataset, leveraging the interactive GUI for feature selection enabled us to isolate the most informative activities and gain deeper insights into the preferences of tourists who might be interested in visiting museums. This understanding facilitated targeted recommendations for increasing museum attendance by focusing on tourists frequenting hiking trails, excursion spots, and shopping centers. Such insights are challenging to extract through automated optimization alone.

Similarly, in the Australian dataset, manual exploration allowed us to refine the understanding of solo travellers by dividing them into three distinct subsegments: highly active, moderately active, and largely inactive tourists. This nuanced segmentation, derived from a blend of automated clustering and manual adjustments, offers a deeper understanding of the varied needs and behaviors within this group, enabling more effective marketing strategies.

While some plots supported by the `lionfish` package are specifically designed for analyzing

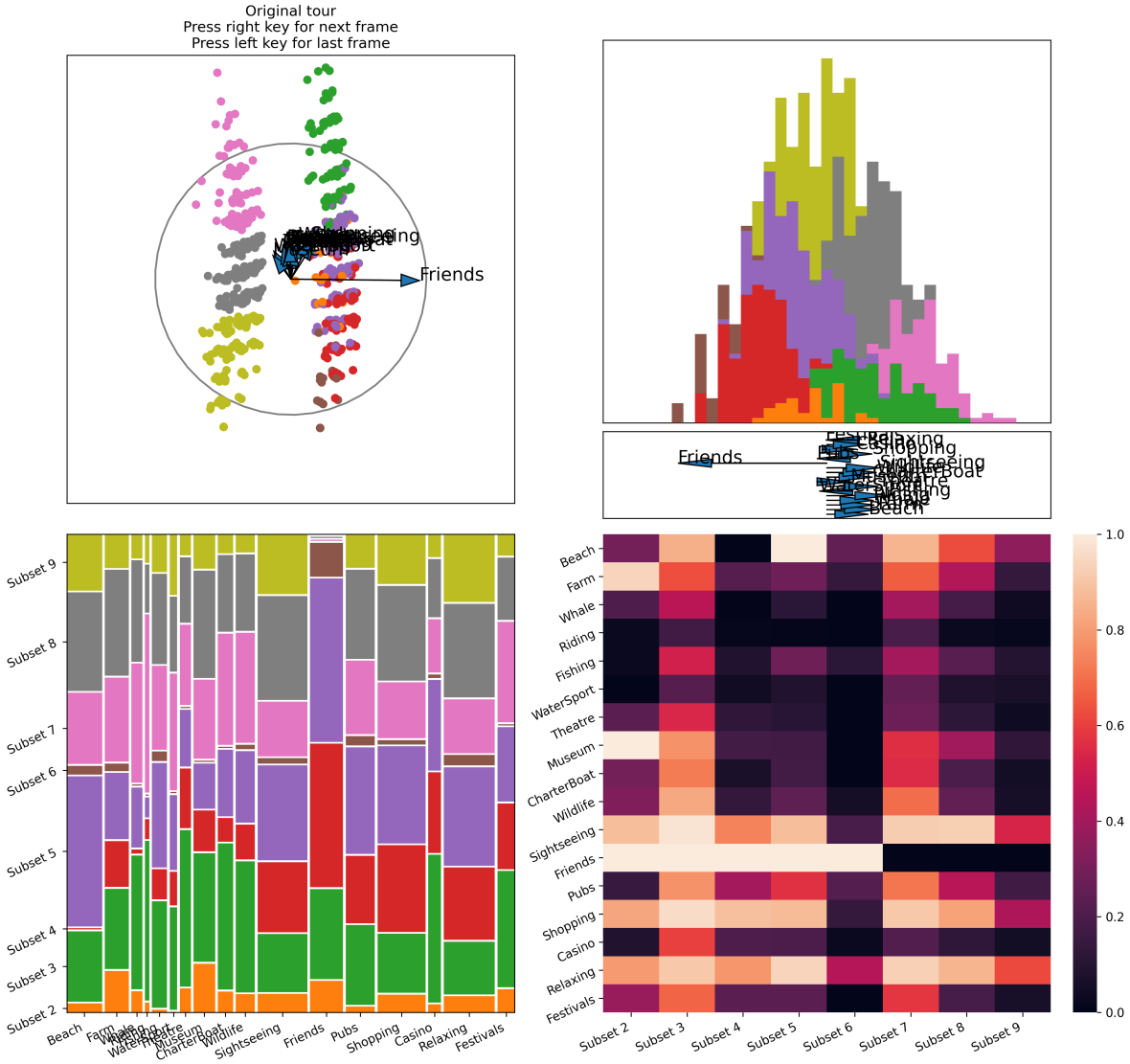

Figure 14: Interactive tour GUI as seen in 13, but after sub-selection of subsets 7 (pink), 8 (grey) and 9 (gold). Top left: 2D tour. Top right: 1D tour. Bottom left: Mosaic plot. Bottom right: Heatmap with the intra cluster fraction. We can observe the preferences of three new subsegments, which can be interpreted as very active (pink), moderately active (grey) and inactive (gold) tourists travelling without their friends.

binary survey data, it is important to emphasize that its capabilities extend far beyond this data type. The versatility of tours has been demonstrated across various applications in the past, showcasing their effectiveness in exploring complex, high-dimensional datasets. The interactive GUI enables seamless exploration of both one- and two-dimensional tours, regardless of the data being analyzed, providing a powerful tool for uncovering patterns and insights across diverse domains.

6. Conclusion

Ultimately, manual data exploration serves as a useful complement to automated methods, providing the flexibility to incorporate context, expert judgment, and specific analytical goals. This approach enables analysts to refine initial results and adapt them to the complexities

of real-world scenarios, leading to more nuanced interpretations and actionable insights. The `lionfish` package offers an organised and responsive interactive tool for conducting such analyses, bridging the gap between automated clustering and exploration.

By integrating interactive visualization capabilities, the `lionfish` package empowers users to dynamically engage with their data, making it possible to uncover subtle patterns and relationships that might otherwise remain hidden. This is especially valuable in tackling complex datasets with the mindset that an automated solution needs to be validated. The package has broad applicability across various data types and analytical contexts where clustering is used. The flexibility of setting up the GUI elements from the command line allow it to be tailored for different applications.

Future developments in the software might include expanding the range of interactive features and providing additional visualization methods. Because it is built on R and python, it would also be possible to integrate more closely with the machine learning algorithms, such as generating new cluster solutions directly from the GUI. For more complex shapes, containing concavities, or non-linear boundaries, implementation of sliced tours (Laa *et al.* 2020), would be recommended.

In summary, `lionfish` represents a significant advancement in the toolkit of data analysts, offering a novel way to balance automated analysis with human intuition and domain expertise, thereby facilitating a deeper and more comprehensive understanding of complex datasets.

References

Asimov D (1985). "The Grand Tour: A Tool for Viewing Multidimensional Data." *SIAM Journal of Scientific and Statistical Computing*, **6**(1), 128–143. ISSN 0196-5204. doi:10.1137/0906011. URL http://dx.doi.org/10.1137/0906011.

Cliff K (2009). "A formative Index Of Segment Attractiveness: Optimising Segment Selection for Tourism Destinations."

Cook D, Buja A (1997). "Manual Controls for High-Dimensional Data Projections." *Journal of Computational and Graphical Statistics*, **6**(4), 464–480. ISSN 1061-8600. doi:10.2307/1390747. URL http://www.jstor.org/stable/1390747.

Cook D, Buja A, Cabrera J, Hurley C (1995). "Grand Tour And Projection Pursuit." *Journal of Computational and Graphical Statistics*, **4**(3), 155–172.

Dolnicar S, Leisch F (2003). "Winter Tourist Segments In Austria: Identifying Stable Vacation Styles Using Bagged Clustering Techniques." *Journal of Travel Research*, **41**(3), 281–292.

Hart C, Wang E (2023). "Taking The Scenic Route: Interactive And Performant Tour Animations." *The R Journal*, **15**, 307–329. ISSN 2073-4859. doi:10.32614/RJ-2023-052. Https://doi.org/10.32614/RJ-2023-052.

Hunter JD (2007). "Matplotlib: A 2D Graphics Environment." *Computing in Science & Engineering*, **9**(3), 90–95. doi:10.1109/MCSE.2007.55.

Laa U, Aumann A, Cook D, Valencia G (2023). "New And Simplified Manual Controls for Projection And Slice Tours, With Application To Exploring Classification Boundaries In High Dimensions." *Journal of Computational and Graphical Statistics*, **32**(3), 1229–1236.

Laa U, Cook D, Valencia G (2020). "A Slice Tour for Finding Hollowness In High-Dimensional Data." *Journal of Computational and Graphical Statistics*, **29**(3), 681–687. doi:10.1080/10618600.2020.1777140.

Lee EK, Cook D, Klinke S, Lumley T (2005). "Projection Pursuit for Exploratory Supervised Classification." *Journal of Computational and Graphical Statistics*, **14**(4), 831–846.

Lee S, Cook D, da Silva N, Laa U, Spyrison N, Wang E, Zhang HS (2022). "The State-Of-The-Art On Tours for Dynamic Visualization Of High-Dimensional Data." *WIREs Computational Statistics*, **14**(4), e1573. doi:https://doi.org/10.1002/wics.1573. https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1573, URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1573.

Leisch F (2006). "A Toolbox for k-Centroids Cluster Analysis." *Computational Statistics and Data Analysis*, **51**(2), 526–544. doi:10.1016/j.csda.2005.10.006.

Leisch F, Dolnicar S, Grün B (2018). *Market Segmentation Analysis: Understanding It, Doing It, and Making It Useful*.

Lundh F (1999). "An Introduction To Tkinter." *URL: www. pythonware. com/library/tkinter/introduction/index. htm*.

Murtagh F, Legendre P (2014). "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?" *Journal of classification*, **31**, 274–295.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Schimansky T (2024). "CustomTkinter." https://github.com/TomSchimansky/CustomTkinter.

Ushey K, Allaire J, Tang Y (2024). *reticulate: Interface To 'Python'*. R package version 1.38.0, https://github.com/rstudio/reticulate, URL https://rstudio.github.io/reticulate/.

Van Rossum G, Drake FL (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA. ISBN 1441412697.

Wickham H, Cook D, Hofmann H, Buja A (2011). "tourr: An R Package for Exploring Multivariate Data With Projections." *Journal of Statistical Software*, **40**(2), 1–18. URL https://doi.org/10.18637/jss.v040.i02.

Affiliation:

Matthias Medl
Institute of Statistics
BOKU University Vienna
E-mail: matthias.medl@boku.ac.at