"*tourr*": An R Package for Exploring Multivariate

Daniel Felbah, Yao Chen, and Gaurav Sitaula
MATH 6820: Statistical Computing
December 11, 2017

Published By: Hadley Wickham, Dianne Cook, Heike Hofmann, and Andreas Buja (2011)

1. Introduction

Our project is to introduce a R package which produces tours of multivariate data. The package includes functions for creating different types of tours, including grand, guided, and little tours, which project multivariate data (p-D) down to 1, 2, 3, or, more generally, d ($\leq$ p) dimensions.

As the student of Mathematics and Statistics, we have to play around with the multivariate data most of the time. Many of us still struggle to explore multivariate data. We are looking for a method to tell us about all the structure in the data. From the article, we learned that the tour method, which shows a smooth sequence of projections of high-dimensional data. The authors use the flea [1] dataset for the multivariate exploration. Most of the in-class examples for this class were performed on the iris [3] data. So, for the final project, we chose the same multivariate iris data since we are more familiar to this data. The tour is most useful when looking for clusters, outliers, non-linear dependence, and to get an overview of the types of structures present in multivariate data. The tour gets us beyond the single static data projection produced by many statistical methods like principal component analysis, linear discriminant analysis, multidimensional scaling, projection pursuit or independent components analysis [5]. With the tour the data analyst sees many data projections, including ones revealing many different aspects of the data and how these are related to each other. We can use the tour package to draw the graph of the multivariate data which can tell us about all the structure in the data.

2. Results

2.1 The tour method

The different methods in **tourr** package reflect the construction of the tour: To create a tour we need to combine a dataset with a type of tour path and display method.

The syntax for the tour function is shown below:

```
R> tour_function(data, tour_path, display_method)
```

From the function we can see there are three arguments inside:

i. A data matrix (n × p), with real-valued elements.
ii. A tour path that produces a smooth sequence of projection matrices (p × d).
iii. A display method that renders the projected data.

First, the most important step is make an n x p matrix. To construct a multivariate data, in our project we chose the iris data which in MASS package.

Second argument is using the p-dimensional data onto d projections.

The third argument is a display of the data, we will introduce later in the report.

The structure of the tourr package reflects this construction of the tour: To create a tour we need to combine a dataset with a type of tour path and display method.

Some examples for the multivariate data:

```
R> animate(iris[, 1:4], grand_tour(d = 2), display = display_xy())
```

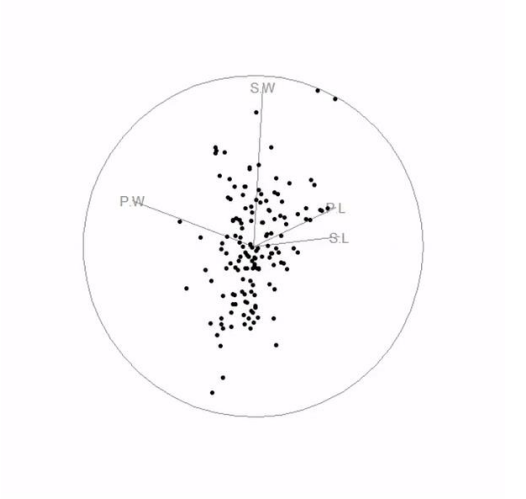

Fig 1: A 2-D tour of iris data displayed with a scatterplot

```
R> animate(iris[, 1:4], grand_tour(d = 3), display = display_depth())
```

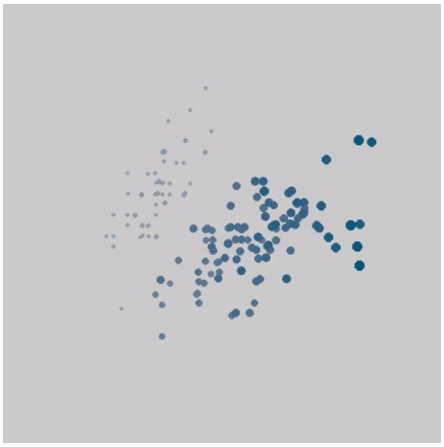

Fig 2: A 3-D tour of iris data displayed with simulated depth

It Uses red-blue anaglyphs to display a 3-D tour path. You'll need a red- blue glasses to visualize this kind of displays in 3-D.

```
R> animate(iris[, 1:4], grand_tour(d = 4), display = display_pcp())
```

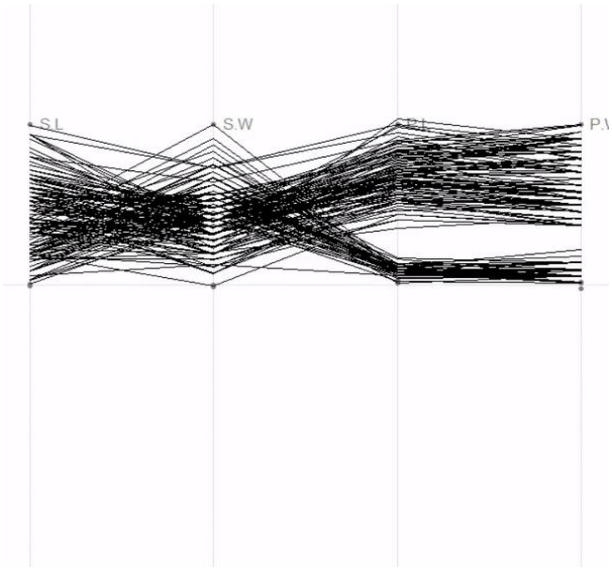

Fig 3: A 4-D tour of iris data displayed with a parallel coordinates plot

By using the different display methods, we produced three different dimensional data graphs.

The tour path is made up of two parts:

i. Interpolator: smoothly interpolates between pairs of projections
ii. Generator: produce the projections

2.1.1 Interpolator

Interpolation is the method to produce a smooth interpolation between planes. And all the generators rely on it currently. We any two frames that rotate the same plane will generate d-D

projections within the p-D data which have equivalent information. Our goal is to avoid unnecessary rotation within the plane. To get a new view of the data from different angles, we should move the plane, makes us see how is the structure of the data and the true relationship between the data. Which we showed the graphs at the previous examples.

2.1.2 Generator

There are five generators in tour package

1. The `grand_tour`

This generator picks a p × d projection matrix randomly. It allows a curve filling the space of projections, ensuring show every possible projection of the data. It is useful for getting a comprehensive overview of a dataset, even for a large number of dimensions. After taking a long time, we still can see the output of the higher projections.

2. In the `guided_tour`

Instead of picking a new projection completely at random, we pick one that is more interesting. Over time, this leads to picking projections that are closer to the current projection, so that we eventually converge to a single maximally interesting projection, in a spirit like simulated annealing.

3. The `planned_tour`

This is the most constrained tour. If we already know which way we want to rotate, the `planned_tour` method enable the data to go through a previous set of frames. The planned tour is most useful if you have a sequence of projections saved by an earlier tour for later replay.

4. The `dependence_tour`

This one combines an n independent 1-D tours. It has a single argument, a numeric vector that specifies which 1-D tour each variable should be assigned to. For example, c (1, 1, 2, 2) specifies that the first two variables will be displayed with a 1-D tour on the first axis, and the second two with a 1-D tour on the second axis.

5. The `local_tour`:

Alternates between a specified starting position and nearby random projections. This allows us to inspect the local neighborhood of a projection.

In our project, the first two generators are most useful because they allow to draw graph from the random data set.

2.2 Display methods

The display methods produce a visual rendering of the tour, there are nine methods for displaying the tour. We classified them into four sections:

a. 1-D

A 1-D projection of the data can be thought of like the first principal component in principal component analysis, or even the linear combination of variables forming a regression equation.

– `animate_dist` (histogram, average shifted histogram, density plot)

– `animate_image` (image plot)

– `animate_ts` (time series)

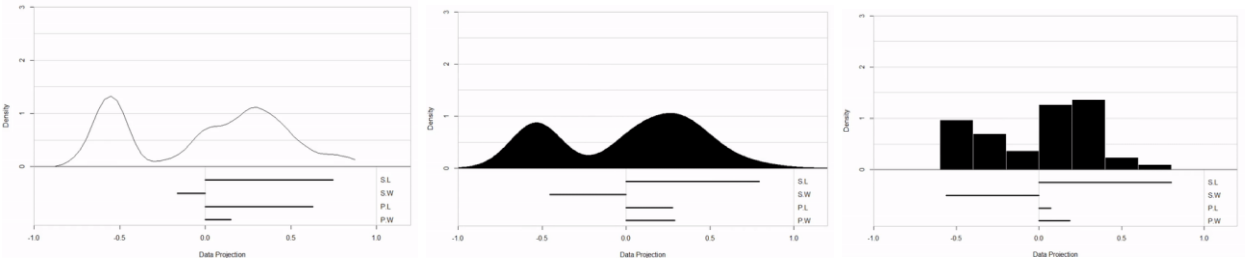

Fig 4: Three visualizations of a 1-D projection. The left plot shows the average shifted histogram. The middle plot shows the density plot. The right plot shows the histogram.

In each display the projection coefficients, that range between -1 and 1, are displayed as line segments underneath the plot.

b. 2-D

– `animate_xy (data, tour_path = grand_tour(), ...)`

Arguments:

Data: matrix, or data frame containing numeric columns `tour_path` tour path generator, defaults to 2-D grand tour.

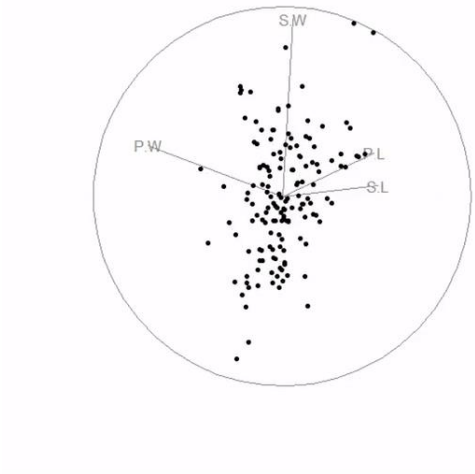

Fig 5: Visualization of a 2-D projection

c. 3-D

It provides two methods:

– `animate_stereo` (anaglyphs)

We will need red-blue glasses to see this tour graph.

– `animate_depth` (3-D depth cues)

This display uses depth cues which is closer points occlude further away points, size (closer points are bigger) and saturation (distant points are hazier and less saturated). The illusion of depth is less convincing than with anaglyphs, but it does not require any special equipment.

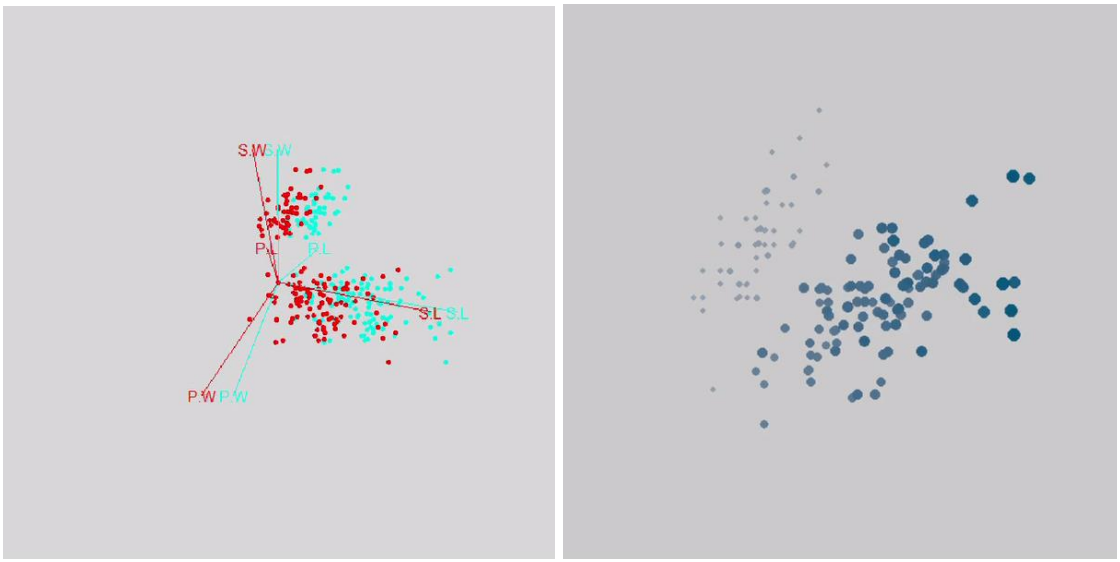

Fig 6: Two visualizations of a 3-D projection. The left plot shows the stereo plot of iris data. The right plot shows the depth plot.

d. K-D

– `animate_andrews` (Andrews curves)

– `animate_faces` (Chernoff faces)

– `animate_pcp` (parallel coordinates)

– `animate_scatmat` (scatterplot matrix)

– `animate_stars` (star glyphs)

Animation function:

The animate_** functions use animate (data, tour_path, display) to produce the tour animations.

There are three main argument in the function, and further arguments to control the speed, frame rate, length of the tour and data scaling.

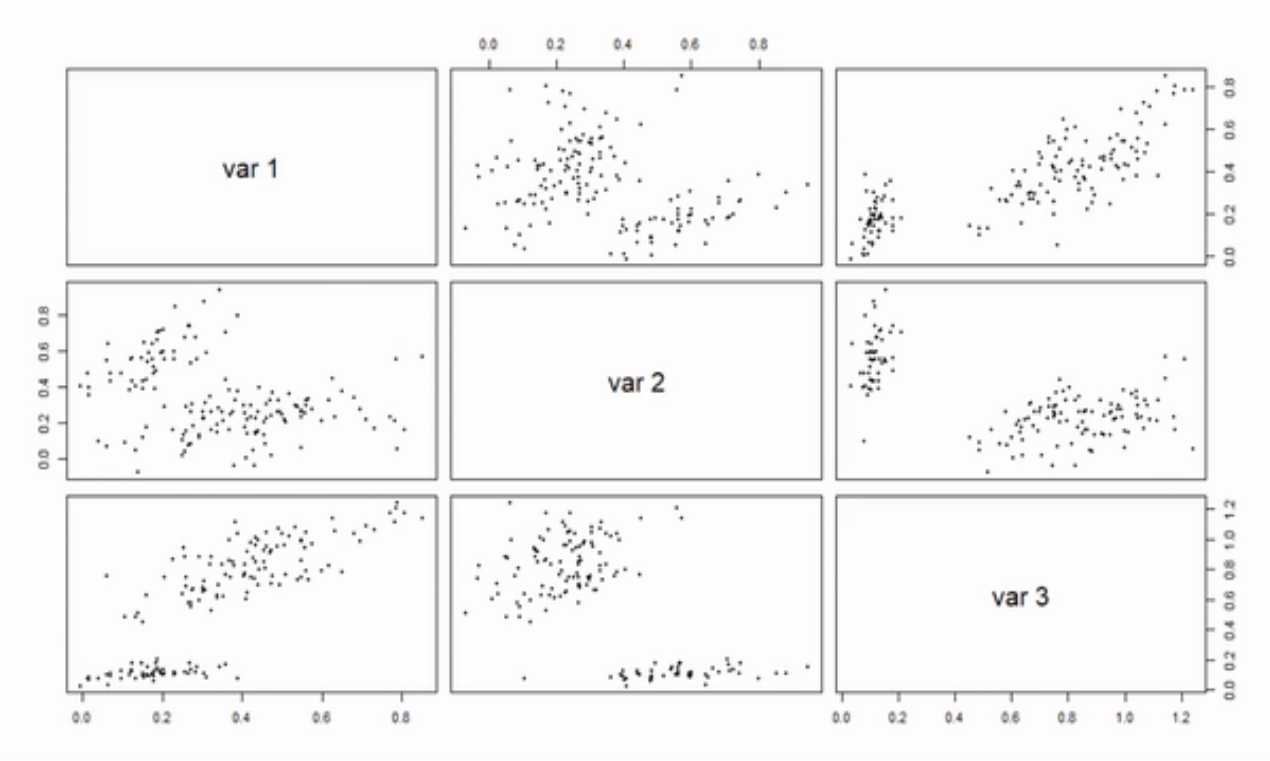

Fig 7: Scatterplot matrix of iris data in a 3-D projection

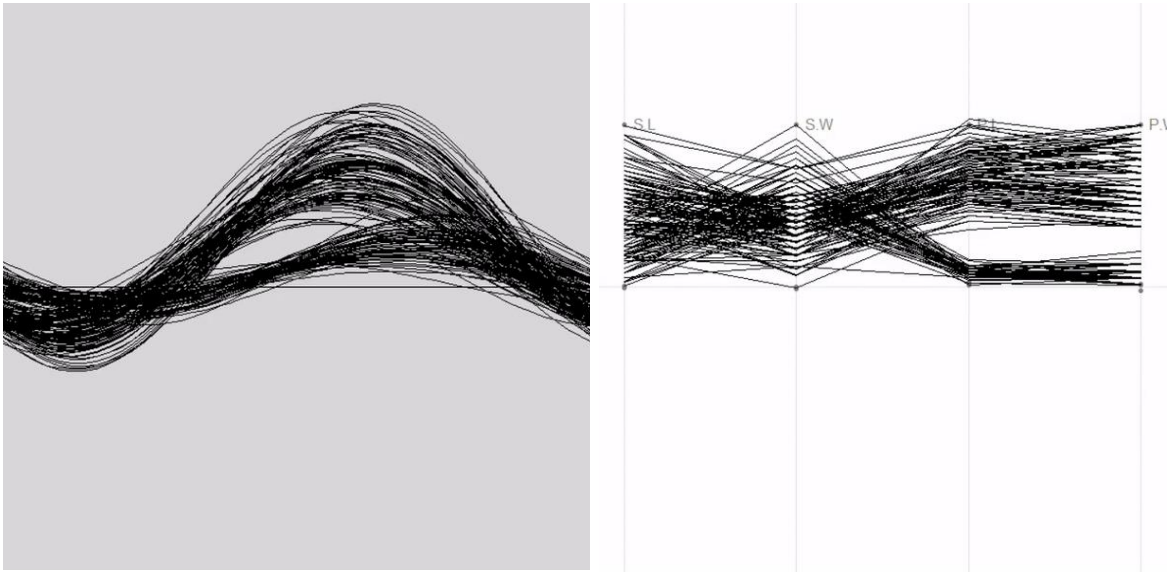

Fig 8: Two visualizations of a k-D projection. The left plot shows the Andrews curve. The right plot shows the parallel coordinate plot of iris data.

2.3 The Data:

We are working on a matrix or data frame of p continuous variables. By default, these variables are scaled to each have range [0,1], but if your variables are measured on a common scale already, you can turn this off by setting rescale = FALSE.

Optionally, the data can also be sphered prior to display with the tour. Sphering rotates and scales the data so that it has a diagonal variance-covariance matrix, this is useful because it removes typically obvious correlation effects and makes it easier to see subtler non-linear patterns.

2.4 Options

2.4.1 Frozen Geodesics

The frozen geodesics is the additional feature in this package that allows the coefficient of some variables to be constant in the projection. These were used in **XGobi [2].** A frozen tour fixes some of the values of the orthonormal projection matrix and allows the others to vary freely according to any of the other tour methods. This frozen tour is a frozen grand tour. The components of this option are: Freezer matrix, Freeze operation, and thaw operation. The freezer matrix is a matrix of frozen values with the missing values of warm components. A freeze operation zeroes out the frozen variable values in the input projection matrix and finally the thaw operation thaws the input projection matrix by replacing the 0's of the frozen variables.

Usage:

```
frozen_tour(d = 2, frozen)
```

Arguments

d	target dimensionality
frozen	matrix of frozen variables, as described in freeze

2.4.2 Optimization

The purpose of the Optimization is to compress an interesting projection.

- `search_better, search_better_random`: inspired by simulated annealing, these methods have been modified for better behavior in the interactive case.
- `search_geodesic`: a new method for stochastic coordinate-wise search.

Properties for a good optimization algorithm should have:

1. Monotonicity: Index values (measure of interestingness) for projections in the interpolation between starting and target bases increase monotonically.
2. Variable step-size: when we approach the optimal point, the algorithm should analyze those neighborhood points accurately with the small step-size

3. Local stopping criterion: A mechanism exists that allows to jump out of a local maximum to explore global maximum rather than local maximum

2.4.3 History:

This function is useful for saving and replaying tour paths that already have been explored.

- `save_history` for options to save history

The following example [4] illustrates how `save_history` saves the tour path and can be used to explore the path in the graphs.

```
fl_holes <- save_history(iris[, 1:4], guided_tour(holes), sphere = TRUE)
path_index(fl_holes, holes)
path_index(fl_holes, cmass)
plot(path_index(fl_holes, holes), type = "l")
plot(path_index(fl_holes, cmass), type = "l")
# Use interpolate to show all intermediate bases as well
hi <- path_index(interpolate(fl_holes), holes)
hi
plot(hi)
```

The `save_history()` calculates the new basis based on the path and the path indices are saved in the `fl_holes. path_index` computes index values for a tour history. Path indices for the holes and central mass are determined and plotted. The interpolated `fl_holes` data also plotted.

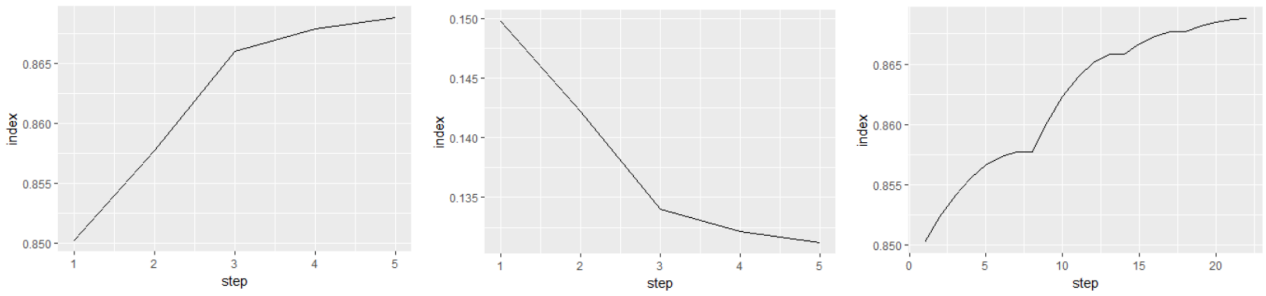

Fig 9: Visualizing the history of a tour path. The left plot shows the index plot of holes. The middle plot shows the index plot for central mass. The right plot shows the interpolated plot for the holes.

3. Summary:

This package is very useful to practice and develop tour methods for high-dimensional multivariate data analysis. This package is more than a magic button that assembles different

methods derived from other packages such as XGobi [2]. This paper uses the iris [3] data for the exploration of this package. The different tour methods were implemented to visualize the multivariate iris [3] data in different angles. Helpful for looking clusters, outliers, non-linear dependencies. This package also comes with different options which makes it possible to make some coefficients of variables as constant in the projection, optimize the search methods, save and replay tour paths that already have been explored and so on. The different extension options of the package would help to customize the display types, interpolation algorithms, projection pursuit indices and optimization methods, and tours for analyzing large *p*, small *n* data [1]. Overall, this package is a boon to those multivariate data researchers which provides exciting new possibilities for tour research.

4. References:

[1] Wickham, H., et al. "tourr: An R Package for Exploring Multivariate Data with Projections."*Journal of Statistical Software*, vol. 40, no. 2, 2011, pp. 1-18.

[2] Swayne, Deborah F., Dianne Cook, and Andreas Buja. "XGobi: Interactive Dynamic Data Visualization in the X Window System." *Journal of Computational and Graphical Statistics*, vol. 7, no. 1, 1998, pp. 113-130.

[3] Fisher, R. A. (1936). “The use of multiple measurements in taxonomic problems.” *Annals of Eugenics*, **7**, Part II, 179–188.

[4] Cook, Dianne, and Hadley Wickham. “Implement Tour Methods in R Code.” Accessed August 2, 2017. https://github.com/ggobi/tourr.

[5] Johnson RA, Wichern DW (2002). “Applied Multivariate Statistical Analysis.” 5th edition. Prentice-Hall, Englewood Cliffs, NJ.