# Dynamic visualization of high-dimensional data via low-dimension projections and sectioning across 2D and 3D display devices

Candidature confirmation report

Nicholas S Spyrison

Supervisors:

Prof. Kimbal Marriott,

Prof. Dianne Cook,

Prof. German Valencia

Faculty of Information Technology

Monash University

Australia

March 2019

# Contents

# Chapter 1

# Using animation to explore the sensitivity of structure in a low-dimensional projection of high-dimensional data with user controlled steering

*The content contained in this appendix document is work done in the last year of my research and currently formatted as a paper to be submitted to the R Journal.*

## 1.1 Abstract

The class of dynamic linear projections known collectively as tours provide a unique exploration and feature identification of high-dimensional data. Tours are particularly useful for understanding the structure held within multivariate data, and in association with techniques for dimension reduction, supervised, and unsupervised classification. The *R* package *tourr* offers a variety of path generators and geometric displays for conducting tours on. This paper discusses an extension package, *spinifex*, that adds support for the

path generation of manual tours and graphic display of tours. Manual tours are used to explore the sensitivity of structure as the contributions of a manipulation variable are changed. *Spinifex* utilizes the animation packages *plotly* and *gganimate* to display tours under animation frameworks.

Keywords: manual tour, guided tour, grand tour, projection pursuit, high dimensional data, multivariate data, data visualization, statistical graphics, data science.

## 1.2 Introduction

A tour is a multivariate data analysis technique in which a sequence of linear (orthogonal) projections are viewed as an animation while the orientation of the projection basis is rotated across time. Each frame of the sequence corresponds to a small change in the projection for a smooth transition that perseveres continuity.

There are many ways that a tour path can be generated, we will focus on the manual tour. The manual tour was described in Cook and Buja (1997) and allows a user to control the contribution of a select variable has in the 2D projection. We call such user-defined manipulation as user-controlled steering (UCS). The primary purpose of UCS is to determine the sensitivity of structure visible to the contributions of a selected manipulation variable. This makes UCS particularly useful once a feature of interest has been identified, for example, by a guided tour (Cook et al., 1995). The path of a guided tour is selected via projection pursuit, the optimization of an index function on the projection via a hill climbing algorithm. The algorithm for a manual tour allows rotations in horizontal, vertical, oblique, angular and radial directions as defined by the values of two angles. Rotation in a radial direction pulls a variable directly into and out of the projection. This type of manual rotation illustrated in this paper.

A manual tour relies on user input and thus has been difficult to implement dynamic control in R. Ideally, the mouse movements of the user are captured, and passed to the computations, driving the rotation interactively. However, this type of interactivity is not simple in R. This has been the reason that the algorithm was not incorporated into the *tourr* package. Spinifex utilizes two new animation packages, *plotly* (Sievert, 2018) and *gganimate* (Pedersen and Robinson, 2019), to display tours, manual or other saved tours. From a given projection, the user can choose which variable to control, and the animation sequence is generated to remove the variable from the projection, and then extend its contribution to be the sole variable in one direction. This allows the viewer to assess the change in structure induced in the projection by the variable's contribution.

The paper is organized as follows. Section 1.3 explains the algorithm using a toy dataset. Section 1.4 discussed the display of the animation after the path has been generated. Section 1.5 illustrates how this can be used for sensitivity analysis applied to contemporary high energy physics. The last section, 1.7 summarizes the work and discusses future research.

## 1.3 Algorithm

Generating the path for a manual tour requires the iteration of these steps:

1. Provided with a 2D projection, choose a variable to explore. This is called the "manip" variable.
2. Create a 3D manipulation space, where the manip variable has the full contribution.
3. Generate a rotation sequence which zero's the norm of the coefficient and increases it to 1.

The steps are described in more detail below.

### 1.3.1 Notation

This section describes the notation used in the algorithm description. The data is an $n \times p$ numeric matrix to be embedded into a $d-$dimensions.

$$\mathbf{X}_{[n,\ p]} = \begin{bmatrix} X_{1,\ 1} & \dots & X_{1,\ p} \\ X_{2,\ 1} & \dots & X_{2,\ p} \\ \vdots & \ddots & \vdots \\ X_{n,\ 1} & \dots & X_{n,\ p} \end{bmatrix}$$

An orthonormal $d-$dimensional basis set is describing the projection from $p-$ to $d-$ space.

$$\mathbf{B}_{[p,\ d]} = \begin{bmatrix} B_{1,\ 1} & \dots & B_{1,\ d} \\ B_{2,\ 1} & \dots & B_{2,\ d} \\ \vdots & \ddots & \vdots \\ B_{p,\ 1} & \dots & B_{p,\ d} \end{bmatrix}$$

The algorithm primarily operates on the projection basis and utilizes the data only when making a display.

### 1.3.2 Toy data set

The flea data from Lubischew (1962), available in the R package *tourr* (Wickham et al., 2011), is used to illustrate the algorithm. The data contains 74 observations across 6 variables, physical measurements of the flea beetles. Each observation belonging to one of three species.

A guided tour on the flea data is conducted by optimizing on the `holes` index (Cook, Swayne, and Buja, 2007). In a guided tour the projection sequence is selected by optimizing an index of interest. In this case, the holes index is selected. The holes index is maximized by when the projected observations are furthest from the center. Figure 1.1, shows a locally optimized projection for this data. The left plot displays the reference axes of the projection basis, a visual indication of the magnitude and direction each variable contributed to the projections. The right plot shows the data projected through this basis. The variables `tars1` and `aede2` contribute mostly in the horizontal direction mostly orthogonal to the other four variables which explain more of the variation in the verticle direction of the projection. Data points are colored and given geoms according to the species of the flea, although the guided tour was unsupervised with this information. The question that will be explored in the explanation of the algorithm is how important is the variable `aede2` to the separation of the clusters.

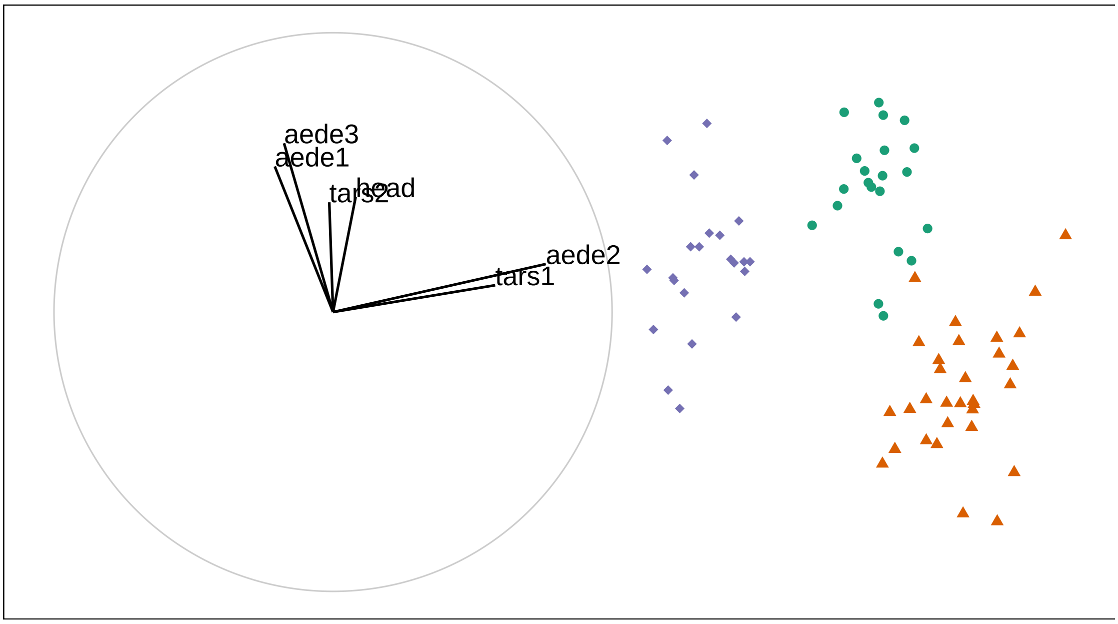


**Figure 1.1:** *Basis reference axes (left) and projected data (right) of standardized flea data. Basis identified by holes-index guided tour. The variables 'aede2' and 'tars1' contribute mostly in the horizontal direction, whereas the other variables contribute mostly in the y direction. We'll select 'aede2' as our manipulation variable to see how the structure of the projection changes as we rotate 'aede2' into and out of the projection.*

Call `view_basis()` on a basis to produce a similar image as a `ggplot2` object. The right side shows how the data looks projected through this basis. If data is passed to the function, the projection will also be displayed. You can project a single basis at any time through the matrix multiplication $\mathbf{X}_{[n,\ p]} * \mathbf{B}_{[p,\ d]} = \mathbf{P}_{[n,\ d]}$ to this effect.

### 1.3.3 Step 1 Choose variable of interest

Select a manipulation variable, $k$. Initialize a zero vector $e$ and set the $k$-th element set to 1. This gives the manip variable full contribution in this dimension.

$$\mathbf{e}_{[p,\ 1]} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

In figure 1.1, above, notice that the variables `tars1` and `aede2` are almost orthogonal to the other 4 variables and control almost all of the variation in the x-axis of the projection. `Aede2` has a larger contribution on this basis, so we'll select it as the manip variable.

### 1.3.4 Step 2 Create the manip space

Use the Gram-Schmidt process to orthonormalize the concatenation of the basis and $e$ yielding the manipulation space.

$$\begin{aligned} \mathbf{M}_{[p,\ d+1]} &= Orthonormalize_{GS}(\mathbf{B}_{[p,\ d]}|\mathbf{e}_{[p,\ 1]}) \\ &= Orthonormalize_{GS}\left( \begin{bmatrix} B_{1,\ 1} & \dots & B_{1,\ d} \\ B_{2,\ 1} & \dots & B_{2,\ d} \\ \vdots & \ddots & \vdots \\ B_{k,\ 1} & \dots & B_{k,\ d} \\ \vdots & \ddots & \vdots \\ B_{p,\ 1} & \dots & B_{p,\ d} \end{bmatrix} \middle| \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \right) \end{aligned}$$

Adding an extra dimension to our basis plane allows for the manipulation of the specified variable. For example, lifting a piece of paper, rather than manipulating on a table top. Orthonormalizing rescales the new vector while leaving the first $d$ variables identical to the basis. An illustration of such can be seen below in figure 1.2.

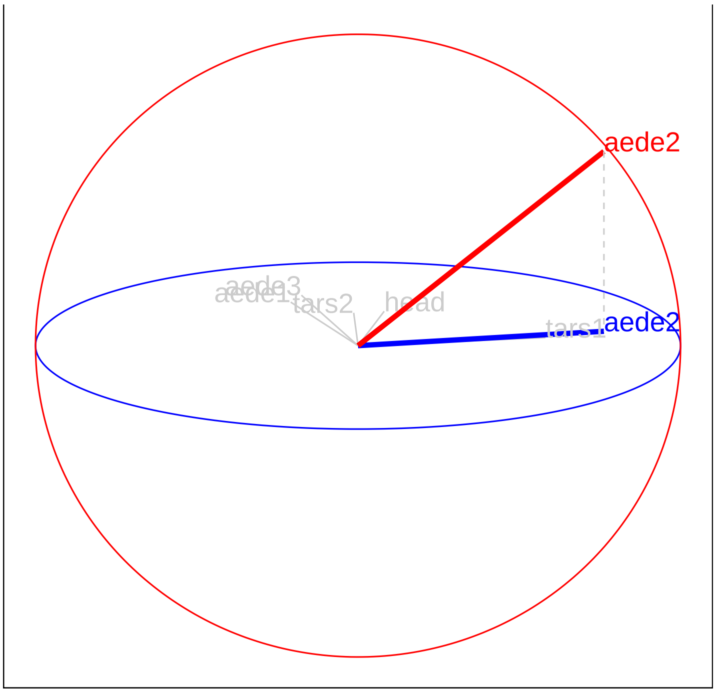


**Figure 1.2:** *Manipulation space for controlling the contribution of aede2 of standardized flea data. Basis was identified by holes-index guided tour. The out of plane axis, in red, shows how the manipulation variable can be rotated, while other dimensions stay embedded within the basis plane.*

Imagine being able to grab hold of the red axis and rotate it changing the projection onto the basis plane. This is what happens in a manual tour. For a radial tour, fix $\phi$, the angle within the blue plane, and vary the $\theta$, the angle between the red and blue lines. The user controlling these angles changes the values of the coefficient the manip variable and performs a constrained rotation on the remaining variables.

### 1.3.5 Step 3 Generate rotation

Define a set of values for $\phi_i$, the angle of out of the plane rotation, orthogonal to the projection plane. This corresponds to the angle between the red manipulation axis and the blue plane in figure 1.2.

**For** $i$ **in 1 to n_slides:**

Post-multiply the manipulation space by a rotation matrix, producing, **RM**, the rotated manip space.

$$\mathbf{RM}_{[p,\ d+1,\ i]} = \mathbf{M}_{[p,\ d+1]} * \mathbf{R}_{[d+1,\ d+1]}$$

For the $d = 2$ case:

$$= \begin{bmatrix} M_{1,\ 1} & \dots & M_{1,\ d} & M_{1,\ d+1} \\ M_{2,\ 1} & \dots & M_{2,\ d} & M_{2,\ d+1} \\ \vdots & \ddots & \vdots & \\ M_{p,\ 1} & \dots & M_{p,\ d} & M_{p,\ d+1} \end{bmatrix}_{[p,\ d+1]} * \begin{bmatrix} c_\theta^2 c_\phi s_\theta^2 & -c_\theta s_\theta (1 - c_\phi) & -c_\theta s_\phi \\ -c_\theta s_\theta (1 - c_\phi) & s_\theta^2 c_\phi + c_\theta^2 & -s_\theta s_\phi \\ c_\theta s_\phi & s_\theta s_\phi & c_\phi \end{bmatrix}_{[3,\ 3]}$$

Where:

$\theta$ is the angle that lies on the projection plane, the *XY*-scatterplot

$\phi$ is the angle orthogonal to the projection plane, in the *Z* direction relative to the *XY*-scatterplot

$c_\theta$ is the cosine of $\theta$

$c_\phi$ is the cosine of $\phi$

$s_\theta$ is the sine of $\theta$

$s_\phi$ is the sine of $\phi$

In application: compile the sequence of $\phi_i$ and create an array (or long table) for each rotated manipulation space. $\phi$ is the angle relative to the initial value of $\phi$, we find the transformation $\phi_i$ - $\phi_1$ useful to think about $\phi$ relative to the basis plane. If the manip variable doesn't move as expected this is the first place to check.

```
for (phi in seq(seq_start, seq_end, phi_inc_sign)) {
  slide <- slide + 1
  tour[,, slide] <- rotate_manip_space(manip_space, theta, phi)[, 1:2]
}
```

In figure 1.3 we illustrate the sequence with 15 projected bases and highlight the manip variable on top while showing the corresponding projected data points on the bottom.

A dynamic version of this tour can be viewed online at `https://nspyrison.netlify.com/thesis/flea_manualtour_mvar5/`, which may take a moment to load. The format of this figure and linking to a dynamic version will be used again in section 1.5.

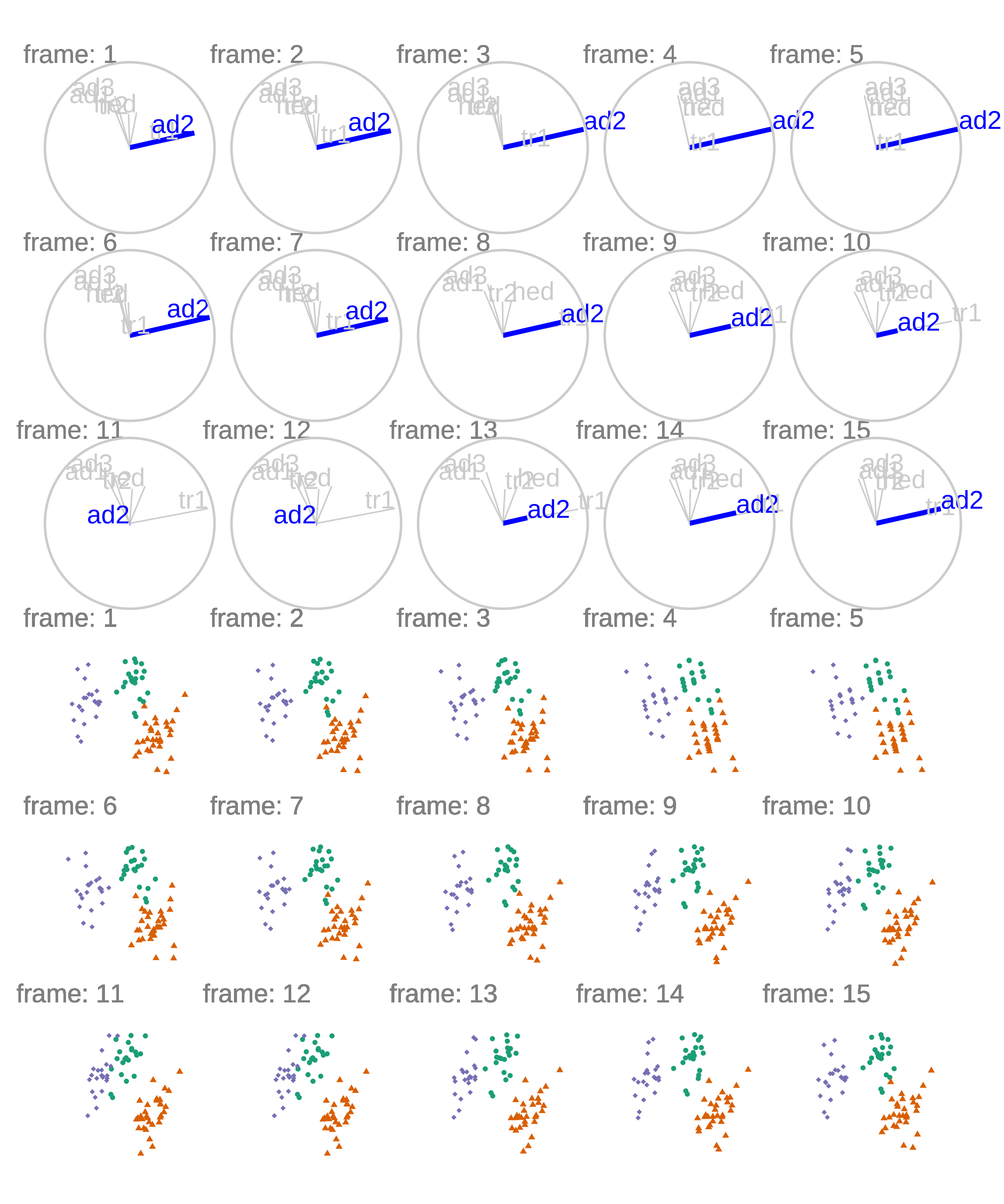


**Figure 1.3:** *Rotated manipulation spaces, a radial manual tour controlling the contribution from* `aded2` *of standardized flea data. The contribution of* `aede2` *extends from its initial contribution to a full contribution to the projection before decreasing to zero and then returning to its initial value. A dynamic version can be viewed at* https://nspyrison.netlify.com/thesis/flea_manualtour_mvar5/.

## 1.4 Display projection sequence

To get back to data-space pre-multiply each rotated manip space by the data for the projection in data-space.

$$\mathbf{P}_{[n,\ d+1]} = \mathbf{X}_{[n,\ p]} * \mathbf{RM}_{[p,\ d+1]} \tag{1.1}$$

$$= \begin{bmatrix} X_{1,\ 1} & \dots & X_{1,\ p} \\ X_{2,\ 1} & \dots & X_{2,\ p} \\ \vdots & \vdots & \vdots \\ X_{n,\ 1} & \dots & X_{n,\ p} \end{bmatrix}_{[n,\ p]} * \begin{bmatrix} RM_{1,\ 1} & RM_{1,\ 2} & RM_{1,\ 3} \\ RM_{2,\ 1} & RM_{2,\ 2} & RM_{2,\ 3} \\ \vdots & \vdots & \vdots \\ RM_{p,\ 1} & RM_{p,\ 2} & RM_{p,\ 3} \end{bmatrix}_{[p,\ d+1]} \tag{1.2}$$

Plot the first 2 variables from each projection in sequence for an $XY$ scatterplot. The remaining variable is sometimes linked to a data point aesthetic to produce depth cues used in conjunction with the $XY$ scatterplot.

*tourr* utilizes R's base graphics for the display of tours. Use `render_plotly()` to display as an dynamic `plotly` Sievert (2018) object or `render_gganimate()` for a `gganimate` Pedersen and Robinson (2019) graphic. Both of which build off of `ggplot2` plotting in internal functions.

Interaction with graphics in R is limited. Traditionally, all commands are passed to the R via calls to the console, conflicting with user engagement. Some recent packages have made advancement into this direction such as with the use of the R package `shiny` (Chang et al., 2018), which allows applications can be hosted either locally or remotely and communicate with the R console, allowing for developers to code dynamic content interaction. To a lesser extent, `plotly` offers static interactions with the contained object, such as tooltips, brushing, and linking without communicating back to the R console.

### 1.4.1 Storage and sharing

Storing each data point for every frame of the animation is very inefficient. In the same way that we gain efficiency by performing math on the bases, that is the same approach suggested for storage and sharing tours. Consider a radial manual tour, we can store the salient features in 3 bases, where $\phi$ is at its starting, minimum, and maximum values. The frames in between can be interpolated by supplying angular speed. By using the function `tourr::save_history()` we can do just that. Save such tour path history and a single set of data offers performant storage and transfer.

## 1.5 Application

In a recent paper, Wang et al. (2018), the authors aggregate and visualize the sensitivity of hadronic experiments, in the field of high energy physics. The authors introduce a new tool, PDFSense, to aid in the visualization of parton distribution functions (PDF). The parameter-space of these experiments lies in 56 dimensions, $\delta \in \mathbb{R}^{56}$, and are presented in as 2D subspaces of the 6 and 10 first principal components in linear and non-linear embeddings.

The work in Cook, Laa, and Valencia (2018) applies manual tours to discern the finer structure of this sensitivity. Table 1 of Cook et. al. summaries the key findings of PDFSense & TFEP (TensorFlow embedded projections) and those from manual tours. The authors selected the 6 first principal components, containing 48% of the variation held within the full data when centered, but not sphered. This data contained 3 clusters: jet, DIS, and VBP. The initial basis sets used below are obtained from projections used in figures 7 and 8 of the previous study (jet and DIS clusters respectively) of the previous study and apply manual tours to explore the local structure with finer precision.

### 1.5.1 Jet cluster

The jet cluster is of particular interest as it contains the largest data sets and is found to be important in Wang et al. (2018). The jet cluster resides in a smaller dimensionality than the full set of experiments with 4 principal components explaining 95% of its variation (Cook, Laa, and Valencia, 2018). We subset the data down to ATLAS7old and ATLAS7new to narrow in on 2 groups with a reasonable number of observations and occupy different parts of the subspace. Below, we perform radial manual tours on various principal components within this scope. In PC3 and PC4 are manipulated in figure 1.4 and figure 1.5 respectively.

Manipulating PC3, where varying the angle of rotation brings interesting features into and out of the center mass of the data, is more interesting than the manipulation of PC4, where the features are mostly independent of the contribution of PC4.

Jet cluster manual tours manipulating each of the principal components can be viewed from the links: PC1, PC2, PC3, and PC4.

### 1.5.2 DIS cluster

We perform a manual tour on this data, manipulating PC6 as depicted in figure 1.6. Looking at several frames we see that DIS HERA data lies mostly on a plane. When PC6 has full contributions, we see the dimuon SIDIS in purple is almost orthogonal to the DIS HERA (green). Yet the contribution of PC6 has zeroed the dimuon SIDIS data occupy the same space as the DIS HERA data. A dynamic version of this manual tour can be found at: `https://nspyrison.netlify.com/thesis/discluster_manualtour_pc6/`. The page may some time to load, as the animation is several megabytes.

The selection of the correct manip variable is important as the manipulation spaces convey different information. For example, in figure 1.7 we select PC2 as the manip variable finding it to be less insightful than PC6.

DIS cluster manual tours manipulating each of the principal components can be viewed from the links: PC1, PC2, PC3, PC4, PC5, and PC6.

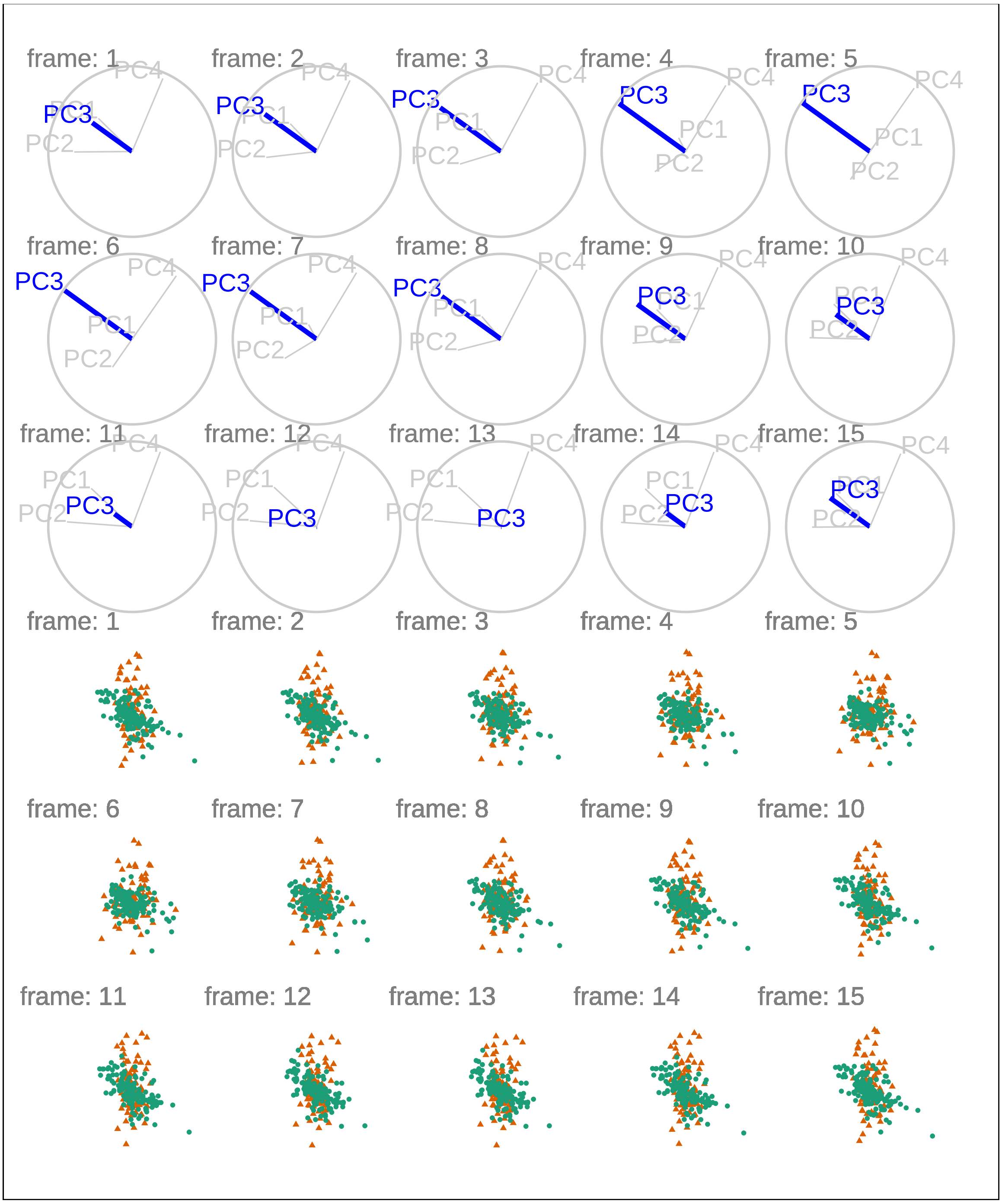


**Figure 1.4:** *Jet cluster, a radial manual tour of PC3. Colored by experiment type: 'ATLAS7new' in green and 'ATLAS7old' in orange. When PC3 fully contributes to the projection ATLAS7new (green) occupies unique space and several outliers are identifiable. Zeroing the contribution from PC3 to the projection hides the outliers and indeed all observations with ATLAS7new are contained within ATLAS7old (orange). A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc3/`.

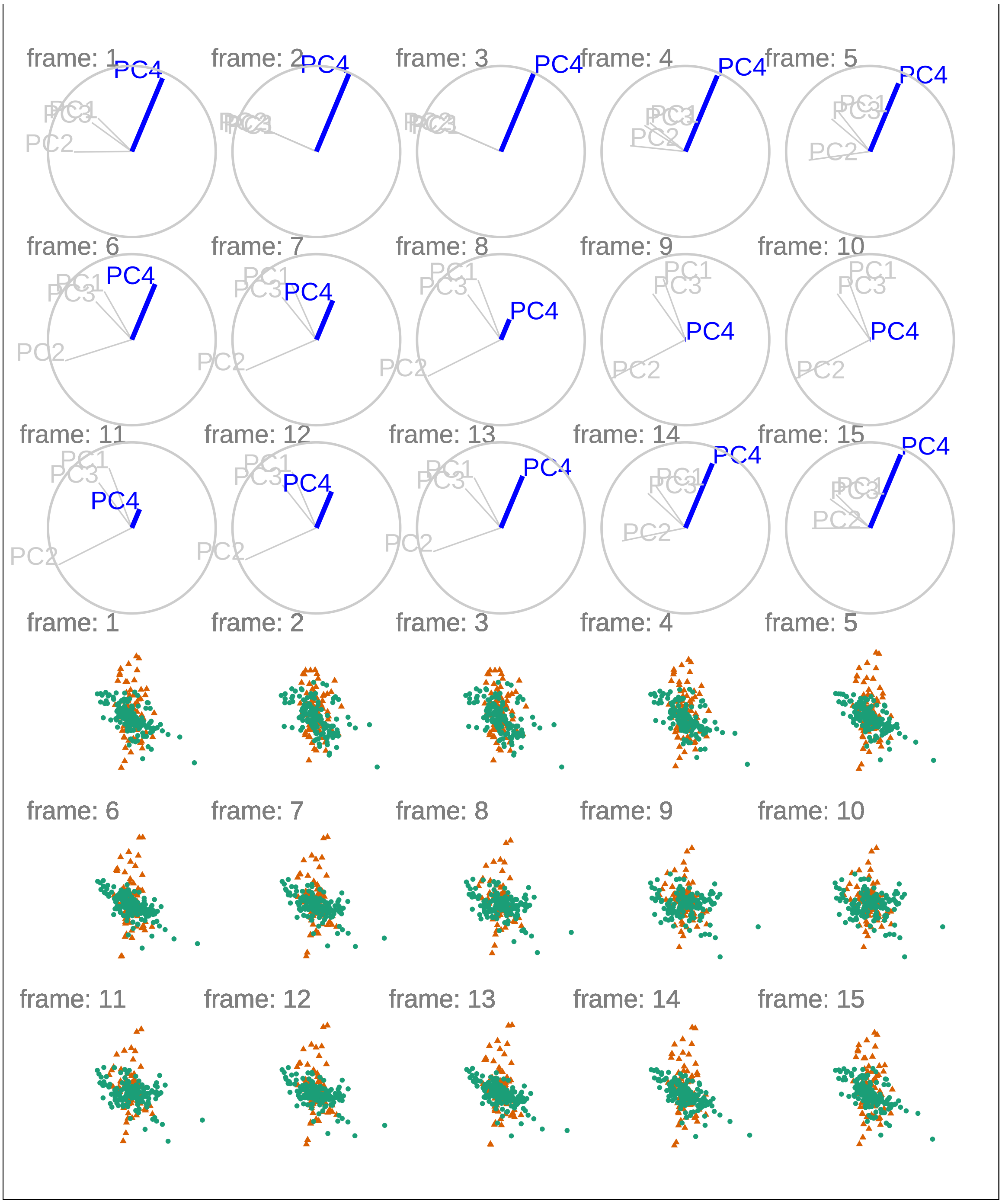


**Figure 1.5:** *Jet cluster, a radial manual tour of PC4. Colored by experiment type: 'ATLAS7new' in green and 'ATLAS7old' in orange. This tour contains less interesting information ATLAS7new (green) has points that are right and left of ATLAS7old, while most points occupy the same projection space, regardless of the contribution of PC4. A dynamic version can be viewed at* [`https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc3/`](https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc3/).

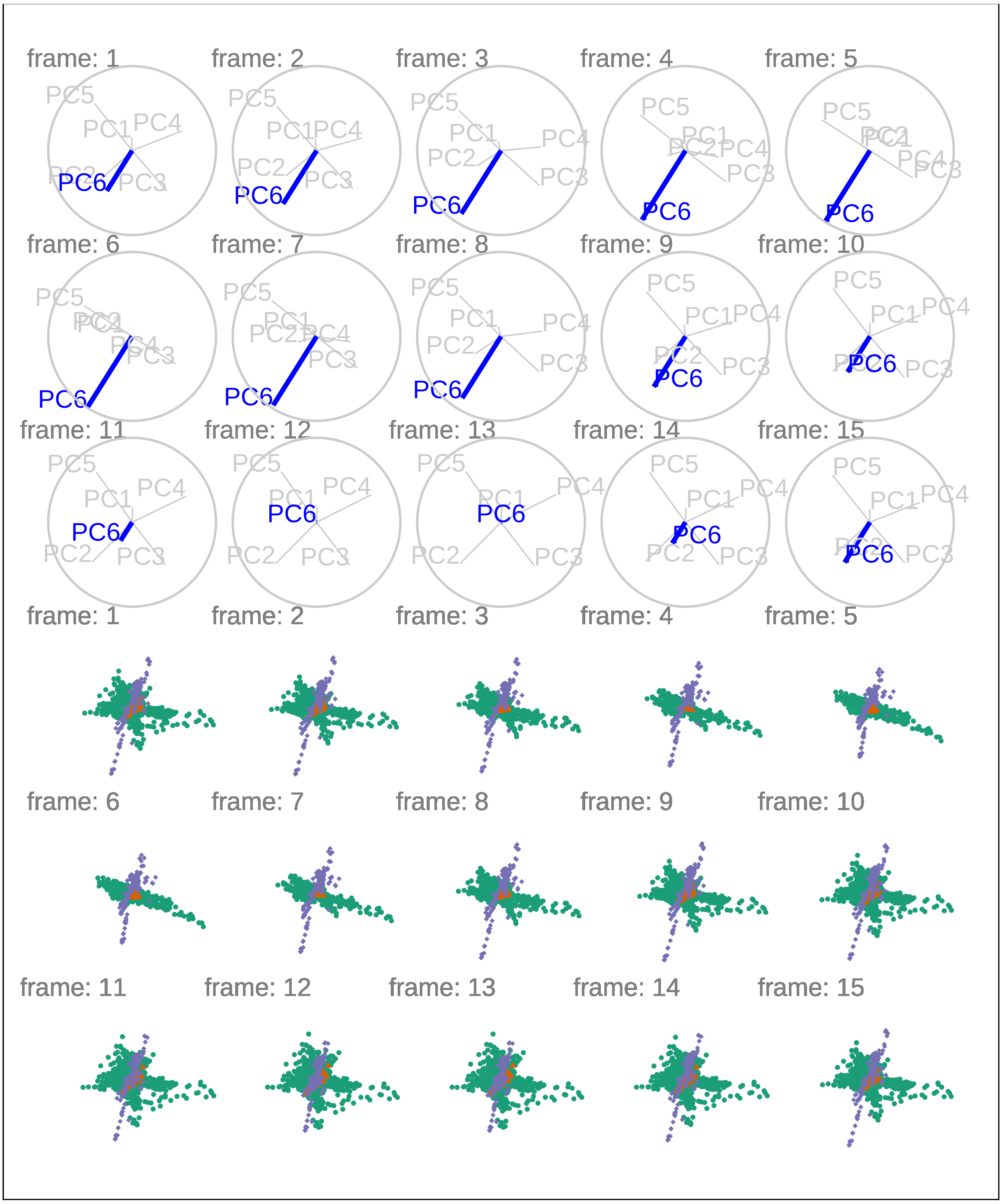


**Figure 1.6:** *DIS cluster, a radial manual tour of PC6. colored by experiment type: 'DIS HERA1+2' in green, 'dimuon SIDIS' in purple, and 'charm SIDIS' in orange. When the contribution PC 6 is large we see that dimuon SIDIS (purple) data are nearly orthogonal to DIS HERA (green) data. As the projection is rotated, we can also see that DIS HERA (green) practically lies on a plane in this 6-d subspace. When the contribution of PC6 is near zero, dimonSIDIS (purple) occupies the same space as the DIS HERA data. A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/discluster_manualtour_pc6/`.

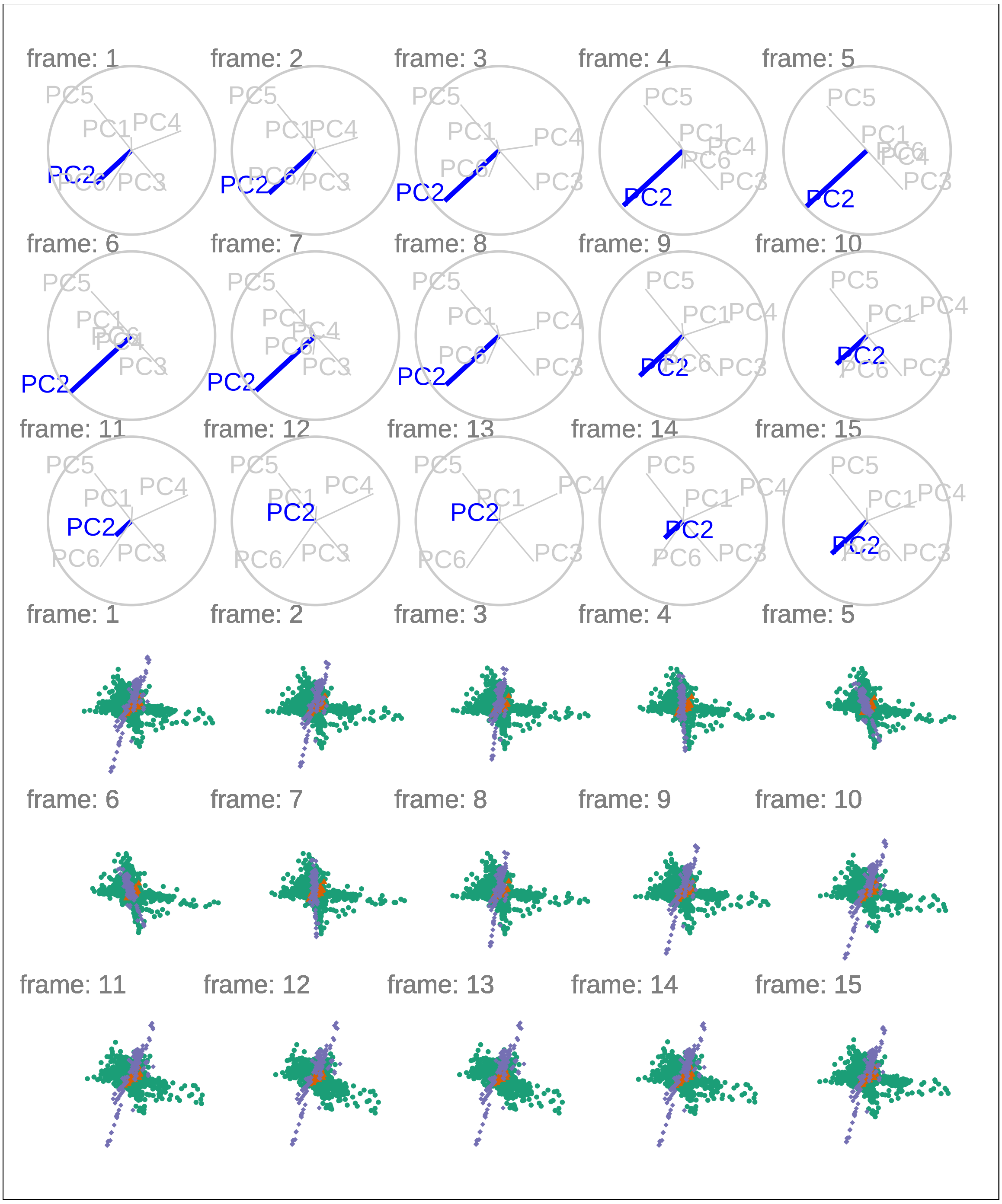


**Figure 1.7:** *DIS cluster, a radial manual tour of PC2. Colored by experiment type: 'DIS HERA1+2' in green, 'dimuon SIDIS' in purple, and 'charm SIDIS' in orange. The structure of previously described plane of DIS HERA (green) and nearly orthogonal dimuon SIDIS (purple) is present, however, the manipulating PC2 does not give a head-on view of either, a less useful manual tour than that of PC6. A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/discluster_manualtour_pc2/`.

## 1.6 Source code and usage

This article was created in R (R Core Team, 2018), using bookdown (Xie, 2016) and rmarkdown (Xie, Allaire, and Grolemund, 2018), with code generating the examples inline. The source files can be found at github.com/nspyrison/confirmation/.

The source code for the spinifex package can be found at github.com/nspyrison/spinifex/. To install the package in R, run:

```
# install.package("devtools")
devtools::install_github("nspyrison/spinifex")
```

## 1.7 Discussion

This research has modified the algorithm producing manual tours in extends animations of tours to other graphics packages. Tour paths generated in tourr can also be viewed using these frameworks.

Future research on the algorithm would include extending it for use in 3D projections. This would allow for projections in immersive virtual reality, which may allow for a better perception of structure and enable function visualization. The tourr package provides many other geometric displays with the tourr::display_*() family. These other geoms should be integrated into the ggplot2 framework for display on plotly and gganimate.

The Givens rotations and Householder reflections as outlined in Buja et al. (2005) could also be added. Currently, Gram-Schmidt is the only form of frame interpolation used. In a Givens rotation, the $x$ and $y$ components (for example $\theta = 0,\ pi/2$) of the in-plane rotation are calculated separately and would be applied sequentially to produce the radial rotation. Householder reflections define reflection axes to project points on to the axes and generate rotations.

The development of a dynamic graphical user interface, perhaps with the use of a shiny app, would allow analysts to rapidly try manual tours with a more intuitive interaction than the command line. The user could easily switch between variables to control, adjust the step size to make smoother rotation sequences, or save any state to continue to explore the contributions of other variables. The `animation` package Xie et al. (2018) could be implemented for another graphics framework. However, `animation` builds from base graphs while `spinifex` current utilizes `ggplot2` graphics.

# Bibliography


Buja, A, D Cook, D Asimov, and C Hurley (2005). "Computational Methods for High-Dimensional Rotations in Data Visualization". en. In: *Handbook of Statistics*. Vol. 24. Elsevier, pp.391–413. http://linkinghub.elsevier.com/retrieve/pii/S0169716104240147 (visited on 04/15/2018).

Chang, W, J Cheng, JJ Allaire, Y Xie, and J McPherson (2018). *shiny: Web Application Framework for R*. https://CRAN.R-project.org/package=shiny.

Cook, D and A Buja (1997). Manual Controls for High-Dimensional Data Projections. *Journal of Computational and Graphical Statistics* **6**(4), 464–480. (Visited on 04/15/2018).

Cook, D, A Buja, J Cabrera, and C Hurley (1995). Grand Tour and Projection Pursuit. en. *Journal of Computational and Graphical Statistics* **4**(3), 155. (Visited on 05/27/2018).

Cook, D, U Laa, and G Valencia (2018). Dynamical projections for the visualization of PDFSense data. *Eur. Phys. J. C* **78**(9), 742.

Cook, D, DF Swayne, and A Buja (2007). *Interactive and Dynamic Graphics for Data Analysis: With R and GGobi*. en. Google-Books-ID: 34DL7lR_4CoC. Springer Science & Business Media.

Lubischew, AA (1962). On the use of discriminant functions in taxonomy. *Biometrics*, 455–477.

Pedersen, TL and D Robinson (2019). *gganimate: A Grammar of Animated Graphics*. http://github.com/thomasp85/gganimate.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Sievert, C (2018). *plotly for R*. https://plotly-book.cpsievert.me.

Wang, BT, TJ Hobbs, S Doyle, J Gao, TJ Hou, PM Nadolsky, and FI Olness (2018). Visualizing the sensitivity of hadronic experiments to nucleon structure. *arXiv preprint arXiv:1803.02777*.

Wickham, H, D Cook, H Hofmann, and A Buja (2011). **tourr** : An *R* Package for Exploring Multivariate Data with Projections. en. *Journal of Statistical Software* **40**(2). (Visited on 11/23/2018).

Xie, Y (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. Boca Raton, Florida: Chapman and Hall/CRC. `https://github.com/rstudio/bookdown`.

Xie, Y, JJ Allaire, and G Grolemund (2018). *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman and Hall/CRC. `https://bookdown.org/yihui/rmarkdown`.

Xie, Y, C Mueller, L Yu, and W Zhu (2018). *animation: A Gallery of Animations in Statistics and Utilities to Create Animations*. `https://yihui.name/animation`.