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

Canidature confirmation document for the degree of

Doctor of Philosophy

by

Nicholas S Spyrison

B.Sc. Statistics, Iowa State University

Department of Information Technology

Monash University

Australia

February 2019

Contents

Acknowledgements **v**

Declaration **vii**

Preface **ix**

Abstract **xi**

1 Introduction **1**

2 Literature review **3**

2.1 Touring . . . 3

2.2 Virtual reality . . . 9

3 *spinifex*: An R package that provides manual rotations in high-dimensions **11**

3.1 Abstract . . . 11

3.2 Introduction . . . 12

3.3 Algorithm . . . 13

3.4 Display projection sequence . . . 20

3.5 Application . . . 21

3.6 Source code and usage . . . 27

3.7 Discussion . . . 27

4 Touring across display dimensionality **29**

5 Human-computer interaction with 3d projections **31**

Bibliography **33**

Acknowledgements

Firtsly, I would like to express my sincere gratitude to my advisors professor Dianne Cook and professor Kimbal Marriott for their support of my Ph.D studies and research, for their patience, motivation, and immense knowledge. Their guidance was an indispensible help through my research and writing.

I thank my fellow Ph.D students in for the stimulating discussions, the countless hours we spent working all hours of the week, and their unwaivering support. I could not have asked for better company.

Last but not the least, I would like to thank my family: my parents and to my brother and sister for supporting me in this thesis and in my life in general.

Declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any university or equivalent institution, and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Nicholas S Spyrison

Preface

The contribution in Chapter 3 will soon be submitted to the R journal and the accomanying R package *spinifex* will be submitted to CRAN.

Abstract

This thesis is about …

Chapter 1

Introduction

This is where you introduce the main ideas of your thesis, and an overview of the context and background.

The chapter 2 covers a literature review, currently in particular covering tour methodology, while work remains to be done for stereo scopically true 3d space. 3 encompasses implementing manual tours in R. Think of each of these as potential papers to be submitted to journals. 3 Finally, Chapter 6 provides some concluding remarks, discussion, ideas for future research, and so on. Appendixes can contain additional material that don't fit into any chapters, but that you want to put on record. For example, additional tables, output, etc.

Chapter 2

Literature review

2.1 Touring

2.1.1 Overview

In univariate data sets histograms, or smoothed density curves are employed to visualize data. In bivariate data scatterplots and contour plots (2-d density) can be employed. In three dimensions the two most common techniques are: 2-d scatter plot with the 3rd variable as an aesthetic (such as, color, size, height, *etc.*) or rendering the data in a 3-d volume using some perceptive cues giving information describing the seeming depth of the image [1]. When there are 4 variables: 3 variables as spatial-dimensions and a 4th as aesthetic, or a scatterplot matrix consisting of 4 histograms, and 6 unique combinations of bivariate scatterplots.

Let p be the number of numeric variables; how do we visualize data for even modest values of p (say 6 or 12)? It's far too common that visualizing in data-space is dropped altogether in favor of modeling parameter-space, model-space, or worse: long tables of statistics without visuals (Wickham, Cook, and Hofmann, 2015). Yet, we all know of the risks inherent in relying too heavily on parameters alone (Anscombe, 1973; Matejka and Fitzmaurice, 2017). So why do we move away from visualizing in data-space? Scalability,

[1] Graphs of data depicting 3 dimension are typically printed on paper, or rendered on a 2-d monitor, they are intrinsically 2-d images. They are sometimes referred to as 2.5-d, or more frequently erroneously referred to as 3-d, more on this later.

in a word, we are not familiar with methods that allow us to concisely depict and digest $p \geq 5$ or so dimensions. This is where dimensionality reduction comes in. Specifically, we will be focusing on a specific group called touring. In the interest of time I will not belabor the diversity of dimensionality reduction, (see [Grinstein, Trutschl, and Cvek (2002); Carreira-Perpinán (1997); heer_tour_2010] for a quick summary). Suffice it to say that touring has a couple of salient features: linear transformations such that we can interpolate back to the original variable space and does not discard dimensions, something that is common to other linear techniques. By employing the breadth of tours we are able to preserve the visualization of data-space, and with it, the intrinsic understanding of structure and distribution of data that is more succinct or beyond the reach of statistic values alone.

Touring is a linear dimensionality reduction technique that orthogonally projects p-space down to $d(\leq p)$ dimensions. Many such projections are interpolated, each making local rotations in p-space. These frames are then viewed in order to the effect of watching an animation of the lower dimensional embedding changing as p-space is manipulated. Shadow puppets offer a useful analogy to aid in conceptualizing touring. Imagine a fixed light source facing a wall. When a hand or puppet is introduced the 3-dimensional object projects a 2-dimensional shadow onto the wall. This is a physical representation of a simple projection, that from $p = 3$ down to $d = 2$. If the object rotates then the shadow correspondingly changes. Observers watching only the shadow are functionally watching a 2-dimensional tour as the 3-dimensional object is manipulated.

Terminology

Terminology varies accross author and implementation. In my work, I use the following

- $\mathbf{X}_{[n,\ p]}$, a data set
  - n, numerber of observations
  - p, number of numeric variables, the dimensionality of data space, $\mathbf{X} \in \mathbb{R}^p$
- $B_{[p,\ d]}$, orthonormal basis vector, defines in the orientation of projection
  - p, dimensionality of data space

  - d, dimensionality of projection space

- Geometric objects are refered to in generalized dimensions. Planes are actually hyperplanes in projection space.

2.1.2 History

Touring was first introduced by Asimov in 1985 with his purposed Grand Tour(Asimov, 1985) at Stanford University. In which, Asimov suggested three types of Grand Tours: torus, at-random, and random-walk. The specifics of which will be discussed below in the Typology section.

The grand tours methods have no input from the user aside from the starting basis. The bulk of touring development since has largely been around dynamic display, user interaction, geometric representation, and application.

This works well when the number of dimensions being toured is small (in the neighborhood of 5-10), yet the number of view, or 2-frames and we can produce from p-space suffers from the so called blessing/curse of dimensionality. In which the plethora of degrees of freedom either offer many (non-unique) solutions to a problem or something that becomes ever increasing unlikely,

2.1.3 Tour path

A fundamental aspect of touring is the path of rotation. Of which there are four primary distinctions(Buja et al., 2005): random choice, precomputed choice, data driven, and manual control.

- *grand tour*, a constrained random choice p-space. Paths are constrained for changes in direction small enough to maintain continuity and aid in user comprehension

  - torus-surface (Asimov, 1985)
  - Geodesic
  - at-random
  - random-walk

  - *local tour*, a sort of grand tour on leash, such that it goes to a nearby random projection before returning to the original position and iterating

- *guided tour*, data driven tour optimizing some objective function via (stochastic) gradient descent (Hurley and Buja, 1990).

  - holes (Cook, Buja, and Cabrera, 1993) - iterates projections that add more white space to the center of the projection.
  - cmass (Cook, Buja, and Cabrera, 1993) - find the projection with the most density or mass in the center.
  - lda (Lee et al., 2005) - linear discriminant analysis, seeks a projection where 2 or more classes are most separated.
  - pda - principal component analysis finding where the data is most spread (1d only).
  - other user-defined objective function (Wickham et al., 2011).

- *planned tour*, Precomputed choice, In which the path has already been generated or defined.

  - *little tour* (McDonald, 1982), where every permutation of variables is stepped through in order, analogous to a brute-force or exhaustive search.
  - a saved path of any other tour

- *manual tour* - Manual control, a constrained rotation on selected manipulation variable and magnitude(Cook and Buja, 1997). Typically used to explore the local area after identifying an interesting feature from another tour.

- *dependance tour*, combination of n independent 1d tours. A vector describes the axis each variable will be displayed on. **ie** $c(1, 1, 2, 2)$ is a 4 to 2d tour with the first 2 variables on on the first axis, and the remaining on the second.

  - *correlation tour* (Buja, Hurley, and McDonald, 1987), a special case of the dependence tour, analogous to canonical correlation analysis

Evaluating

TODO:: add density and rapidity as a new section?

2.1.4 Geometrics and display dimension

Up to this point we have been talking about 2d scatterplots, which offer the first and a simple case for viewing lower-dimensional embeddings of p-space. However, other geometrics (or geoms) offer perfectly valid orthonormal projections as well.

- 1d geoms
  - 1-d densities: such as histogram, average shifted histograms(**scott85**), and kernel density(**scott95**).
  - image: (**Wegman**)
  - time series: where multivariate values are independently lagged to view peak and trough alignment. Currently no package implementation, but use case is discussed in (Cook and Buja, 1997).
- 2d geoms
  - 2-d density (**NS**)
  - scatterplot
  -
- 2.5d, 3d geoms {ADD FOOTNOTE ABOUT 2.5d vs 3d}
  - Anaglyphs, sometimes called stereo, where (typically) red images are positioned for the left channel and cyan for the right, when viewed with corresponding filter glasses give the depth perception of the image.
  - Depth, which use some subset of depth cues, most commonly size and/or color of data points.
- d-dim geoms
  - Andrews curves (Andrews, 1972), smoothed variant of parallel coordinate plots, discussed below.
  - Chernoff faces (Chernoff, 1973), variables linked to size of facial features for rapid cursory like-ness comparison of observations.

- Parallel coordinate plots (Ocagne, 1885), where any number of variables are plotted in parallel with observations linked to their corresponding variable value by polylines.
- Scatterplot matrix (Becker and Cleveland, 1987), showing a triangle matrix of bivariate scatterplots with 1-d density on the diagonal.
- Radial glyphs, radial variants of parallel coordinates including radar, spider, and star glyphs (Siegel et al., 1972).

2.1.5 Implementations

Below is a non-exhaustive list of software implementing touring in some degree, ordered by descending year:

- Spinifex (**spinifex**) – for Linux, Unix, and Windows.
- Tourr (Wickham et al., 2011) – for Linux, Unix, and Windows. R package.
- CyrstalVision (Wegman, 2003) – for Windows.
- GGobi (Swayne et al., 2003) – for Linux and Windows.
- DAVIS (Huh and Song, 2002) – Java based, with GUI.
- VRGobi (Nelson, Cook, and Cruz-Neira, 1998) – for use with the C2 in stereoscopic 3d displays.
- ExplorN (Carr, Wegman, and Luo, 1996) – for SGI Unix.
- XGobi (Swayne, Cook, and Buja, 1991) – for Linux, Unix, and Windows (via emulation).
- XLispStat (Tierney, 1990) – for Unix, and Windows.
- Prim-9 (Asimov, 1985; Fisherkeller, Friedman, and Tukey, 1974) – on an internal operating system.

Support and maintenance of such implementations give them a particularly short life span, while conceptual abstraction and technically heavier implementations have hampered user growth. There have been notable efforts to diminish the barriers to entry and make touring more approachable as a data exploration tool [Huh and Song (2002); Swayne et al. (2003); Wegman (2003); Wickham et al. (2011); huang_tourrgui:_2012].

2.2 Virtual reality

Chapter 3

spinifex: An R package that provides manual rotations in high-dimensions

3.1 Abstract

The tour algorithm, and its various versions provide a systematic approach to viewing low-dimensional projections of high-dimensional data. It is particularly useful for understanding multivariate data, and useful in association with techniques for dimension reduction, supervised and unsupervised classification. The *R* package *tourr* provides many methods for conducting tours on multivariate data. This paper discusses an extension package which adds support for the manual tour, called *spinifex*. It is particularly usefully for exploring the sensitivity of structure discovered in a projection by a guided tour, to the contribution of a variable. *Spinifex* utilizes the animation packages *plotly* and *gganimation* to allow users to rotate a variable into and our of a chosen projection.

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

3.2 Introduction

A tour is a multivariate data analysis technique in which is a sequence of orthogonal projection into a lower subspace are viewed in order. each frame of the sequence corresponds to a small change in the projection for a smooth transition.

Multivariate data analysis can be broken into 2 groups: linear and non-linear transformations. Similar to PCA and LDA, touring uses linear dimension reduction with inter-operability back to the original parameter-space. They differ from non-linear transformations such as t-SNE (t-distributed stochastic nearest neighbor embeddings), MDS (multi-dimension scaling), and LLE (local linear embedding), which distort parameter-space for more opaque interpretations

There are many ways that a tour path can be generated, we will focus on one in particular, the manual tour. The manual tour was described in Cook and Buja (1997), and allows a user to rotate a variable into and out of a 2D projection of high-dimensional space. The primary purpose is to determine the sensitivity of structure visible in a projection to the contributions of a variable. Manual touring can also be useful for exploring the local structure once a feature of interest has been identified, for example, by a guided tour (Cook et al., 1995). The algorithm for a manual tour allows rotations in horizontal, vertical, oblique, angular and radial directions. Rotation in a radial direction, would pull a variable into and out of the projection, which allows for examining the sensitivity of structure in the projection to the contribution of this variable. This type of manual rotation is the focus of this paper.

A manual tour relies on user input, and thus has been difficult to program 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 packages for conducting animations, *plotly* (Sievert, 2018) and *gganimate* (Pedersen and Robinson, 2019), to conduct a manual tour. 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 contribution.

The paper is organized as follows. Section 3.3 explains the algorithm using a toy dataset. Section 3.5 illustrates how this can be used for sensitivity analysis. The last section, **??** summarizes the work and discusses future research.

3.3 Algorithm

Creating a manual tour animation requires 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 full contribution.
3. Generate a rotation sequence which zero's the norm of the coefficient and also increases it to 1.

These steps are described in more detail below.

3.3.1 Notation

This section describes the notation used in the algorithm description. The data to be displayed is an $n \times p$ numeric matrix.

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

and an orthonormal d-dimensional projection matrix is

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

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

3.3.2 Toy data set

The flea data from the R package *tourr* (Wickham et al., 2011), is used to illustrat the algorithm. The data, originally from Lubischew (1962),contains 74 observations across 6 variables, which physical measurements of the insects. Each individual belonged 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 data the projection sequence is shown by optimizing an index of interest. The holes index is maximized by when the projected data has a lack of observations in the center. Figure 3.1, shows an optimal projection of this data. The left plot displays the projection basis, while the right plot shows the projected data. The display of the basis has a unit circle with lines showing the horizontal and vertical contributions of each variable in the projection. Here is is primarily tars1 and aede2 contrasting the other four variables. In the projected data it can be seen that there are three clusters, which have been colored, although not used in the optimization. The question that will be explored in the explanation of the algorithm is how important is aede2 to the separation of the clusters.

The left frame of 3.1 shows the reference frame for the basis. It describes the X and Y contributions of the basis as it projects from the 6 variable dimensions down to 2. Call `view_basis()` on a basis to produce a similar image as a `ggplot2` object. The right side

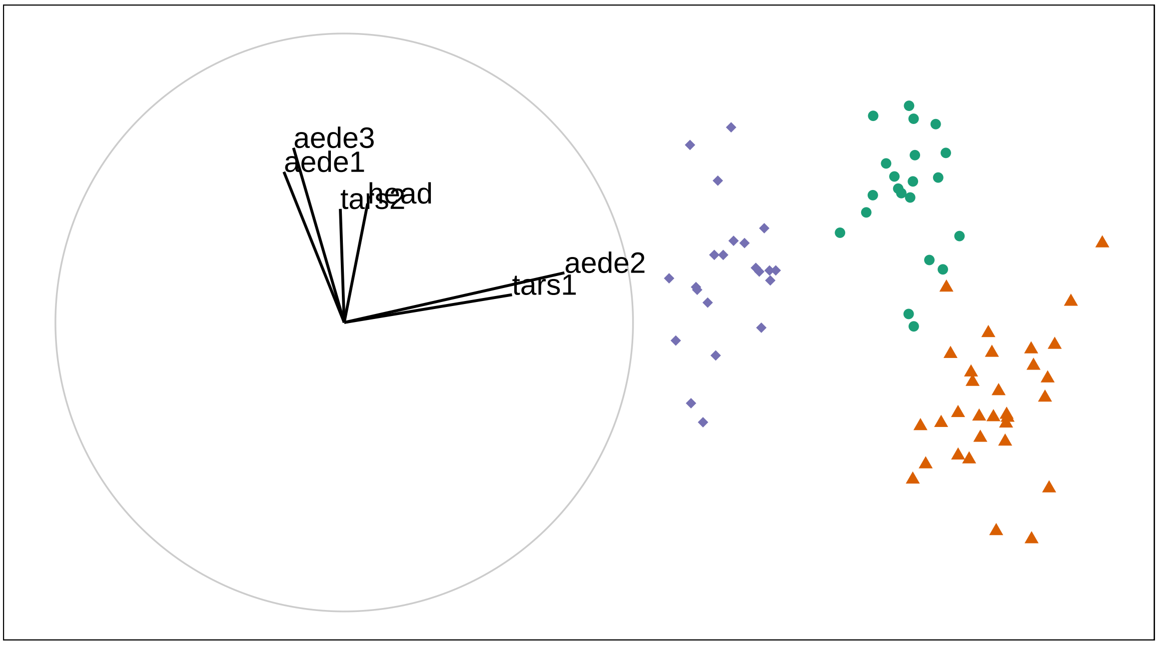

Figure 3.1: *Basis reference frame (left) and projected data (right) of standardized flea data. Basis identified by holes-index guided tour. The variables* `aede2` *and* `tars1` *contribute mostly in the x direction, whereas the other variables contribute mostly in the y direction. We'll select* `aede2` *as our manipulation variable to see how the structure of the projection changes as we rotate* `aede2` *into and out of the projection.*

shows how the data looks projected through this basis. You can project a single basis at any time through the matrix multiplication $\mathbf{X}_{[n,\ p]} * \mathbf{B}_{[p,\ d]} = \mathbf{P}_{d[n,\ d]}$ to such effect.

3.3.3 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.

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

In figure 3.1, 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 in this basis, so we'll select it

3.3.4 Step 2 Create the manip space

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

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

In R it looks like the below chunk. `tourr::orthonormalise()` uses the Gram Schimidt process (rather than Householder reflection) to orthonormalize.

```
e             <- rep(0, len = nrow(basis))
e[manip_var] <- 1
manip_space  <- tourr::orthonormalise(cbind(basis, e))
```

Adding an extra dimension to our basis plane allows for the manipulation of the specified variable while the others are kept fully within the basis plane. orthonormalizing rescales the matrix without bringing the other variables into this new axis. An illustration of such can been seen below in 3.2.

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. By controlling the angle between

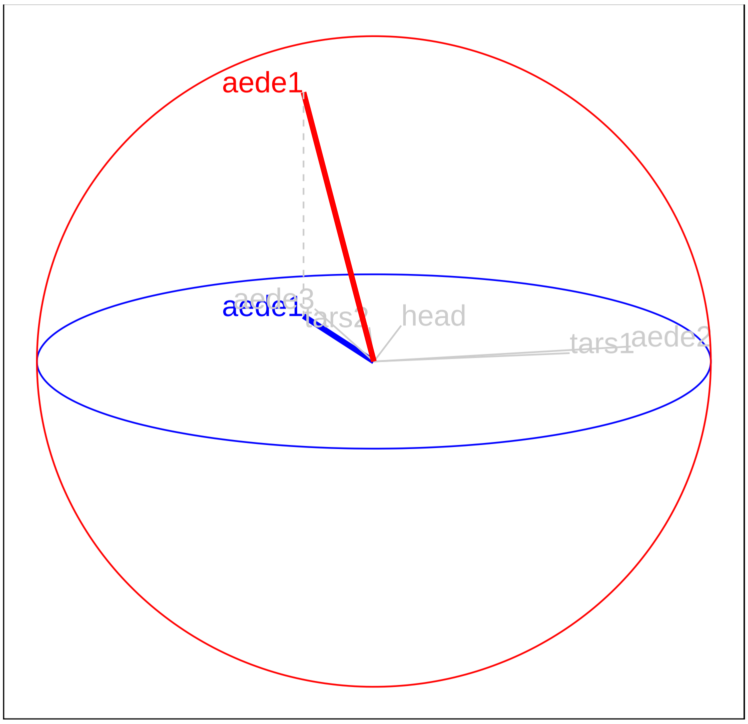

Figure 3.2: *Manipulation space for controlling the contribution of aede2 of standardized flea data. Basis was identified by holes-index guided tour. The out of plane axis, in red, shows how the manipulation variable can be rotated, while other dimensions stay embedded within the basis plane.*

the axis and the basis plane we change the contribution of the manipulation variable on the projection.

3.3.5 Step 3 Generate rotation

Define a set of values for ϕ_i, the angle of out-of plane rotation, orthogonal to the projection plane. This corresponds to the angle between the red manipulation axis and the blue plane in 3.2.

For i in 1 to n_slides:

For each ϕ_i, post multiply the manipulation space by a rotation matrix, producing as many basis-projections.

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

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:

θ is the angle that lies on the projection plane (*ie*. on the XY plane)

ϕ is the angle orthogonal to the projection plane (*ie*. in the Z direction)

c_θ is the cosine of θ

c_ϕ is the cosine of ϕ

s_θ is the sine of θ

s_ϕ is the sine of ϕ

In application: compile the sequence of ϕ_i and create an array (or long table) for each rotated manipulation space. ϕ is actually the angle relative to the ϕ_1, we find the transformation ϕ_i - ϕ_1 useful to discuss ϕ relative to the basis plane.

```
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 3.3 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_mvar4/, will take a moment to load. This format of this figure and linking to dynamic version will be used again the 3.5 section.

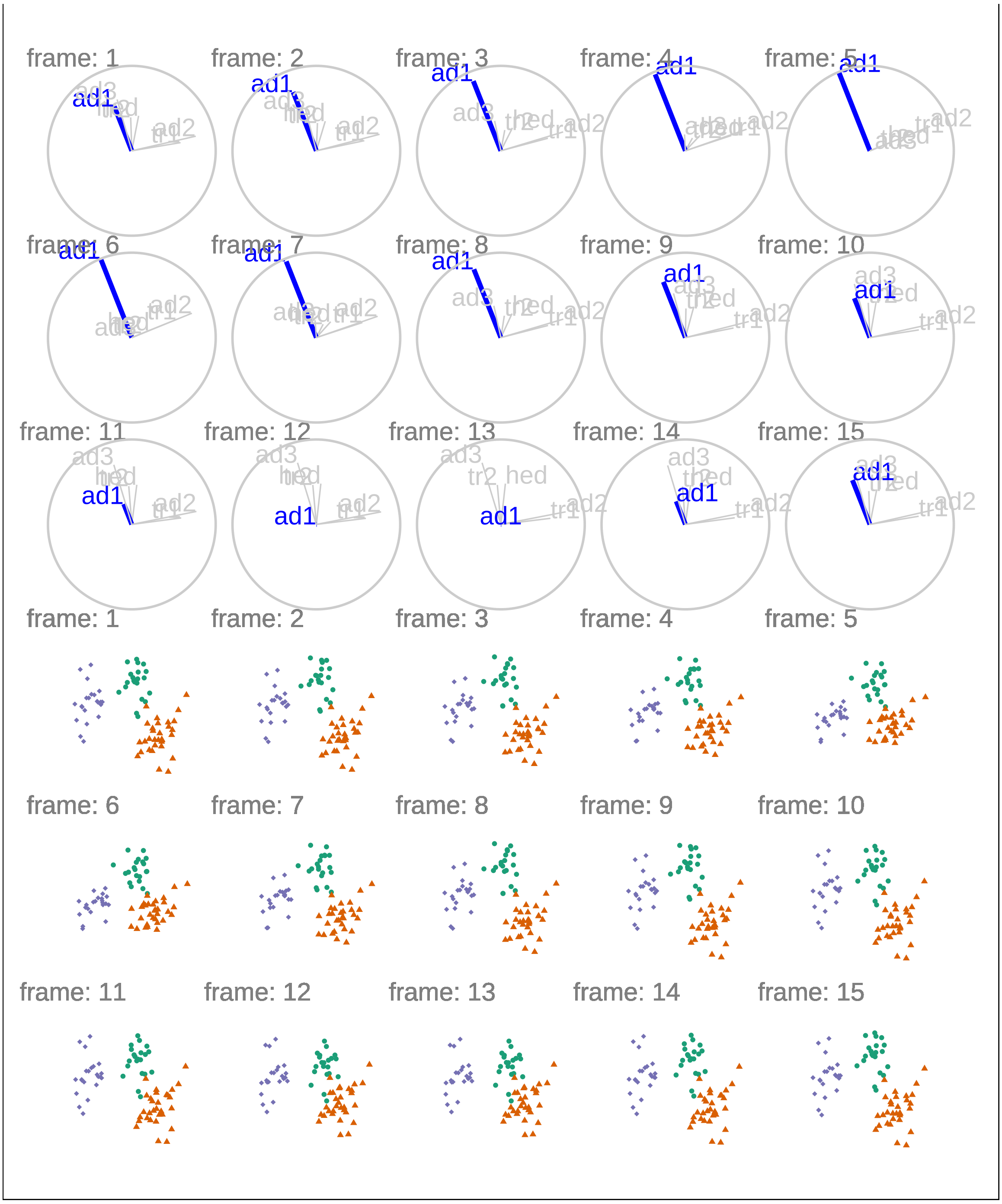

Figure 3.3: *Rotated manipulation spaces, a radial manual tour manipulating aded2 of standardized flea data. The manipulation variable, aede2, extends from it's initial contribution to a full contribution to the projection before decreasing to zero, and then returning to it's initial state. A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/flea_manualtour_mvar4/` .

3.4 Display projection sequence

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

$$\mathbf{P}_{d[n,\ d+1]} = \mathbf{X}_{[n,\ p]} * \mathbf{P}_{b[p,\ d+1]} \tag{3.1}$$

$$= \begin{bmatrix} X_{1,\ 1} & \dots & X_{1,\ p} \\ X_{2,\ 1} & \dots & X_{2,\ p} \\ \vdots & \vdots & \vdots \\ X_{n,\ 1} & \dots & X_{n,\ p} \end{bmatrix}_{[n,\ p]} * \begin{bmatrix} P_{b:1,\ 1} & P_{b:1,\ 2} & P_{b:1,\ 3} \\ P_{b:2,\ 1} & P_{b:2,\ 2} & P_{b:2,\ 3} \\ \vdots & \vdots & \vdots \\ P_{b:p,\ 1} & P_{b:p,\ 2} & P_{b:p,\ 3} \end{bmatrix}_{b[p,\ d+1]} \tag{3.2}$$

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. A third notable animation related package is `animation` Xie et al. (2018). It's not yet implemented in `spinifex` as it uses base graphics, whereas the former two are compatible with `ggplot2`.

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 `shinny`, which custom-made applications can be hosted either locally or remotely and interact with the R console, allowing for developers to code dynamic content interaction. To a lesser extent `plotly` offers static interactions with contained object, such as tool tips, brushing, and linking without communicating back to the R console.

Storing the each data point and all of the overhead though goes into dynamic graphics if very inefficient. In the same way that we performed math the bases, that is the same approach storage and sharing tours. Consider the manual tour, we can store the salient features in 3 basis, where ϕ is at it's starting, minimum, and maximum values. The frames in between can be interpolated by supplying angular speed or number of desired frames. By using the `tourr::save_history()` we can do just that. Save such tour path history and a single set of the data offers a performant storage and transferring.

3.5 Application

In a recent paper, Wang et al. (2018), the authors aggregate and visualize the sensitivity of hadronic experiments. 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 this work in 3-d subspaces of the 10 first principal components and non-linear embeddings.

The work in Cook, Laa, and Valencia (2018) applies touring for discern finer structure of this sensitivity. Table 1 of Cook et. al. summaries the key findings of PDFSense & TFEP (tensorflow embedded projection) and those from touring. 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. Below pick up from the projections used in their figures 7 and 8 (jet and DIS clusters respectively) and apply manual tours to explore the local structure with finer precision.

3.5.1 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 it's 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 the this scope. In PC3 and PC4 are manipulated in 3.4 and 3.5 respectively. Manipulating PC3, where varying the angle of rotation brings interesting features in-to and out-of the center mass of the data, is interesting than the manipulation of PC4, where features are mostly independent of the manip var.

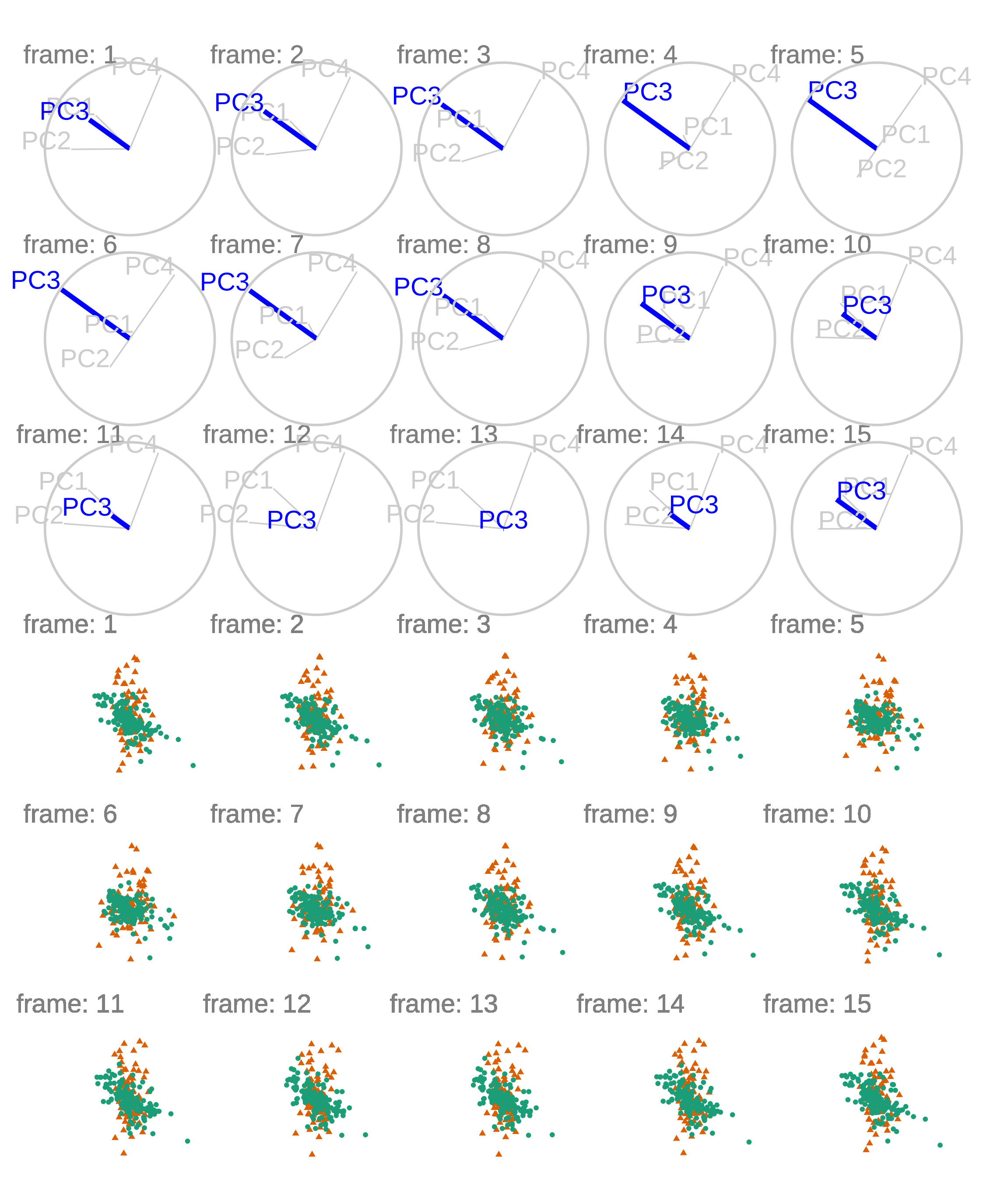

Figure 3.4: *Jet cluster, radial manual tour of PC3. Colored by experiment type: 'ATLAS7new' in green and 'ATLAS7old' in orange. When PC3 fully contributes to the projection ATLAS7new (green) occupies unique space and several outliers are identifiable. Zeroing the contribution from PC3 to the projection 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/`.

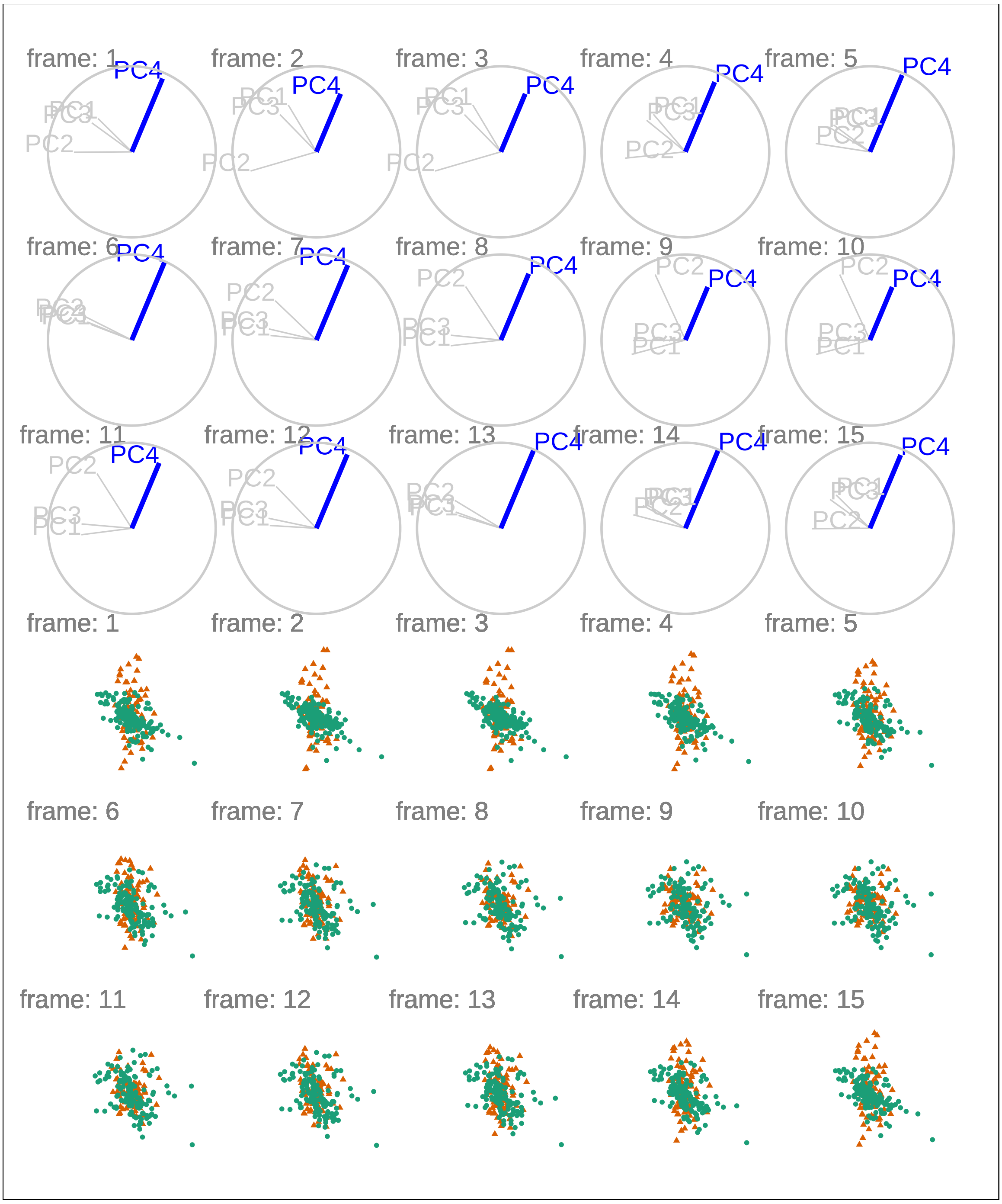

Figure 3.5: *Jet cluster, radial manual tour of PC4. Colored by experiment type: 'ATLAS7new' in green and 'ATLAS7old' in orange. This tour contain less interesting information ATLAS7new (green) has points that are right and left of ATLAS7old, while most points occupy the same projection space, regardless 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/).

Jet cluster manual tours manipulating each of the principal components can be viewed from the links below:

- PC1 - https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc1/
- PC2 - https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc2/
- PC3 - https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc3/
- PC4 - https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc4/

3.5.2 DIS cluster

We perform a manual tour on this data, manipulating PC6 as depicted in 3.6. Looking at several frames we see that DIS HERA lie 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 is 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 take a bit to load, as the animation is several megabytes.

This is different story than if we had selected a different variable to manipulate. In 3.7 we manipulate PC2.

DIS cluster manual tours manipulating each of the principal components can be viewed from the links below:

- PC1 - https://nspyrison.netlify.com/thesis/discluster_manualtour_pc1/
- PC2 - https://nspyrison.netlify.com/thesis/discluster_manualtour_pc2/
- PC3 - https://nspyrison.netlify.com/thesis/discluster_manualtour_pc3/
- PC4 - https://nspyrison.netlify.com/thesis/discluster_manualtour_pc4/
- PC5 - https://nspyrison.netlify.com/thesis/discluster_manualtour_pc5/
- PC6 - https://nspyrison.netlify.com/thesis/discluster_manualtour_pc6/

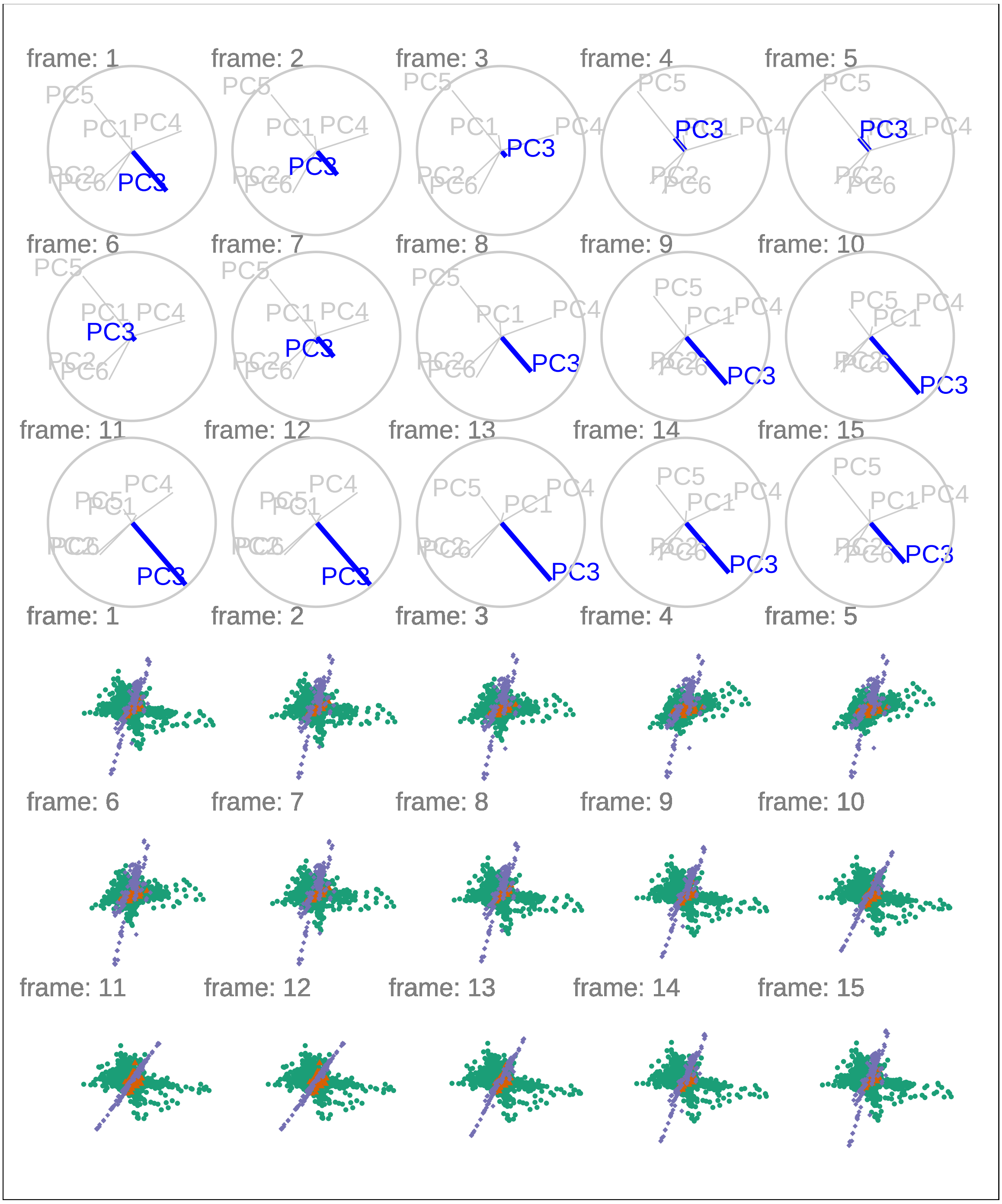

Figure 3.6: *DIS cluster, radial manual tour of PC6. colored by experiment type: 'DIS HERA1+2' in green, 'dimuon SIDIS' in purple, and 'charm SIDIS' in orange. When the contribution PC 6 is large we see that dimuon SIDIS (purple) data are nearly orthogonal to DIS HERA (green) data. As the data is rotated, we can also see that DIS HERA (green) practically lie 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/`.

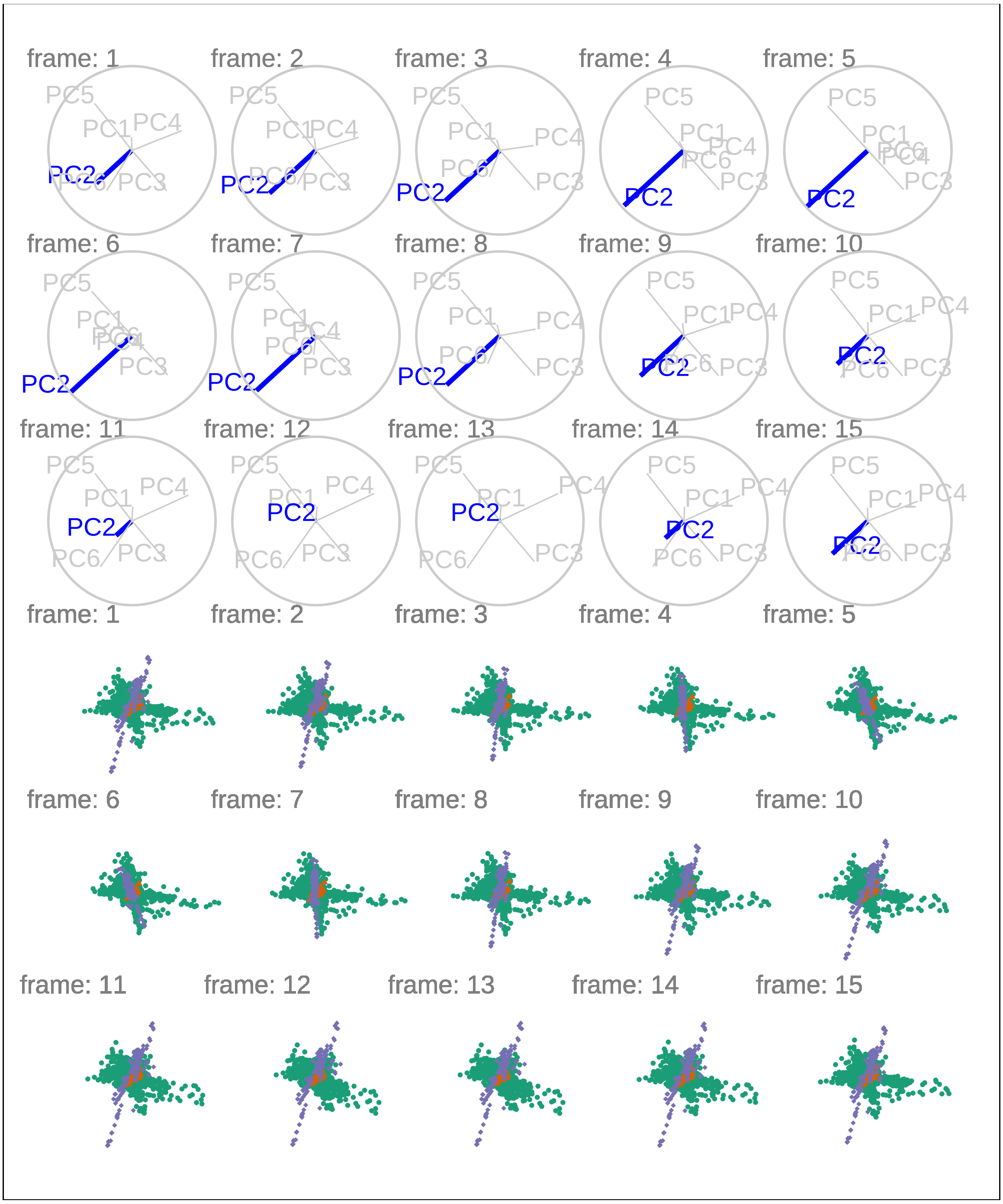

Figure 3.7: *DIS cluster, radial manual tour of PC2. Colored by experiment type: 'DIS HERA1+2' in green, 'dimuon SIDIS' in purple, and 'charm SIDIS' in orange. The structure of previously described plane of DIS HERA (green) and nearly orthogonal dimuon SIDIS (purple) is present, however the manipulating 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/`.

3.6 Source code and usage

This article was created `bookdown` (Xie (2016)) using `rmarkdown` (Xie, Allaire, and Grolemund (2018)), with code generating the examples inline, and 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")
```

3.7 Discussion

This chapter has described an algorithm and package for exploring conducting a manual tour, from a 2D projection, to explore the sensitivity of structure to the contributions of a variable.

Future work on the algorithm and package would include developing it to work with arbitrary projection dimension, enabling the method to operate on other displays like parallel coordinates, and implementing the unconstrained manual control, called oblique in Cook and Buja (1997).

The Givens rotations and Householder reflections as outlined in Buja et al. (2005) may provide a way to conduct higher dimensional manual control. In a Givens rotations, the x and y components ($ie.\theta = 0, pi/2$) 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 project points on to the axes and generate rotations.

The *tourr* package provides a number of d-dimensional graphic displays including andrews curves, chernoff faces, parallel coordinate plots, scatterplot matrix, and radial glyphs. Having manual controls available for these types of displays requires a general algorithm.

Development of a graphical user interface, e.g. *shiny* app, would make the *spinifex* package more flexible. 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 continue to explore the contributions of other variables.

Chapter 4

Touring across display dimensionality

- XGobbi vs the C2

Chapter 5

Human-computer interaction with 3d projections

- Tour in 3D
- ImAxes / IATK

Bibliography

Andrews, DF (1972). Plots of High-Dimensional Data. *Biometrics* **28**(1), 125–136. (Visited on 12/19/2018).

Anscombe, FJ (1973). Graphs in Statistical Analysis. *The American Statistician* **27**(1), 17–21. (Visited on 12/19/2018).

Asimov, D (1985). The grand tour: a tool for viewing multidimensional data. *SIAM journal on scientific and statistical computing* **6**(1), 128–143.

Becker, RA and WS Cleveland (1987). Brushing Scatterplots. *Technometrics* **29**(2), 127–142. (Visited on 01/10/2019).

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).

Buja, A, C Hurley, and JA McDonald (1987). A data viewer for multivariate data. In: *Colorado State Univ, Computer Science and Statistics. Proceedings of the 18 th Symposium on the Interface p 171-174(SEE N 89-13901 05-60)*.

Carr, D, E Wegman, and Q Luo (1996). ExplorN: Design considerations past and present. **129**.

Carreira-Perpinán, MA (1997). A review of dimension reduction techniques. *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09* **9**, 1–69.

Chernoff, H (1973). The Use of Faces to Represent Points in K-Dimensional Space Graphically. *Journal of the American Statistical Association* **68**(342), 361–368. (Visited on 01/05/2019).

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, and J Cabrera (1993). Projection Pursuit Indexes Based on Orthonormal Function Expansions. *Journal of Computational and Graphical Statistics* **2**(3), 225–250. (Visited on 01/07/2019).

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.

Fisherkeller, MA, JH Friedman, and JW Tukey (1974). PRIM-9: An Interactive Multidimensional Data Display and Analysis System.

Grinstein, G, M Trutschl, and U Cvek (2002). High-Dimensional Visualizations. en, 14.

Huh, MY and K Song (2002). DAVIS: A Java-based Data Visualization System. en. *Computational Statistics* **17**(3), 411–423. (Visited on 01/06/2019).

Hurley, C and A Buja (1990). Analyzing High-Dimensional Data with Motion Graphics. *SIAM Journal on Scientific and Statistical Computing* **11**(6), 1193–1211. (Visited on 11/27/2018).

Lee, EK, D Cook, S Klinke, and T Lumley (2005). Projection Pursuit for Exploratory Supervised Classification. *Journal of Computational and Graphical Statistics* **14**(4), 831–846. (Visited on 01/07/2019).

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

Matejka, J and G Fitzmaurice (2017). Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing. en. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. Denver, Colorado, USA: ACM Press, pp.1290–1294. http://dl.acm.org/citation.cfm?doid=3025453.3025912 (visited on 12/19/2018).

McDonald, JA (1982). INTERACTIVE GRAPHICS FOR DATA ANALYSIS.

Nelson, L, D Cook, and C Cruz-Neira (1998). XGobi vs the C2: Results of an Experiment Comparing Data Visualization in a 3-D Immer- sive Virtual Reality Environment with a 2-D Workstation Display. en. *Computational Statistics* **14**(1), 39–52.

Ocagne, Md (1885). *Coordonnées parallèles et axiales. Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles, par Maurice d'Ocagne, ...* French. OCLC: 458953092. Paris: Gauthier-Villars.

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

Siegel, JH, EJ Farrell, RM Goldwyn, and HP Friedman (1972). The surgical implications of physiologic patterns in myocardial infarction shock. English. *Surgery* **72**(1), 126–141. (Visited on 01/05/2019).

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

Swayne, DF, D Cook, and A Buja (1991). *Xgobi: Interactive Dynamic Graphics In The X Window System With A Link To S*.

Swayne, DF, DT Lang, A Buja, and D Cook (2003). GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*. Data Visualization **43**(4), 423–444. (Visited on 12/19/2018).

Tierney, L (1990). *LISP-STAT: An Object Oriented Environment for Statistical Computing and Dynamic Graphics*. eng. Wiley Series in Probability and Statistics. New York, NY, USA: Wiley-Interscience.

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*.

Wegman, EJ (2003). Visual data mining. en. *Statistics in Medicine* **22**(9), 1383–1397. (Visited on 12/19/2018).

Wickham, H, D Cook, and H Hofmann (2015). Visualizing statistical models: Removing the blindfold: Visualizing Statistical Models. en. *Statistical Analysis and Data Mining: The ASA Data Science Journal* **8**(4), 203–225. (Visited on 03/16/2018).

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.