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

Abstract

Visualizing data space is crucial to exploratory and general data analysis yet doing so quickly becomes difficult as the dimensionality of the data increases. Traditionally, static, low-dimensional linear embeddings are used to identify clustering, outliers, and structure. Observing one such embedding often misses a significant amount of variation, and hence, information held within the data. *Tours* are a class of dynamic linear projections that animates many linear projections as the orientation in data space changes. User-controlled steering (UCS) of the original dimensionality offers fine control of the local structure of projections.

Data visualization has lagged behind in utilizing 3D and virtual spaces after the overhype of the 1980s and '90s gave way to some unpromising results. Modern mixed reality hardware has significantly improved the quality and simultaniously reduced the barrier to entry. Contemporary studies have regularly shown increased accuracy of perception of visuals displayed in 3D over 2D, including in projected subspaces. It's time to further explore dynamic projections in virtual spaces.

Multivariate data is ubiquitous and viewing it in data-space is a crucial aspect of data analysis and consumption. This research is four-fold and allows for fine exploration of the data structure in embeddings of high dimensional spaces, contrasts UCS with traditional static techniques, extends UCS & creates surface projections in 3D space, and quantifies the benefits of dynamic projections across display devices.

Chapter 1

Introduction

1.1 Exploratory data analysis

The term exploratory data analysis was coined in Tukey (1977), who leaves it as a broad term to encompass the initial summarization and visualization of a data set. This is a critical first step of checking for realistic values and validating assumptions made by prospective methodology. Visualization is crucial to a clear understanding of the data. Things can go awry when data is summarized via numeric statistics alone (Anscombe, 1973) as demonstrated in figure 1.1 (Matejka and Fitzmaurice, 2017). In these studies, bivariate data have the same summary statistics (such as mean and standard deviation), yet contains obvious visual trends and shapes that could go completely unheeded if plotting is foregone. Because there are inherent dangers to relying on statistics alone, this requirement for looking at visuals necessitates *human-in-the-loop* analysis, defined as any model that requires human interaction (Karwowski, 2006).

It is clear that data-space visualization is needed but it becomes complex as data dimensionality increases. Embedding (or projecting) $p-$dimensional data on to a lower, d-dimensional subspace is a common dimension reduction approach to visualize multivariate data spaces. Traditionally a single static projection is used to summarize a space, which necessarily shows a subset of the variation of the data. Asimov (1985) suggested the use of viewing projections dynamically across a changing projection basis allows for more

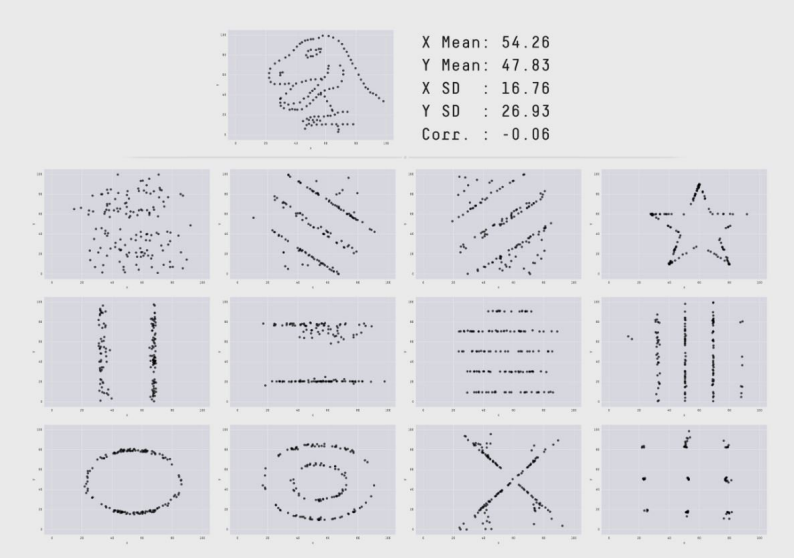


Figure 1.1: *12 data sets created from the datasaurus by simulated annealing. Each is restrained to the same summary statistics but given shapes with visual peculiarity to mutate into (Matejka and Fitzmaurice, 2017).*

variation to be contained and viewed temporally. This dynamic view of many changing projections is known as *tours*. While, there are different methods of generating tour paths, human-in-the-loop user-controlled steering (UCS) offers the finest control for navigating the local structure.

Tours are typically viewed from standard 2D monitors, and most commonly viewed as a projection down to 2D. A notable exception being Nelson, Cook, and Cruz-Neira (1998), where 3D embeddings were viewed in 3D head-tracked VR. Data visualization studies generally show benefits in 3D visuals over 2D, especially when adequate depth cues are provided. The state of modern hardware has made VR more affordable and available to wider audiences, at ever increasing resolutions of display than previously possible. It is therefore timely for research to be conducted to compare the structure and speed of comprehension of dynamic linear projections across 2- and 3D display devices.

1.2 Research objectives

Data and models are typically high-dimensional, with many variables and parameters. Developing new methods to visualize high dimensions has been a pursuit of statisticians, computer scientists and visualization researchers for decades. As technology evolves

examining, extending, and assessing current techniques, in new environments, for new data challenges, is an important endeavor. The primary objectives of this Ph.D. research can then be summarized as the following.

Research objectives (RO):

1. **How can UCS be generalized to work within graphic-specific environments for 2D projections?**
   (Work in progress, chapter 3.)
   Building from the UCS algorithm in Cook and Buja (1997), the algorithm should be modified for generalized use with graphic-specific environments. This enables fine control of exploring the local structure of data in 2D embeddings and sets the foundation to be used in the remaining objectives.

2. **Does 2D UCS tours provide benefits over alternatives?**
   (Future work, chapter 4.)
   The quality and effectiveness of 2D UCS will be compared with alternatives of static, single, linear and non-linear projection techniques. They will be quantified by the measurement of structure, variation, and clustering across on benchmark datasets.

3. **How can UCS be extended to 3D?**
   (Future work, chapter 4.)
   The addition of a 3rd dimension potentially allows for the improved perception of the structure of the data in dynamic UCS. To investigate this UCS algorithm needs to be extended to a third dimension. This would also allow for novel application mutli-parameter function projection. This will involve the addition of a new angle and it controls to the projection space, reference axes, and manipulation space. In particular, the manipulation space, now in 4D, will be hard to visualize, but it should be able stand as a mathematical construct facilitated through interaction with a point (the projection coeffcients of the selected manipulation variable) on the on the now 3D reference axes volume.

4. **Does UCS in 3D displays provide benefits over 2D displays?**
   (Future work, chapter 4.)

The addition of a 3rd dimension has previously been shown to provide benefits. The extension of UCS into 3D should be used to explore the potential benefits of UCS projections as well. Interactive, time-varying tours theoretically allow for improved understanding and comprehension speed of the structure of the data. These metrics will be measured across display device (including a 2D standard monitor, 3D head tracked monitor, and 3D head-mounted display).

Contributions:

The intended contributions and scope of this research can be summarized as:

1. A modified UCS algorithm and new implementation applied to contemporary high energy physics and astrophysics applications in 2D animation frameworks.
2. A performance comparison of static and interactive UCS projection techniques assessed on benchmark data sets from the recent literature.
3. A new algorithm for UCS in 3D. With new applications to function visualization in 3D.
4. Quantitative understanding of the relative benefits of UCS across 2- and 3D display devices.

1.3 Methodology

This research is interdisciplinary; it stems from a linear dimension reduction technique developed by statisticians and extended with information technology into 3D including VR technologies, with applications in high energy physics identified(Cook, Laa, and Valencia, 2018). Experts in these fields correspondingly supervise the research.

The research corresponding with RO #1 entails a work in progress **algorithm design** following the work in Cook and Buja (1997). The proposed algorithm discusses the generalized application of UCS for use across animation-specific frameworks. The outcome of this is an *R* package, `spinifex`, which will be submitted to CRAN and for hosting and distribution. This forms the foundation for future work in the remaining objectives.

The second objective is addressed with a benchmark dataset **performace comparison** between dynamic linear projections and alternatives (static linear and static non-linear projections such as principal component analysis, multi-dimensional scaling, and t-distributed neighbor embeddings, described in more detail in chapter 4). Benchmark datasets will be compared across techniques, measurements will include variation explained, transparency to the original variable space, clustering identification, and outlier identification.

The research for RO #3 involves **algorithm design**, where the work in RO #1 will be extended to display with the use of a third spatial dimension. Surface and function projections will also be developed. This forms the calculation base for the work. Several difficulties may arise when bringing dynamic projection into 3D spaces, especially when exploring 3D surfaces (discussed in more detail in chapter 4).

The research resulting from RO #4 is a controlled **usability study** to explore the efficacy of bringing UCS into 3D as compared across various display devices, in a standardized interface allowed by the work stemming from RO # 3. In this design, the factors are user tasks (such as separation of clusters and ranking of manipulation variable) across the treatment of display device (including 2D standard monitor, 3D head-tracked monitor, and head-mounted display). Quantitative measurements include participant speed and accuracy of tasks, biometric readings, and subjective Likert surveys of participants. A lineup-type model as outlined in Hofmann et al. (2012) may also be employed for assessing the quality of display types.

1.4 Workflow and reproducibility

Figure 1.2 depicts the general data analysis workflow (Wickham and Grolemund, 2016). Where data first must be imported into a tool, the structure of the data must be tidied and ordered neatly into the correct use format. After the data enters a repeating cycle, where values may be transformed, visualized, and modeled with communication going to the appropriate recipients. The research proposed in this document aids exploratory data analysis as well as the visualization aspect of this workflow. Mature analysis workflow is also made reproducible with the use of programmatic scripts.

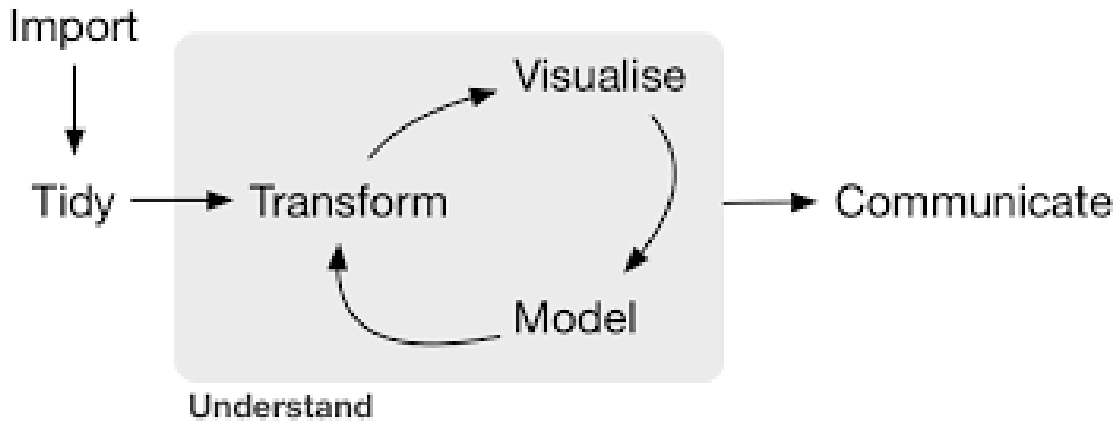


Figure 1.2: *Data analysis workflow (Wickham and Grolemund, 2016). This research aids visualization in exploratory data analysis and workflow.*

The programing language, *R*, is used in the work described below to import, tidy, and transform data. It can be used directly to visualize 2D tours (RO #1 & 2) or be consumed into the game engine *Unity* to visualize 3D tours (RO #3 & 4). Doing analysis and writeup in such programmatic ways allow work to be done reproducibly. Where data, analysis, and code are stored in the same directory. Reproducible work facilitates validation, maintains transparency and minimizes the chance for human error. Reproduction of work is a key feature to validate and defend the claims and methodology held within a work. Directories of current and planned work are/will be hosted publicly on GitHub, including this report. Accessing the source files for this report is discussed in section 6.

1.5 Project overview

Figure 1.3 depicts a schematic flow chart that the research objectives will be executed in. The research stemming from RO #1, the application of 2D user-controlled steering (UCS), sets the foundation for which the other objectives can be researched. RO #3, the application of 3D UCS, must precede RO #4, as it explores the efficacy of 3D UCS across display devices. RO #2, the comparison of 2D UCS vs alternatives, must come after RO #1, but is of lower priority to RO #3 & 4, and so will be conducted last, in the event of a time crunch.

In this report, the related literature is discussed in chapter 2. A brief overview of the research is given in chapter 1.5, followed by ongoing work and future work in chapters

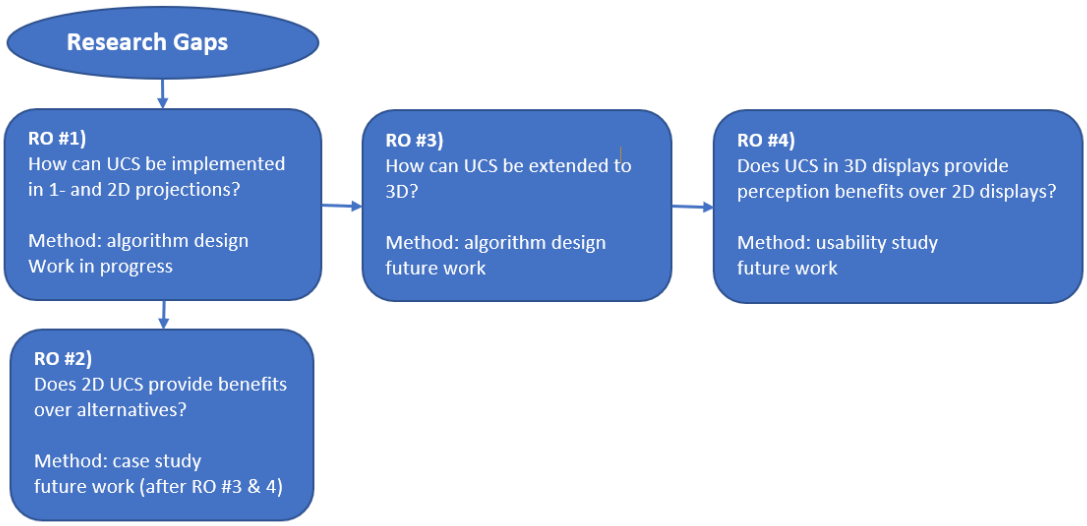


Figure 1.3: *Flow chart of research objective dependencies, work order, and methodology.*

3 and 4 respectively. A prospective timeline is listed in chapter 5. Notation for dynamic projections and VR data visualization can be found in appendix A and an excerpt of a paper to be submitted to the R Journal can be found in appendix B.

Chapter 2

Literature review

The following chapter discusses the current research in two primary areas: dynamic linear projections (collectively known as tours) and multivariate data visualization in stereoscopic 3D. After each section, research gaps are highlighted.

2.1 Dynamic linear projections of multivariate data (tours)

2.1.1 Overview

The introduction established that visualizing data-space is an important aspect of exploratory data analysis and data analysis in general. Yet, there is an inherent difficulty as the dimensionality of data increases. In univariate data sets histograms or smoothed density curves are employed to visualize data. In bivariate data, XY scatterplots and contour plots (2D density) can be employed. In three dimensions, the two most common techniques are 3D scatterplot[1] or 2D scatterplot with the 3rd variable as an aesthetic (such as color, size, or height). These aesthetic cues convey some information but are not a sufficient replacement for axes for use with continuous variables.

[1] Graphs depicting three dimensions are typically viewed on a 2D display, in print or with a standard monitor. These are 2D images of monocular 3D spaces, sometimes referred to as 2.5D data visualization, more on this in section A.2.

As dimensionality of the data, p, increases the visualization of data-space quickly becomes complex. It is common that visualizing data-space is dropped in favor of graphing model-space (for example residuals), parameter-space (in fewer dimensions), or worse yet: long tables of statistics without visuals (Wickham, Cook, and Hofmann, 2015). To preserve the visualization of data-space, a solution that scales with the dimensionality of data is needed; this is where dimensionality reduction comes in. This work will focus on a group of dynamic linear projection techniques collectively known as *tours*. The scope of the project lies within the dynamic linear projections; a broader review of dimensionality reduction techniques is discussed in the review paper by Grinstein, Trutschl, and Cvek (2002). Tours are used for a couple of salient features: the use of linear projections maintains transparency back to the original variable space (which non-linear projections lose) and the use of many projections shows more variation than a single linear projection. Employing the breadth of tours extends the dimensionality of visualizations, and with it, the intrinsic understanding of the structure and distribution of data that is more succinct or beyond the reach of summary statistics alone.

Let p be the dimensionality of the data, and d be the dimension of the projection space. Tours perform linear dimensionality reduction, orthogonally projecting p-space down to $d(\leq p)$ dimensions. Many such projections are interpolated, each making small rotations in p-space. These frames are then viewed in order as an animation of the lower dimensional embedding changing as the original variable space is manipulated. Shadow puppets offer a useful analogy to aid in conceptualizing tours. Imagine a fixed light source facing a wall. When an object is introduced it projects a 2D 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 2D tour as the 3D object is manipulated. Some orientations explain more information about the shape of the object than others while watching an animation of the shadow changing gives a more robust understanding than looking at any one frame. More complex structures generally require more time to comprehend the nature of the geometry. These features hold for tours as well.

An extended tour notation is listed in the appendix section A.1.

2.1.2 History

The first tour was introduced was the *grand tour* in Asimov (1985) at the Stanford Linear Accelerator, Stanford University. Asimov suggested three types of grand tours: torus, at-random, and random-walk. The original application of tours was performed on high energy physics on the PRIM-9 system.

Before choosing projection paths randomly, an exhaustive search of $p-$space was suggested by McDonald (1982), also at the Stanford Linear Accelerator. This was later coined *little tour*.

Friedman and Tukey (1974) and later Huber (1985) recommended projection pursuit (also referred to as PP). Projection pursuit optimizes an objective function before it removes a single component/variable and then iterates in this newly embedded subspace. Within each subspace, the projection seeks for a local extremum via a hill climbing algorithm on an objective function. This formed the basis for *guided tours* suggested by Hurley and Buja (1990).

The grand and little tours have no input from the user aside from the starting basis. Guided tours allow for an index to be selected. The bulk of tour development since has largely been around the dynamic display, user interaction, geometric representation, and application.

2.1.3 Path generation

A fundamental aspect of tours is the path of rotation. There are four primary distinctions of tour path generation (Buja et al., 2005): random choice, data-driven, precomputed choice, and manual control.

- Random choice, *grand tour*, constrained random walks p-space. Paths are constrained for changes in direction small enough to maintain continuity and aid in user comprehension
  - torus-surface (Asimov, 1985)
  - at-random (Asimov, 1985)

  - random-walk (Asimov, 1985)
  - *local tour* (Wickham et al., 2011), a sort of grand tour on a leash, such that it goes to a nearby random projection before returning to the original position and iterating to a new nearby projection.

- Data-driven, *guided tour*, optimizing some objective function/index within the projection-space, called projection pursuit (PP) (Hurley and Buja, 1990), including the following indexes:

  - holes (Cook, Buja, and Cabrera, 1993) - moves points away from the center.
  - cmass (Cook, Buja, and Cabrera, 1993) - moves points toward the center.
  - lda (Lee et al., 2005) - linear discriminant analysis, seeks a projection where 2 or more classes are most separated.
  - pda (Lee and Cook, 2010) - penalized discriminant analysis for use in highly correlated variables when classification is needed.
  - convex (Laa and Cook, 2019) - the ratio of the area of convex and alpha hulls.
  - skinny (Laa and Cook, 2019) - the ratio of the perimeter distance to the area of the alpha hull.
  - stringy (Laa and Cook, 2019) - based on the minimum spanning tree (MST), the diameter of the MST over the length of the MST.
  - dcor2D (Grimm, 2017; Laa and Cook, 2019) - distance correlation that finds linear and non-linear dependencies between variables.
  - splines2D (Grimm, 2017; Laa and Cook, 2019) - measure of non-linear dependence by fitting spline models.
  - other user-defined objective indices can be applied to the framework provided in the *tourr* package Wickham et al. (2011).
  - Another data-drive tour is the *dependence tour*, a combination of n independent 1D tours. A vector describes the axis each variable will be displayed on. for example $c(1,1,2,2)$ is a 4- to 2D tour with the first 2 variables 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.

- Precomputed choice, *planned tour*, 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 brute-force or exhaustive search.
  - a saved path of any other tour, typically an array of basis targets to interpolate between.
- Manual control, *manual tour*, 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, perhaps via guided tour.

2.1.4 Path evaluation

Consider projection down to 2D, then each projection is called a 2-frame (each spanning a 2-plane). Mathematically, a Grassmannian is the set of all possible unoriented 2-frames in p-space, $\mathbf{Gr}(2,\ p)$. Asimov (1985) pointed out that the unique 2-frames of the grand tour approaches $\mathbf{Gr}(2,\ p)$ as time goes to infinity. The *density* of a tour is defined as the fraction of the Grassmannian explored. Ideally, an exploring tour will be dense, but the time taken to become dense vastly increases as variable space increases dimensionality. *Rapidity* is then defined as how quickly a tour encompasses the Grassmannian. Due to the random selection of a grand tour, it will end up visiting homomorphisms of previous 2-frames before all unique values are visited, leading sub-optimal rapidity.

The little tour introduced in McDonald (1982), on the other hand, is necessarily both dense and rapid, performing essentially an exhaustive search on the Grassmannian. However, this path uninteresting and with long periods of similar projections strung together. The Grassmannian is necessarily large and increases exponentially at the rate of p. Viewing of the whole Grassmannian is time-consuming, and interesting projections are sparse, there was a clear space for computers to narrow the search space.

Guided tours (Hurley and Buja, 1990) optimize an objective function generating path will be a relatively small subset of the Grassmannian. As they are not used for exploration, density and rapidity are poor measures. On the other hand, they excel at finding interesting

projections quickly. Recently, Laa and Cook (2019), compared projection pursuit indices with the metrics: *smoothness*, *squintability*, *flexibility*, *rotation invariance* and *speed*.

2.1.5 Geometric display by dimensionality

Up to this point, 2D scatterplots have been the primary display discussed, they offer a logical display for viewing embeddings of high-dimensional point clouds. However, other geometrics offer perfectly valid projections as well.

- 1D geometrics (geoms)
  - 1D densities: such as histogram, average shifted histograms (Scott, 1985), and kernel density (Scott, 1995).
  - image (pixel): (Wegman, Poston, and Solka, 2001).
  - time series: where multivariate values are independently lagged to view peak and trough alignment. Use case is discussed in (Cook and Buja, 1997).
- 2D geoms
  - 2D density (available on GitHub at `https://github.com/nspyrison/tourr`)
  - XY scatterplot
- 3D geoms
  - Anaglyphs, sometimes called stereo, where red images are positioned for the left channel and cyan for the right when viewed with corresponding filter glasses give a perception of depth to the image.
  - Depth, which gives depth cues via aesthetic mappings, most common size and/or color of data points.
- d-dimensional geoms
  - Andrews curves (Andrews, 1972), smoothed variant of parallel coordinate plots, discussed below.
  - Chernoff faces (Chernoff, 1973), variables linked to the size of facial features. The idea is to apply the human facial-perception for rapid, cursory, like-ness comparisons between 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 (SPLOM) (Becker and Cleveland, 1987), showing a triangle matrix of bivariate scatterplots with 1D density on the diagonal.
  - Radial glyphs, radial variants of parallel coordinates including radar, spider, and star glyphs (Siegel et al., 1972).

2.1.6 Tour software

Tours have yet to be widely adopted, due in part, to the fact that print and static .pdf output does not accommodate dynamic viewing. Conceptual abstraction and technically density have also hampered user growth. Due to low levels of adoption and the rapid advancement of technology support and maintenance of such implementations give them a particularly short life span. Despite the small user base, there have been a fair number of tour implementations, including:

- spinifex github.com/nspyrison/spinifex – R package, all platforms.
- tourr (Wickham et al., 2011) – R package, all platforms.
- CyrstalVision (Wegman, 2003) – for Windows.
- GGobi (Swayne et al., 2003) – for Linux and Windows.
- DAVIS (Huh and Song, 2002) – Java based, with GUI.
- ORCA (Sutherland et al., 2000) – Extensible toolkit build in Java.
- VRGobi (Nelson, Cook, and Cruz-Neira, 1998) – for use with the C2, tours 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.
- Explor4 (Carr and Nicholson, 1988) – Four-dimensional data using stereo-ray glyphs.
- Prim-9 (Asimov, 1985; Fisherkeller, Friedman, and Tukey, 1974) – on an internal operating system.

2.1.7 Research gaps

Dynamic projections offering UCS have not incorporated recent animation frameworks (**RO #1**). This leaves the class of dynamic linear projections without the most precise, fine-scale control of rotating p-space. This should be implemented with an eye on extensibility and maintainability.

A comparative study outlining the benefits of UCS vs alternatives is also absent from the literature (**RO #2**). The benefits of dynamic linear projections hold in theory, but a direct comparison with popular alternatives should be made. Barriers to adoption and sharing should also be kept in mind as the dynamic display is not easy to display on print and in .pdf documents.

2.2 Multivariate data visualization in 3D

The scope of this research pertains to numeric multivariate data, a wider overview of 3D data visualization is discussed in chapter 2 of Marriott et al. (2018). Terminology for 3D visuals is in the glossary section A.2.

2.2.1 A rocky start

Scientific visualization has readily adopted mixed realities as a large amount of the science exists in three spatial dimensions, lending itself well to virtual immersion (Marriott et al., 2018). Data visualization, on the other hand, has been slow to utilize graphics above 2.5D, (and haptic interaction) primarily due to the mixed results of over-hyped of 3D visuals from the 1980s and '90s (Munzner, 2014). Since then, however, there have been several promising studies suggesting that it is time for data visualization to revisit and adopt the use of 3D visuals for specific combinations of displays and depth cues.

2.2.2 3D rotated projections vs 3 2D orthogonal projections

Three orthogonal 2D views can represent the three face-on views of 3D shapes. When 3D representations are used with binocular cues, they are found to have more accurate perception than 2D counterparts (Lee, MacLachlan, and Wallace, 1986).

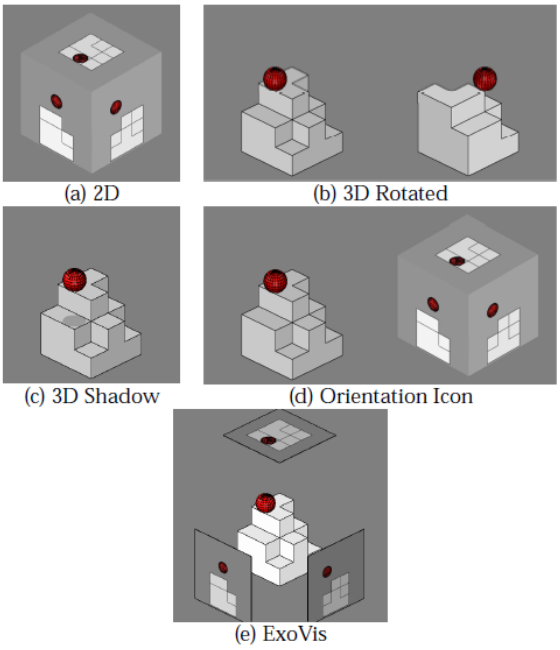


Figure 2.1: *Screen capture from Tory et al. (2006): "fig. 1 (a) 2D, (b) 3D Rotated, (c) 3D Shadow, (d) Orientation Icon, and (e) ExoVis displays used in Experiment 1 (position estimation). Participants estimated the height of the ball relative to the block shape. In this example, the ball is at height 1.5 diameters above the block shape."*

Between 3D and split view 2D of three-dimensional economics, data Wickens, Merwin, and Lin (1994) asked participants questions integrating several variables, finding that 3D visuals resulted in faster answers when questions involved three dimensions, while the speed was similar when questions involved fewer dimensions.

Using 3D rotated projections gives more precise (relative to 2D) estimates of the height a ball is suspended above complex box shapes, while combinations of 2D and 3D give the most precise orientation and positioning information (Tory et al., 2006, depicted in figure 2.1).

Sedlmair, Munzner, and Tory (2013), depicted in figure 2.2, asked users about cluster separation across 2D scatterplots, 2D scatterplot matrices (SPLOMs) and interactive 3D scatterplots, PCA (linear projection), and t-SNE (non-linear projection) as viewed in monocular 3D from a standard monitor. They conclude that interactive 3D scatterplots perform worse for class separation. This result is surprising as the extra dimension theoretically allows for the clustering structure to be seen and explored more clearly.

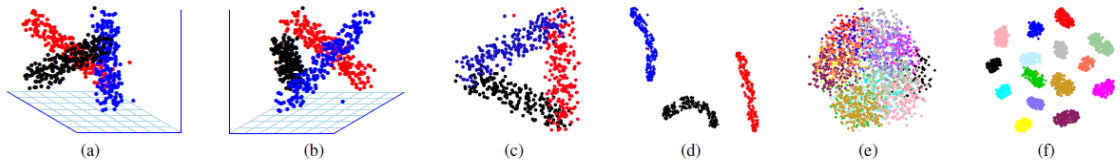


Figure 2.2: *Screen capture of "figure 5. example of a mesh display" from Sedlmair, Munzner, and Tory (2013): "fig. 5. (a)-(d): Screenshots of the entangled dataset* `entangled1-3d-3cl-separate` *designed to show the most possible benefits for i3D. (a),(b) two viewpoints of the same i3D PCA scatterplot. An accompanying video shows the full 3D rotation. (c) 2D PCA projection. (d) t-SNE untangles this class structure in 2D. (e)-(f): 2D scatterplots of the reduced* `entangled2-15d-adjacent` *dataset which we designed to have a ground truth entangled class structure in 15D. (e) Glimmer MDS cannot untangle the classes, neither can PCA and robPCA (see supplemental material). (f) t-SNE nicely untangles and separates the ground truth classes in 2D."*

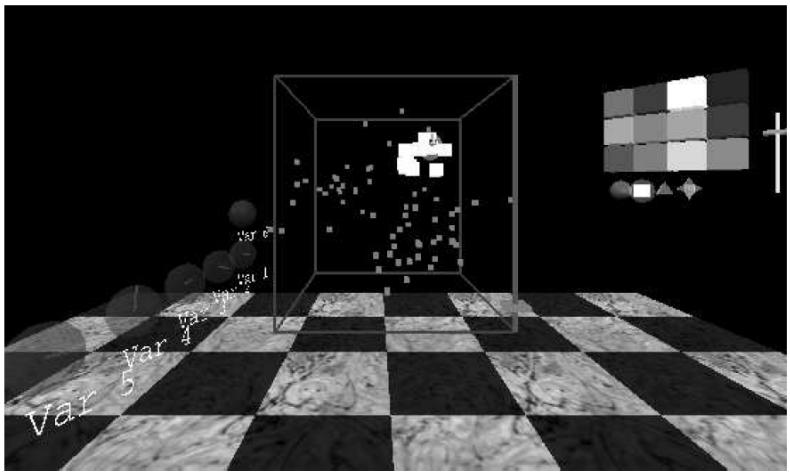


Figure 2.3: *Screen capture from Nelson, Cook, and Cruz-Neira (1998): "figure 4: This is a picture of a 3-D room, running VRGobi. Data is plotted in the center, with painting tools to the right and variable spheres to the left. In the viewing box, the data can be seen to contain three clusters, and one is being brushed."*

2.2.3 Comparing 3D and 2D embeddings of multivariate data

Nelson, Cook, and Cruz-Neira (1998), depicted in figure 2.3, had $n = 15$ participants perform brushing and identification tasks (of clusters, structure, and data dimensionality) in 3D with head-tracked binocular VR. 3D proved to have a substantial advantage for cluster identification and some advantage in identifying the shape. Brushing did take longer in VR, perhaps due to the lower familiarity of manipulating 3D spaces.

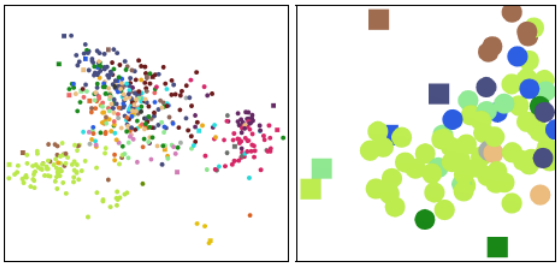


Figure 4: In the *2D* condition, data points are distributed along screen space (left), and the user is allowed to zoom and pan (right).

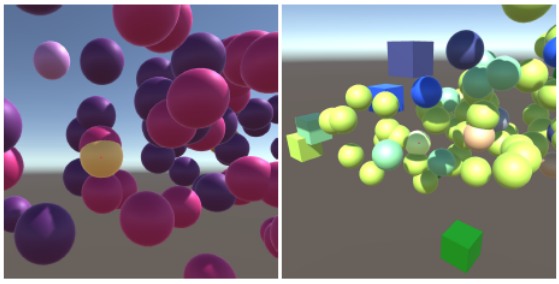


Figure 5: In the *3D* conditions, the user is allowed to freely navigate through the data, which is distributed along a 3D virtual environment.

Figure 2.4: *Screen capture from Wagner Filho et al. (2018), original captions contained in the capture.*

Another study, Gracia et al. (2016) performed dimensionality reduction down to 2- and 3D scatterplots, both displayed in monocular 3D on a standard monitor. Users were found to more accurately compare distances between points and identify outliers on 3D scatterplots. However, both tasks were performed slower with the use of the 3D scatterplots and statistical significance was not reported.

Wagner Filho et al. (2018), depicted in figure 2.4, performed an $n = 30$ empirical study on PCA embedded projections, measuring perception error across 4 tasks and 3 display types: 2D, static 3D, and immersive 3D. Overall task error was less in static and immersive 3D relative to 2D. According to the user Likert-scale survey, 2D is slightly easier to navigate and slightly more comfortable, while, static and immersive 3D displays are slightly easier to interact with and moderately easier to find information on.

2.2.4 Immersive analytics platform in VR

Immersive analytics is an emerging field, where data visualization and analysis is facilitated in an intuitive, immersive virtual reality environment. An example of which is shown in Cordeil et al. (2017) introduces a collaborative space for immersive data analysis.

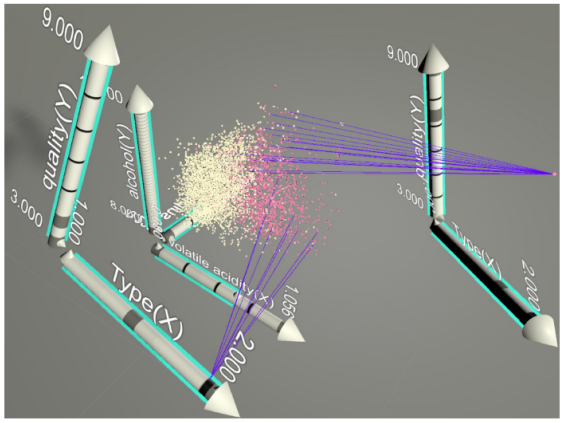


Figure 2.5: *Screen capture of figure 15 from Cordeil et al. (2017).*

Where axes are displayed and intuitively interacted with while responding to proximity to other variable axes and popping into place changing the resulting geometric display. For example, three variables can be placed as the x, y, $z-$ axes for a 3D scatterplot or stood up right next to each other for a parallel coordinate plot. The subsequent work in Cordeil (2019) builds from the previous reference and refines it for the next iteration in interactive, scalable data visualization in virtual spaces.

2.2.5 Research gaps

When comparing between 2- and 3D orthogonal views, in general, studies show that perception accuracy is better in 3D, though manipulation speed is generally slower. The speed discrepancy is confounded by the difference in users familiar with manipulating 2D vs 3D spaces (Lee, MacLachlan, and Wallace, 1986; Wickens, Merwin, and Lin, 1994; Tory et al., 2006; counterexample Sedlmair, Munzner, and Tory, 2013).

Similar results have been shown in static, 3D projected spaces (Gracia et al., 2016; Wagner Filho et al., 2018) and in dynamic 2D embedded spaces depicted in immersive 3D (Nelson, Cook, and Cruz-Neira, 1998). Modern VR hardware brings about a steady improvements to quality, resolution while driving down cost, making VR more easily accessible than ever. It's timely to review dynamic 3D projections and do so in immersive spaces to quantify the corresponding benefits (**RO #3 & 4**).

Chapter 3

Work in progress

Extending the algorithm for UCS in 2D for generalized use in graphics environments is fundamental to the extension of the UCS into 3D space. This forms the foundation for future work addressed in the remaining research objectives. This is a synthesis of the work that will be submitted to the R Journal. A wider discussion of implementation as an *R* package and application to high energy physics data (Wang et al., 2018; Cook, Laa, and Valencia, 2018) is attached in appendix B formatted as a paper.

3.1 RO #1) How can UCS be generalized to work within graphic-specific environments for 2D projections?

This chapter builds off of *manual tours* (Cook and Buja, 1997) and modifies the algorithm for generalized use in graphics implementations. Manual tours allow users to rotate a selected variable into and out of a 2D projection of high-dimensional space for user-controlled steering (UCS). The primary use is to determine the sensitivity of the structure to the contributions of a selected manipulation variable. This is particularly powerful for exploring the local structure once a feature of interest has been identified, for example, by a guided tour.

3.2 Algorithm

The algorithm for generating a manual tour path involves these steps:

1. Given a 2D projection basis, choose a variable to explore. This is called the "manip" variable.
2. Create a 3D manipulation space, where the manip variable has a full contribution.
3. Generate a rotation sequence which zeros coefficient and increases it to 1 before returning to the initial position.

3.2.1 Step 1 Choose variable of interest

Start with multivariate data in an $n \times p$ numeric matrix, and an orthonormal (orthogonal and each column normalized to have a norm of 1) $p \times d$ basis set is describing the projection from $p-$ to $d-$space. Select a manipulation (manip) variable. The coefficients of this variable will be operated on.

3.2.2 Step 2 Create the manip space

Use the Gram-Schmidt process to orthonormalize another dimension on the basis with the full contribution on the manip variable. This brings the basis into a 3D manip space. The new dimension giving the basis room to rotate 'out-of-plane', akin to being able to lift a piece of paper, rather than being confined to the top of a table.

3.2.3 Step 3 Generate rotation

Define or capture a sequence of values for two angles, the 'in-plane' and 'out-of-plane', picture within an XY plane, and out of Z-axis. For the corresponding value in each set, rotate (post multiply) the manip space according to a rotation matrix.

3.3 Display projection sequence

Once the path has been generated, return to data-space matrix multiply the data with the rotated manip space. Plot the first two components as an XY-scatterplot for each

frame. The projected data is commonly shown with reference axes, the coefficients of each variable extending from a unit circle showing the direction and magnitude of contribution to the 2D projection space.

Chapter 4

Future work

4.1 RO #2) Does 2D UCS provide benefits over alternatives?

Dimensionality reduction is important to viewing high dimensional data spaces. There are various techniques have been developed to view projections of data. The research answering this question would quantify the potential benefits of dynamic 2D UCS over commonly used alternatives. All comparison groups would be unsupervised (agnostic of clustering), static, single embeddings in a lower dimension, and would include:

- **Principal Component Analysis (PCA)**, a linear transformation that forms orthogonal linear combinations of the variables by maximizing the amount of variation that is independent of all preceding components. That is, the first principal component is the linear combination that explains the most variation in one direction, the second component explaining the most of the remaining variation and is orthogonal to the first, and so on.
- **Multi-Dimensional Scaling (MDS)**, non-linear dimension reduction that compares pairwise distances between observations.
- **t-distributed neighbor embeddings (tSNE)**, a nonlinear technique that iterates epochs of 1) constructing a probability distribution for selecting neighboring data and 2), minimizing Kullback-Leibler divergence (a measure of relative entropy).

Unfortunately, static linear projections necessarily lose the variation of the components not displayed, while non-linear techniques lose transparency back to the original variable space. On the other hand, dynamic linear projections keep variation in tack (at the expense of viewing over time) and preserve transparency to variable-space. User-controlled steering of tours allows for finer exploration local exploration, which should be reflected in the benefits over alternative options.

The methodology for this future work is a **performace comparison** across technique, as assessed across contemporary benchmark datasets. Differences in the techniques make for an uneven comparison, but measurements will be made where applicable comparing at least variation, clustering, and structure. The design space includes data sets, techniques, and measures of comparison.

4.2 RO #3) How can UCS be extended to 3D?

The literature has shown positive results for improved perception of 3D graphics over 2D. The additional dimension theoretically allows for the improved structural perception in 3D scatterplots and would allow for the novel application of the dynamic projection of multidimensional function spaces. The research answering RO #1 will be extended for these uses.

The work presented in Cordeil et al. (2017) creates immersive space for users to explore data in a virtual environment. Users can actively create different visualization by spatial manipulation of virtual variable-axes. Bringing dynamic projection into an environment that offers immersive interaction could be a boon to interfacing with something so dynamic in nature. The ability to render in 3D could also act as a common interface that can be used across various display devices in RO #4.

The research addressing this objective applies **algorithm design**, first, the *R* package spinifex will be extended to 3D and function/surface projections will be developed. After the projections are computed, *Unity* will be used to render the embeddings in 3D VR and act as a compatible front end to be used across display devices.

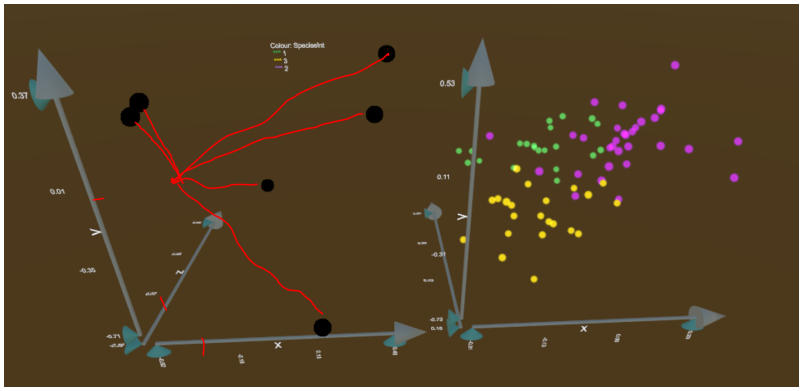


Figure 4.1: *A mockup of a 3D basis reference space (left) and scatterplot projection (right). Basis axes should extend from the origin to allow a better perception of magnitude and direction each variable contributes to the projection.*

Manipulating 3D spaces may not be straight forward. In section {sec:algorithm} the manipulation space was in 3D, where 2 angles defined a point that was projected back to the 2D reference frame. The now 4D manipulation space should only be necessary for internal mathematics, where the 3 angles spanning it, could be controlled through manipulation of a selected variable on the now 3D reference volume. Navigating 3D space may be very intuitive or may require a slide ruler for each axis may offer more control. An additional concern of user interaction is the potential for objects in peripheral vision to cause discomfort. The angular speed of the projections should be regulated for continuity of observation and to mitigate potentially nauseating movement.

In a function projection, a multi-parameter function surface would be projected rather than individual data points. Imagining the projection of a unit grid is a useful middle ground. Viewing projected functions may have several difficulties in 3D. The first is occulation, the surfaces in the foreground blocks the view behind it. Opacity, wire mesh, and projection sectioning are potential ways to address this issue. A second issue is that it may be disorienting or nauseating to watch surfaces folding into each other in seemingly non-euclidean movements. Changing opacity or focus in the vicinity of these areas may mitigate the potential concern.

The design space for this research includes the path generators (outlined in section 2.1.3), geometric display (section 2.1.5), layout in virtual space, dynamic interactions. Tour paths are conceptually straight-forward mapping between values and 3D rendering. Each

geometric display will need unique recreation, though 3D scatterplot, parallel coordinate plots, and scatterplot matrices (SPLOMs) are currently supported in the respective packages.

4.3 RO #4) Does UCS in 3D displays provide perception benefits over 2D displays?

The bulk of previous tours were performed in 2D, with the exceptions of Nelson, Cook, and Cruz-Neira (1998) and Arms, Cook, and Cruz-Neira (1999) who conducted an $n = 15$ experimental study comparing tasks performed across 2D and 3D tour displays. The XGobi interface was used on a standard 2D monitor while VRGobi (on the C2 setup) was used with head-tracked, binocular VR. The three accuracy tasks: clustering, intrinsic data dimensionality, and radial sparseness were recorded along with the speed of brushing data. Accuracy was the same for the dimensionality task, while 3D display outperformed 2D on clustering, and even more so on the radial sparsity task. However, the time taken to brush a cluster was less than half the time in 2D displays as compared with 3D.

Wagner Filho et al. (2018) performed a user study on the perception of linear projections between 2D, 3D, and immersive 3D. The $n = 30$ user study created 3D embeddings of multidimensional data via principal component analysis (PCA, described in RO #2, above). Users performed three tasks across two data sets and three displays; 2D, 3D, and immersive 3D. Data sets were chosen to have vastly different amounts of information contained in the 3rd principal component. They find that the introduction of a 3rd dimension in visualization improves task performance (perception error, task error, and completion time) regardless of immersion for only the dataset containing large amounts of variation in the 3rd principal component. Independent of the dataset, immersive 3D display led to a larger subjective perception of accuracy and engagement.

The results of Wagner Filho et al. (2018), Nelson, Cook, and Cruz-Neira (1998) and, Arms, Cook, and Cruz-Neira (1999) cast positive light on 3D spaces improving the perception of embeddings of high-dimensional data. Others have found the same for tri-variate data.

After tours have been extended in 3D spaces (RO #3), the effects of viewing dynamic projections should be quantified across the display type.

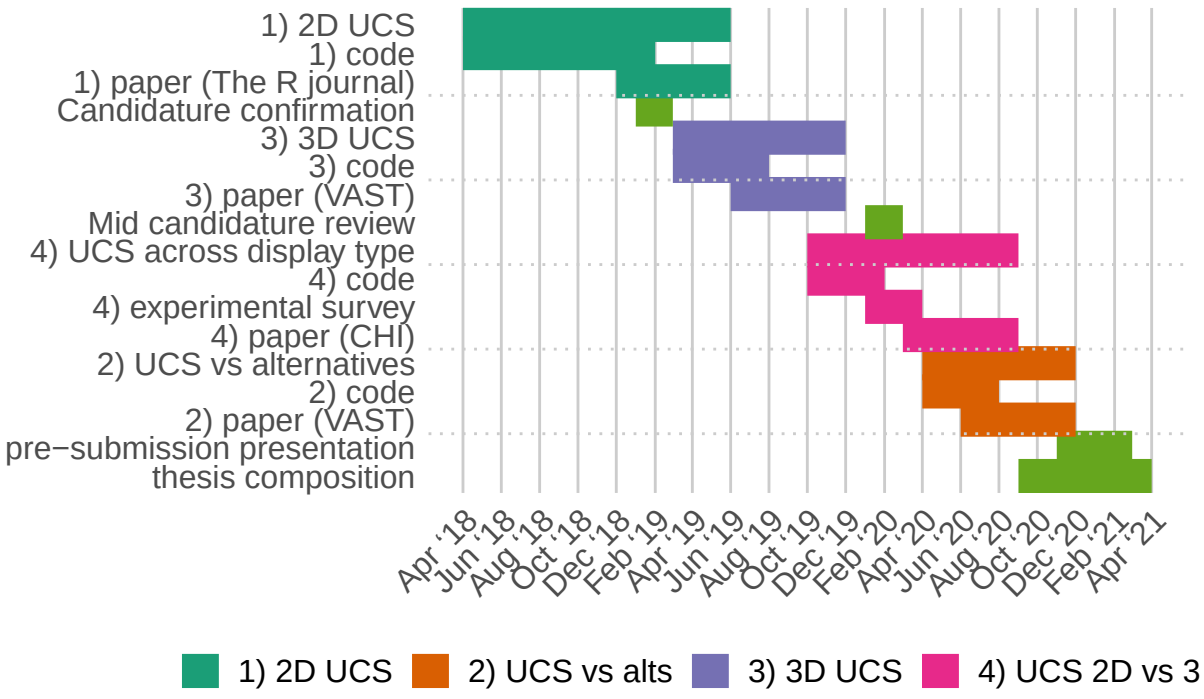
A controlled **usability study** will be performed to measure the benefits of 3D over 2D display devices. Every participant will complete every task on every display device. Task order and display device will be randomly assigned to minimize learning bias. Correctness and speed of tasks will be recorded alongside demographic data and subjective 5-point Likert scale survey. A lineup model as outlined in Hofmann et al. (2012) may be employed to quantify the "best" display device. This lineup model is a visual variant of statistical p-test where participants are asked to pick the real data set as pitted against data generated from the null hypothesis. Running such a lineup within all display devices and comparing accuracy may rank the quality of the display device.

The factors of the study are user tasks: the perception of cluster structure, surface structure, and ranking of manipulation variables. All factors will be tested across the treatment of at least three display devices: standard 2D monitor, stereoscopic 3D monitor (on a zSpace 200), and head-mounted VR goggles (HTC VIVE). The user interface will be standardized across display devices. The data explored will be of high energy physics experiments already being discussed in publication (Wang et al., 2018; Cook, Laa, and Valencia, 2018) and looked at in 2D UCS in appendix B.

Chapter 5

PhD schedule

5.1 Timeline



Note RO #2 logically would fit before RO #3 & 4, but it is of lower impact and retrospective performance comparison. I move this research to the end of the timeline to give the other research objectives priority in the event of time constraints.

5.2 Accompanying documents

- FIT 5144 hours
  - >120 hours **Tracked, awaiting mandatory events**, due at the mid-candidature review
- WES Academic record
  - FIT6021: 2018 S2, **Completed** with Distinction
  - FIT5144: 2019 S1+2, **Upcoming**, due at mid-candidature review
  - FIT5113: 2018 S2, **Exemption submitted**, forwarded 14/02/2019
- myDevelopment - IT: Monash Doctoral Program - Compulsory Module
  - Monash Graduate Research Student Induction: **Completed**
  - Research Integrity - Choose the Option most relevant: **Completed** (2 required of 4)
  - Faculty Induction: **Content unavailable** (25/02/2019: "Currently being updated and will be visible in this section soon")

Chapter 6

Source code

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.

Following best practices of version control, transparency and reproducibility, the source files can be found at github.com/nspyrison/confirmation/.

Appendix A

Glossary

A.1 Tour notation

Tour notation varies across articles and authors. In my work, I use the following:

- n, number of observations in the data.
- p, number of numeric variables, the dimensionality of data space.
- d, the dimensionality of projection space.
- $\mathbf{X}_{[n,\ p]}$, a data matrix in variable-space, $\mathbf{X} \in \mathbb{R}^p$. Typically centered, scaled, and optionally sphered.
- $\mathbf{B}_{[p,\ d]}$, orthonormal (orthogonal and each column normalized to a norm of 1) basis set, defining the axes directions for the projection from $p-$ to $d-$space.
- $\mathbf{Y}_{[n,\ d]}$, projected data matrix in projection-space, $\mathbf{Y} \in \mathbb{R}^d$.
- Reference axes (or reference frame), a display showing how each variables coefficient(s) contribute to a projection. Either on its own axis (1D) or relative to a unit circle (2D).
- Geometric objects are referred to in generalized dimensions; the use of the term plane is not necessarily a 2D surface, but a hyperplane in the arbitrary dimensions of the projection space.

A.2 Data visualization terminology

- 2D - representation of data in 2 dimensions, without the use of depth perception cues and minimal aesthetic mapping (such as color, size, and height) to data points.
- 2.5D - following the definition given in Ware (2000): visualizations that are essentially 2D but select depth cues are used to provide some suggestion of 3D. However, the term 2.5D is commonly used for several meanings *due to the ambiguous use of 2.5D, this document errs on the side stating 3D with descriptions of depth cues used*.
- 3D - visualizations of 3 dimensions with liberal use of depth cues unless otherwise qualified.
- Depth perception cues - an indication that indicates the depth to an observer, including:
  - linear perspective - the property of parallel lines converging on a vanishing point.
  - aerial perspective - objects that far away have lower contrast and color saturation due to light scattering in the atmosphere.
  - occultation (or interposition) - where closer objects partially block the view of further objects.
  - motion perspective/parallax - closer objects, move across the field of view faster than further objects.
  - accommodation - the change of focal length due to change in the shape of the eye. Effective for distances of less than 2 meters.
  - binocular stereopsis/disparity - the use of 2 images of slightly varied angles from the horizontal distance of the eyes. The disparity for distant objects is small, but it is significant for nearby objects.
  - binocular convergence - The ocular-motor cue due to stereopsis focusing on the same objects. Convergence is effective for distances up to 10 meters.
- Virtual reality (VR) - an immersive experience of computer-generated sensory input.
- Augmented reality (AR) - view of physical spaces with augmenting/ supplementing sensory input of information.

- Mixed reality (MR) - a mix of physical and virtual realities with objects from both interacting in real time. This differs from AR by the flow of interaction; AR augments physical reality while MR has reciprocating interactions.
- Extended reality (XR) - any degree of virtual, augmented, or mixed reality.
- Scatterplot matrices (SPLOMs) - matrix display of pair-wise 2D scatterplots, sometimes with 1D density on the diagonal.
- Human-in-the-loop - any model that requires human interaction (Karwowski, 2006).

Appendix B

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.

B.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 that 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 *gganimate* to allow users to rotate the selected variable into and out of a chosen projection.

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

B.2 Introduction

A tour is a multivariate data analysis technique in which is a sequence of linear (orthogonal) projections into a lower subspace in which $p-$space is rotated across time. Each frame of the sequence corresponds to a small change in the projection for a smooth transition to persevere the object continuity.

Multivariate data analysis can be broken into 2 groups: linear and non-linear transformations. Like PCA and LDA, tours use linear dimension reduction that maintains transparency back to the original variable space. PCA and LDA are typically represented with single static projection as a 2- or 3D scatterplot, inherently losing the variation held with the high components, whereas tours keep the information in tack by showing the other components across time. Non-linear transformations such as tSNE (t-distributed stochastic nearest neighbor embeddings), MDS (multi-dimension scaling), and LLE (local linear embedding) distort the parameter-space which lacks transparency back to the original parameter-space. They show more extreme separation in embeddings, but the variable opacity can be a non-starter for many uses.

There are many ways that a tour path can be generated, we will focus on one, 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. This will be called user-controlled steering (UCS). The primary purpose is to determine the sensitivity of structure visible in a projection to the contributions of a variable. Making them particularly 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 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 manual tours 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 3.2 explains the algorithm using a toy dataset. Section B.5 illustrates how this can be used for sensitivity analysis. The last section, B.7 summarizes the work and discusses future research.

B.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 the full contribution.
3. Generate a rotation sequence which zero's the norm of the coefficient and increases it to 1.

These steps are described in more detail below.

B.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.

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

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

B.3.2 Toy data set

The flea data from the R package *tourr* (Wickham et al., 2011), is used to illustrate the algorithm. The data, originally from Lubischew (1962), contains 74 observations across 6 variables, which physical measurements of the insects. 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. The holes index is maximized by when the projected data has a lack of observations in the center. Figure **??**, shows an optimal projection of this data. The left plot

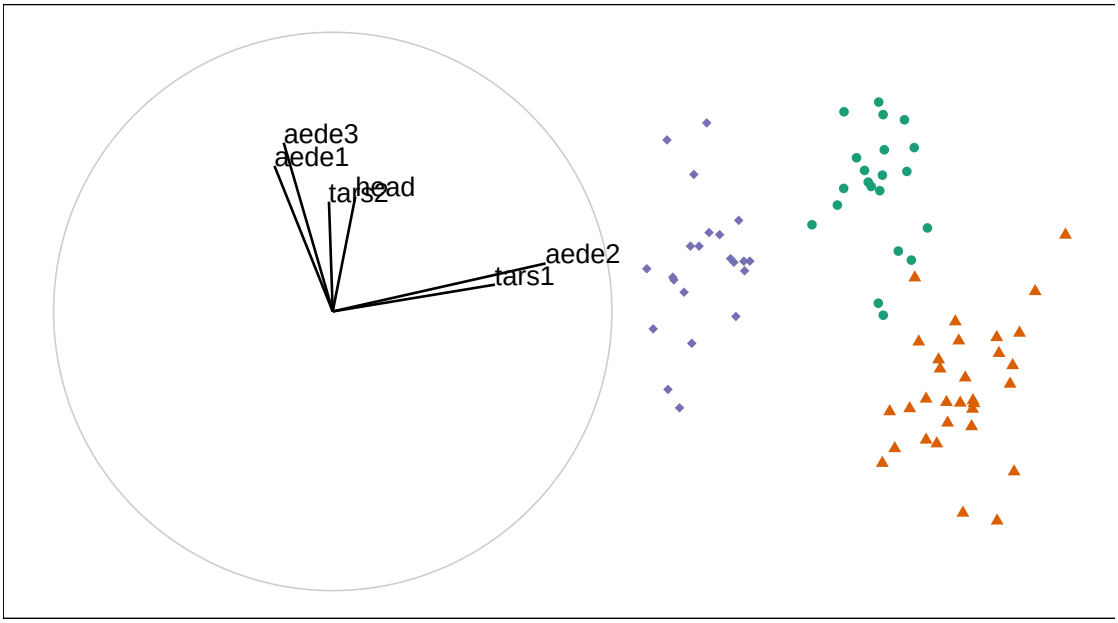


Figure B.1: *Basis reference frame (left) and projected data (right) of standardized flea data. Basis identified by holes-index guided tour. The variables 'aede2' and 'tars1' contribute mostly in the x direction, whereas the other variables contribute mostly in the y direction. We'll select 'aede2' as our manipulation variable to see how the structure of the projection changes as we rotate 'aede2' into and out of the projection.*

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 primarily tars1 and aede2 contrasting the other four variables. In the projected data 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 figure B.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 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.

B.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.

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

In figure B.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.

B.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.

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

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

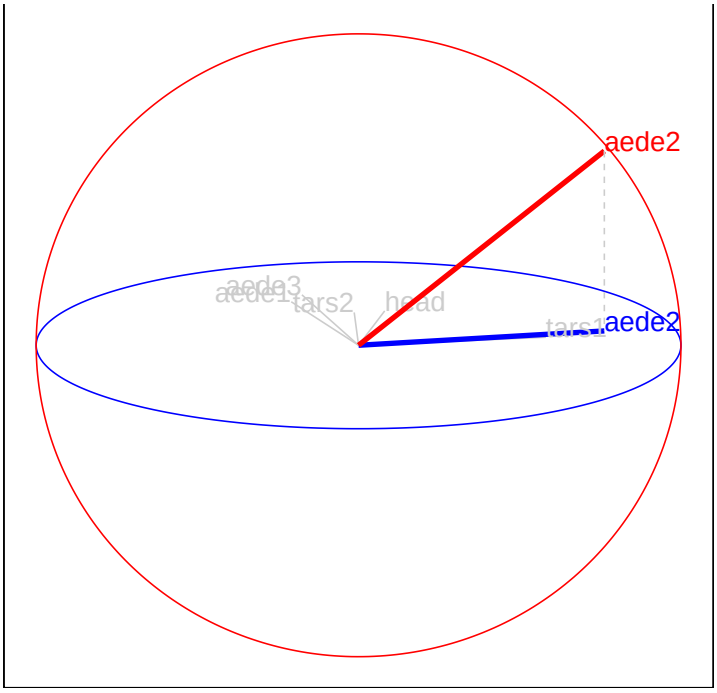


Figure B.2: *Manipulation space for controlling the contribution of aede2 of standardized flea data. Basis was identified by holes-index guided tour. The out of plane axis, in red, shows how the manipulation variable can be rotated, while other dimensions stay embedded within the basis plane.*

```
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. 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 B.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. For a radial tour, fix ϕ, the angle within the blue plain, and vary the θ, 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.

B.3.5 Step 3 Generate rotation

Define a set of values for ϕ_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 B.2.

For i **in 1 to n_slides:**

For each ϕ_i, post multiply the manipulation space by a rotation matrix, producing, **RM**, the rotated manip space.

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

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

For the $d = 2$ case:

$$* \begin{bmatrix} c_\theta^2 c_\phi s_\theta^2 & -c_\theta s_\theta (1 - c_\phi) & -c_\theta s_\phi \\ -c_\theta s_\theta (1 - c_\phi) & s_\theta^2 c_\phi + c_\theta^2 & -s_\theta s_\phi \\ c_\theta s_\phi & s_\theta s_\phi & c_\phi \end{bmatrix}_{[3,\, 3]}$$

Where:

θ is the angle that lies on the projection plane, the *XY*-scatterplot

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

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 the angle relative to the ϕ_1, we find the transformation ϕ_i

- ϕ_1 useful to think about ϕ 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 B.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/, will take a moment to load. This format of this figure and linking to a dynamic version will be used again in section B.5.

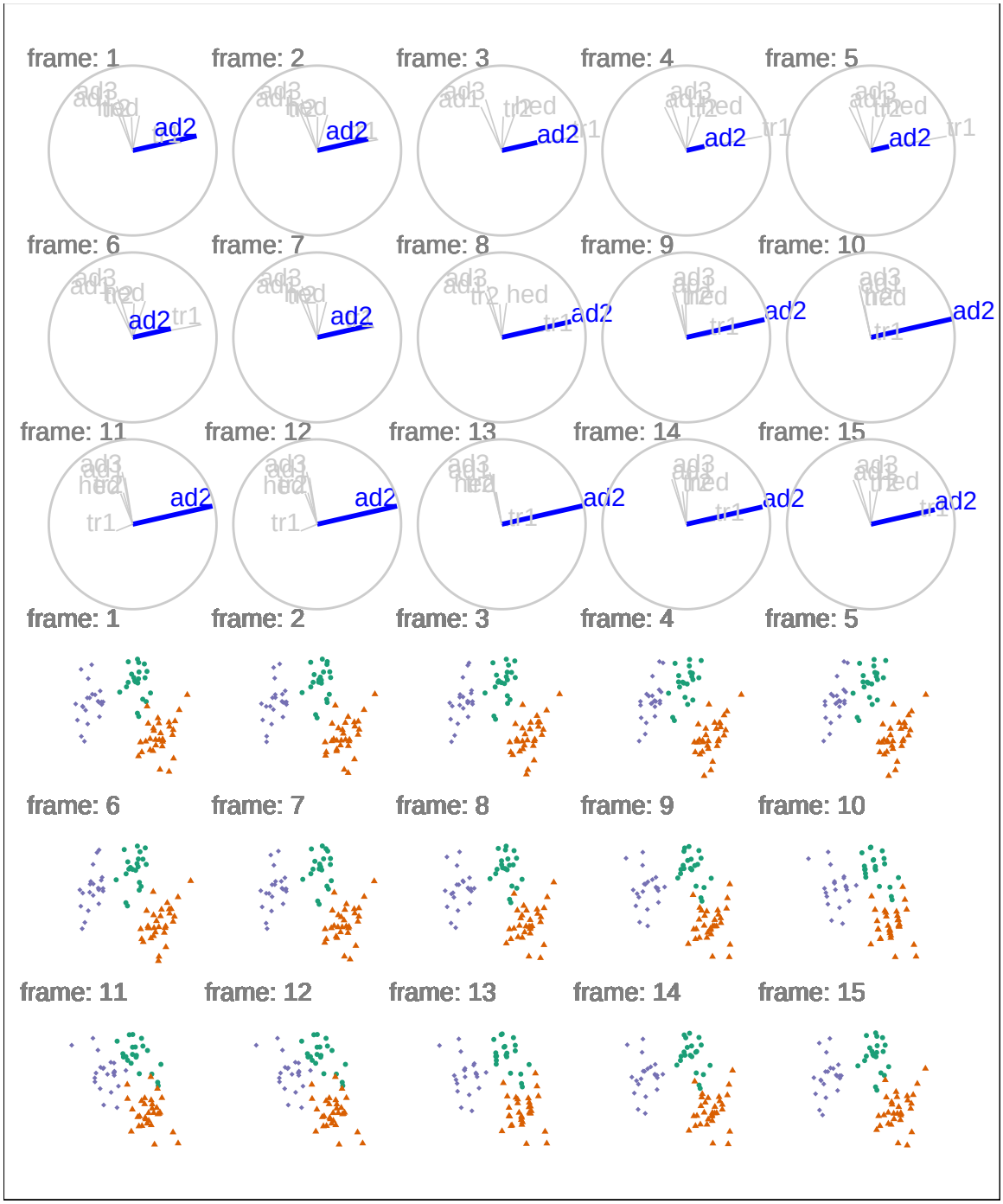


Figure B.3: *Rotated manipulation spaces, a radial manual tour controlling the contribution from* `aded2` *of standardized flea data. The contribution of* `aede2` *extends from its initial contribution to a full contribution to the projection before decreasing to zero and then returning to its initial state. A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/flea_manualtour_mvar5/`.

B.4 Display projection sequence

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

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

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

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`, 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 the contained object, such as tooltips, brushing, and linking without communicating back to the R console.

B.4.1 Storage and sharing

Storing each data point for every frame with the overhead dynamic graphics 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 ϕ is at its starting, minimum, and maximum values. The frames in between can be interpolated by supplying angular speed. By using the `tourr::save_history()` we can do just that. Save such tour path history and a single set of data offers performant storage and transfer.

B.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 3D subspaces of the 10 first principal components 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 projection) 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. 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.

B.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 B.4 and figure B.5 respectively. Manipulating PC3, where varying the angle of rotation brings interesting

features into 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.

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

B.5.2 DIS cluster

We perform a manual tour on this data, manipulating PC6 as depicted in figure B.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 takes a bit to load, as the animation is several megabytes.

This is a different story than if we had selected a different variable to manipulate. In figure B.7 we manipulate PC2.

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

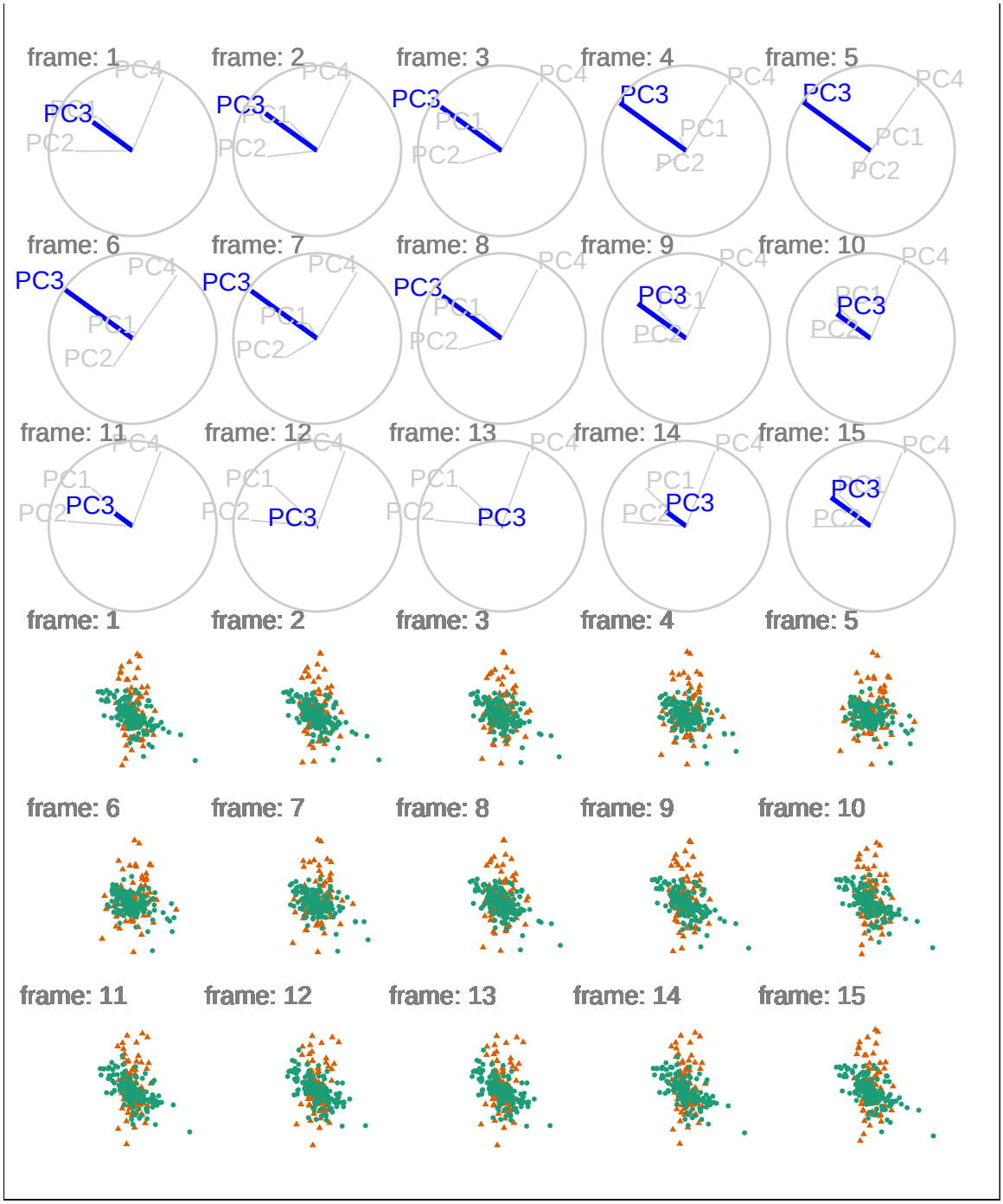


Figure B.4: *Jet cluster, a radial manual tour of PC3. Colored by experiment type: 'ATLAS7new' in green and 'ATLAS7old' in orange. When PC3 fully contributes to the projection ATLAS7new (green) occupies unique space and several outliers are identifiable. Zeroing the contribution from PC3 to the projection 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/`.

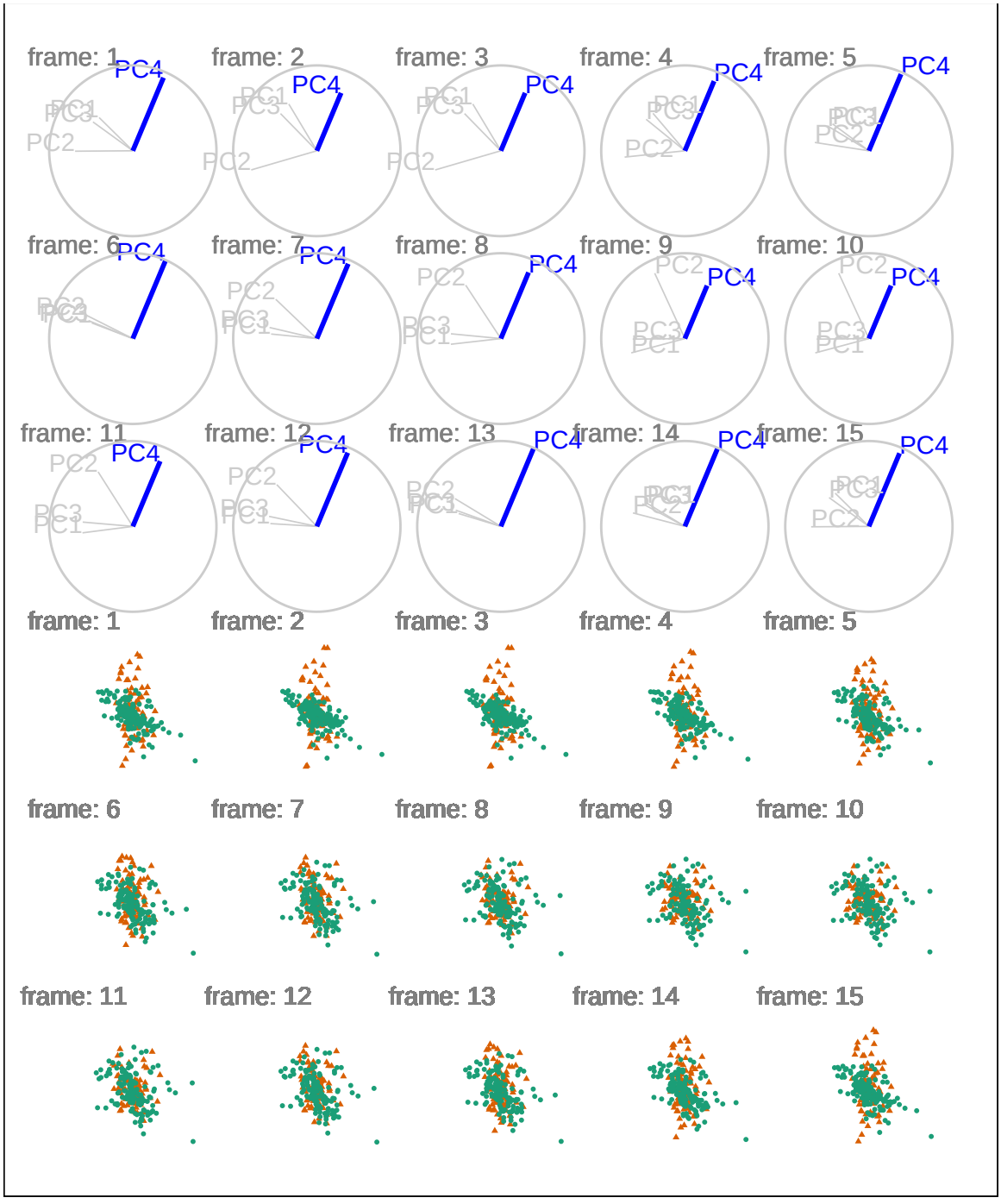


Figure B.5: *Jet cluster, a radial manual tour of PC4. Colored by experiment type: 'ATLAS7new' in green and 'ATLAS7old' in orange. This tour contains less interesting information ATLAS7new (green) has points that are right and left of ATLAS7old, while most points occupy the same projection space, regardless of the contribution of PC4. A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc3/`.

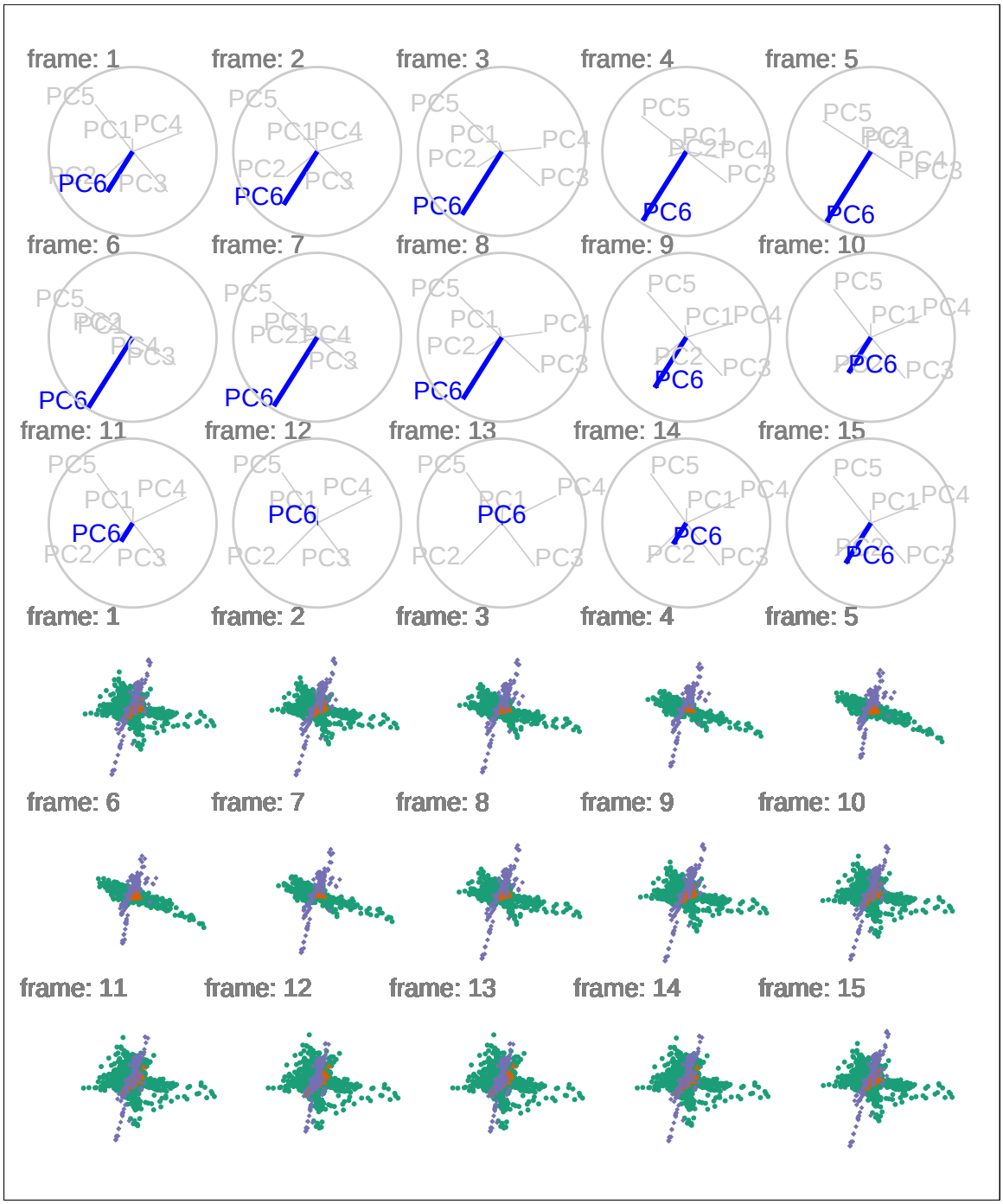


Figure B.6: *DIS cluster, a radial manual tour of PC6. colored by experiment type: 'DIS HERA1+2' in green, 'dimuon SIDIS' in purple, and 'charm SIDIS' in orange. When the contribution PC 6 is large we see that dimuon SIDIS (purple) data are nearly orthogonal to DIS HERA (green) data. As the data is rotated,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

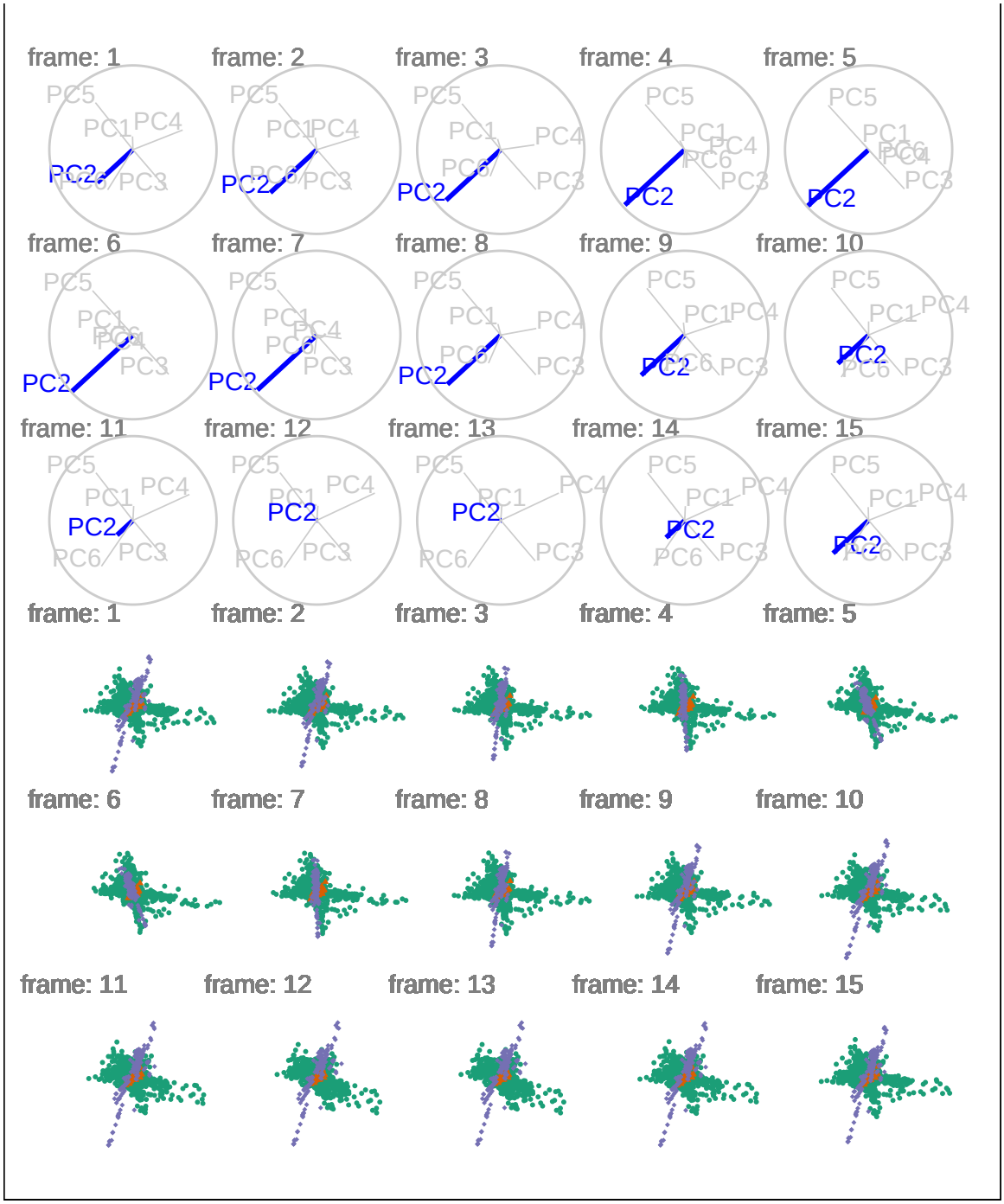


Figure B.7: *DIS cluster, a radial manual tour of PC2. Colored by experiment type: 'DIS HERA1+2' in green, 'dimuon SIDIS' in purple, and 'charm SIDIS' in orange. The structure of previously described plane of DIS HERA (green) and nearly orthogonal dimuon SIDIS (purple) is present, however, the manipulating 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/`.

B.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")
```

B.7 Discussion

This work 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 $p-$ dimensional geoms, such as 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 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 *tourr* package provides several 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 would require a dimensionally-generalized rotation matrix.

The 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 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

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).

Arms, L, D Cook, and C Cruz-Neira (1999). The benefits of statistical visualization in an immersive environment. In: *Virtual Reality, 1999. Proceedings., IEEE*. IEEE, pp.88–95.

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, DB and WL Nicholson (1988). 'Explor4: A Program for Exploring Four-Dimensional Data Using Stereo-Ray Glyphs, dimensional constraints, rotation, and masking. *Cleveland and McGill (1988)*, 309–329.

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

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.

Cordeil, M (2019). *Immersive Analytics Toolkit*. original-date: 2017-02-16T05:25:32Z. https://github.com/MaximeCordeil/IATK (visited on 02/04/2019).

Cordeil, M, A Cunningham, T Dwyer, BH Thomas, and K Marriott (2017). ImAxes: Immersive axes as embodied affordances for interactive multivariate data visualisation. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, pp.71–83.

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

Friedman, J and J Tukey (1974). A Projection Pursuit Algorithm for Exploratory Data Analysis. en. *IEEE Transactions on Computers* **C-23**(9), 881–890. (Visited on 06/22/2018).

Gracia, A, S González, V Robles, E Menasalvas, and T von Landesberger (2016). New insights into the suitability of the third dimension for visualizing multivariate/multidimensional data: A study based on loss of quality quantification. en. *Information Visualization* **15**(1), 3–30. (Visited on 08/20/2018).

Grimm, K (2017). *mbgraphic: Measure Based Graphic Selection*. https://CRAN.R-project.org/package=mbgraphic (visited on 02/07/2019).

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

Hofmann, H, L Follett, M Majumder, and D Cook (2012). Graphical tests for power comparison of competing designs. *IEEE Transactions on Visualization and Computer Graphics* **18**(12), 2441–2448.

Huber, PJ (1985). Projection Pursuit. en. *The Annals of Statistics* **13**(2), 435–475.

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).

Karwowski, W (2006). *International Encyclopedia of Ergonomics and Human Factors, -3 Volume Set*. CRC Press.

Laa, U and D Cook (2019). Using tours to visually investigate properties of new projection pursuit indexes with application to problems in physics. *arXiv:1902.00181 [physics, stat]*. arXiv: 1902.00181. (Visited on 02/04/2019).

Lee, EK and D Cook (2010). A projection pursuit index for large p small n data. en. *Statistics and Computing* **20**(3), 381–392. (Visited on 02/13/2019).

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).

Lee, JM, J MacLachlan, and WA Wallace (1986). The effects of 3D imagery on managerial data interpretation. *MIS Quarterly*, 257–269.

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

Marriott, K, F Schreiber, T Dwyer, K Klein, NH Riche, T Itoh, W Stuerzlinger, and BH Thomas (2018). *Immersive Analytics*. en. Google-Books-ID: vaVyDwAAQBAJ. Springer.

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.

Munzner, T (2014). *Visualization analysis and design*. AK Peters/CRC Press.

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.

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

Scott, DW (1985). Averaged shifted histograms: effective nonparametric density estimators in several dimensions. *The Annals of Statistics*, 1024–1040.

Scott, DW (1995). Incorporating density estimation into other exploratory tools. In: *ASA Proceedings of the Section on Statistical Graphics', American Statistical Association, Alexandria, VA*. Citeseer, pp.28–35.

Sedlmair, M, T Munzner, and M Tory (2013). Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization & Computer Graphics* (12), 2634–2643.

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.

Sutherland, P, A Rossini, T Lumley, N Lewin-Koh, J Dickerson, Z Cox, and D Cook (2000). Orca: A Visualization Toolkit for High-Dimensional Data. *Journal of Computational and Graphical Statistics* **9**(3), 509–529. (Visited on 01/10/2019).

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.

Tory, M, AE Kirkpatrick, MS Atkins, and T Moller (2006). Visualization task performance with 2D, 3D, and combination displays. *IEEE transactions on visualization and computer graphics* **12**(1), 2–13.

Tukey, JW (1977). *Exploratory data analysis*. Vol. 32. Pearson.

Wagner Filho, J, M Rey, C Freitas, and L Nedel (2018). Immersive Visualization of Abstract Information: An Evaluation on Dimensionally-Reduced Data Scatterplots. In:

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*.

Ware, C (2000). Designing with a 2$1/2$D attitude. *Information Design Journal* **10**(3), 258–265.

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

Wegman, E, W Poston, and J Solka (2001). *Pixel Tours*. University of Minnesota. https://ima.umn.edu/2001-2002/W11.12-15.01/18492 (visited on 01/10/2019).

Wickens, CD, DH Merwin, and EL Lin (1994). Implications of graphics enhancements for the visualization of scientific data: Dimensional integrality, stereopsis, motion, and mesh. *Human Factors* **36**(1), 44–61.

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).

Wickham, H and G Grolemund (2016). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. en. Google-Books-ID: I6y3DQAAQBAJ. "O'Reilly Media, Inc.".

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.