Mathematics of the tour

Hadley Wickham
ssistant Professor / Dobelman Family Junior Cha
Department of Statistics / Rice University

1. Projection and orientation
2. Motion and interpolation
3. Basis generation

A. Buja, D. Cook, D. Asimov, and C. Hurley. *Theory of dynamic projections in high-dimensional data visualization*. http://www-stat.wharton.upenn.edu/~buja/PAPERS/paper-dyn-proj-math.pdf. 2004.

Projection

We have a n x p data matrix that we want to project into a d-dimensional view

We accomplish this by multiplying by a p x d **projection matrix**.

This matrix needs to be **orthonormal**: the columns should be independent and have length 1. **Why?**

Example

Projects from 4d to 2d.

Read down to see the linear combination that makes each projected dimension.

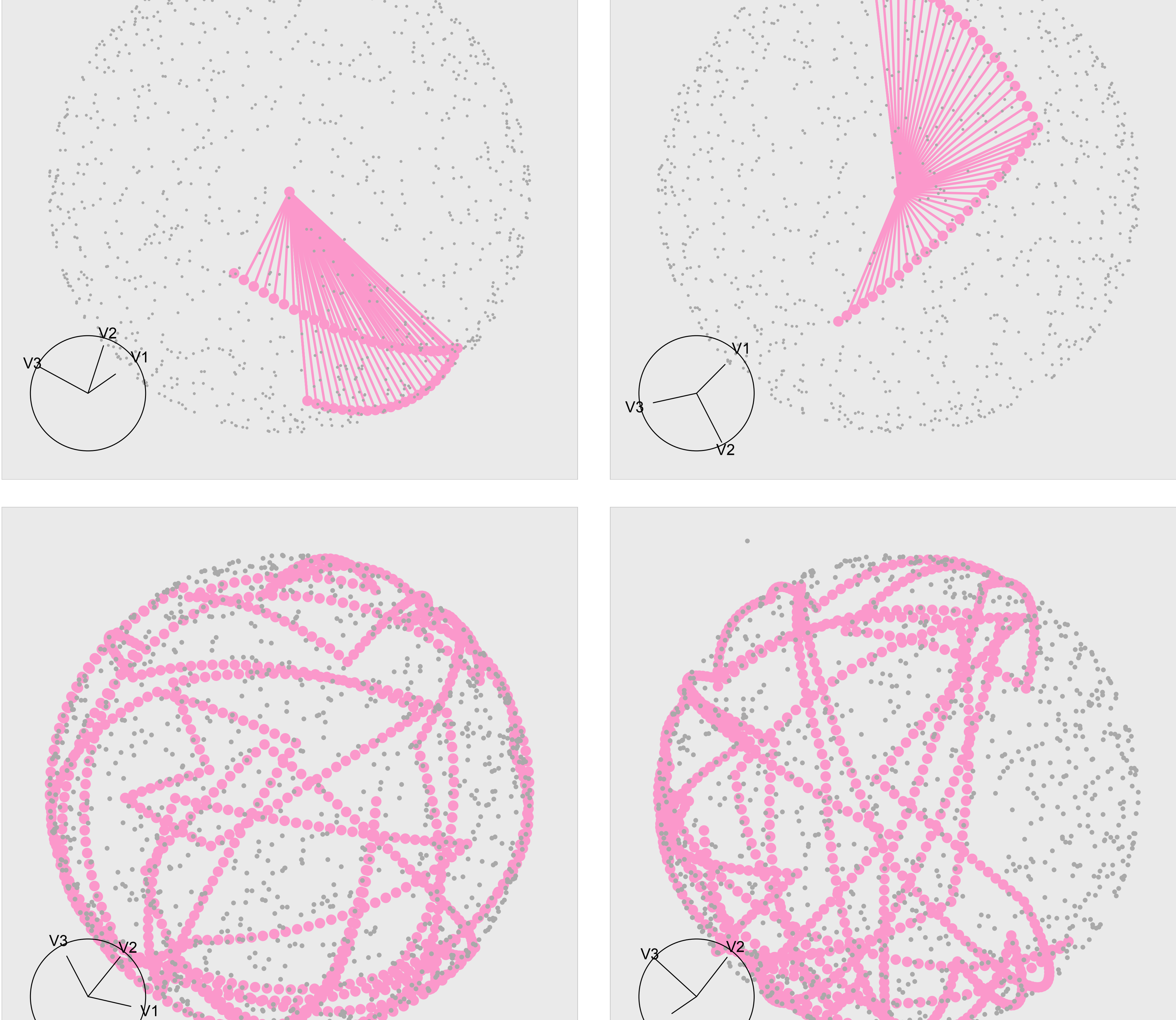
These values define the coordinates of the axes.

0.28	-0.85
-0.19	-0.47
-0.91	-0.20
0.24	-0.11

1d projections

Normality constraint implies that projections can be thought of points on a surface of a sphere.

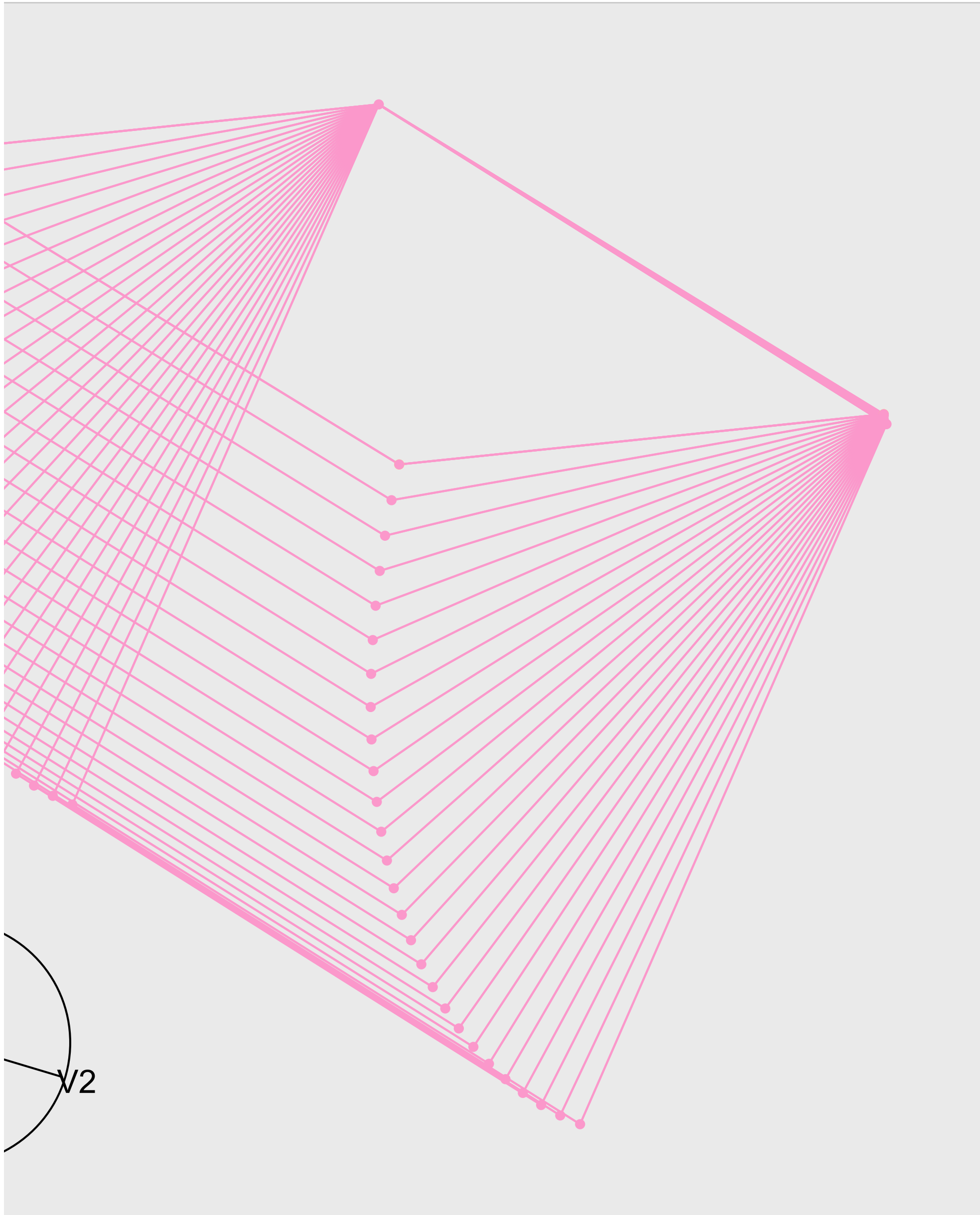
-0.85
-0.34
0.40

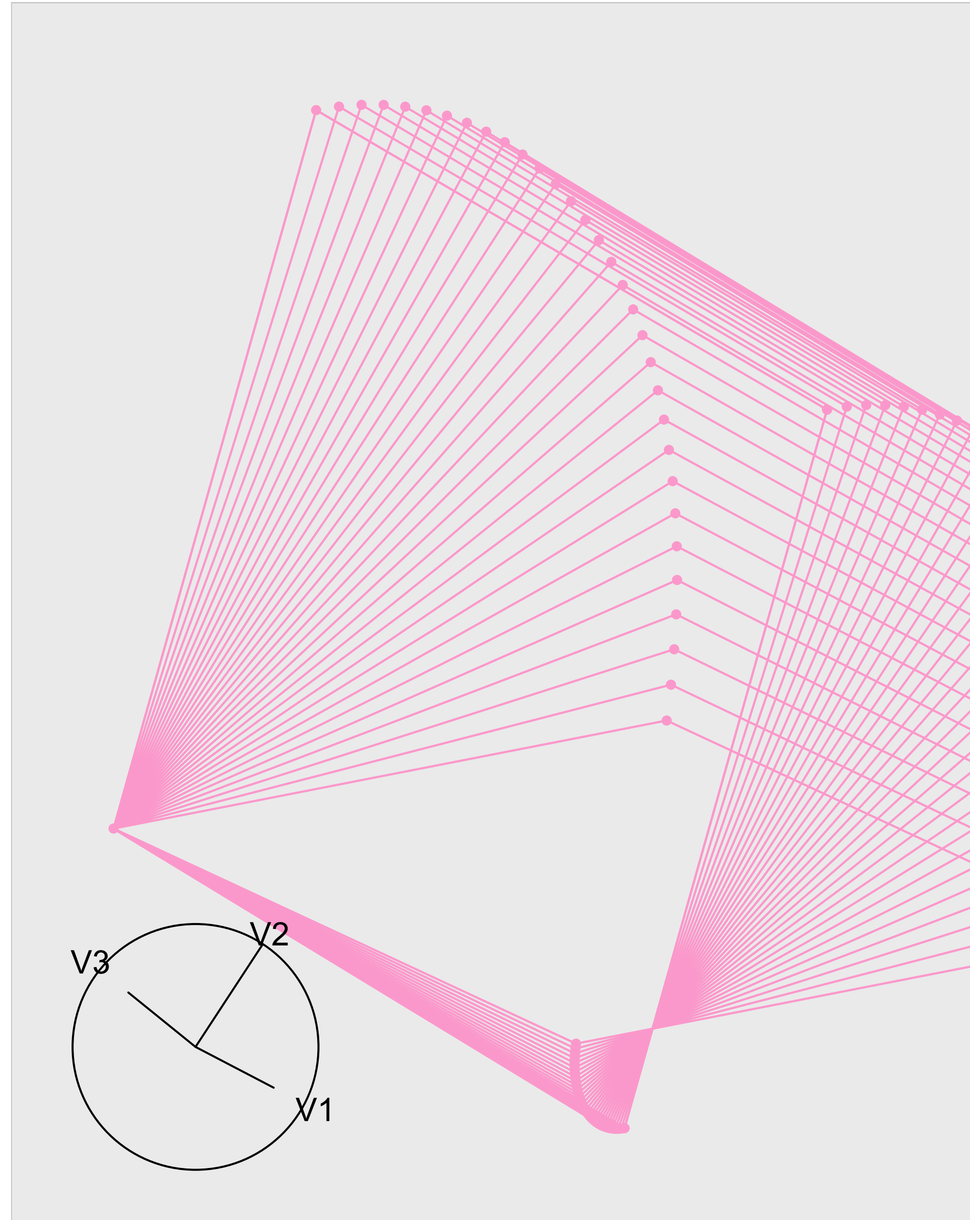

V3
V2
V1
V1
V3
V2
V3
V2
V1
V3
V2
V1

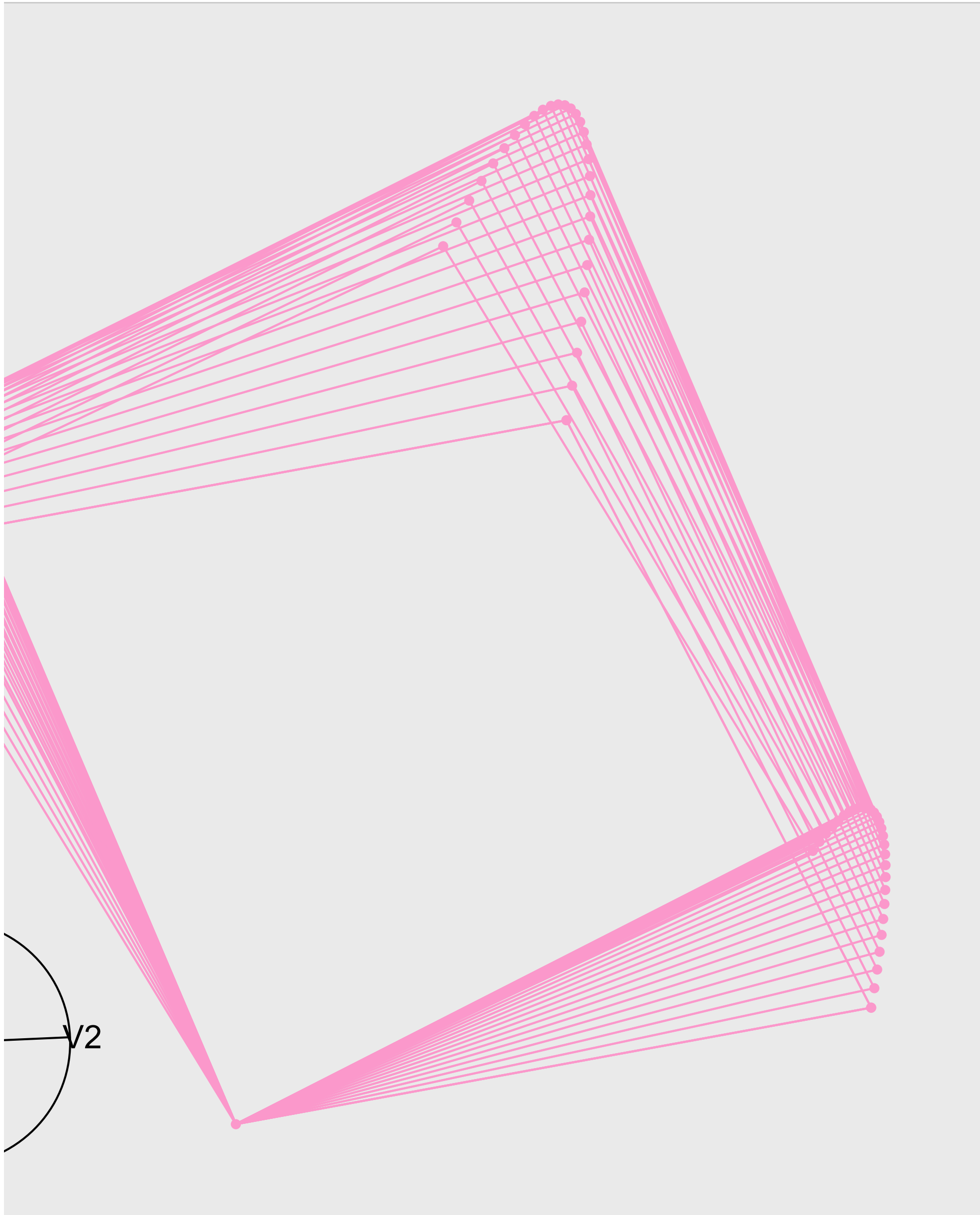
Orientation

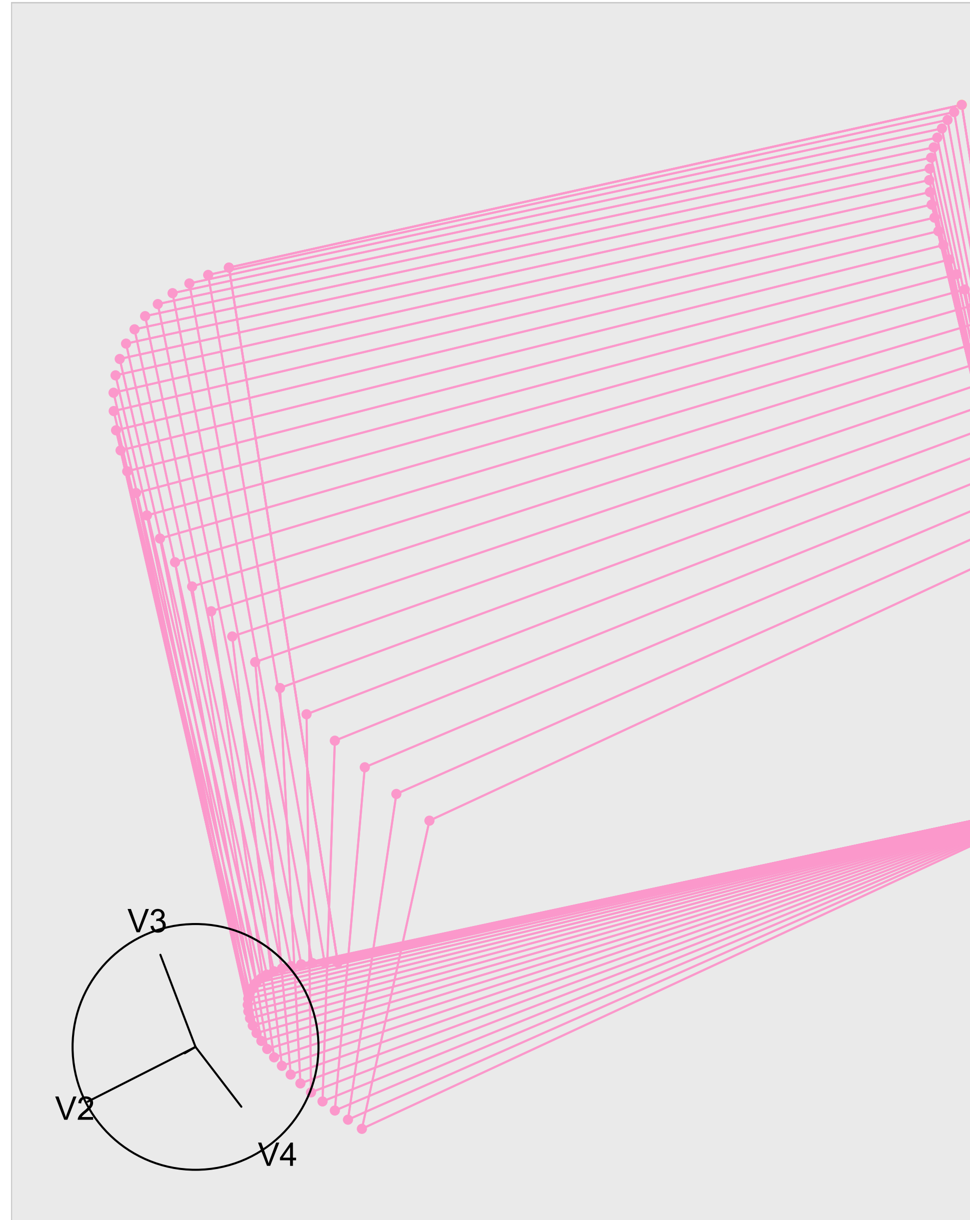
For many visual tasks, orientation doesn't matter - i.e. if you rotate a scatterplot, outliers will be just as obvious.

When orientation is important, we talk about a **frame** (or basis). When it isn't important, we talk about a **plane** (span of the basis).


V2


V3
V2
V1


V2


V3
V2
V4

Motion

Even if orientation doesn’t matter, still need to work with frames, because some orientation is needed for rendering.

Frame-to-frame motion =
pure plane motion +
within-plane (**whip**) spin

Want to minimise whip spin because it’s distracting.

Interpolation

Current frame = F_a. Target frame = F_z.

When orientation doesn't matter, can pick any frame in target **plane**.

So pick a target frame that minimises whip-spin.

What do we see?

Flea beetles example.

Motion adds an additional 2-dimensions velocity in the x and y directions.

Not as easy to perceive as position, but still useful.

```
ary(tourr)

ies <- as.numeric(flea$species)
$species <- NULL

ate_xy(flea, grand_tour(), col = species)
ate_trails(flea, grand_tour(), col = species
```

Velocity

We also want the tour to have constant velocity.

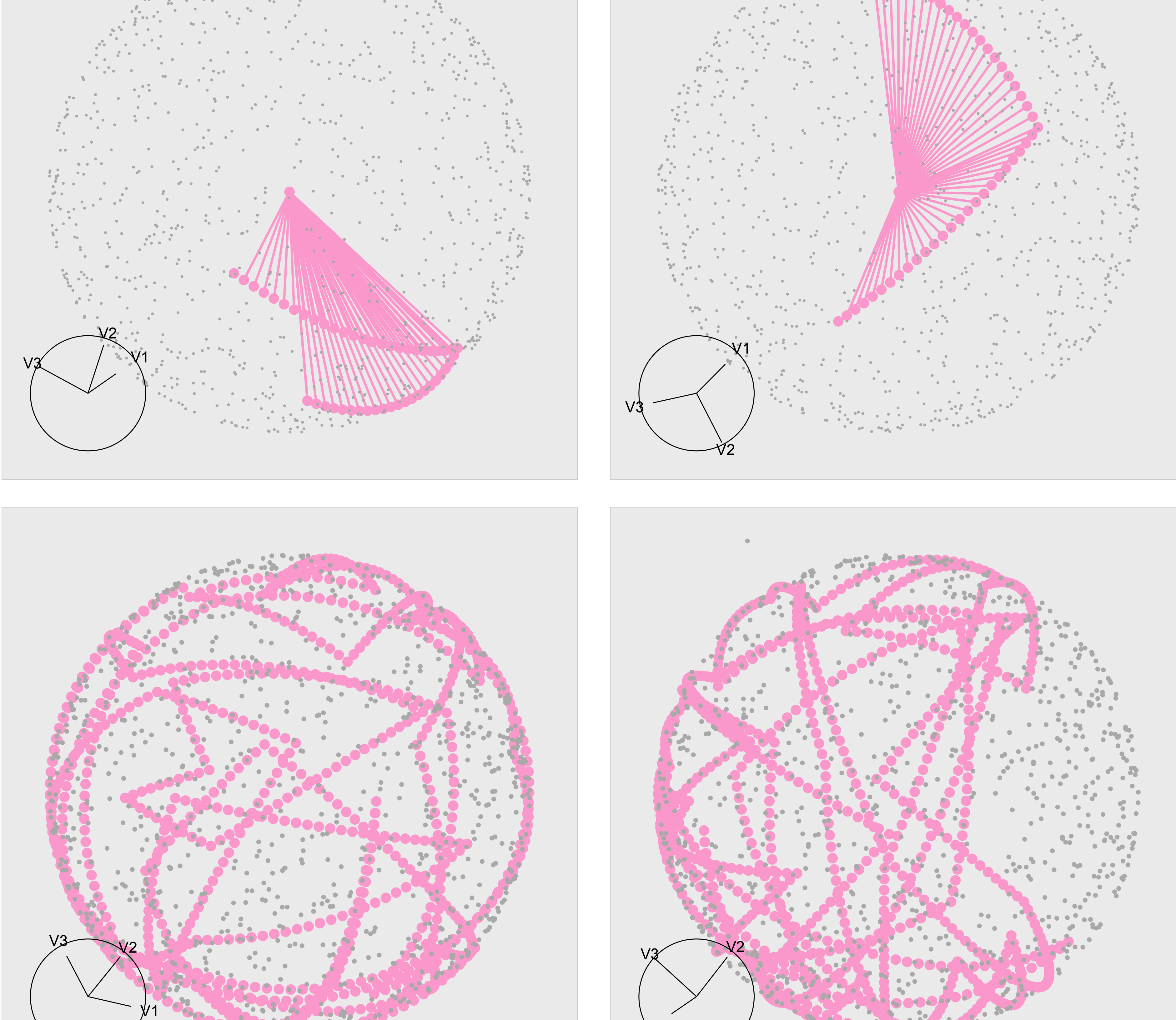
This uniquely defines the path of intermediate projections from the current to the target.

Basically constant angular velocity.

Basis generation

Turns out to be pretty easy!

Take multivariate normal, normalise and orthogonalise - gives uniform distribution on space of projections.


V1
V2
V3
V1
V2
V3
V1
V2
V3
V2
V3