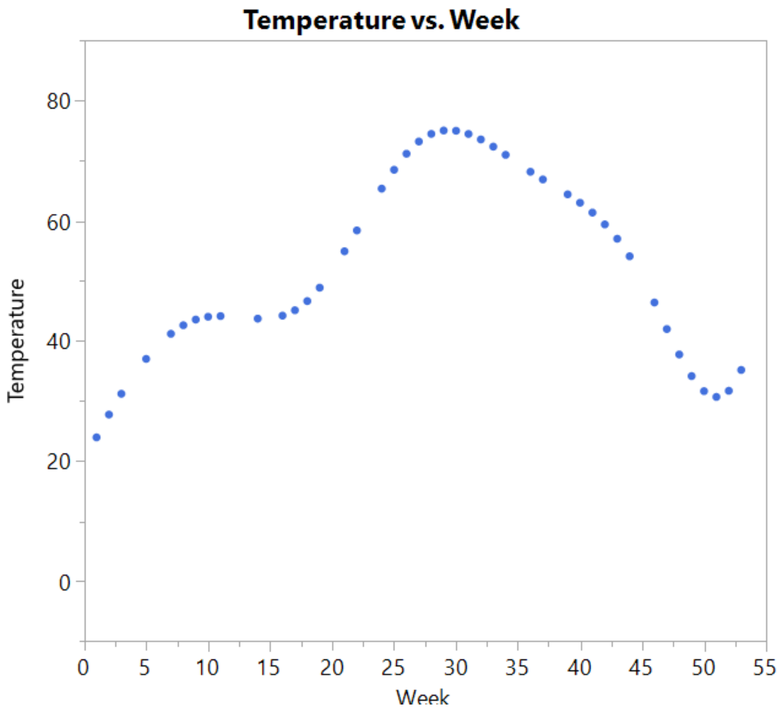
Analyzing functional data with Direct Functional PCA in Functional Data Explorer

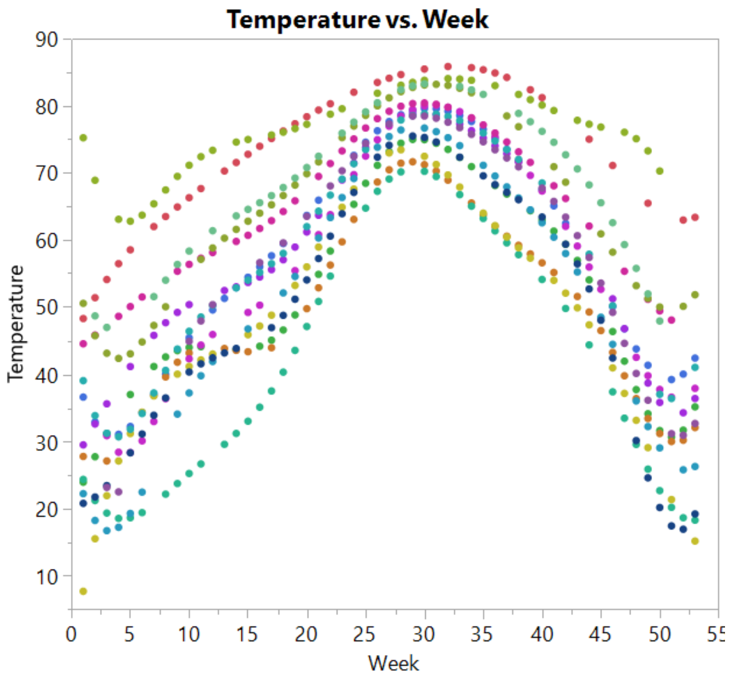
Ryan Parker
Sr. Research Statistician Developer
JMP

JMP STATISTICAL DISCOVERY FROM SAS

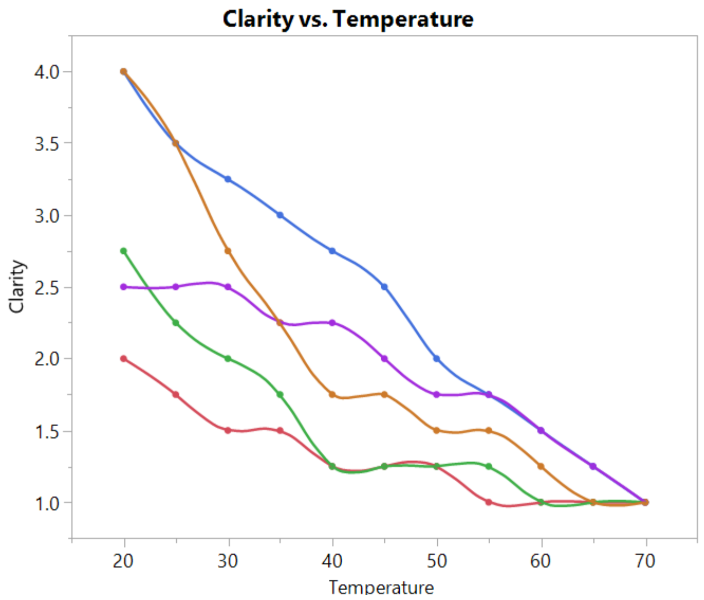
Functional Data



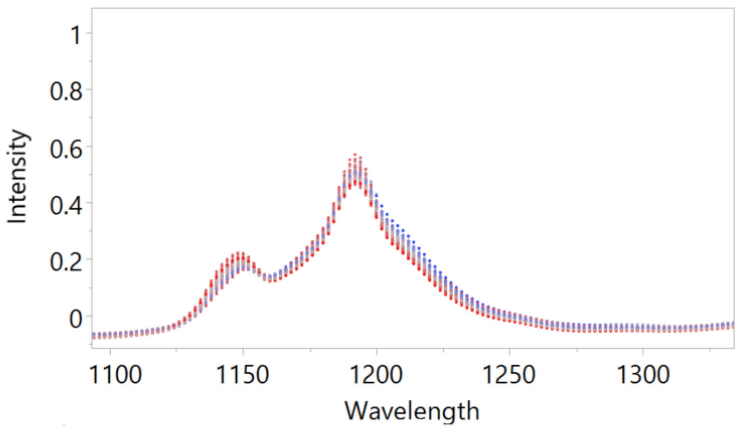
Functional Data



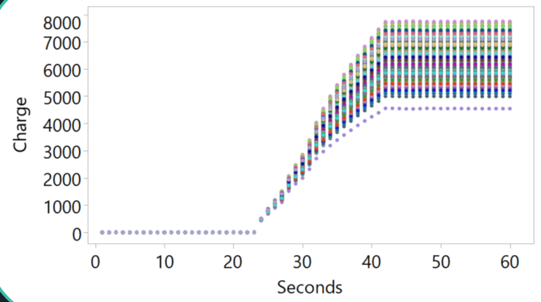
Functional Data

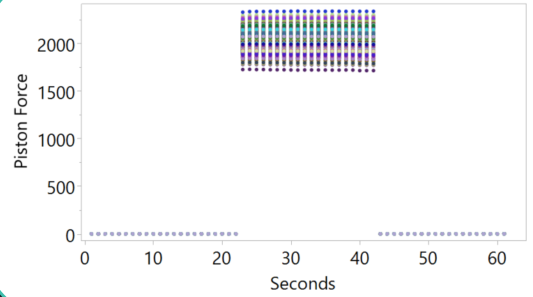


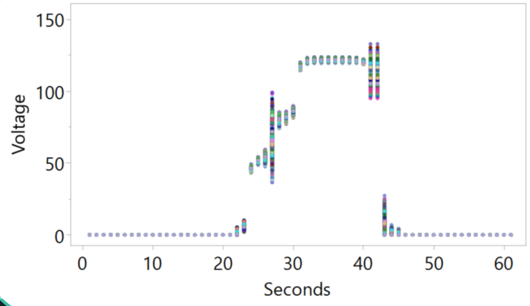
Functional Data

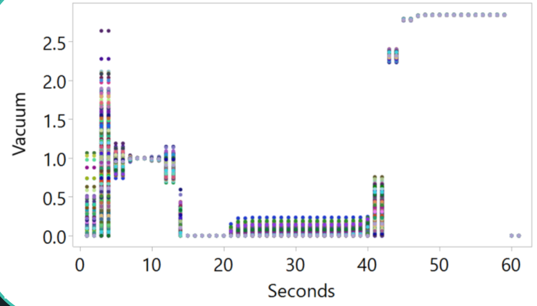


Functional Data







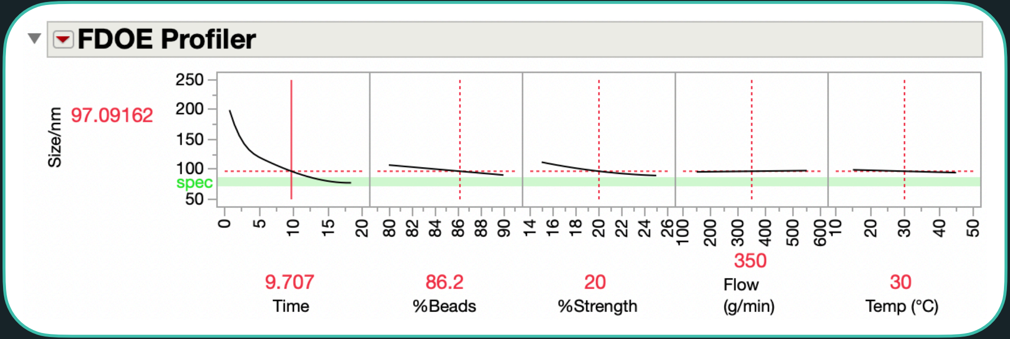


jmp STATISTICAL DISCOVERY FROM SAS

Functional Data Explorer

Functional DOE

- How can we use DOE factors to predict a response function?

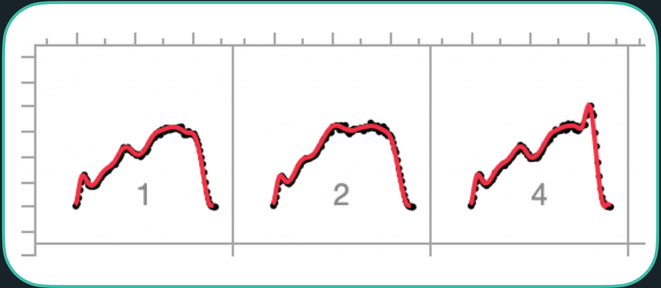


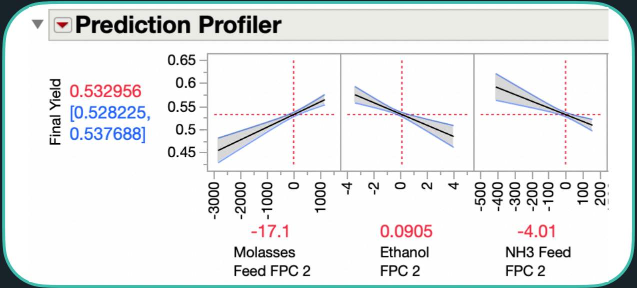
- Want our process to remain in specification as long as possible.

Functional Data Explorer

Functional Machine Learning

- How can we predict the outcome of a process using functions as inputs?





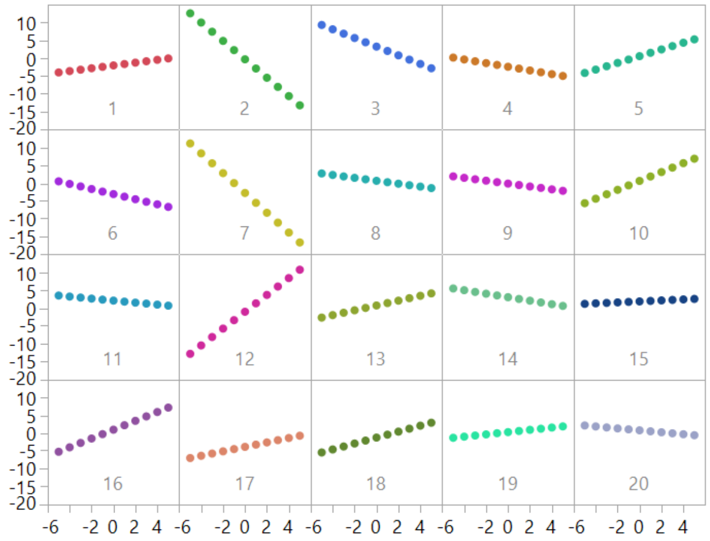
- Predict ***Final Yield*** of a fermentation process using sensor streams.

jmp STATISTICAL DISCOVERY FROM SAS

Functional Data Explorer

Functional PCA

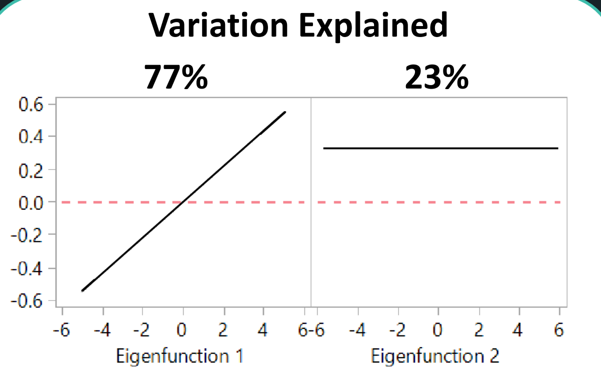
- FPCA decomposes data into orthogonal eigenfunctions
- Explains as much function-to-function variation as possible
- Allows us to extract summaries for predictive modeling

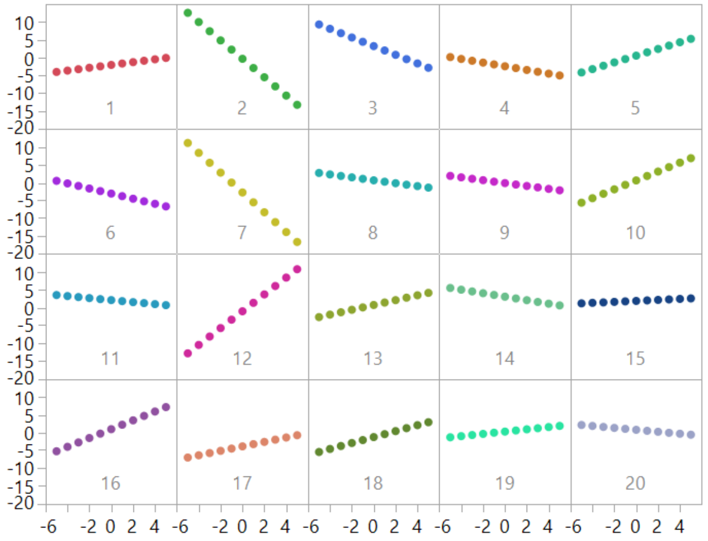


Functional PCA

Eigenfunctions

- Eigenfunction plots highlight features in the functional data

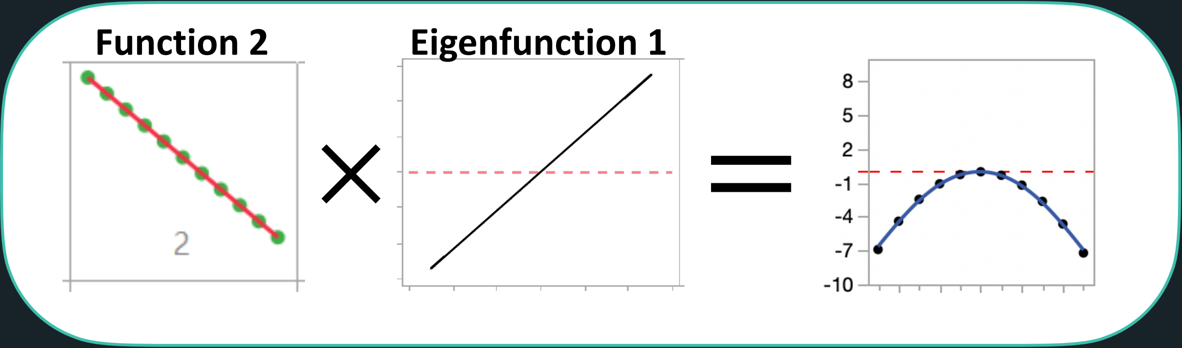




Functional PCA

Functional Principal Components

- FPCs summarize differences between functions



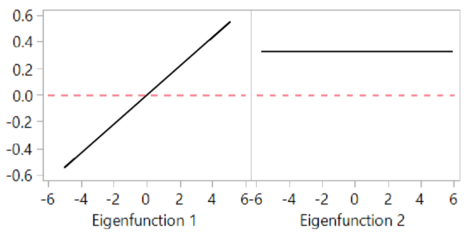
ID	FPC 1	FPC 2
1	3.63	-6.41
2	-23.5	-1.14
3	-11.1	10.15
4	-4.64	-7.63
5	8.58	1.81
6	-6.48	-9.73
7	-25.2	-8.81
8	-3.74	2.08
9	-3.68	-0.41
10	11.35	2.03

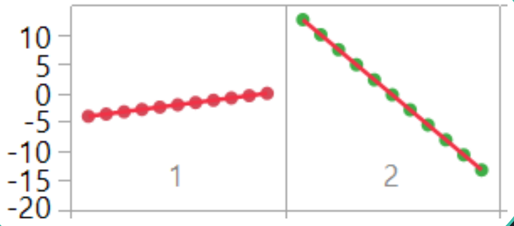
$$FPC_{ij} = \int_X Y_i(x) \times E_j(x)$$

Functional PCA

Function Approximation

ID	FPC 1	FPC 2
1	3.63	-6.41
2	-23.5	-1.14
3	-11.1	10.15
4	-4.64	-7.63
5	8.58	1.81
6	-6.48	-9.73
7	-25.2	-8.81
8	-3.74	2.08
9	-3.68	-0.41
10	11.35	2.03





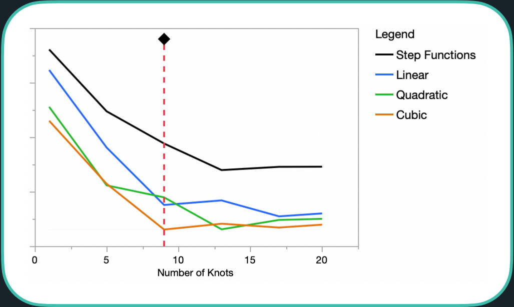
$$Y_1(x) \approx \mu(x) + 3.63 \times E_1(x) - 6.41 \times E_2(x)$$

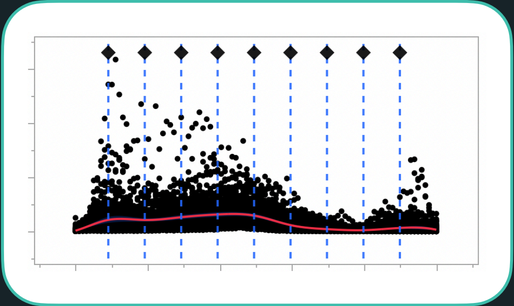
$$Y_2(x) \approx \mu(x) - 23.5 \times E_1(x) - 1.14 \times E_2(x)$$

Direct Functional PCA

Why DFPCA?

- FDE fits basis function models
  - B-splines, P-splines, Fourier
- FPCA on the model coefficients
- For large problems:
  - Unfeasible parameter tuning
  - Computing can be costly

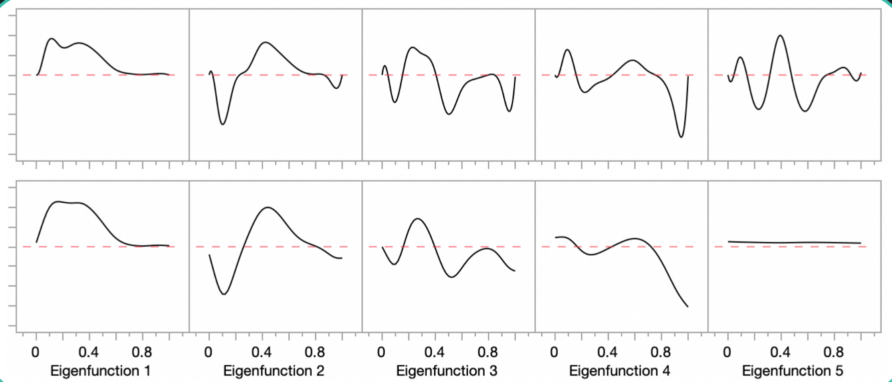




Direct Functional PCA

B-splines

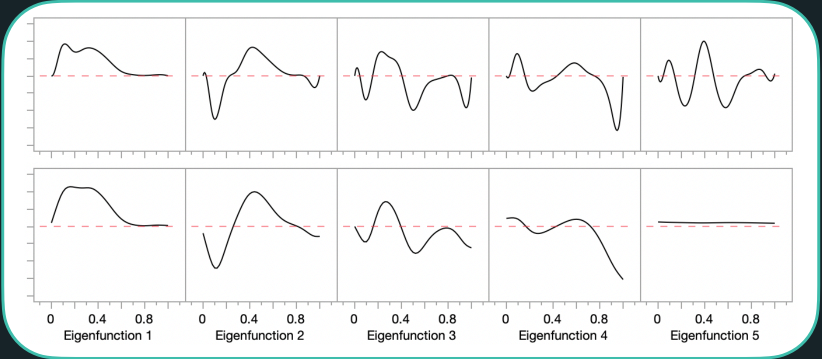
DFPCA

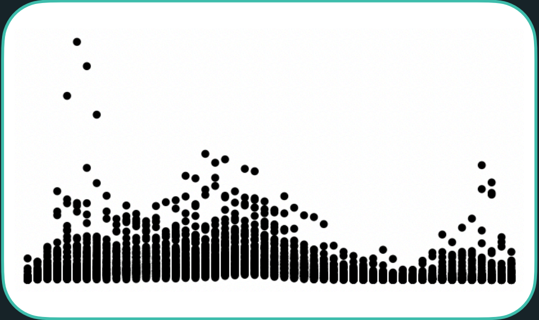


- Operates directly on the data
- Smooths the eigenfunctions

Direct Functional PCA

- Iterative algorithm that identifies one eigenfunction at a time
  - Similar in spirit to Rice and Silverman (1991)
- Data must be on a regular grid
  - We interpolate it for you
  - Can control this with ***Reduce***
- Smoothing simpler eigenfunctions makes DFPCA very fast!

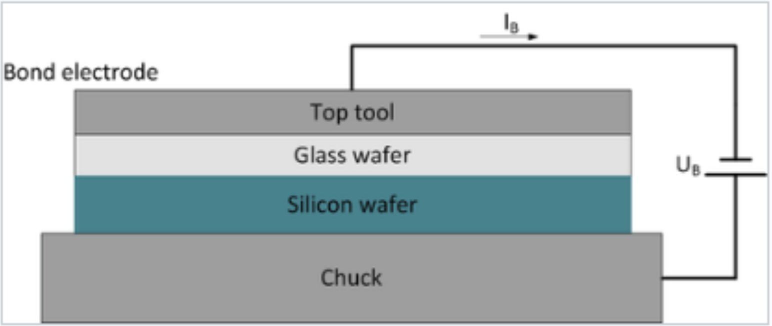




Example

Anodic Bonding

- Semiconductor manufacturing process where glass is bonded to the devices
- Process destroys $\approx 10\,\%$ of wafers, but they are not identified until production process is complete
- **Goal**: use sensor stream functional data to identify wafers to discard early



Scheme of anodic bonding procedure. The top tool works as a cathode and the chuck as anode.

Source: *https://en.wikipedia.org/wiki/Anodic_bonding*

Demo

Direct Functional PCA

Tips on using DFPCA

- Fast computing makes it ideal for larger data sets.
- Data must be gridded
  - Use ***Reduce*** to control the grid
- Diagnostics provided to help identify issues.
  - As with any statistical model, check to make sure it is working well!

Thank you!

JMP STATISTICAL DISCOVERY FROM SAS

jmp.com

References

- Rice, J.A. and Silverman, B.W. (1991). Estimating the mean and covariance structure non parametrically when the data are curves. *Journal of the Royal Statistical Society*, Series B, 53, 233-243.

- Ramsay, J.O and Silverman, B.W. (2005). Functional data analysis, second edition (Springer series in statistics).

Direct Functional PCA

Algorithm

- Direct FPCA algorithm on data matrix M:

  1. Perform Lanczos SVD to capture the i^{th} eigenfunction
  2. Smooth the eigenfunction using penalized splines
  3. Normalize the eigenfunction to have unit norm
  4. Orthogonalize the eigenfunction
  5. Subtract the component from M and go back to step 1