STAT 215A Fall 2021 Week 4

Theo Saarinen

Announcements

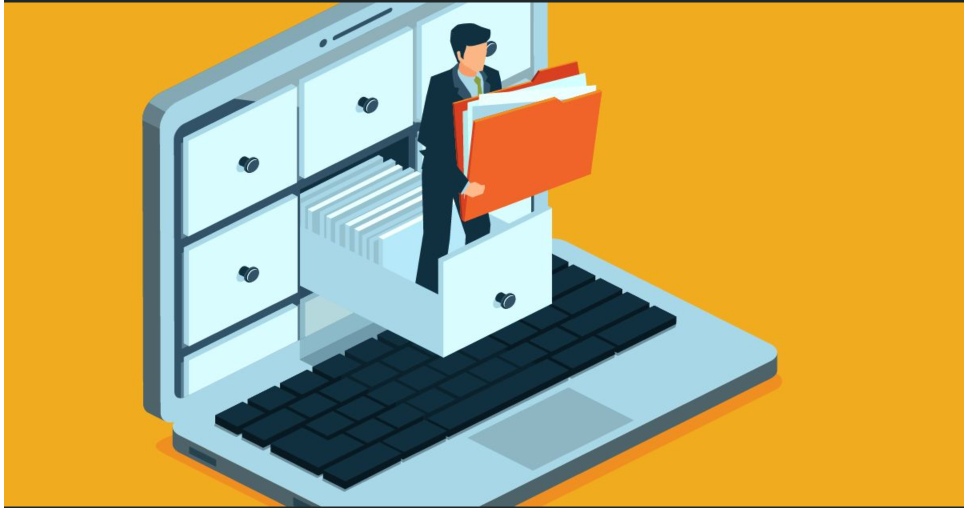
- Lab 1 due next week: **11:59pm Friday Sept. 23**
- I will send out instructions on how to do peer reviews next Friday
    - Completed peer reviews due in one week from the lab deadline at **11:59pm Sunday Sept. 30**
- Lab 2 + Homework 2 will also be released next Friday

Plan for Today:

- PCS documentation
- Kernel density estimation
- Review of PCA
- In-class lab

PCS Documentation

Veridical Data Science
(Karl Kumbier and Bin Yu, 2019)



PCS Documentation

Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

1. Domain question / problem*

PCS Documentation

Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

1. Domain question / problem*
2. Data collection & storage*

PCS Documentation

Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

1. Domain question / problem*
2. Data collection & storage*
3. Data cleaning & preprocessing*

PCS Documentation

Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

1. Domain question / problem*
2. Data collection & storage*
3. Data cleaning & preprocessing*
4. EDA*

PCS Documentation

Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

1. Domain question / problem*
2. Data collection & storage*
3. Data cleaning & preprocessing*
4. EDA*
5. Modeling & post-hoc analysis + PCS inference

PCS Documentation

Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

1. Domain question / problem*
2. Data collection & storage*
3. Data cleaning & preprocessing*
4. EDA*
5. Modeling & post-hoc analysis + PCS inference
6. Interpretation

PCS Documentation

Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

1. Domain question / problem*
2. Data collection & storage*
3. Data cleaning & preprocessing*
4. EDA*
5. Modeling & post-hoc analysis + PCS inference
6. Interpretation

You just did this!

PCS Documentation

Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

1. Domain question / problem*
2. Data collection & storage*
3. Data cleaning & preprocessing*
4. EDA*
5. Modeling & post-hoc analysis + PCS inference
6. Interpretation

You just did this!

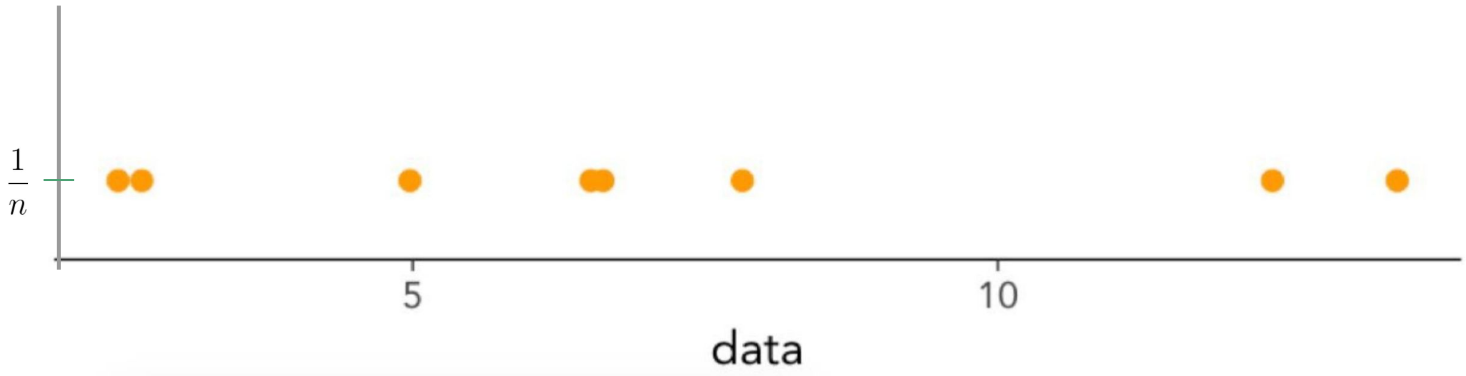
Coming soon...

PCS Documentation Examples

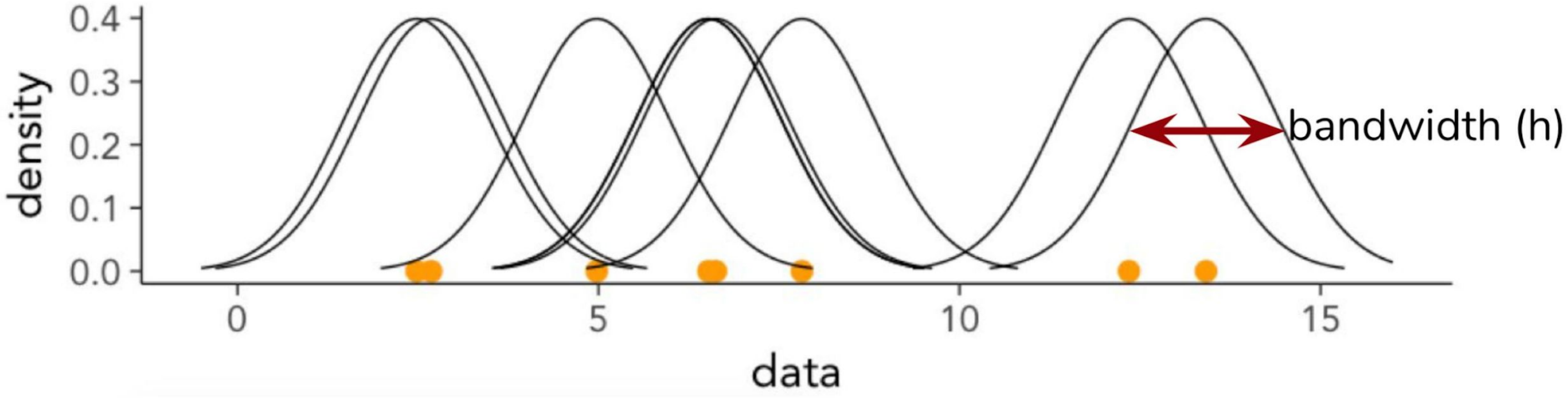
Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

- Ex.: Cancer Cell Line Encyclopedia (Xiao Li, Tiffany Tang and Bin Yu, 2020)
- https://github.com/Yu-Group/stadisc
- This is all about **transparent** and **reproducible** research!

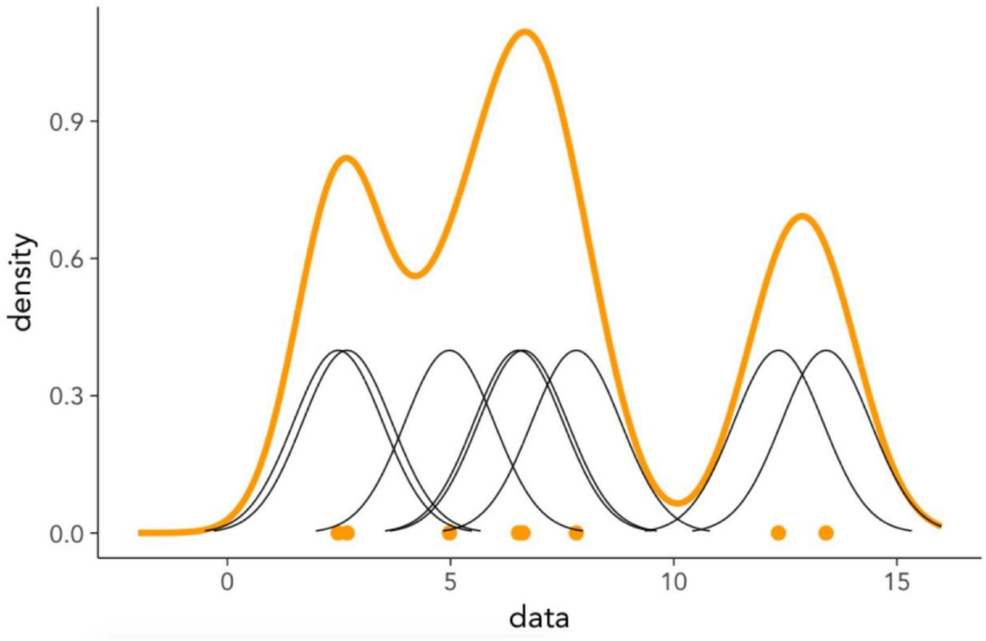
Kernel density estimation



Kernel density estimation



Kernel density estimation



Kernel density estimation

Estimate the density, f, by adding together individual kernel functions

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big)$$

Kernel density estimation

Estimate the density, $\boldsymbol{f}$, by adding together individual kernel functions

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big)$$

Each kernel function is
centered at a data point

Kernel density estimation

The width of the kernel function is defined by the bandwidth $\boldsymbol{h}$

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big)$$

Kernel density estimation

There are many possible kernel functions that you could use:

- Gaussian
- Uniform
- Triangular
- ...

Kernel density estimation

There are many possible kernel functions that you could use:

- Gaussian
- Uniform
- Triangular
- ...

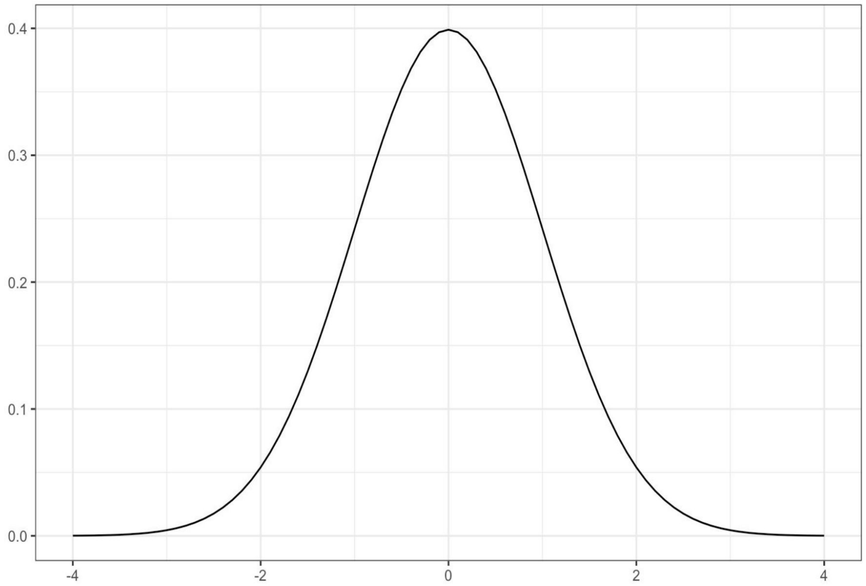
Properties:

- $K(u) \geq 0$
- $\int K(u)du = 1$
- $\int uK(u)du = 0$
- $\int u^2 K(u)du > 0$

Gaussian kernel

Support: $u \in \mathbb{R}$

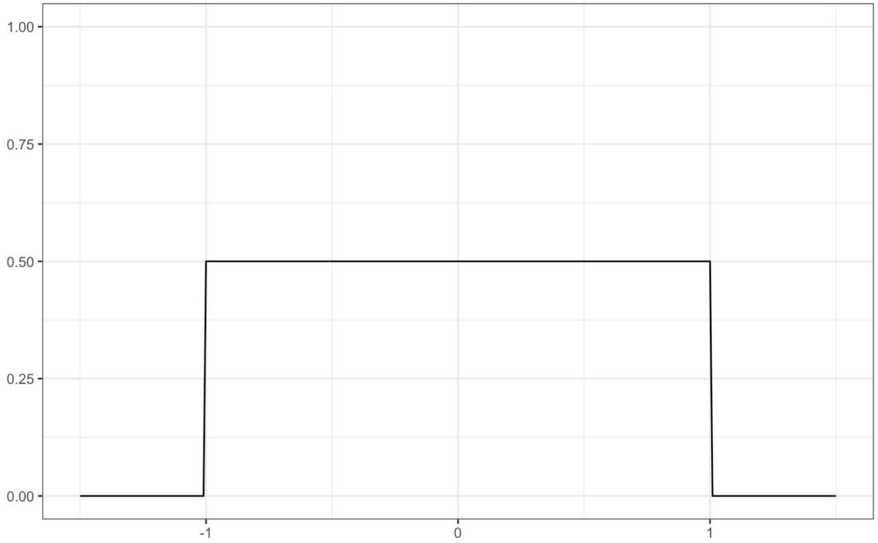
$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$$



Square kernel

Support: $|u| \leq 1$

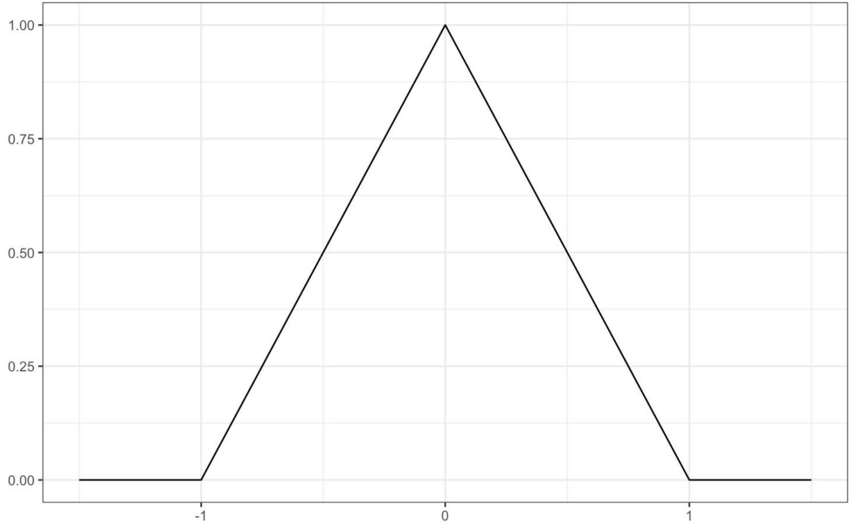
$$K(u) = \frac{1}{2}$$

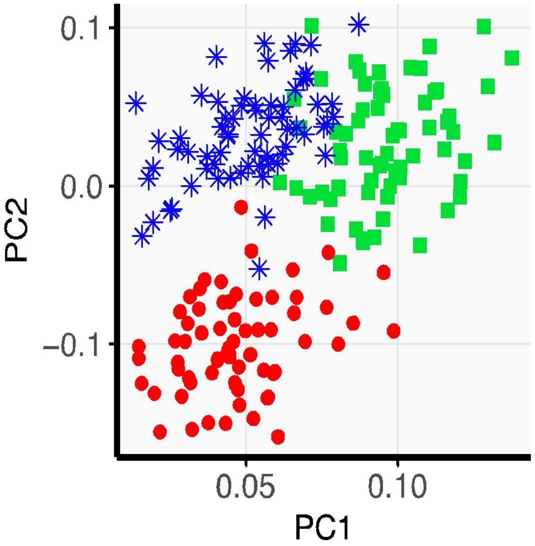


Triangular kernel

Support: $|u| \leq 1$

$$K(u) = 1 - |u|$$



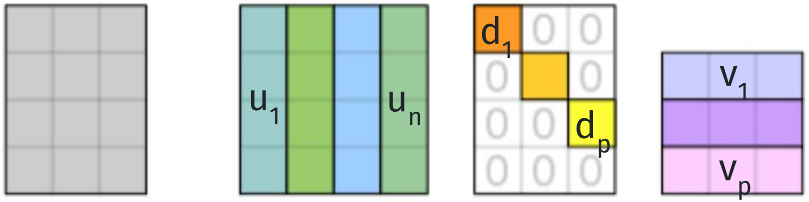


Review of PCA

(Slides in part thanks to Tiffany Tang)

SVD

(assuming n > p)



$$\mathbf{X} = \mathbf{U} \quad \mathbf{D} \quad \mathbf{V}^T$$

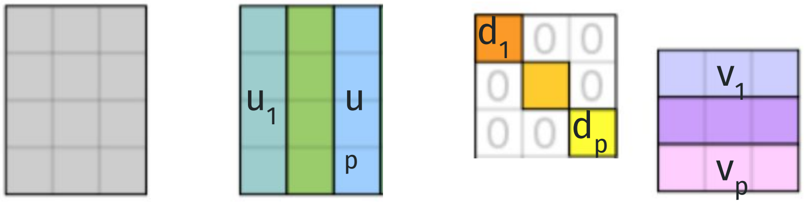
n x p   n x n   n x p   p x p

$$d_1 \geq \cdots \geq d_p$$

$$\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_{nxn}$$

$$\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_{pxp}$$

Economy SVD



$$\mathbf{X} = \tilde{U} \, \tilde{D} \, \mathbf{V}^T$$

n x p = n x p, p x p, p x p

Left singular vectors

Singular values

Right singular vectors

In R: `svd()`

PCA

$$\mathbf{X}^\top\mathbf{X} = \mathbf{V}\tilde{\mathbf{D}}\underbrace{\tilde{\mathbf{U}}^\top\tilde{\mathbf{U}}}_{\mathbf{I}_{p\times p}}\tilde{\mathbf{D}}\mathbf{V}^\top$$

$$= \mathbf{V}\tilde{\mathbf{D}}^2\mathbf{V}^\top$$

Eigenvalues

Eigenvectors

PCA

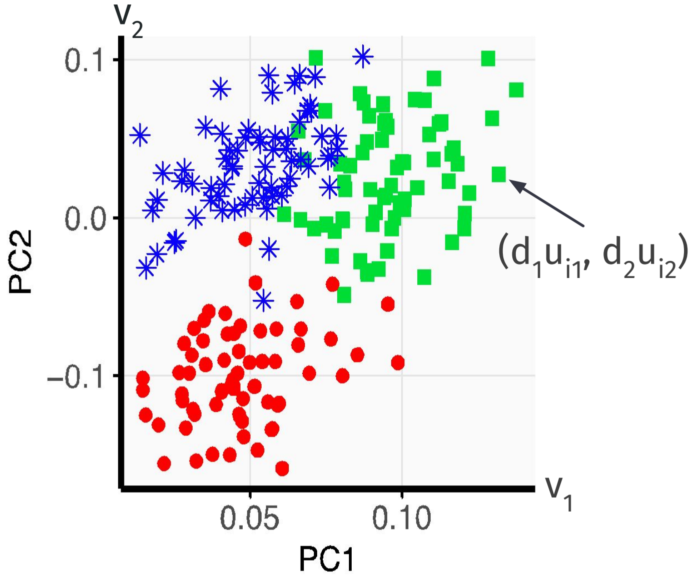
PC directions: dominant feature patterns

$$\mathbf{v}_1 = \text{argmax}_{\mathbf{v}\in\mathbb{R}} \mathbf{v}^\top \mathbf{X}^\top \mathbf{X}\mathbf{v} \quad \text{subject to} \quad \|\mathbf{v}\|_2^2 = 1, \quad \mathbf{v}^\top \mathbf{v}_i = 0 \;\; \forall i \neq j$$

PC scores: dominant observation patterns

$$d_j \, \mathbf{u}_j = \mathbf{X}\, \mathbf{v}_j$$

(projection of data onto directions of maximizing variance)



Proportion of Variance Explained:

$$\frac{\mathbf{v}_j^\top \mathbf{X}^\top \mathbf{X}\, \mathbf{v}_j}{\text{tr}(\mathbf{X}^\top \mathbf{X})} = \frac{d_j^2}{\sum_{i=1}^{p} d_i^2}$$

Practical Considerations for PCA

- PCA is optimal with Gaussian data, but can also work with non-Gaussian data in practice (but not always)
- What to do with categorical data?
  - One-hot encoding

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

- Only need to run PCA once to get all orthogonal, nested components

Image source: https://www.kaggle.com/ayushkaul/using-categorical-data-with-one-hot-encoding

Other Alternatives

- Modifications of PCA:
    - **Sparse PCA:** sparse, interpretable PCs
    - **Kernel PCA:** want non-linear PCs
    - **Functional PCA:** for functional/time series data
    - **Robust PCA:** for grossly corrupted observations
    - Downside: requires additional tuning parameters, which are difficult to tune
- Other methods for dimensionality reduction and pattern recognition
    - **NMF:** https://blog.acolyer.org/2019/02/18/the-why-and-how-of-nonnegative-matrix-factorization/
    - **t-SNE:** https://distill.pub/2016/misread-tsne/
    - **UMAP:** https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668
    - **CCA:** https://en.wikipedia.org/wiki/Canonical_correlation