

# STAT 215A Fall 2021

## Week 4

---

Omer Ronen

# Announcements

- Congrats on finishing Lab 1 !!!!
- I will send out instructions on how to do peer reviews on Sunday
  - Completed peer reviews due in one week at **11:59pm Sunday Sept. 26**
- Lab 2 + Homework 2 will 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)

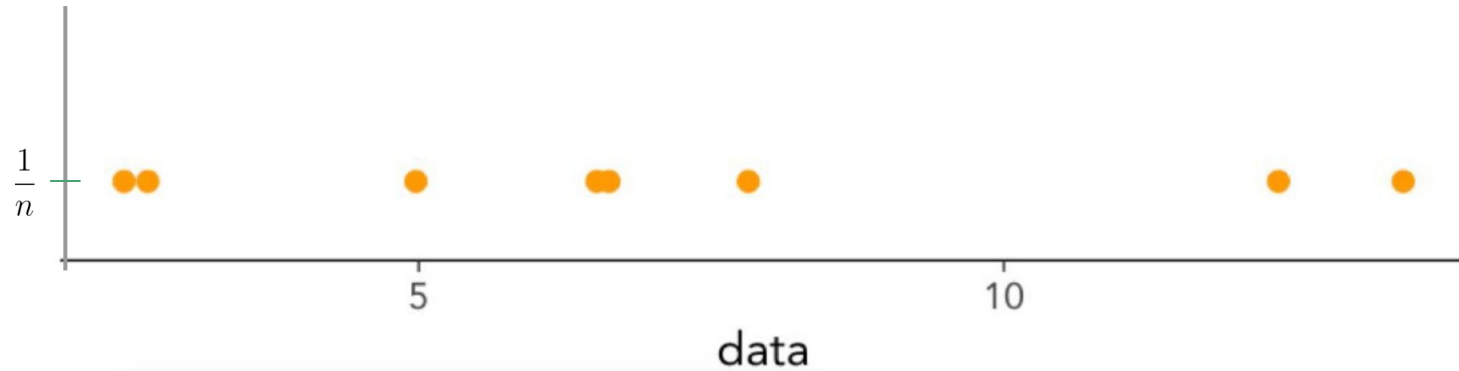
1. Domain question / problem\*
  2. Data collection & storage\*
  3. Data cleaning & preprocessing\*
  4. EDA\*
  5. Modeling & post-hoc analysis + PCS inference
  6. Interpretation
- } You just did this!
- } Coming soon...

# PCS Documentation Examples

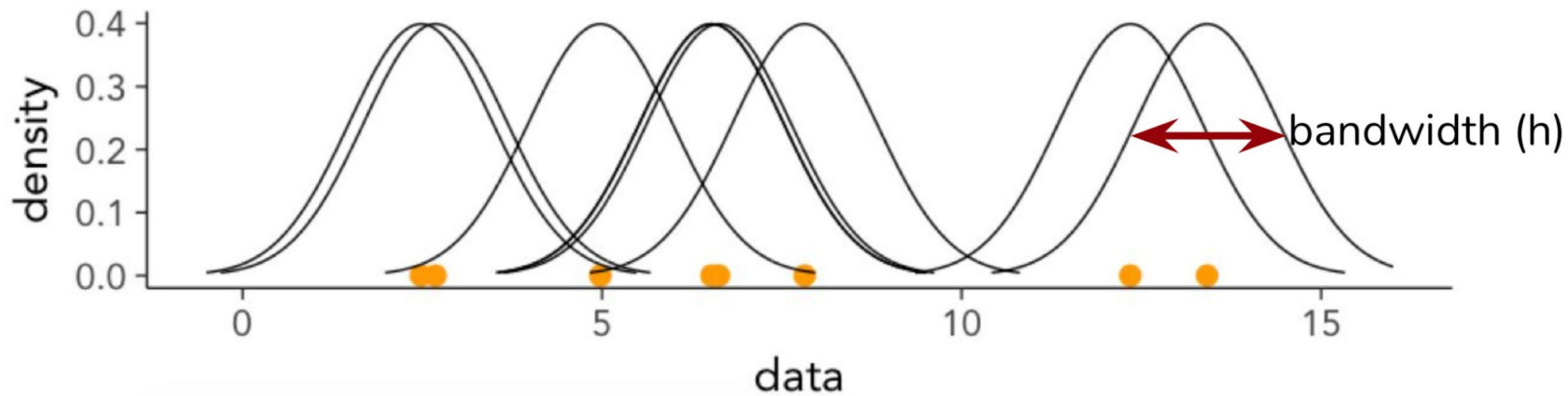
Veridical Data Science (Karl Kumbier and Bin Yu, 2019)

- 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!

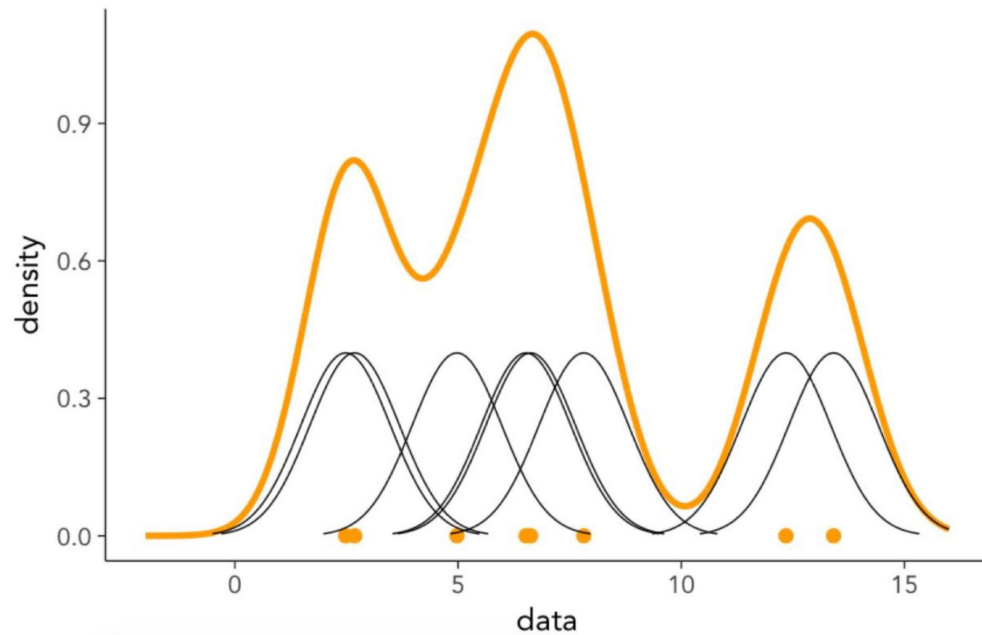
# Kernel density estimation



# Kernel density estimation



# Kernel density estimation





# Kernel density estimation

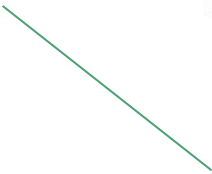
Estimate the density,  $f$ , by adding together individual kernel functions

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

# Kernel density estimation

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

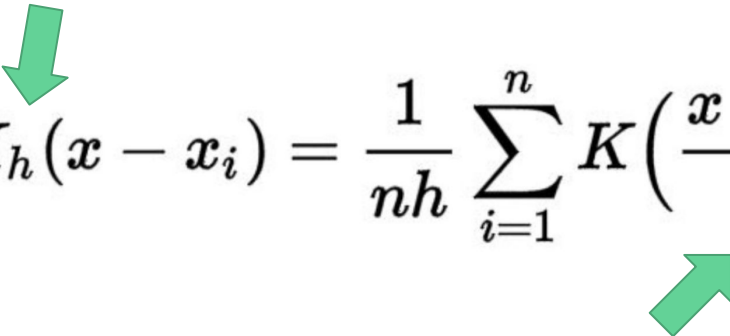
$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$



Each kernel function is centered at a data point

# Kernel density estimation

The width of the kernel function is defined by the bandwidth  **$h$**

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


# 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
- ...

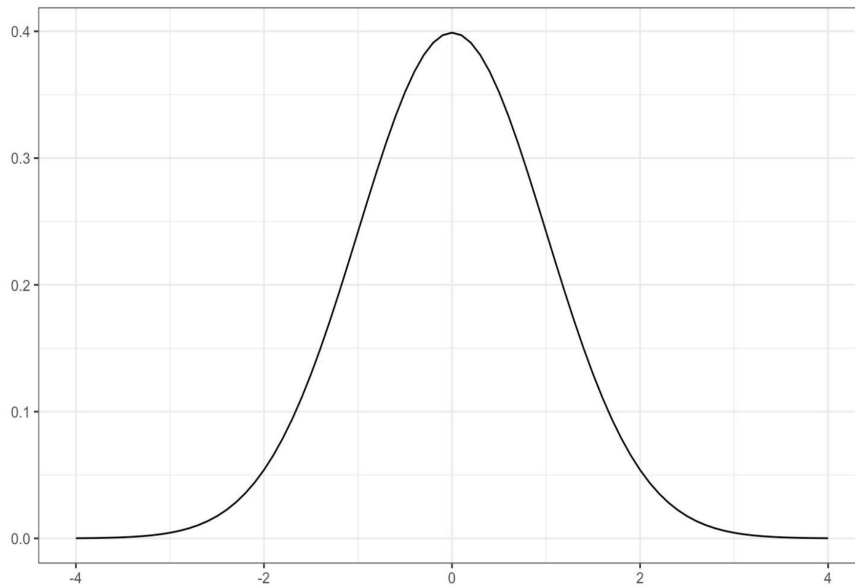
Properties:

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

# Gaussian kernel

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

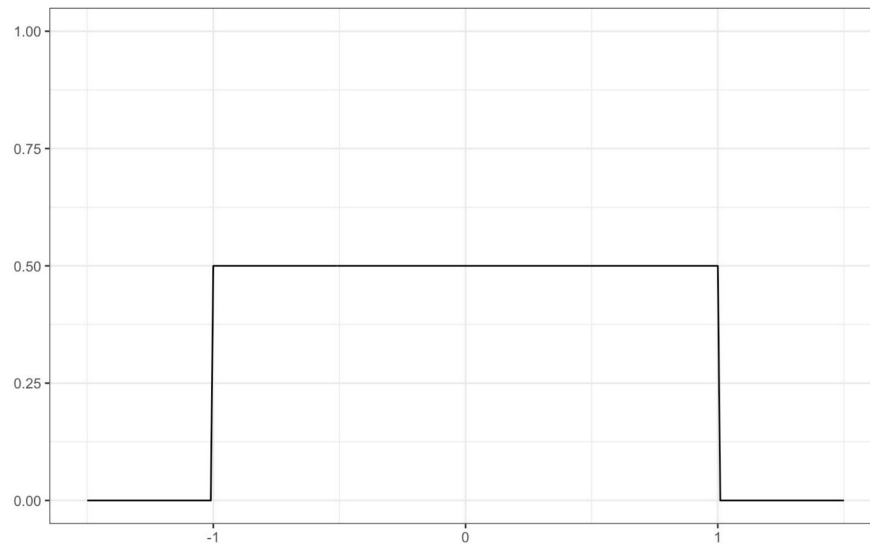
$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$$



# Square kernel

Support:  $|u| \leq 1$

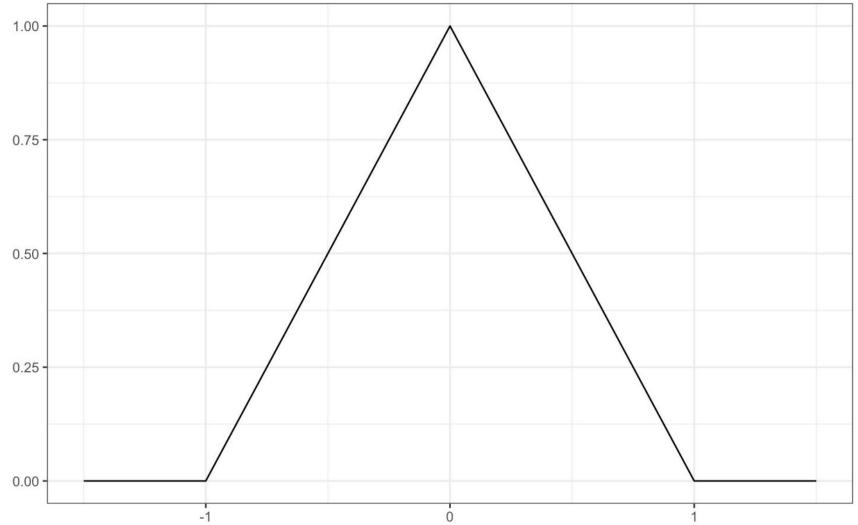
$$K(u) = \frac{1}{2}$$



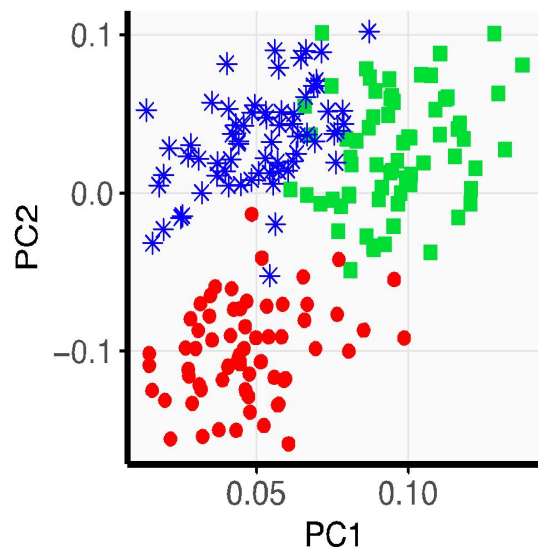
# Triangular kernel

Support:  $|u| \leq 1$

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





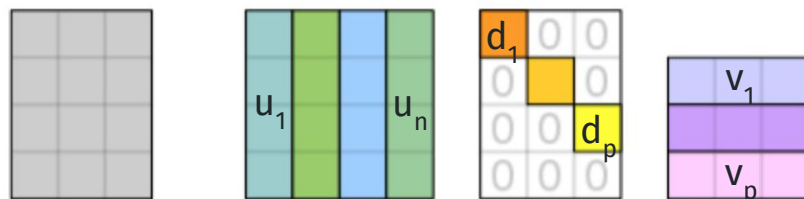


# Review of PCA

(Slides in part thanks to Tiffany Tang)

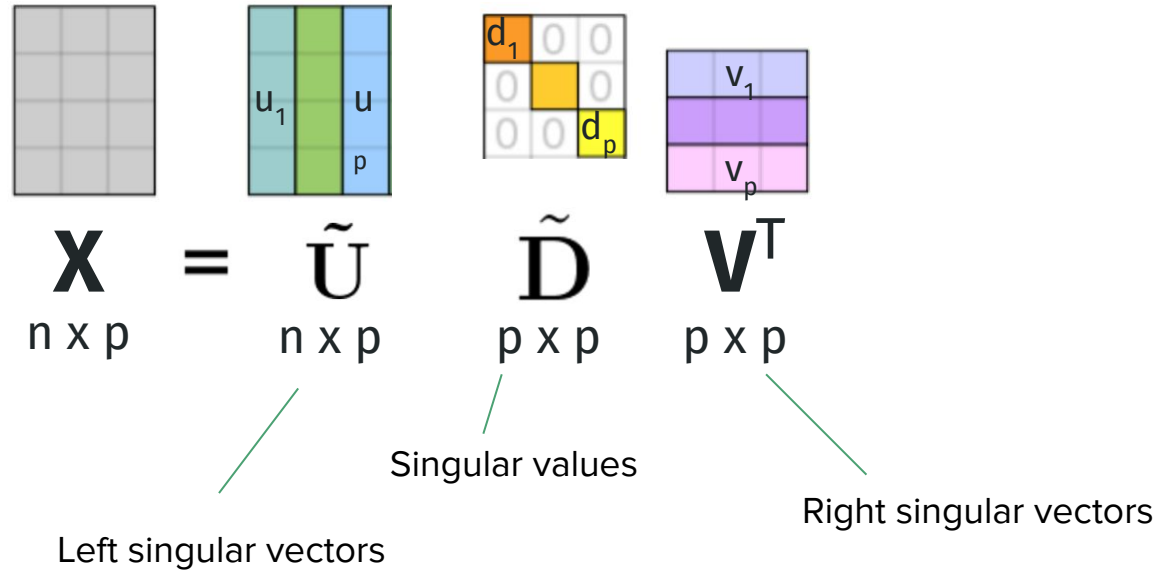
# SVD

(assuming  $n > p$ )


$$\begin{matrix} \mathbf{X} & = & \mathbf{U} & \mathbf{D} & \mathbf{V}^T \\ n \times p & & n \times n & n \times p & p \times p \end{matrix}$$

$$\begin{aligned} d_1 &\geq \dots \geq d_p \\ \mathbf{U}^T \mathbf{U} &= \mathbf{U} \mathbf{U}^T = \mathbf{I}_{n \times n} \\ \mathbf{V}^T \mathbf{V} &= \mathbf{V} \mathbf{V}^T = \mathbf{I}_{p \times p} \end{aligned}$$

# Economy SVD



In R: `svd()`

# PCA

$$\begin{aligned}\mathbf{X}^\top \mathbf{X} &= \mathbf{V} \underbrace{\tilde{\mathbf{D}} \tilde{\mathbf{U}}^\top \tilde{\mathbf{U}} \tilde{\mathbf{D}}}_{\mathbf{I}_{p \times p}} \mathbf{V}^\top \\ &= \mathbf{V} \tilde{\mathbf{D}}^2 \mathbf{V}^\top\end{aligned}$$

Eigenvalues                      Eigenvectors

# PCA

**PC directions:** dominant feature patterns

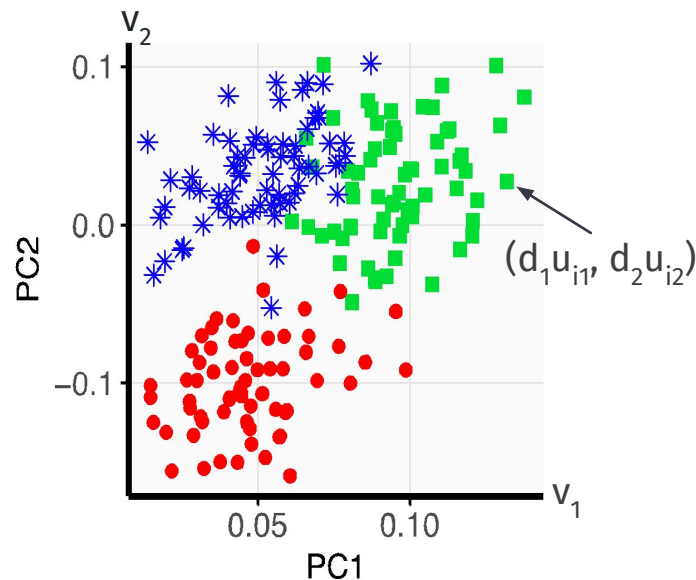
$$\mathbf{v}_1 = \operatorname{argmax}_{\mathbf{v} \in \mathbb{R}} \mathbf{v}^\top \mathbf{X}^\top \mathbf{X} \mathbf{v} \quad \text{subject to} \quad \|\mathbf{v}\|_2^2 = 1, \quad \mathbf{v}^\top \mathbf{v}_i = 0 \quad \forall i \neq j$$

**PC scores:** dominant observation patterns

$$d_j \mathbf{u}_j = \mathbf{X} \mathbf{v}_j \quad (\text{projection of data onto directions of maximizing variance})$$

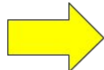
**Proportion of Variance Explained:**

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



# 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

# 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>