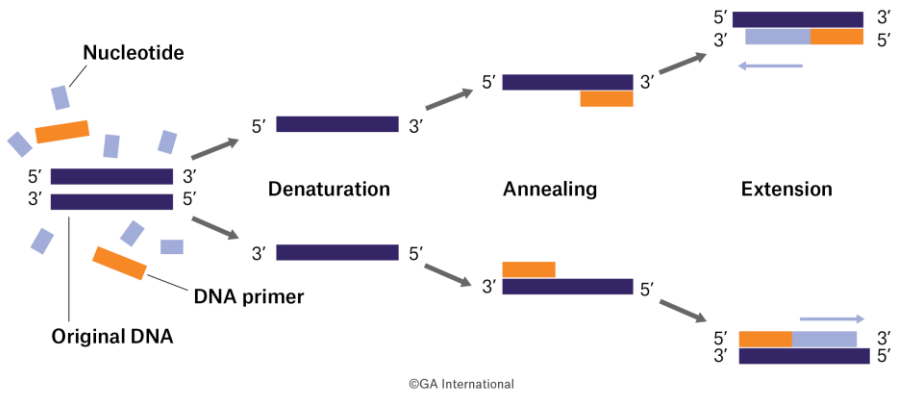
Chemical Engineering 4H03

Principal Component Regression (PCR)

Jake Nease
McMaster University



https://blog.labtag.com/a-brief-history-of-pcr-and-its-derivatives/

Think Back to Our Data Sets…

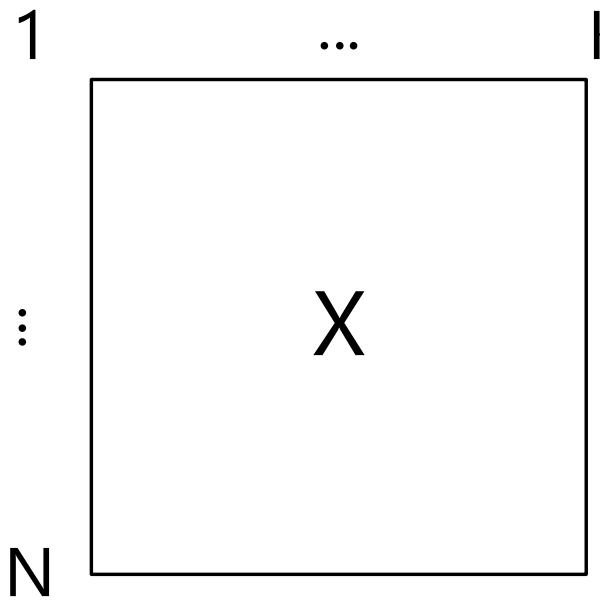
- What kinds of **output** variables did we have?
  - Perhaps user scores or ratings for pastries?
  - Pizza prices?
  - Others?

- Outcomes can be **continuous** or **categorical**
  - Continuous
    - Density
    - Reactor yield
    - Strain required for fracture
    - Absorbance/reflectance of surface material
    - Final course grade (as %)
  - Categorical
    - Pass/fail (also for course grades but also lots of processes!)
    - Good/marginal/bad (or another Likert-style rating)

A Primary Goal of Modeling

- We always want to learn from and visualize data
  - Tools like PCA help us do this nicely
- However, we frequently want to **predict or monitor an outcome based on known quantities**
  - For this, we typically build a model or regression
- Lucky for us, there are TWO things PCA/LVMs can give us in this regard:
  - Dimension reduction → regression
  - Dimension reduction **and** regression simultaneously

A Primary Goal of Modeling



We often *know* values of **Y** in the model building phase

Model Building

Use Model in Future

$\mathbf{x}_{new}^T$ → $\hat{\mathbf{y}}_{new}^T$

We desire the model to *predict* future values of the variables in **Y** since we don't want to measure them again (or may not be able to)

Objectives for this Class

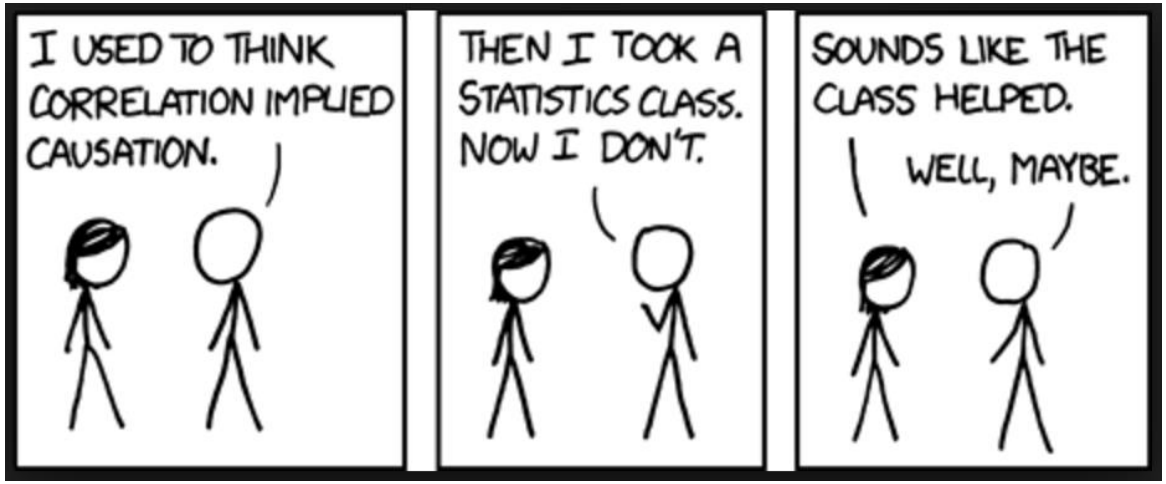
- Now that we know how to compute a PCA model, can we use that information to predict an *outcome* **Y**?

- We sure can. There are two main ways:
  - **P**rincipal **C**omponent **R**egression (PCR) is the poor cousin to…
  - **P**rojection of **L**atent **S**tructures (PLS)

- **This class**: PCR
  1. Mathematical review of covariance
  2. Advantages of PCR to straight-up linear regression
  3. Methods of calculating (long way and shortcuts)

Review: Covariance

… But, like, an applied version of covariance



https://i.pinimg.com/originals/7f/48/66/7f4866f265e67344670e05ba27ccf88f.png

Let's Consider a Data Set

	Temperature (K)	Pressure (kPa)	Humidity (%)
	273	1600	42
	285	1670	48
	297	1730	45
	309	1830	49
	321	1880	41
	333	1920	46
	345	2000	48
	357	2100	48
	369	2170	45
	381	2200	49
MEAN	327	1910	46.1
VARIANCE	1188	38940	7.29

Review: Covariance

- The covariance of two variables x and y is:

$$\mathcal{C}\{x, y\} = \mathcal{E}\{(x - \bar{x})(y - \bar{y})\}$$

Recall $\bar{z}$ is the mean for a random variable

- $\mathcal{E}\{z\}$ is the so-called "expected outcome" of z and is $\bar{z}$
- The covariance of x with itself is its **variance**

$$\mathcal{C}\{x, x\} = \mathcal{V}(x) = \mathcal{E}\{(x - \bar{x})(x - \bar{x})\} \text{ ----- } \frac{\sum_i (x_i - \bar{x})^2}{n}$$

- (Co)variance of an centered vector is precisely equal to the (co)variance of an uncentered vector (why?)
  - Covariance attempts to describe tendencies of two variables

Example: Covariance Matrix $\mathcal{C}\{x,y\} = \mathcal{E}\{(x-\bar{x})(y-\bar{y})\}$

- A covariance matrix is a **real, symmetric** matrix
  - Shows **variances** on the diagonal
  - Shows **covariances** between observations on off-diagonals

- For our observations:

	Temperature	Pressure	Humidity
Temperature	**1188**	6780	35.4
Pressure	6780	**38940**	202
Humidity	35.4	202	**7.29**

```
>> C = cov(A,1)

C =

   1.0e+04 *

    0.1188    0.6780    0.0035
    0.6780    3.8940    0.0202
    0.0035    0.0202    0.0007
```

- Easy to do in `MATLAB`: use `cov(A)` on matrix A
  - **NOTE**: the "1" means population covariance (normalized n) rather than sample covariance (normalized to $n-1$)

Review: Correlation

- Covariance depends on **units** of x and y
- Correlation $r\{x, y\}$ removes this effect by scaling covariance by the root-product of each variance

$$r\{x, y\} = \frac{\mathcal{E}\{(x - \bar{x})(y - \bar{y})\}}{\sqrt{\mathcal{V}(x)\mathcal{V}(y)}} = \frac{\mathcal{C}\{x, y\}}{\sqrt{\mathcal{V}(x)\mathcal{V}(y)}}$$

- A couple of notes:
  - $-1 \leq r\{x, y\} \leq 1$
  - Result is dimensionless

Example: Correlation Matrix

$$r\{x,y\} = \frac{\mathcal{C}\{x,y\}}{\sqrt{\mathcal{V}(x)\mathcal{V}(y)}}$$

- A correlation matrix is a **real, symmetric** matrix
  - Shows **unity** on the diagonal (why?)
  - Shows **correlation** between observations on off-diagonals

- For our observations:

	Temperature	Pressure	Humidity
Temperature	1	0.997	0.38
Pressure	0.997	1	0.379
Humidity	0.38	0.379	1

```
>> R = corrcoef(A)

R =

    1.0000    0.9968    0.3804
    0.9968    1.0000    0.3791
    0.3804    0.3791    1.0000
```

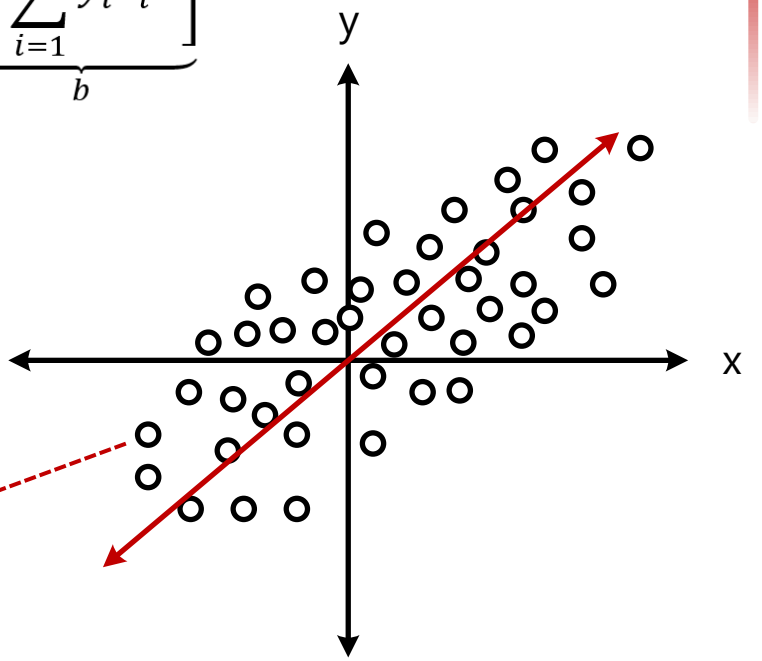
- Easy to do in `MATLAB`: use `corrcoef(A)` on matrix A

Reminder: Linear Regression

- You may recall the LSOE that can be solved for the regression coefficients in our regression $\hat{\boldsymbol{y}} = a_0 + a_1\boldsymbol{x}$:

$$\underbrace{\begin{bmatrix} \sum_{i=1}^{N} 1 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_0 \\ a_1 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} y_i x_i \end{bmatrix}}_{b}$$

- **RECALL**
  - We will center our data
  - Thus $a_0 = 0$



Distance from point to model (line) is ϵ_i, the error between observation y_i and prediction $\hat{y}_i$

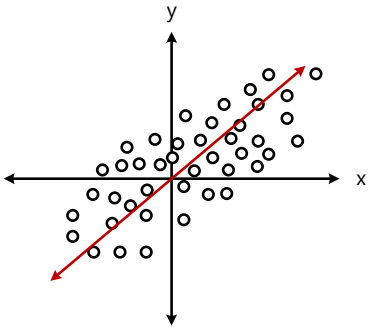
THIS SHOULD BE REVIEW AT THIS POINT. STOP ME IF IT IS STILL UNCLEAR

Reminder: LR in Vector Form

- Our "system of equations" in that case simply boils down to one equation and one unknown:

$$\underbrace{\begin{bmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & N \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} a_1 \\ a_0 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} \sum_{i=1}^{N} y_i x_i \\ \sum_{i=1}^{N} y_i \end{bmatrix}}_{b}$$

$$a_1 = \frac{\sum_{i=1}^{N} y_i x_i}{\sum_{i=1}^{N} x_i^2}$$



- Fun fact, these sums of the elements in each vector can pretty easily be written as dot-products

$$\sum_{i=1}^{N} y_i x_i = y^T x \qquad \sum_{i=1}^{N} x_i^2 = x^T x$$

- We can thus collapse our equation for a_1 into:

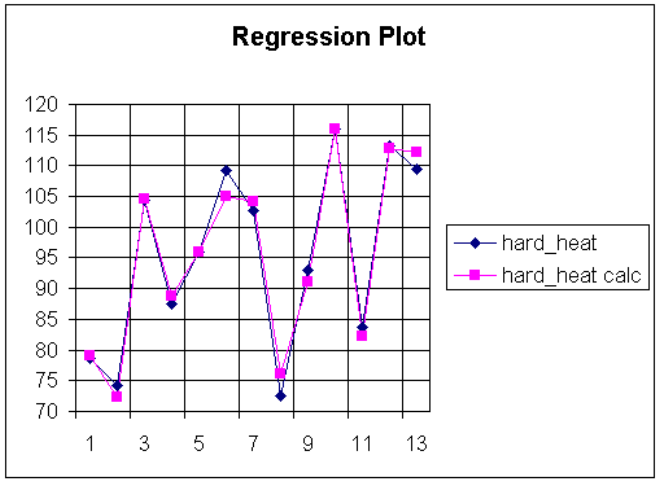
$$a_1 = \frac{x^T y}{x^T x}$$

--------- At this point, I should not need to convince you that this expression is identical to the one for a_1 above

Multi-Linear Regression

It's like what we did before but simplified

BOY there sure are some brutal examples of this stuff 😐



https://www.polymath-software.com/PolymathHelp/mlexpo7.gif

Multi-Linear Regression (MLR)

- The general form of a multi-linear regression (MLR) model for some observation i is a simplified version of our multidimensional basis function regression:
  - Each column of **X** has been centered (maybe scaled)

$$y_i = a_1 x_1 + a_2 x_2 + \cdots + a_K x_K + \epsilon_i$$

$$y_i = \begin{bmatrix} x_1 & x_2 & \cdots & x_K \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix} + \epsilon_i$$

$$y_i = \boldsymbol{x}^T a + \epsilon_i$$

These are all the same

The model for y_i represents a hyperplane in K dimensions (but you may have already known that)

Multi-Linear Regression (MLR)

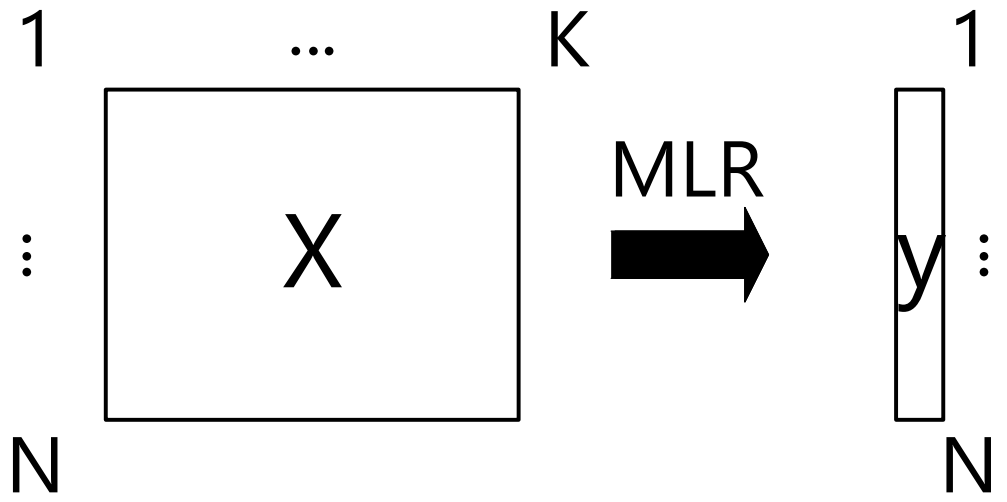
- For multiple rows:

IMPORTANT
X and **a** DO NOT need to have the same length (actually, they SHOULD NOT have the same length)… WHY?

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,K} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,K} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

$$\boldsymbol{y} = X\boldsymbol{a} + \boldsymbol{\epsilon}$$

- $\boldsymbol{y} \in \mathbb{R}^{\boldsymbol{N}\times\boldsymbol{1}}$
- $X \in \mathbb{R}^{\boldsymbol{N}\times\boldsymbol{K}}$
- $\boldsymbol{a} \in \mathbb{R}^{\boldsymbol{K}\times\boldsymbol{1}}$
- $\boldsymbol{\epsilon} \in \mathbb{R}^{\boldsymbol{N}\times\boldsymbol{1}}$



MLR Calculations

- Solving for the coefficients is simple for us now:
- We can do it on the board!

$$\min_{\boldsymbol{a}} \phi = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}$$

$$\min_{\boldsymbol{a}} \phi = (\boldsymbol{y} - X\boldsymbol{a})^T (\boldsymbol{y} - X\boldsymbol{a})$$

$$\min_{\boldsymbol{a}} \phi = \boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{y}^T X\boldsymbol{a} - \boldsymbol{a} X^T X\boldsymbol{a}$$

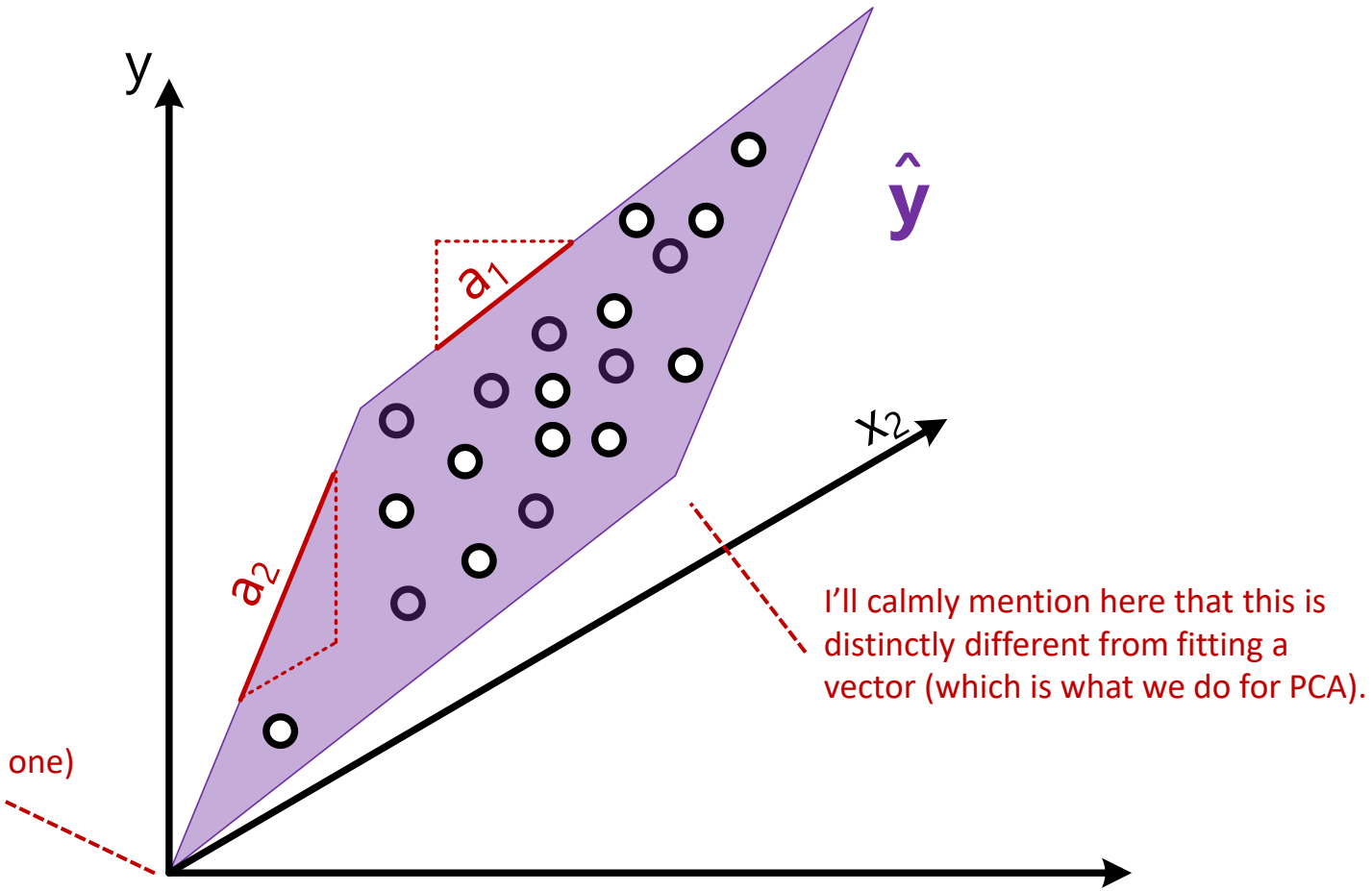
$$\frac{\partial \phi}{\partial \boldsymbol{a}} = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{a} = (X^T X)^{-1} X^T \boldsymbol{y}$$

IMO, when you derive stuff like this from the optimization perspective, things make way more sense than when you are given random equations.

This ought to look VERY familiar with how we computed the loadings **p** from the NIPALS algorithm :3

Interpretation of Coefficients

- Unsurprisingly, the coefficients represent the **projected slope** of variable k on the x_k, y plane



MLR: What Could Go Wrong?

- Missing values
  - What if one of our $\boldsymbol{x}$ values is missing?
  - Do we replace with 0? How about the mean (trick question :3)
    - This could be VERY dangerous
  - In reality, there is nothing we can do
  - **We can't use the model**

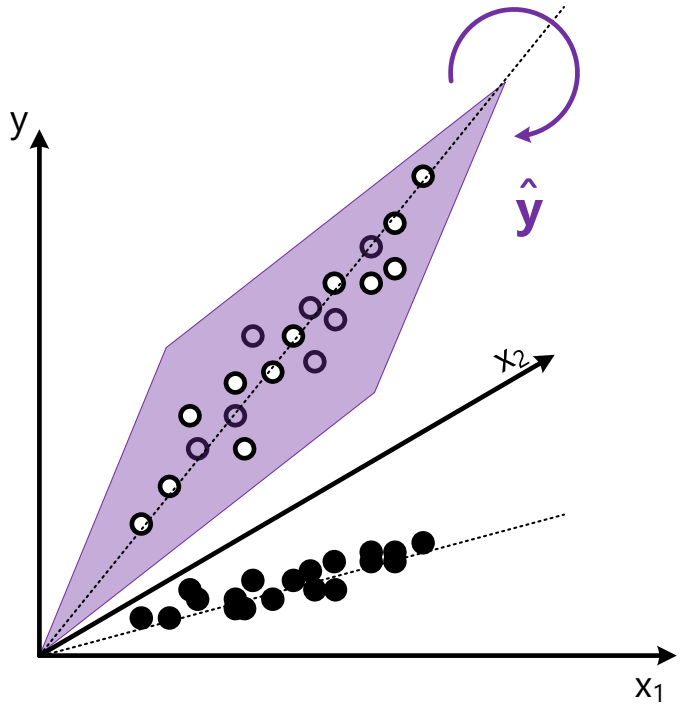
- $\hat{y}_i = a_1 x_1 + a_2 x_2 + \cdots + a_K x_K$

MLR: What Could Go Wrong?

- Correlated variables in **X**
  - Leads to ALL KINDS of problems…
  - Recall $\boldsymbol{a} = (X^T X)^{-1} X^T \boldsymbol{y}$
  - Variance of coefficient vector $\mathcal{V}(\boldsymbol{a}) = (X^T X)^{-1} S_{\boldsymbol{\epsilon}}^2$
  - S_ϵ^2 is the sample variance of ϵ with $N - K$ DOF: $S_{\boldsymbol{\epsilon}}^2 = \frac{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}}{N-K}$

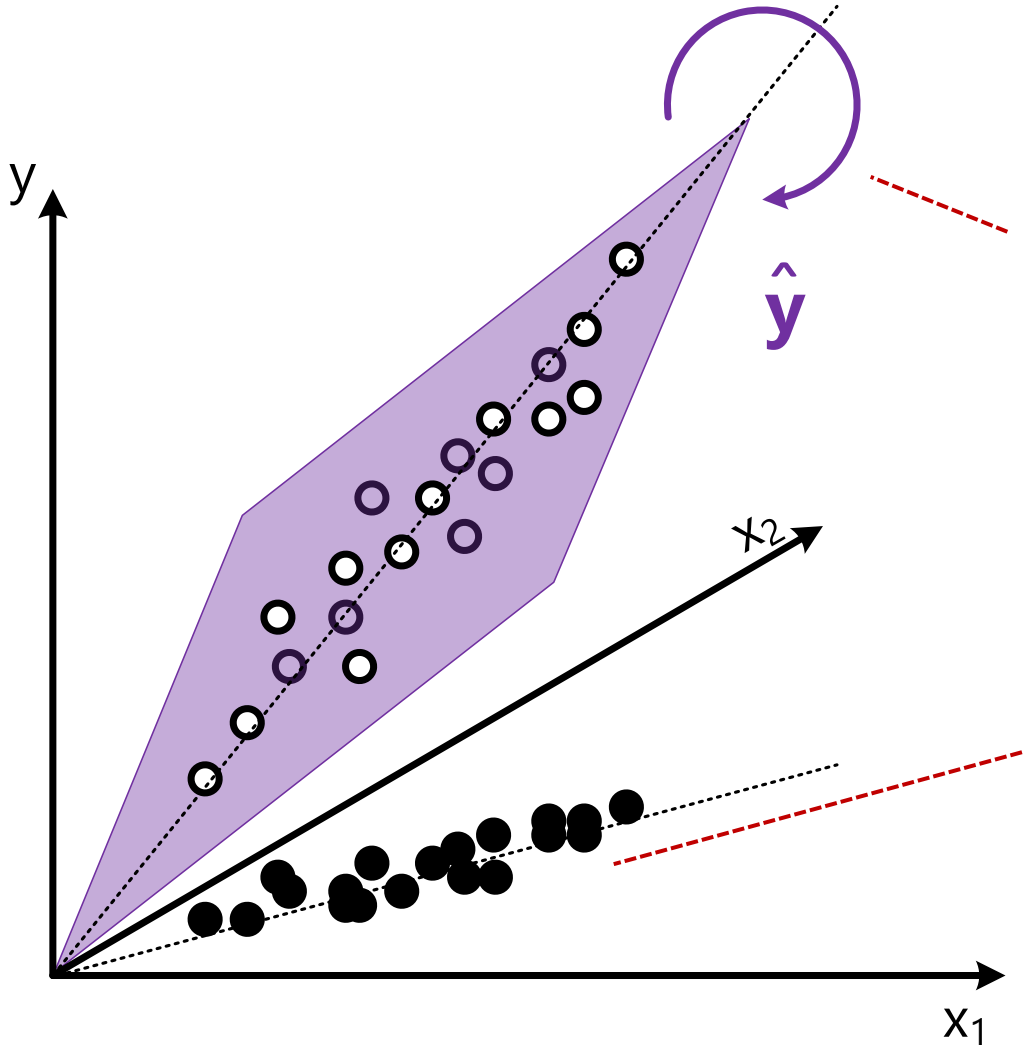
This is one of those things you can just take my word on ☺

- Leads to **lack of confidence** in coefficients of **a**
  - Geometrically, it means the model plane could "spin" around and basically give the same results (see right)
  - Algebraically, it means that our columns are **dependent** and thus $X^T X$ risks singularity (woof!)



MLR: What Could Go Wrong?

- Let's discuss this figure a little more



QUESTION: what could cause this model plane to rotate?

QUESTION: How do you think the dotted line relates to our PCA knowledge?

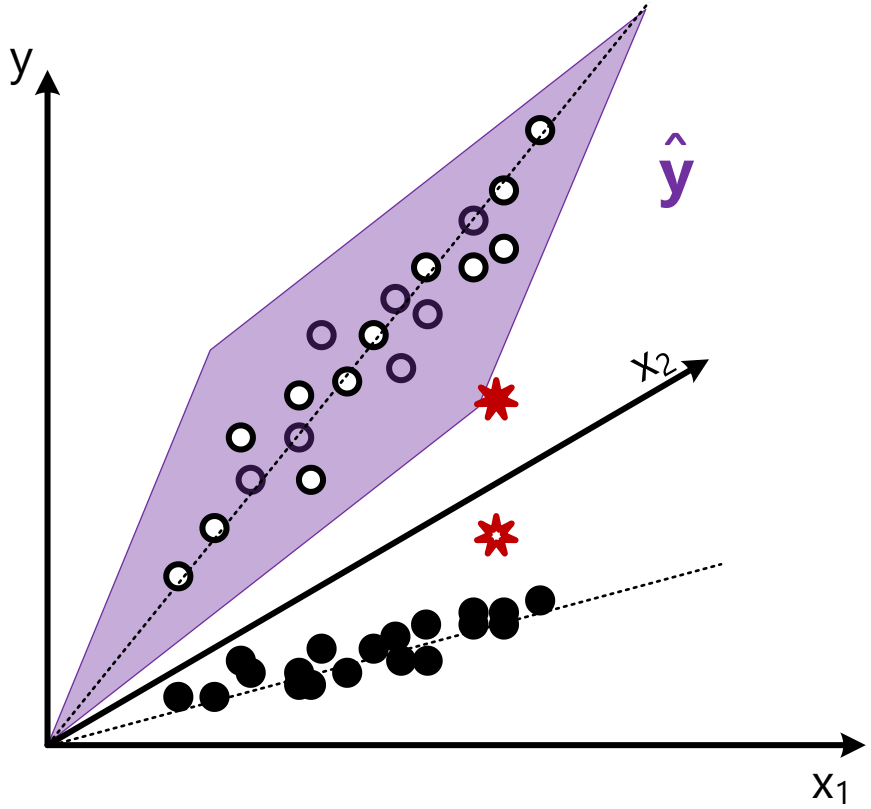
QUESTION: what are we looking at here?

MLR: What Could Go Wrong?

- Noise in **X**
  - Recall the LS model is $\hat{y} = a_1 x + \epsilon$
  - This assumes the error of the model is contained in **y**
- We therefore model our error in the **y**-space
  - We can compare our **y** error to S_ϵ^2
  - Akin to finding the R^2
  - See if something is wrong, and fix it
- CRITICALLY, LS assumes that **X** is exact (no noise)
  - No model space for the **X** error ⇒ loss of model fidelity

MLR: What Could Go Wrong?

- Nonsense observation for **X**
  - Since we assume no error in **X**, can result in:
    1. A "bad" outlier in **y** that is not the model's fault
    2. A prediction for **y** that is completely unrelated to **X**
  - In both cases, it is the measurement that is bad



MLR: What Could Go Wrong?

- MLR requires N > K
  - Some data sets have more columns than rows!
- Multiple **Y** variables
  - Need a separate regression model for each one
- Strategies to improve these issues?
  - Use basis functions if process is nonlinear
  - Visualization
  - Elimination of correlated columns in **X** (how, though?)
- For many data sets, problems solved using…

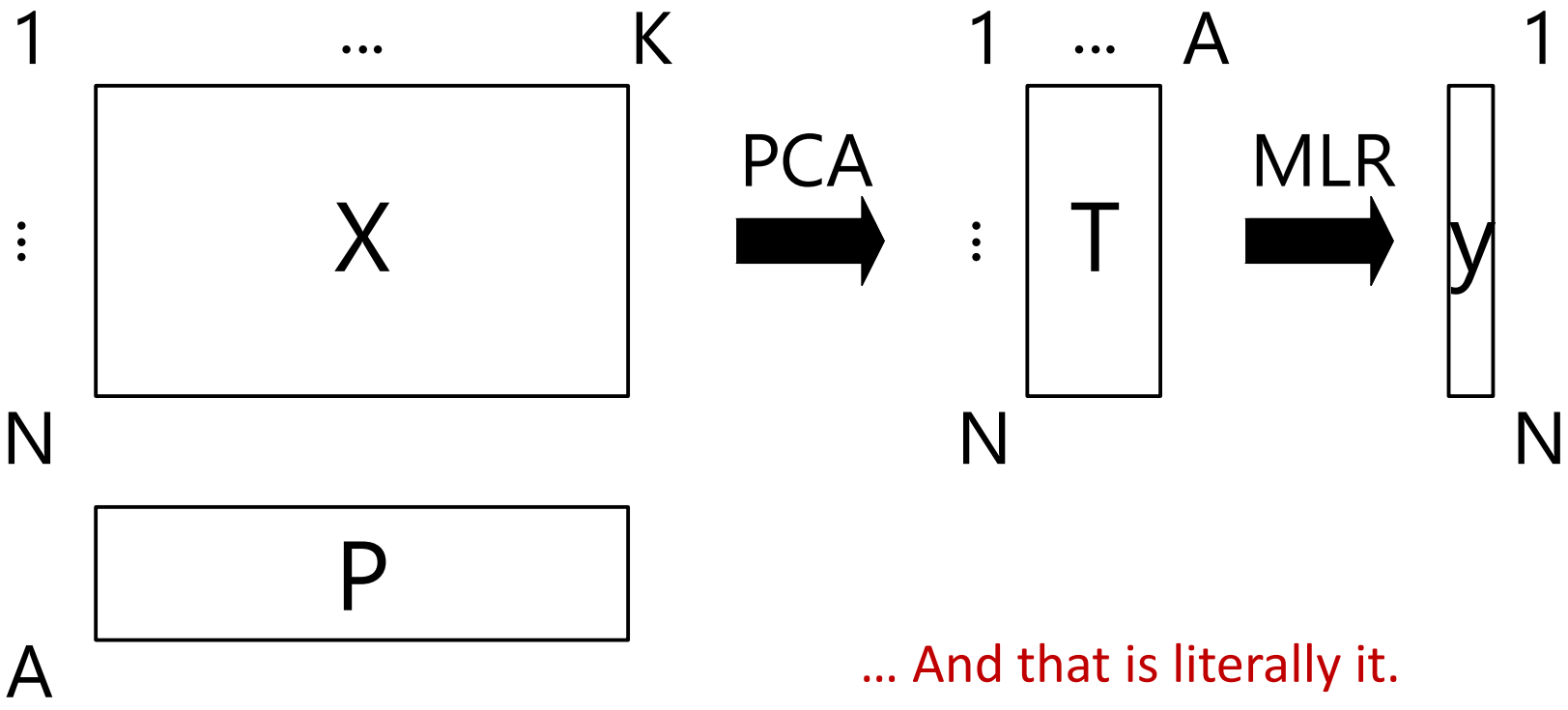
Principal Component Regression

And now for 4 slides on the actual topic :3

https://vignette.wikia.nocookie.net/slay-the-spire/images/e/e3/Distraction.png/revision/latest?cb=20181016211544

PCR in a Nutshell

- The idea (you saw this coming)
  1. Perform PCA on your data set **X** to get loadings **P**
  2. Regress **Y** onto the SCORES **T** to get regression coefficients



… And that is literally it.

PCR in a Nutshell

- Calculations:

$$T = XP$$
$$\hat{y} = T\boldsymbol{a}$$

Don't forget to **center** (x AND y) and **scale** (really only needed for x)

- Coefficients in $\boldsymbol{a}$ are computed as:

$$\boldsymbol{a} = (T^T T)^{-1} T^T \boldsymbol{y}$$

- Future predictions:
  - Use x_{new} to get t_{new}
  - Use t_{new} and $\boldsymbol{a}$ to get $\hat{y}_{new}$

Don't forget to **center** and **scale** x_{new} here as well! Also, your $\hat{y}$ will be centered so add $\bar{y}$ to get the "real" y

PCR ADVANTAGES

- T is orthogonal, thus $(\mathbf{T}^T\mathbf{T})^{-1}$ is easy to compute
  - Related point: correlated columns in **X** will have one **T**
- PCA step (using NIPALS) handles any missing data
- **T** has considerably less error than **X**
  - **Although** we do lose some variance in our PCA model…
  - **ALTHOUGH** we like to believe it is pure noise…
- PCA scores contain A < K columns, thus the N > K requirement for regression is **likely** to be met
  - QUESTION: What happens to the PCR output is K = A?

- CRITICAL: Each new point can be tested against:
  - SPE (is the X data out of whack?)
  - T^2 (is this an extreme point on the PCA model?)

PCR DISADVANTAGES

- PCA components calculated without knowledge of **Y**
  - PCA and regression steps are sequential, not simultaneous
  - The data in **X** are not necessarily predictive of **y**
- To ensure all predictors of **Y** are contained in **X**…
  - We often fit extra components to PCA model
  - Violate our cross-validation heuristics
    - PRESS and Q^2 improvement rules, for example
- Would it not be better to fit ONE model that tried to relate the PCA components to **Y** immediately?
  - It sure would

Final Remarks

- A quick review of what we have done:
  - Review of covariance and correlation
  - Review of linear regression with no intercept
  - Covered methods of computing coefficients to MLR
  - Caveats and shortcomings of MLR
  - PCR → a better method of dealing with high correlation in **X**

- Next up: **Projection of Latent Structures**
  - Idea: fit loadings for **X** and **Y** together
  - Maximize **covariance** between **X** and **Y** columns