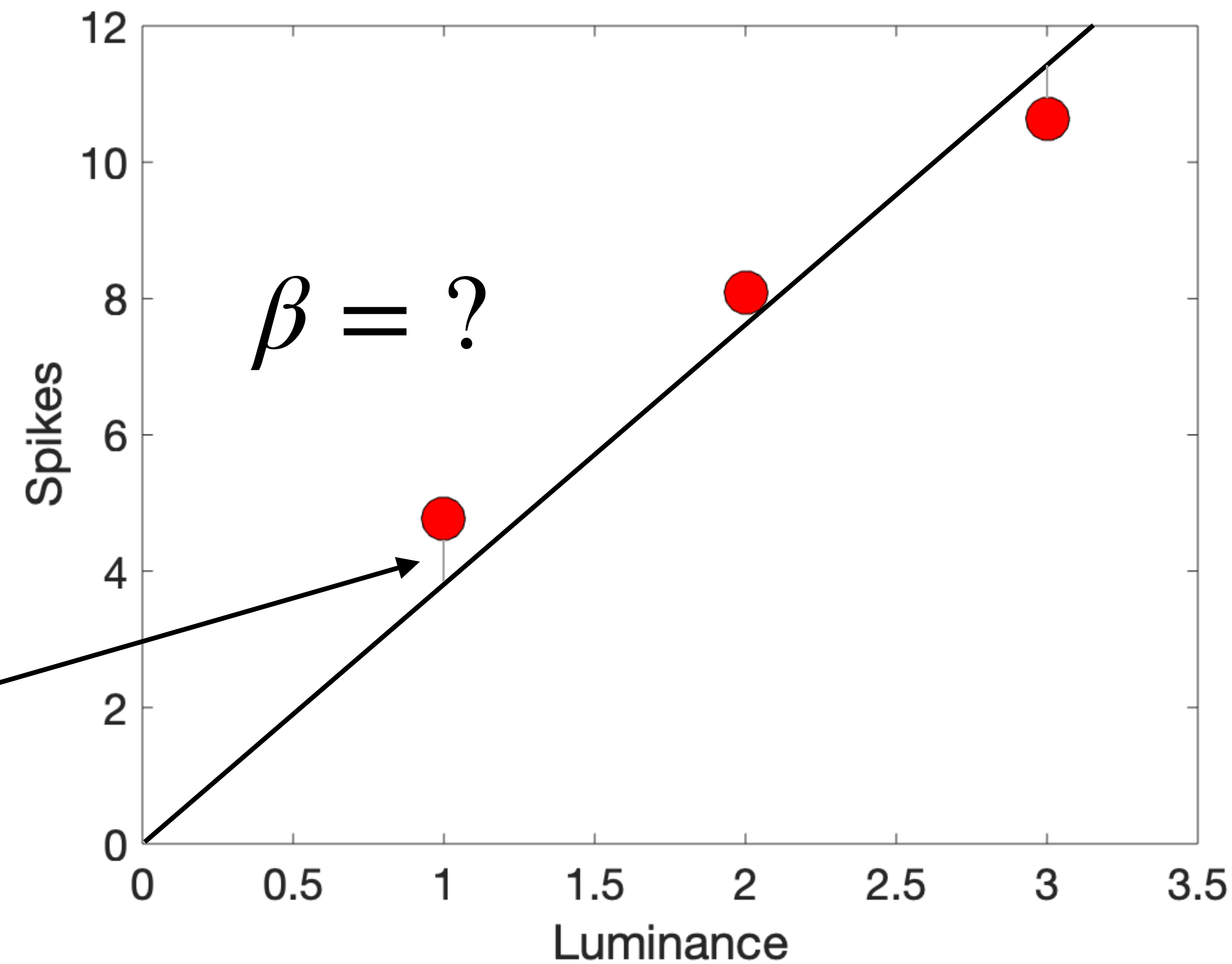
Math Tools

Linear Regression and PCA

02 Oct 2020

Linear Regression

- Let's say we have data from an experiment where we presented light at different luminance levels to a neuron and we measured the spike rate
- We would like to model this neuron as a linear system and use the luminance level to predict the spike rate
- What is the best number to multiply the luminance values tested (the regressor) to get closest to the measured spike rate?
- What is the "best"? measure the error by the squared distance between the actual data, and the predicted values



Linear Regression

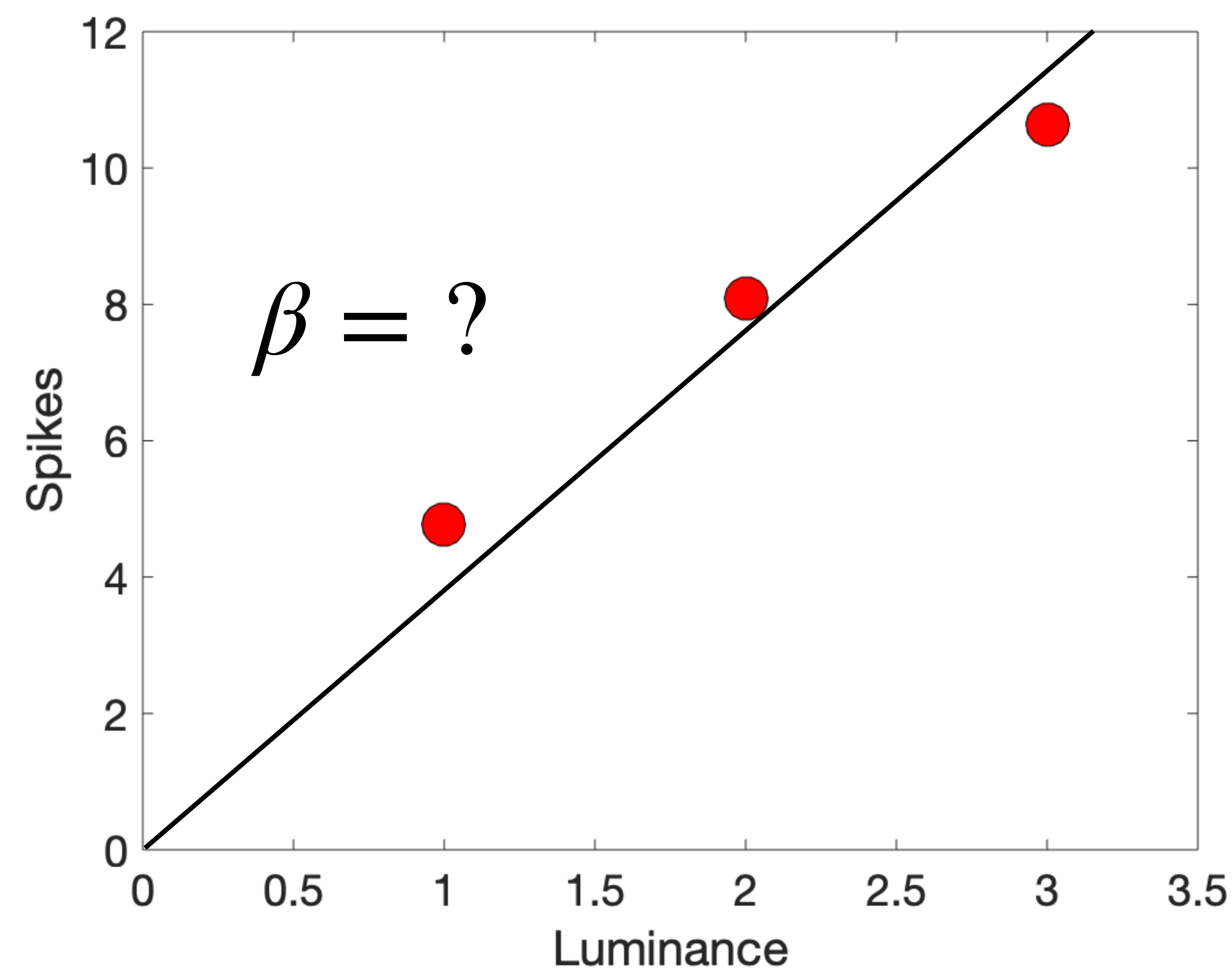
- How do we find the best β?
- Iteratively: try a bunch of different βs and see which one has the minimum error
- But based on our derivations in class, we can find the best β using linear algebra

1. Take SVD of X

$\overrightarrow{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$, matlab: [U,S,V] = svd(x)

2. $\beta_{opt} = VS^{\#}U^{T}\overrightarrow{y}$

$\beta_{opt} = VS^{\#}U^{T}\overrightarrow{y} = 3.78$



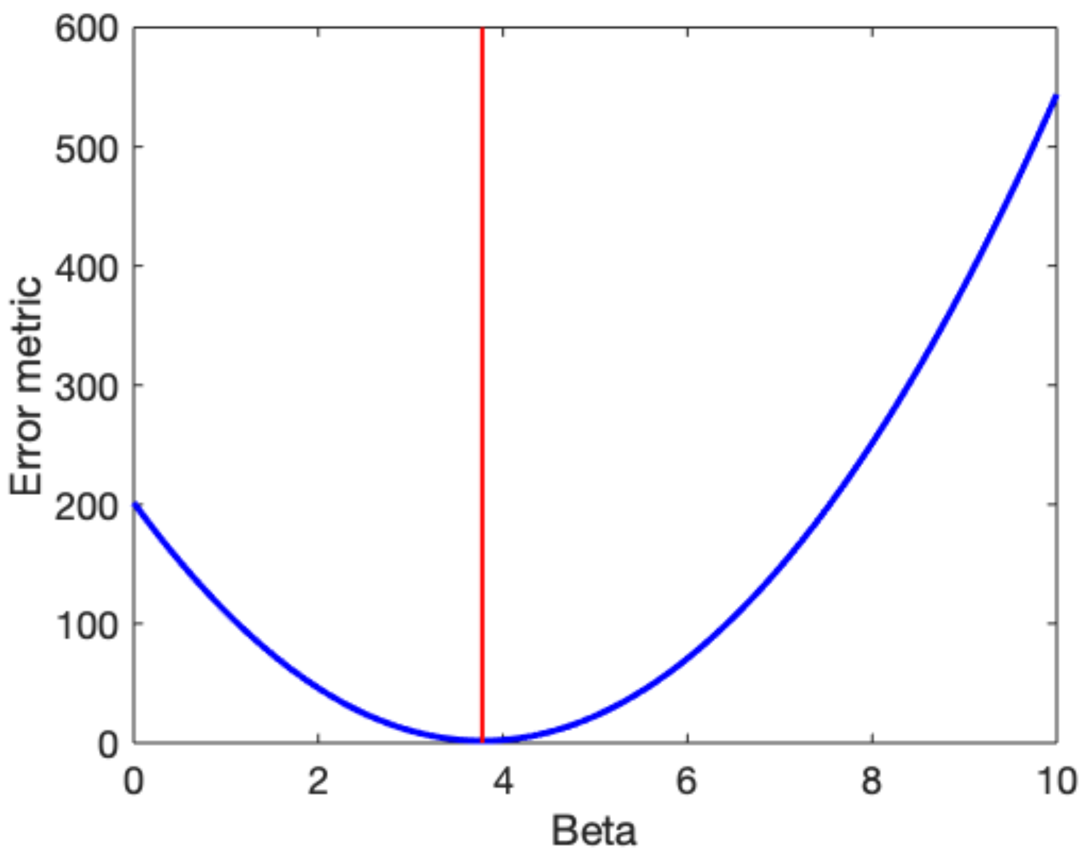
Linear Regression

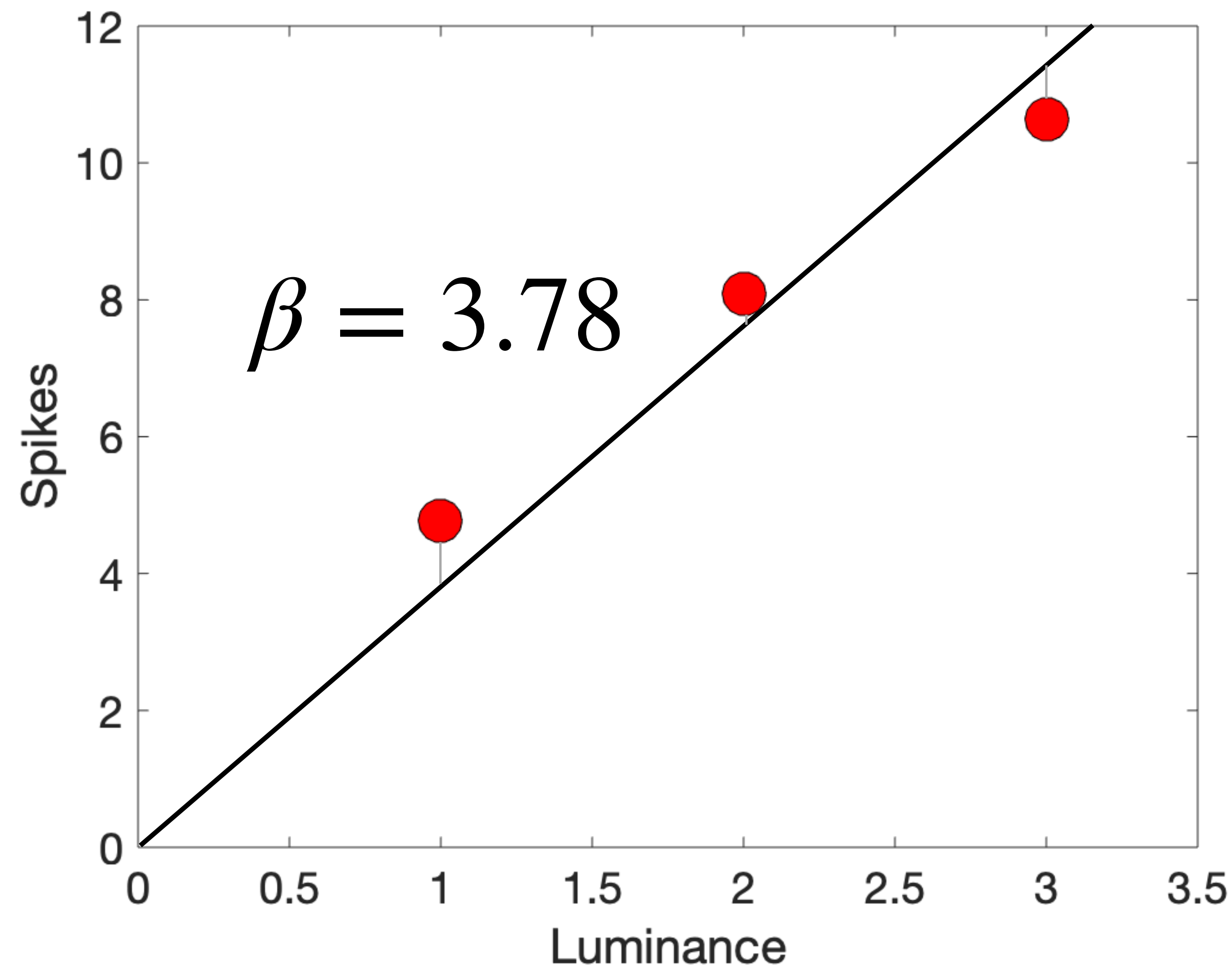
$$\min_{\beta} ||\vec{y} - \beta\vec{x}||^2$$

- Was this the best β?

- Recall: we say it's the optimal β by minimizing the squared distance between the actual data, and the predicted values

- What happens when we vary β?



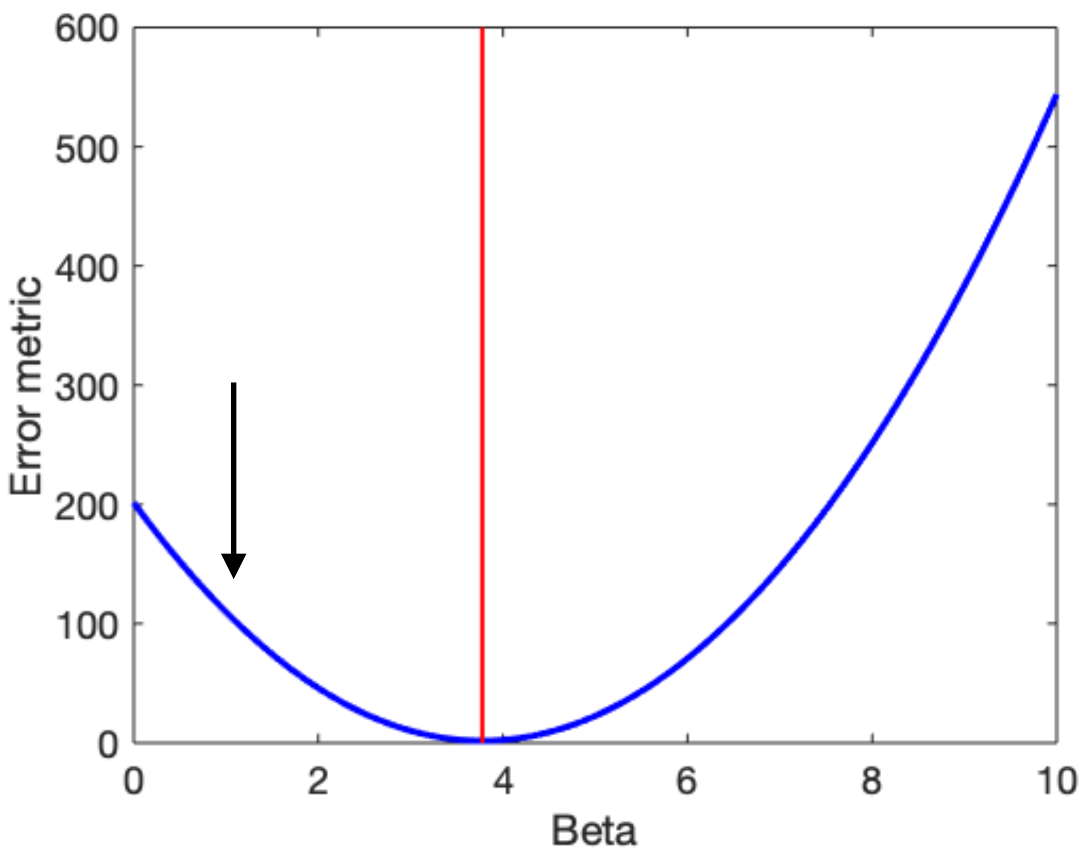


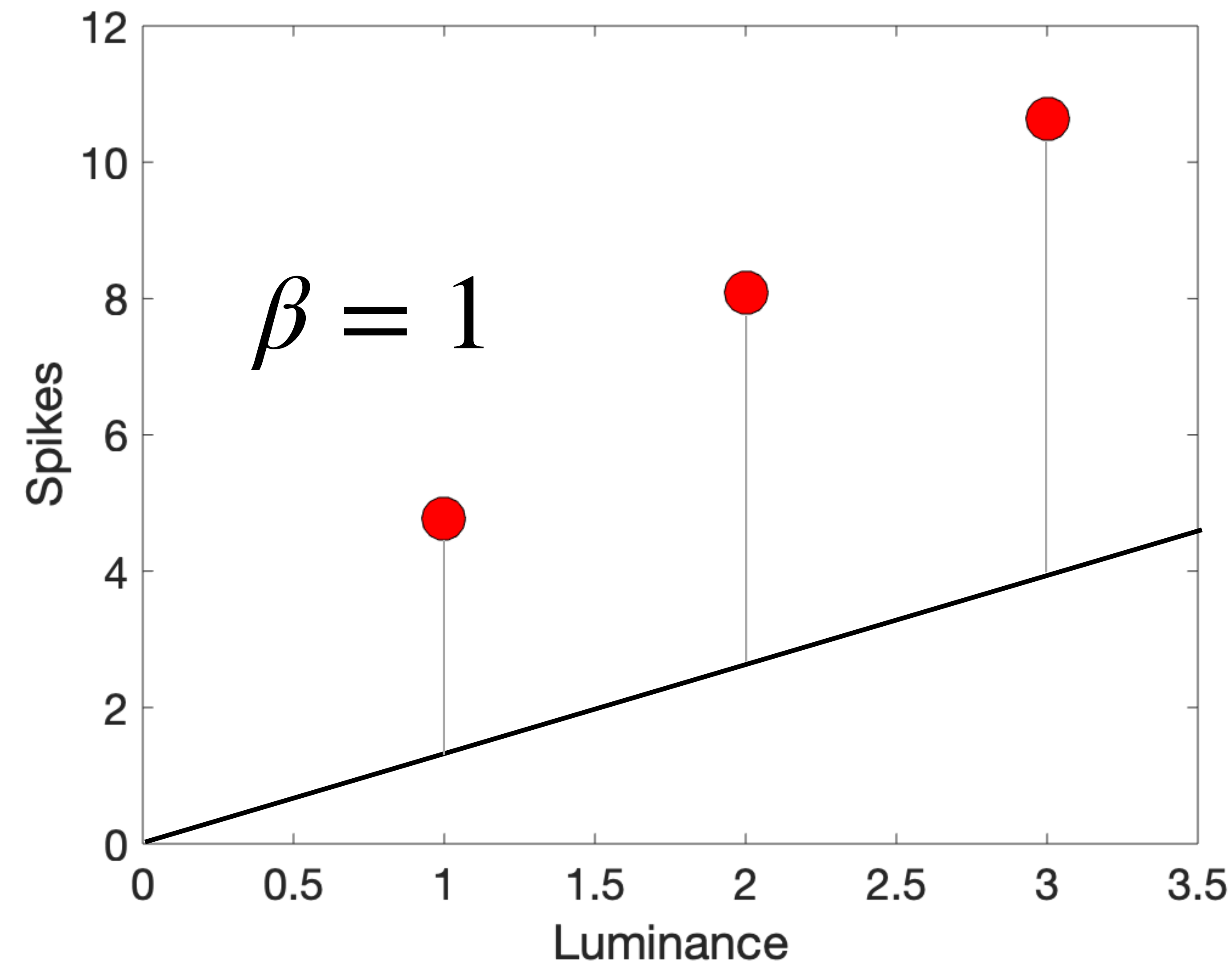
Linear Regression

$$\min_{\beta} ||\vec{y} - \beta\vec{x}||^2$$

- Was this the best β?

- Recall: we say it's the optimal β by minimizing the squared distance between the actual data, and the predicted values

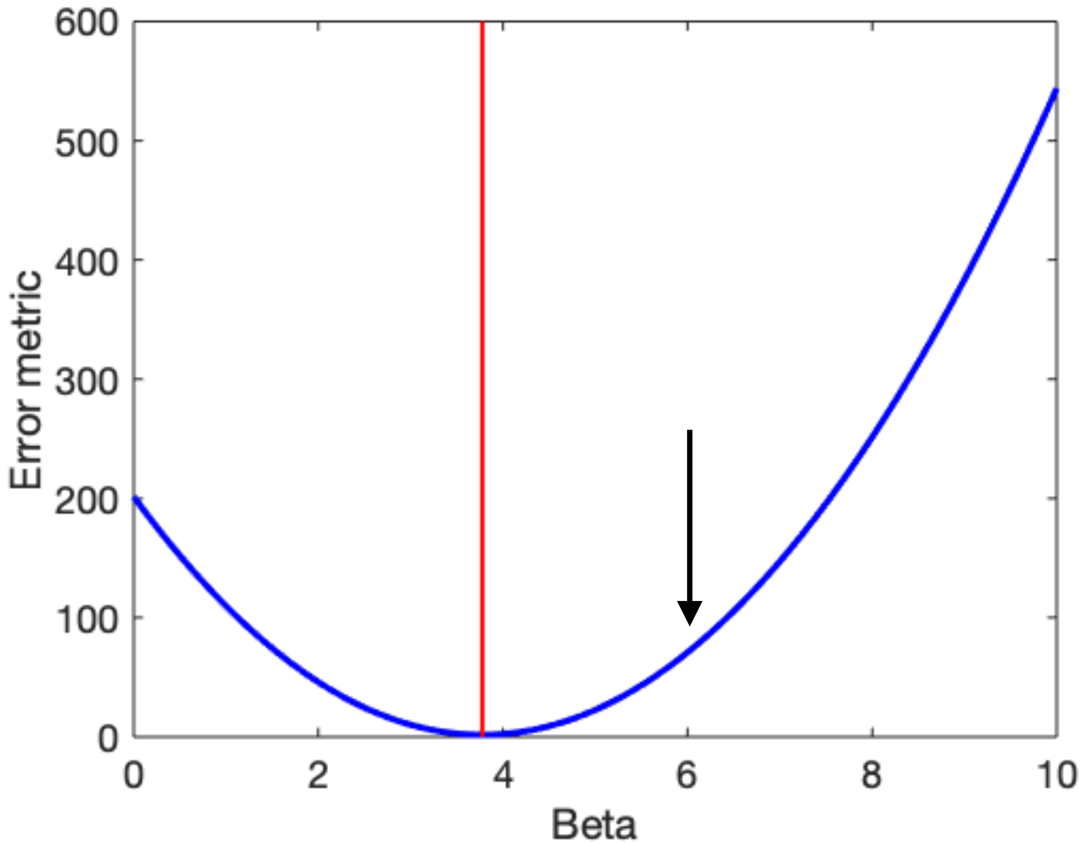


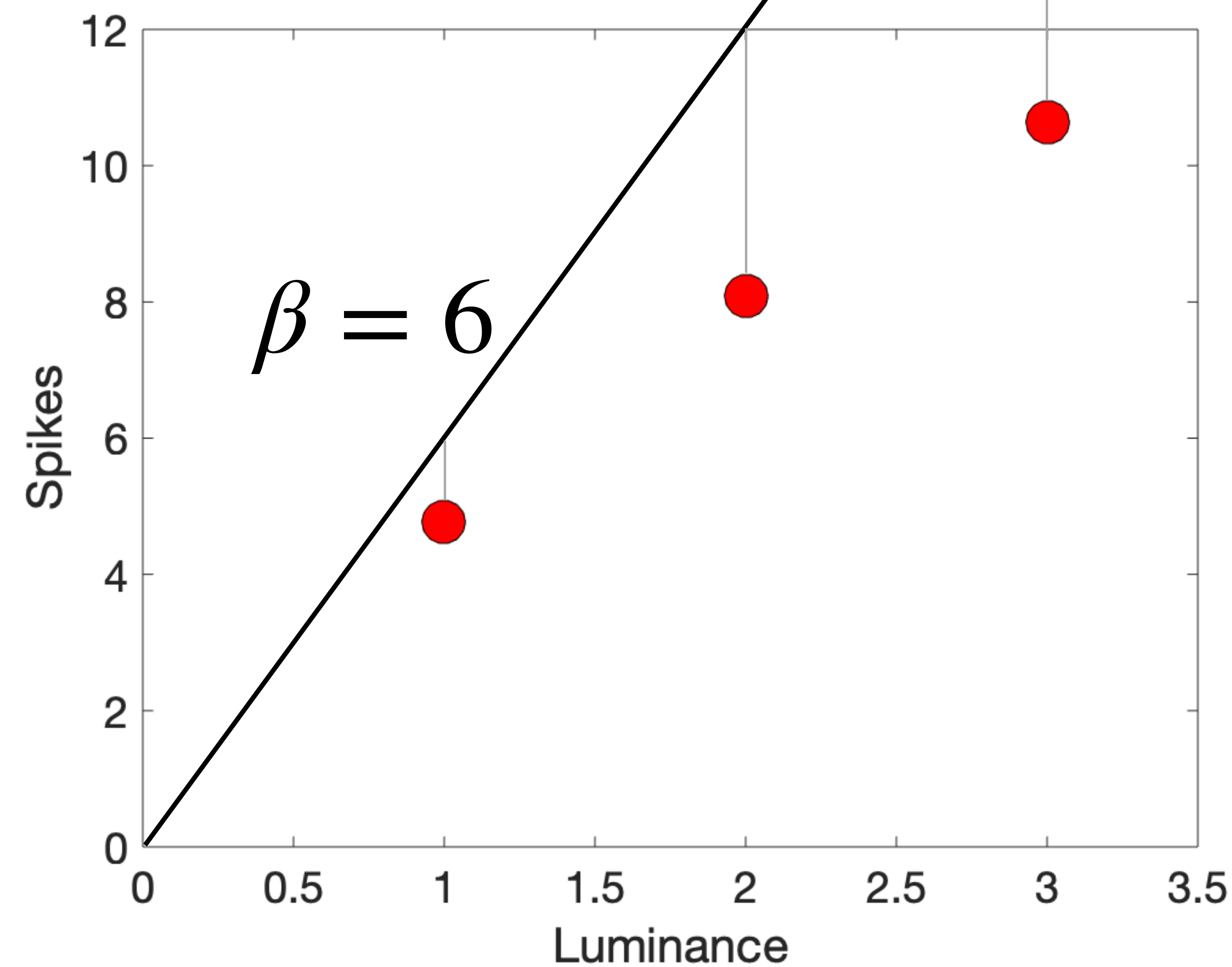


Linear Regression

$$\min_{\beta} ||\vec{y} - \beta \vec{x}||^2$$

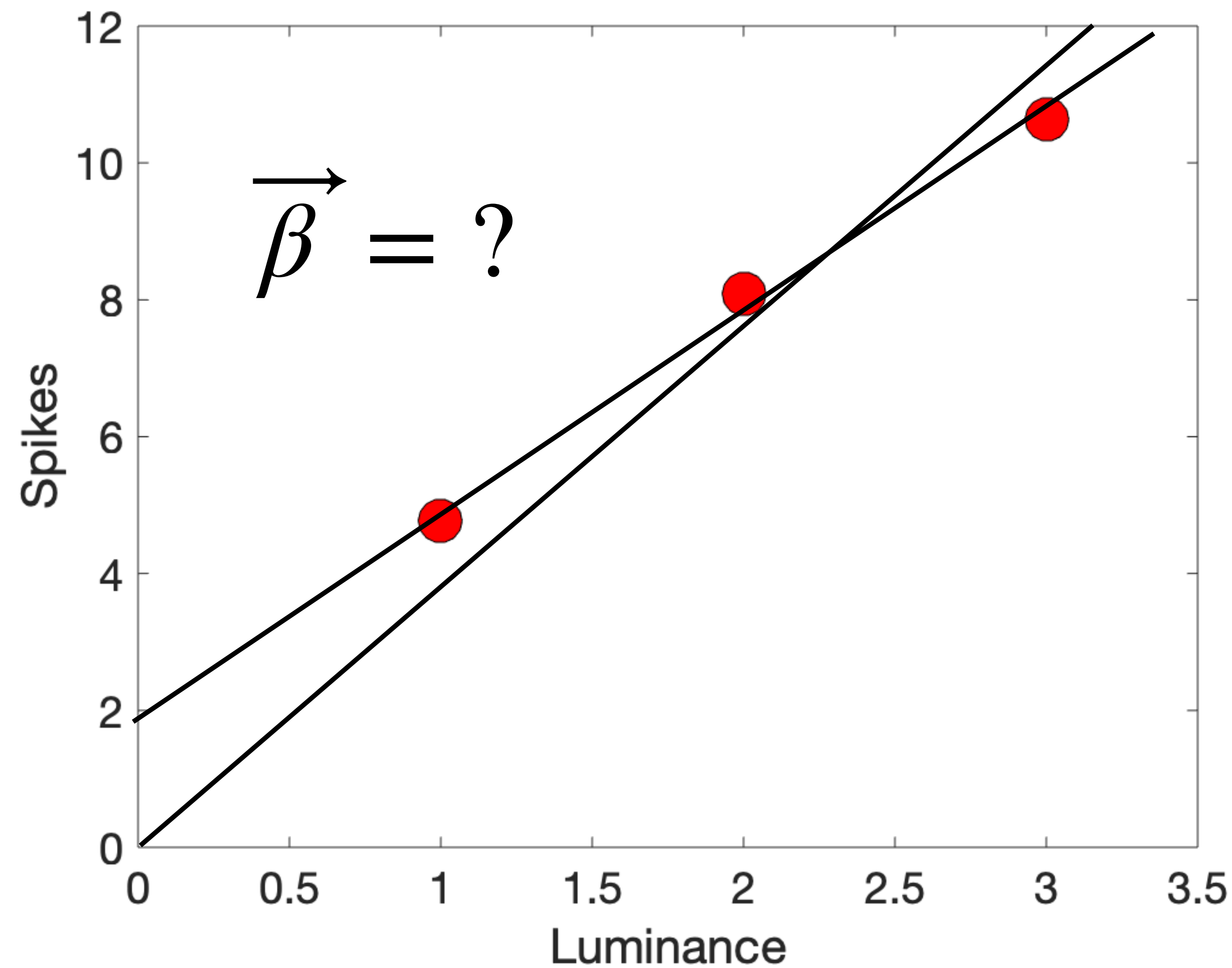
- Was this the best β?
- Recall: we say it's the optimal β by minimizing the squared distance between the actual data, and the predicted values





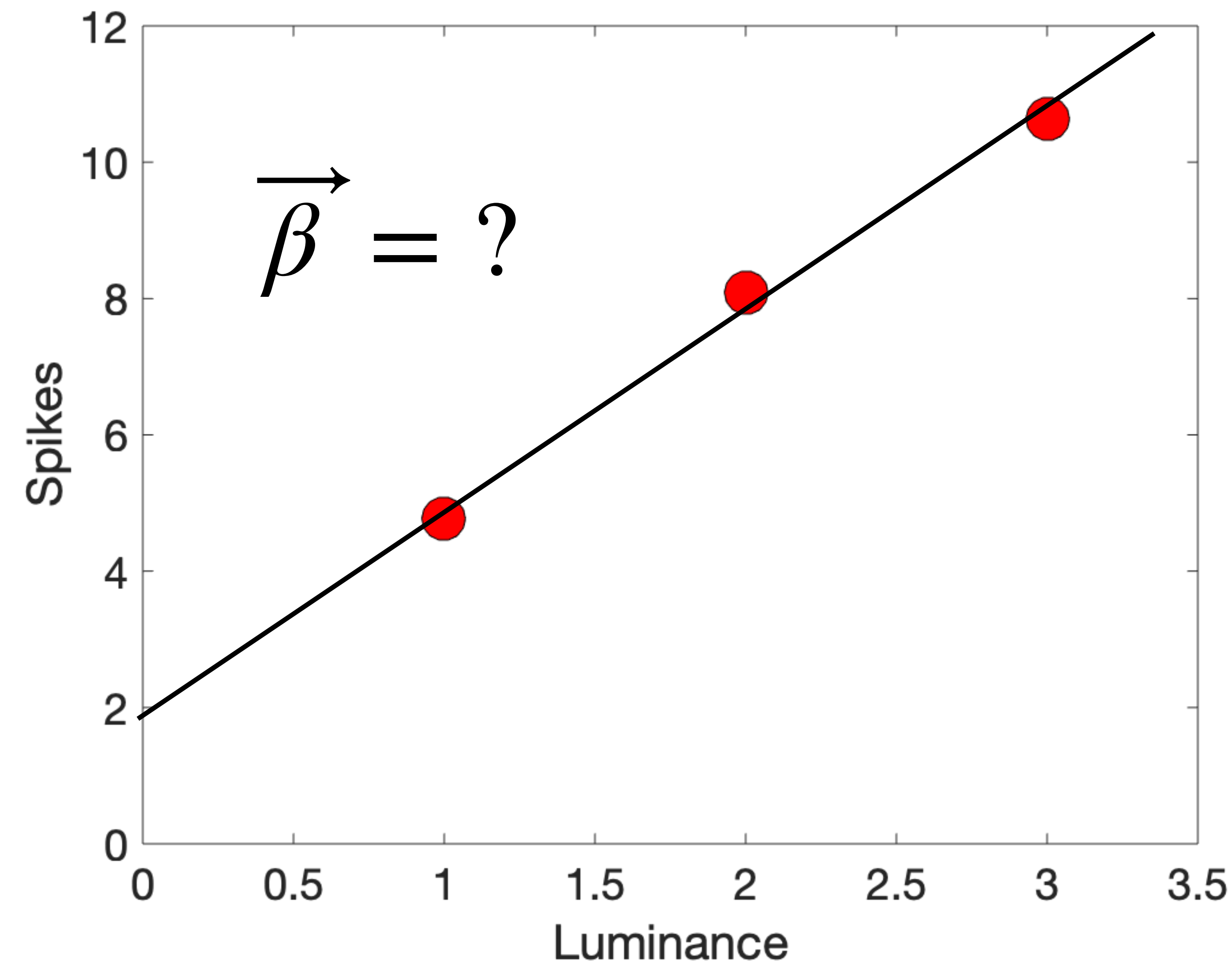
Linear Regression

- **Multiple Regression**: We would now like to take a linear combination of more than one predictor to get the output.
- Looking at the neuron again, what if a better line had a y-intercept?



Linear Regression

- **Multiple Regression**: We would now like to take a linear combination of more than one predictor to get the output.
- Looking at the neuron again, what if a better line had a y-intercept?
- Now we would like to know not only what's the best number to multiply the luminance values but also a number to add to all these values to predict the spikes



Before we had:

$\overrightarrow{y} = \beta \overrightarrow{x}$

Now we have:

$\overrightarrow{y} = \beta_0 + \beta_1 \overrightarrow{x}$

or

$\overrightarrow{y} = [1 \ \overrightarrow{x}]\beta = X\beta$

Linear Regression

- **Multiple Regression**: We would now like to take a linear combination of more than one predictor to get the output.
- Looking at the neuron again, what if a better line had a y-intercept?

1. Take SVD of X

Before we had:

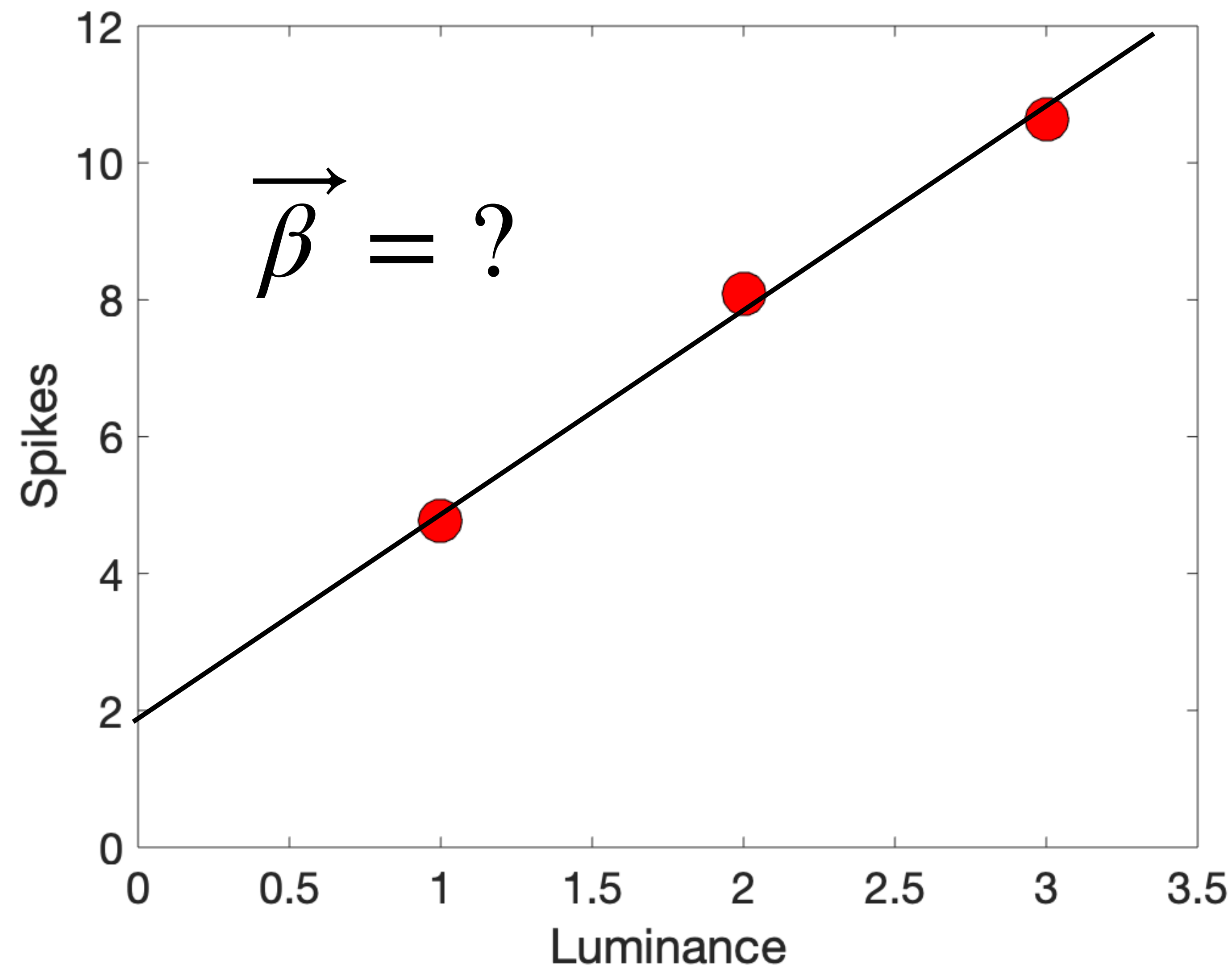
$$\vec{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Now we have:

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

matlab: [U,S,V] = svd(x)

2. $\beta_{opt} = VS^{\#}U^{T}\vec{y}$



Linear Regression

- **Multiple Regression**: We would now like to take a linear combination of more than one predictor to get the output.
- Looking at the neuron again, what if a better line had a y-intercept?

1. Take SVD of X

Before we had:

$$\vec{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

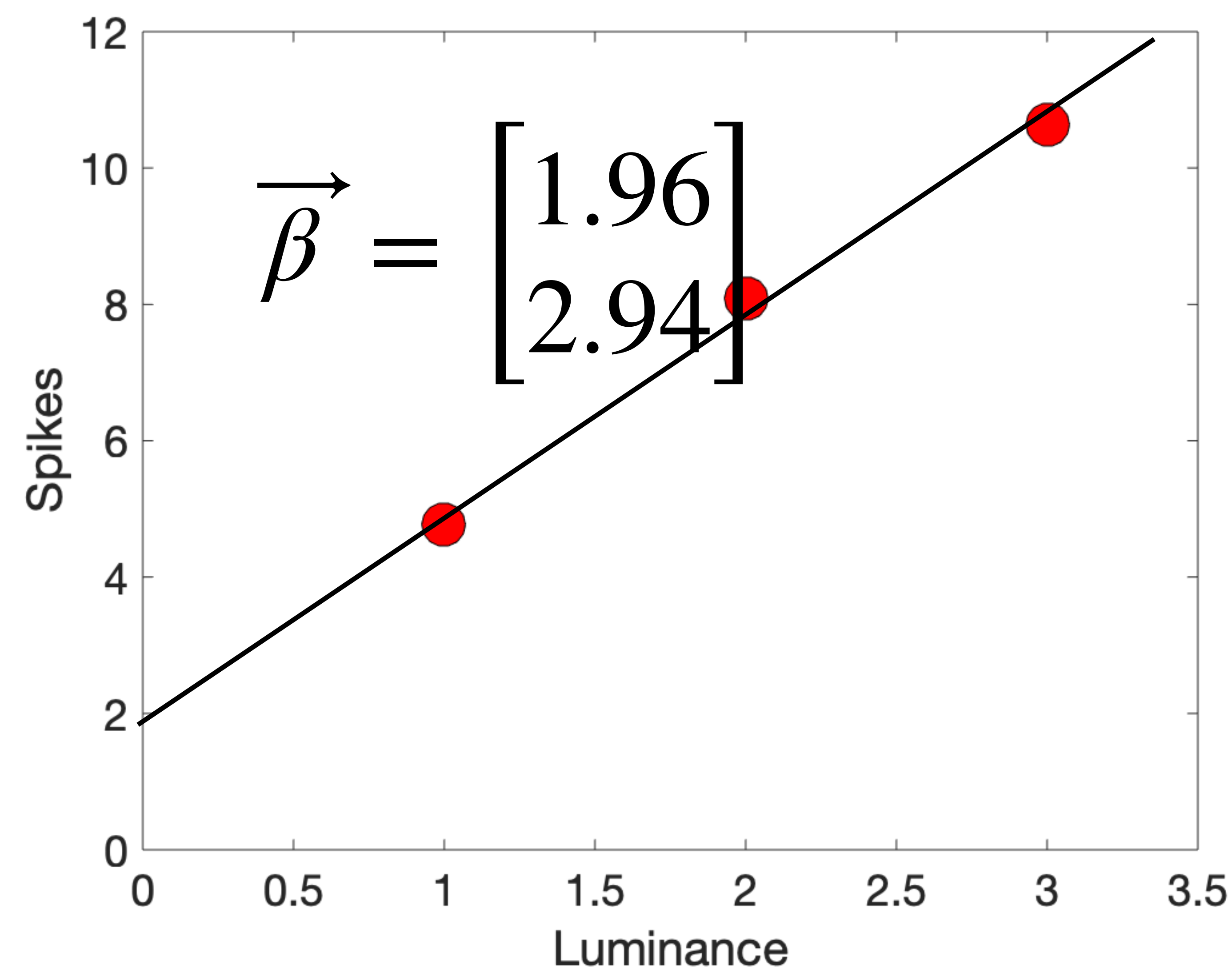
Now we have:

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

matlab: [U,S,V] = svd(x)

2. $\beta_{opt} = VS^{\#}U^{T}\vec{y}$

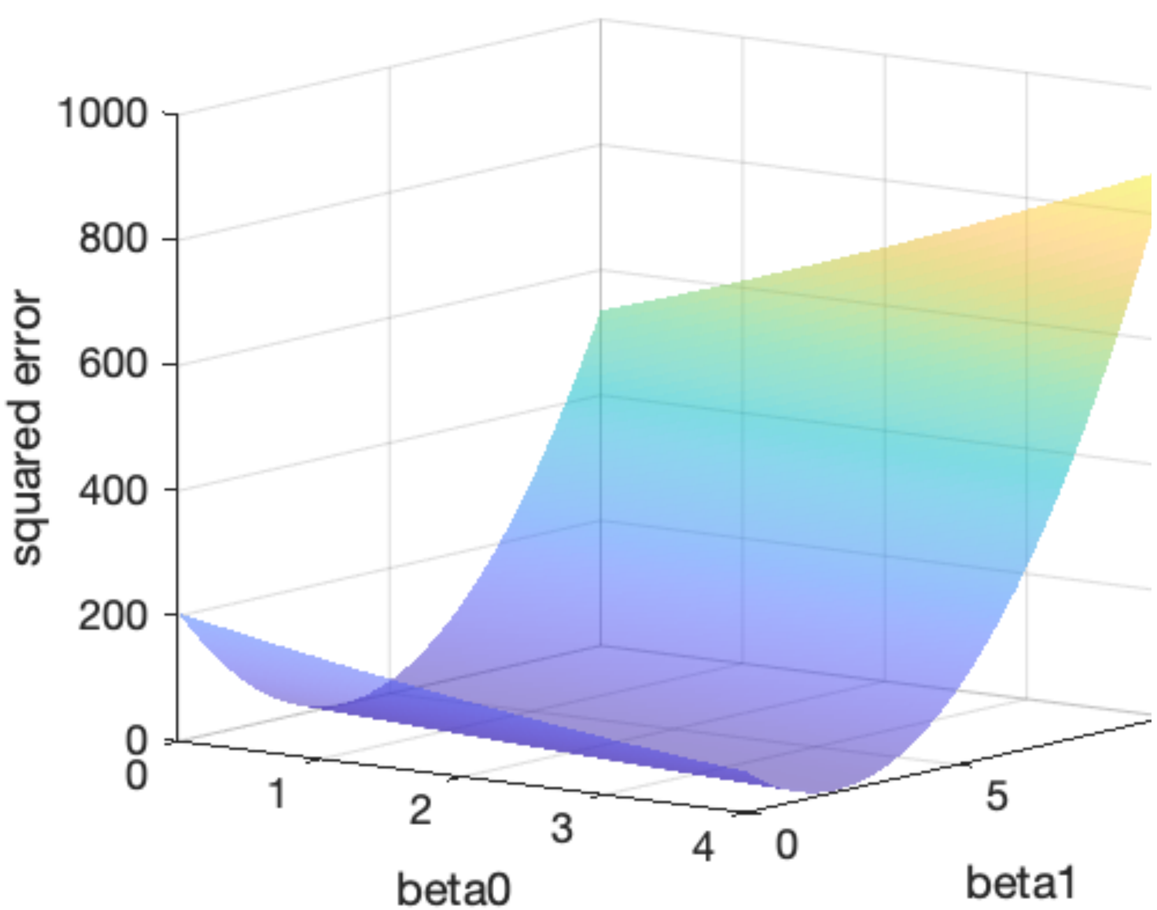
$$\beta_{opt} = VS^{\#}U^{T}\vec{y} = \begin{bmatrix} 1.96 \\ 2.94 \end{bmatrix}$$

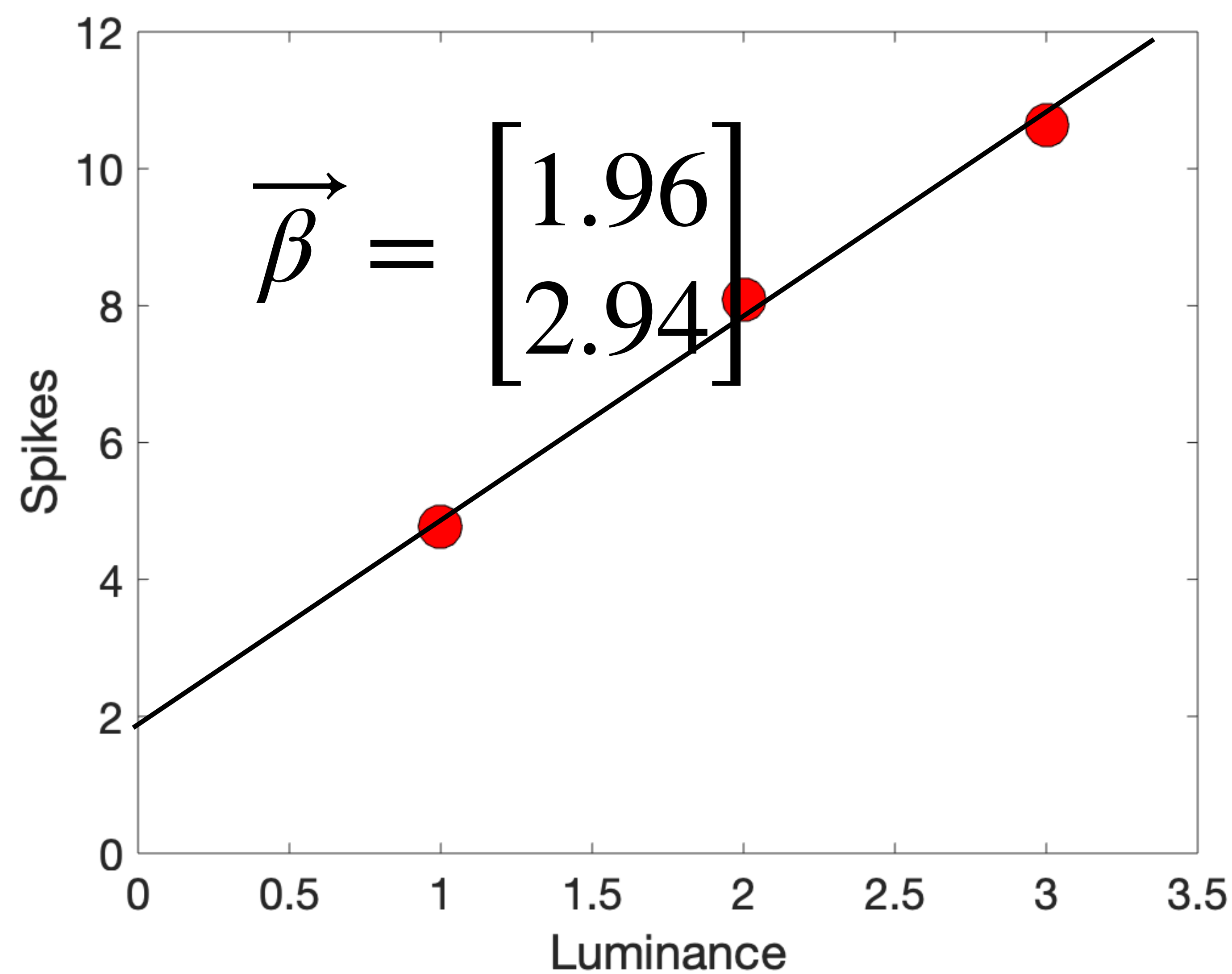


Linear Regression

$$\min_{\vec{\beta}} ||\vec{y} - X\vec{\beta}||^2$$

- Was this the best β?
- Recall: we say it's the optimal β by minimizing the squared distance between the actual data, and the predicted values





Linear Regression

- **Multiple Regression**: We would now like to take a linear combination of more than one predictor to get the output.
- Let's say we are trying to measure the response of an auditory neuron to intensities of two different frequencies (low frequencies are low pitch, high frequencies are high pitch)
- We would like to model this neuron as a linear system, where we use the amount of each frequency to predict the neuron's firing rate

- What linear combination of each frequency best predicts the firing rate?

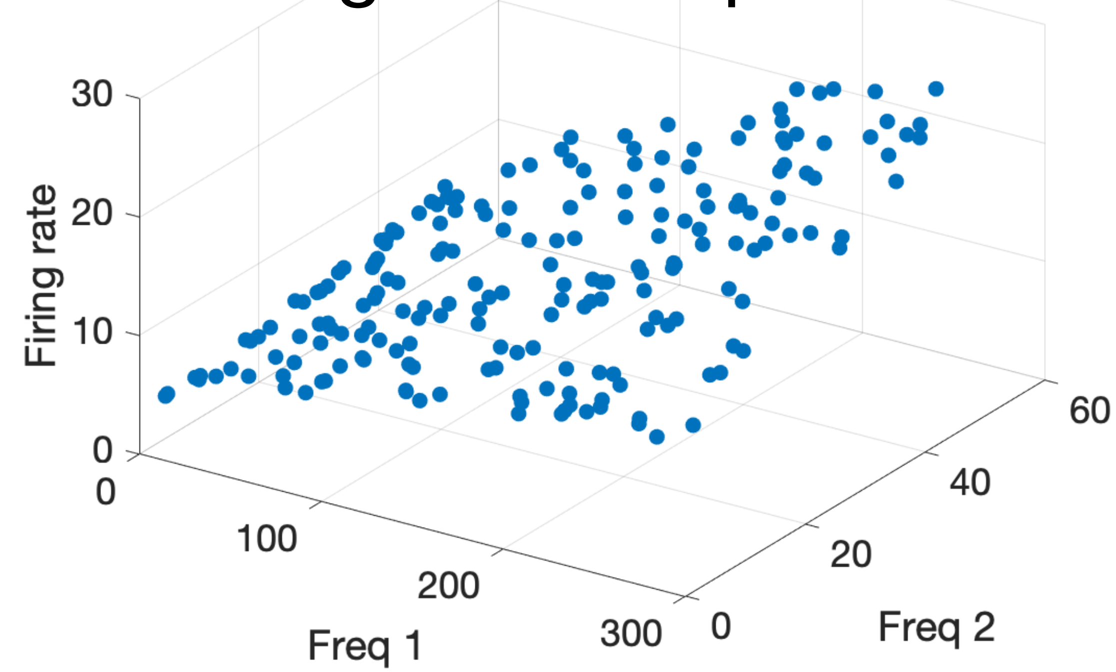
1. Take SVD of X
2. $\beta_{opt} = VS^{\#}U^{T}\overrightarrow{y}$

Intensity freq 1	Intensity freq 2	Firing Rate
230.1	37.8	22.1
44.5	39.3	10.4
17.2	45.9	9.3
151.5	41.3	18.5
180.8	10.8	12.9
8.7	48.9	7.2
57.5	32.8	11.8

Linear Regression

- **Multiple Regression**: We would now like to take a linear combination of more than one predictor to get the output.

- Now we would like to fit a plane that can best describe the data



1. Take SVD of X

$$X = \begin{bmatrix} 230.1 & 37.8 \\ 44.5 & 39.3 \\ 17.2 & 45.9 \\ \vdots & \vdots \end{bmatrix}$$

matlab: [U,S,V] = svd(x)

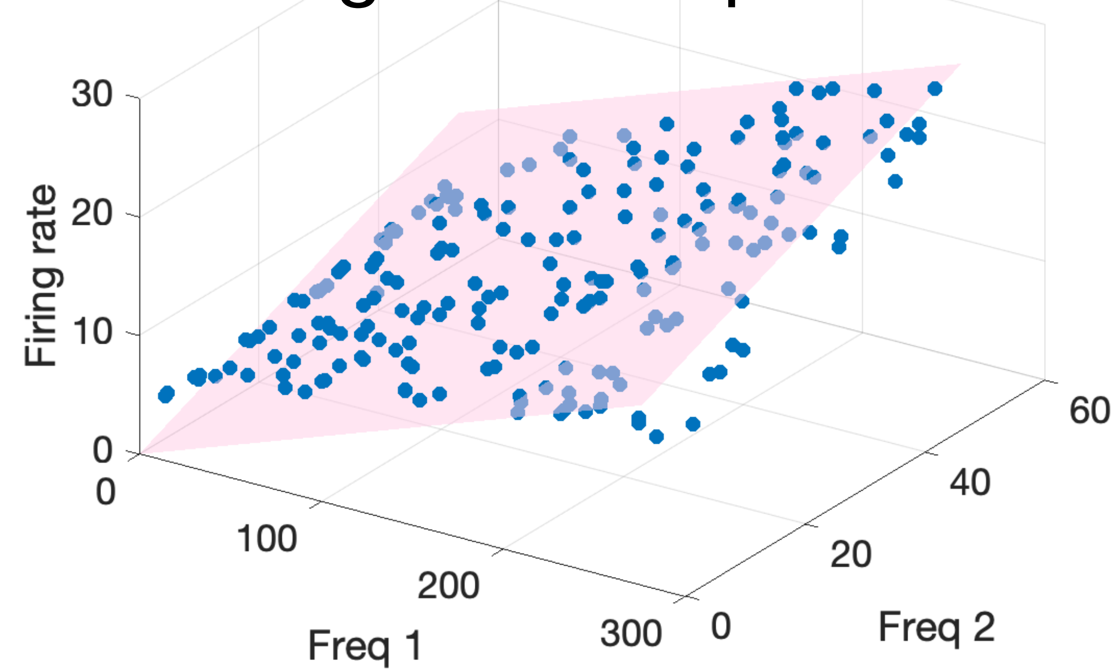
2. $\beta_{opt} = VS^{\#}U^T\overrightarrow{y}$

Intensity freq 1	Intensity freq 2	Firing Rate
230.1	37.8	22.1
44.5	39.3	10.4
17.2	45.9	9.3
151.5	41.3	18.5
180.8	10.8	12.9
8.7	48.9	7.2
57.5	32.8	11.8

Linear Regression

- **Multiple Regression**: We would now like to take a linear combination of more than one predictor to get the output.

- Now we would like to fit a plane that can best describe the data



1. Take SVD of X

$$X = \begin{bmatrix} 230.1 & 37.8 \\ 44.5 & 39.3 \\ 17.2 & 45.9 \\ \vdots & \vdots \end{bmatrix}$$

matlab: [U,S,V] = svd(x)

*how would you add a z-intercept?

2. $\beta_{opt} = VS^{\#}U^{T}\overrightarrow{y}$

$$\beta_{opt} = VS^{\#}U^{T}\overrightarrow{y} = \begin{bmatrix} 0.06 \\ 0.24 \end{bmatrix}$$

Intensity freq 1	Intensity freq 2	Firing Rate
230.1	37.8	22.1
44.5	39.3	10.4
17.2	45.9	9.3
151.5	41.3	18.5
180.8	10.8	12.9
8.7	48.9	7.2
57.5	32.8	11.8

Coding Exercises:

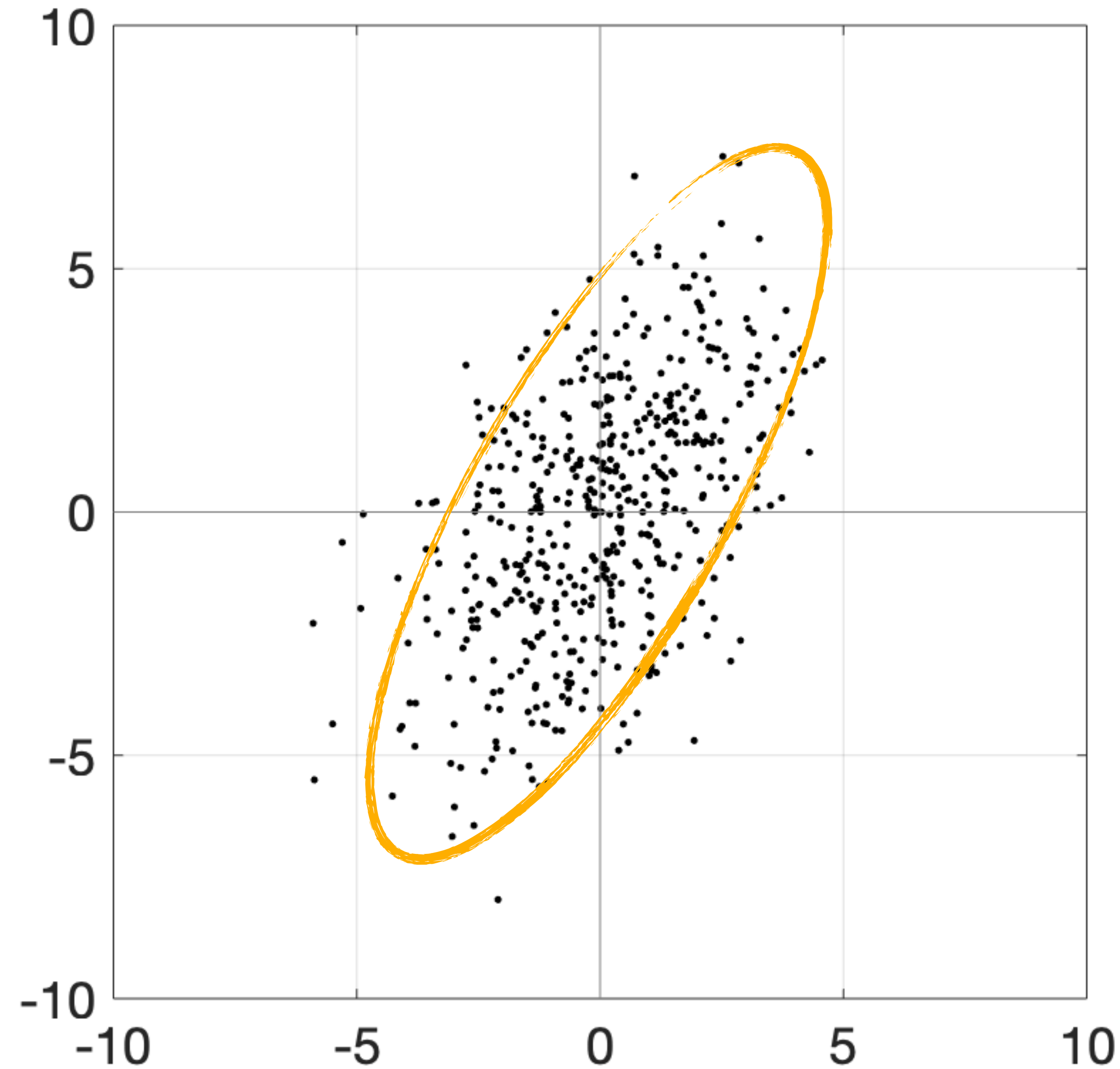
Multiple linear regression: visual neuron, auditory neuron

Principal Component Analysis (PCA)

- Often times we work with data that lives in high dimensional space
  - eg: neuron recordings for 150 ms, at 1 ms intervals — each recording is a vector living in a 150 dimensional space!)
- We would like to figure out along which dimensions we can best describe the data. By reducing to these dimensions, we then might be able to answer these questions more easily:
  - If you have 400 recordings, is there a way to distinguish how many neurons you might have recorded from?

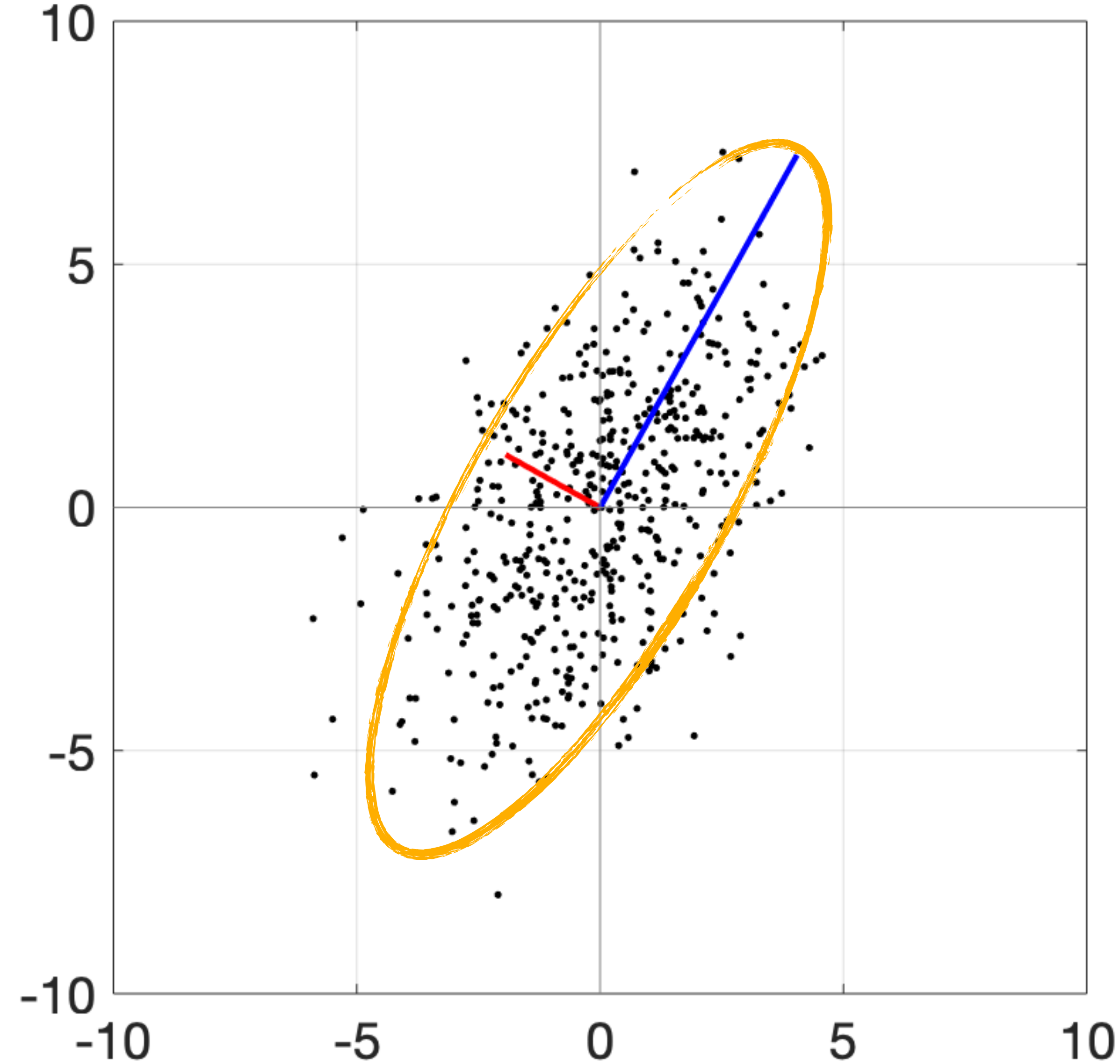
Principal Component Analysis (PCA)

- What ellipse does the best job at capturing this data?
  - If you started with a ball of data points, which directions was it squeezed/stretched?



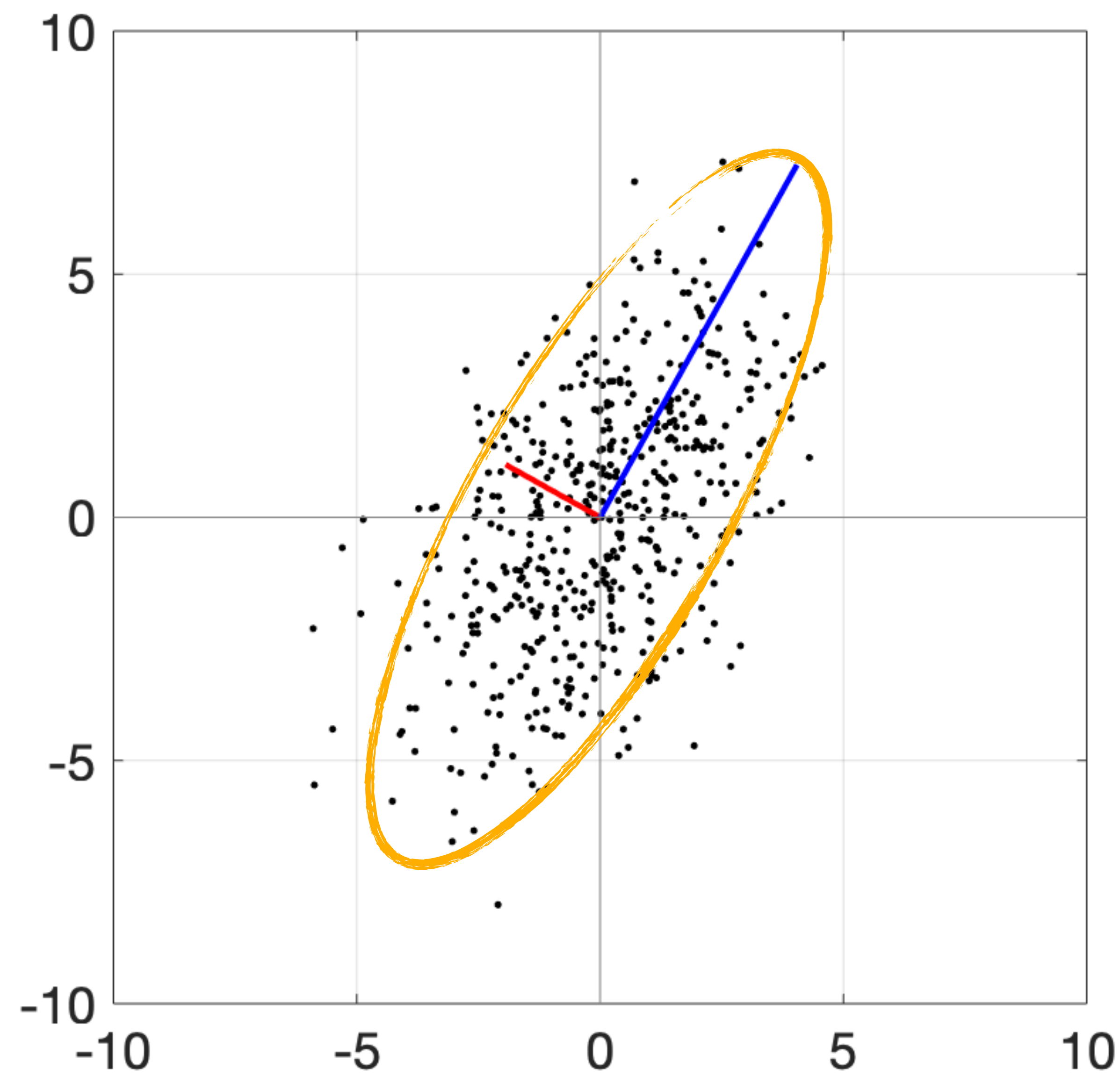
Principal Component Analysis (PCA)

- What ellipse does the best job at capturing this data?
- PCA find the directions the data was most stretched



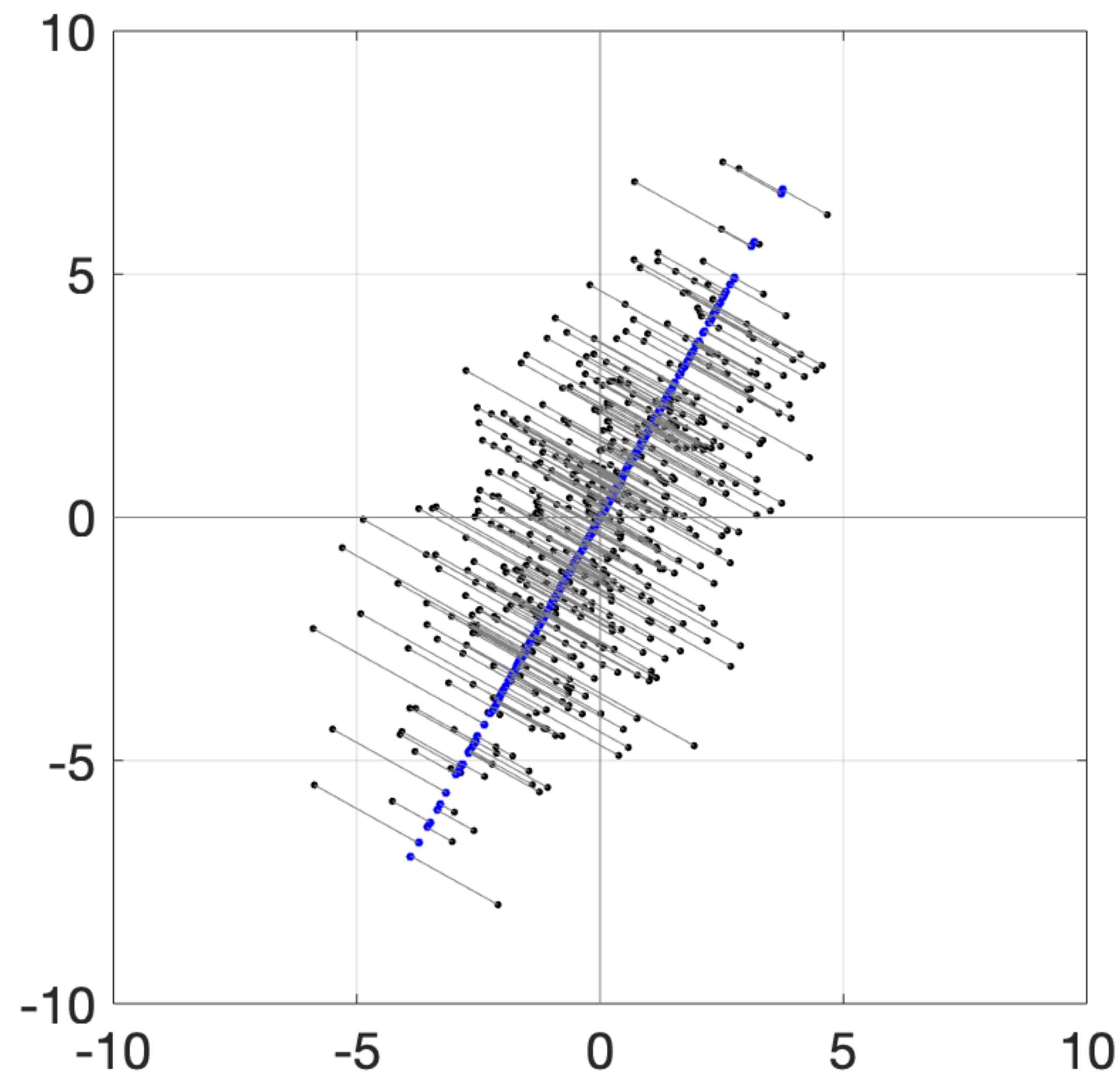
Principal Component Analysis (PCA)

- What ellipse does the best job at capturing this data?
- PCA find the directions the data was most stretched
- The directions minimize the distance to line by total least squares
  - Which is the equivalent of maximizing the spread of the points if we projected all the points down to the line
- Why is this useful?



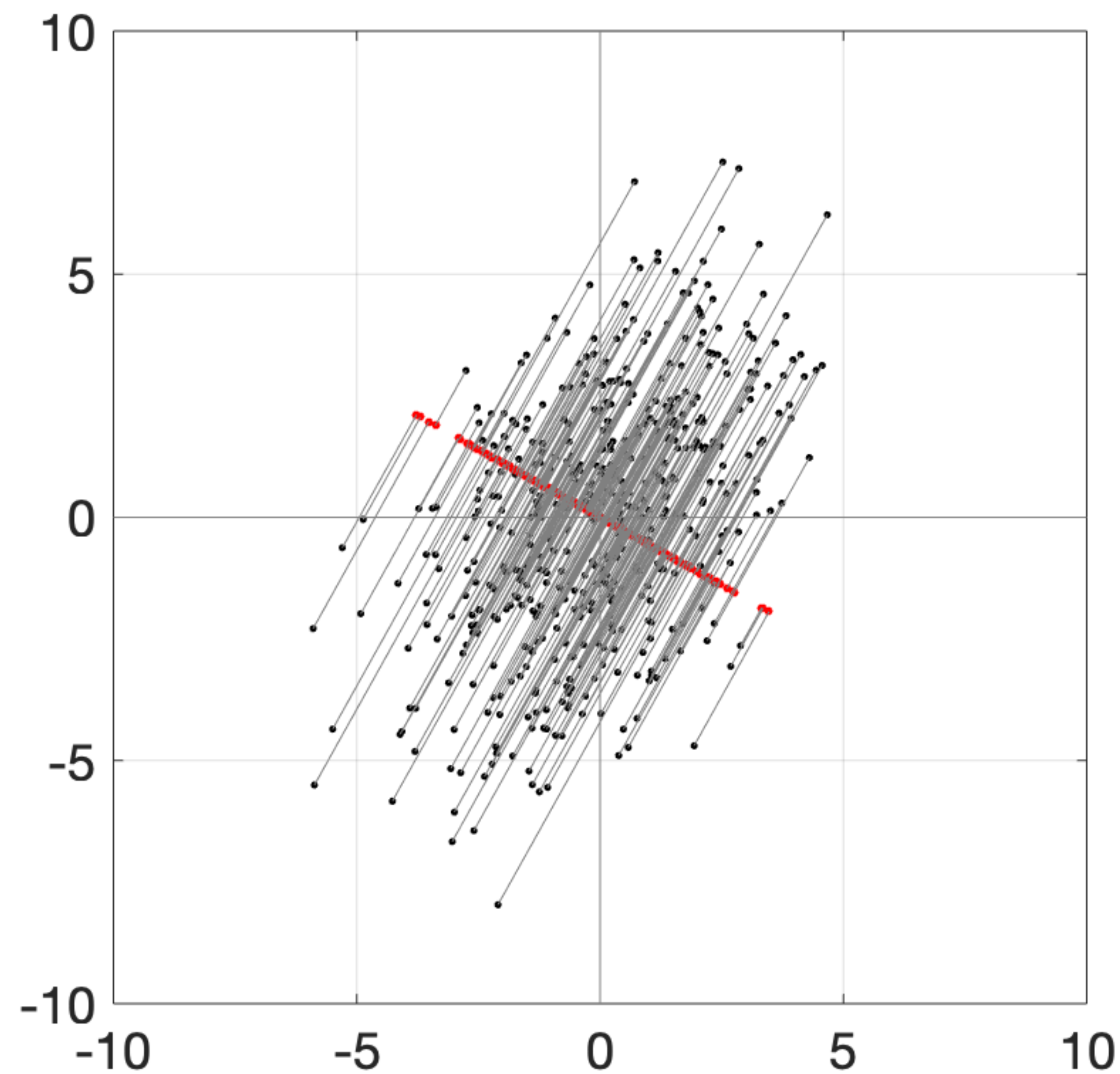
Principal Component Analysis (PCA)

- Why is this useful?
  - If we project all our data onto the first principal component, we lose information, since we have reduced the number of dimensions
  - But it seems like we have captured the direction the points are the most spread out
  - We can still distinguish between certain points



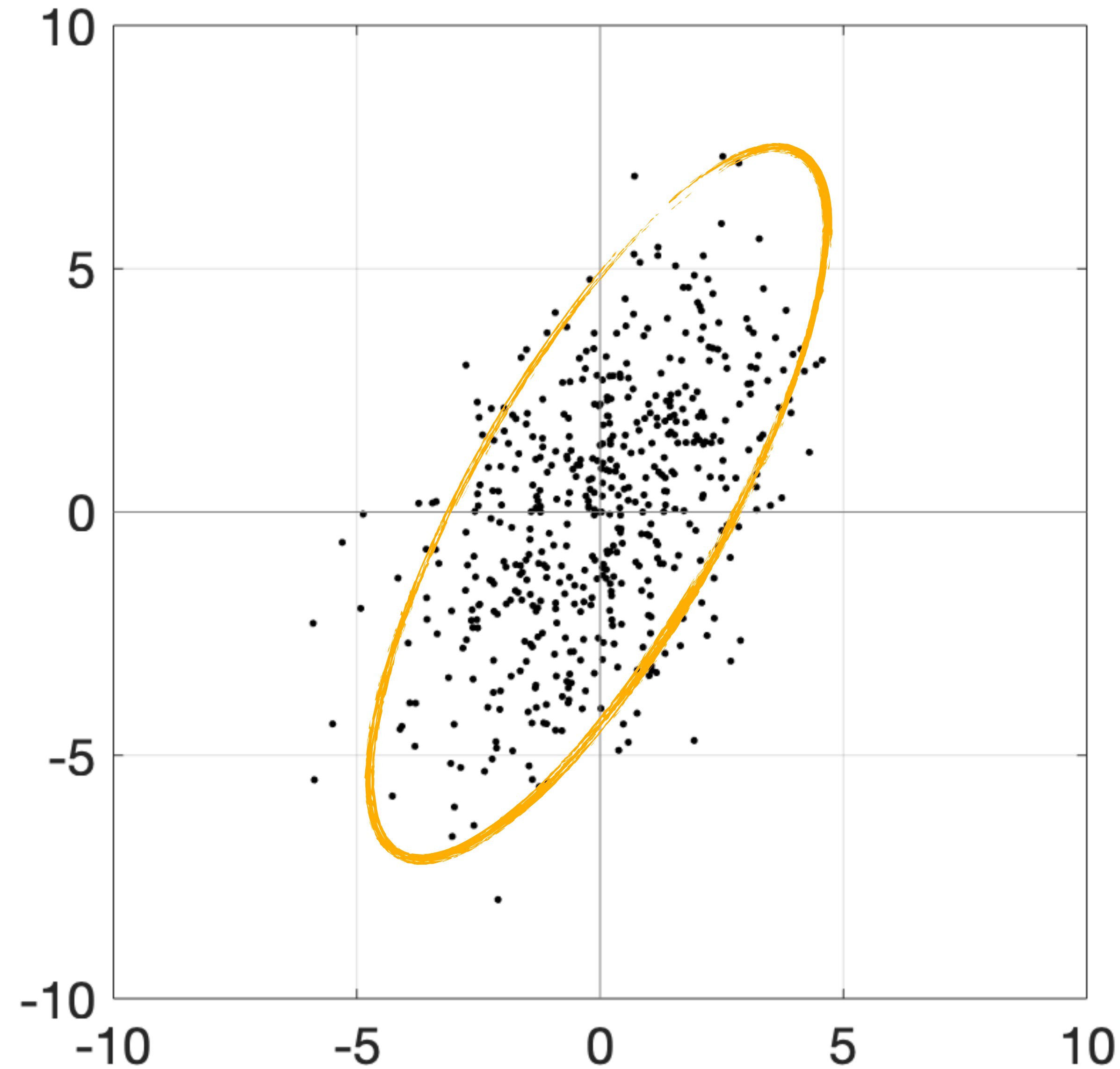
Principal Component Analysis (PCA)

- Why is this useful?
  - But if we project down to the second component somehow we've lost more information
  - When we look at the data from this perspective, there is less space between each point so they're less distinguishable from each other



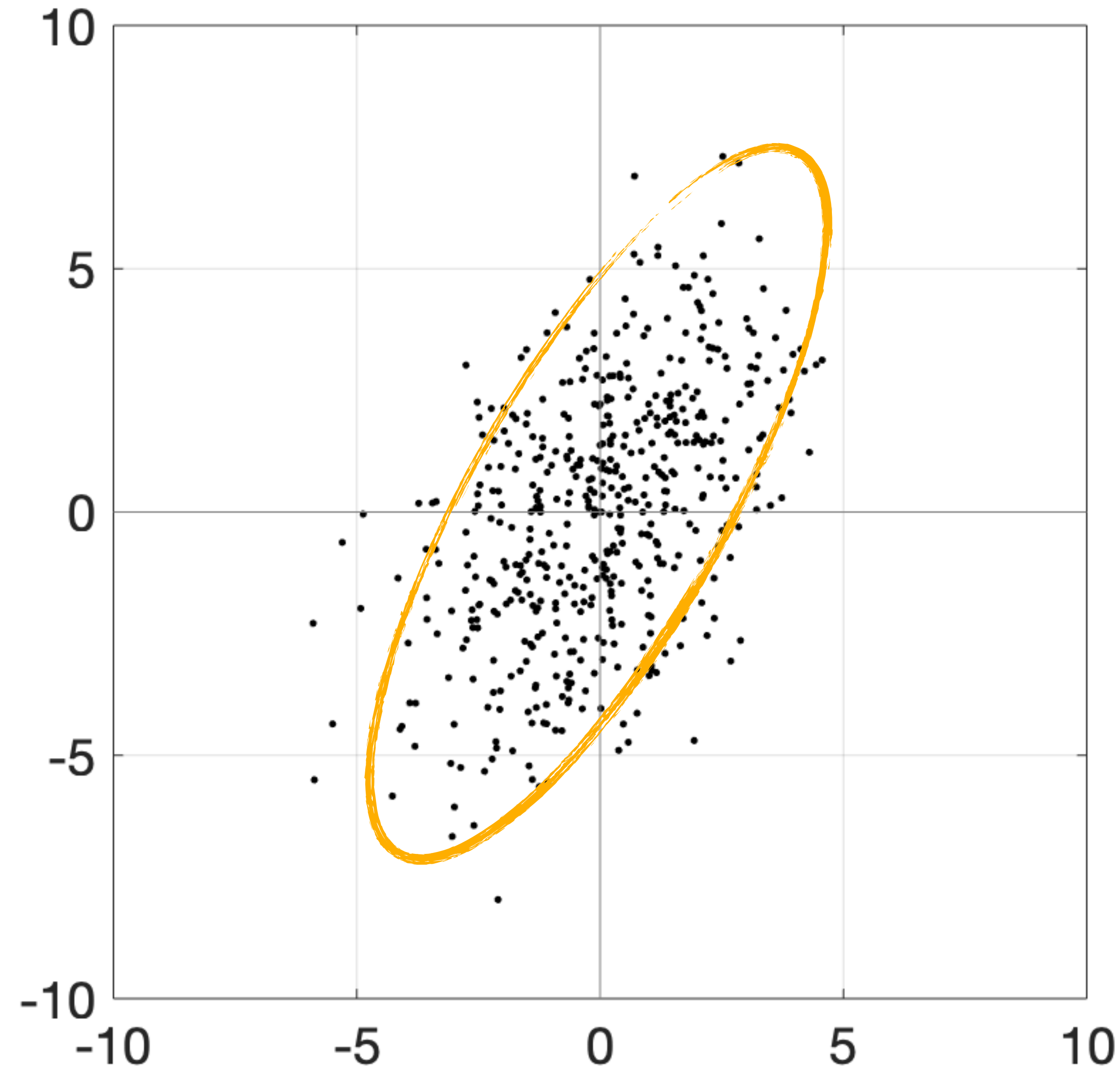
Principal Component Analysis (PCA)

- Why is this useful?
  - It is easy for us to visualize 2D data, so reducing the dimensionality down to 1D seems a little silly
  - But if we are looking at data that lives in very high dimensional space - like those neuron recordings (150 dimension) it might be nice to be able to analyze the data in a lower dimensional space



Principal Component Analysis (PCA)

- What ellipse does the best job at capturing this data?
  - If you started with a ball of data points, which directions was it squeezed/stretched?



PCA Algorithm

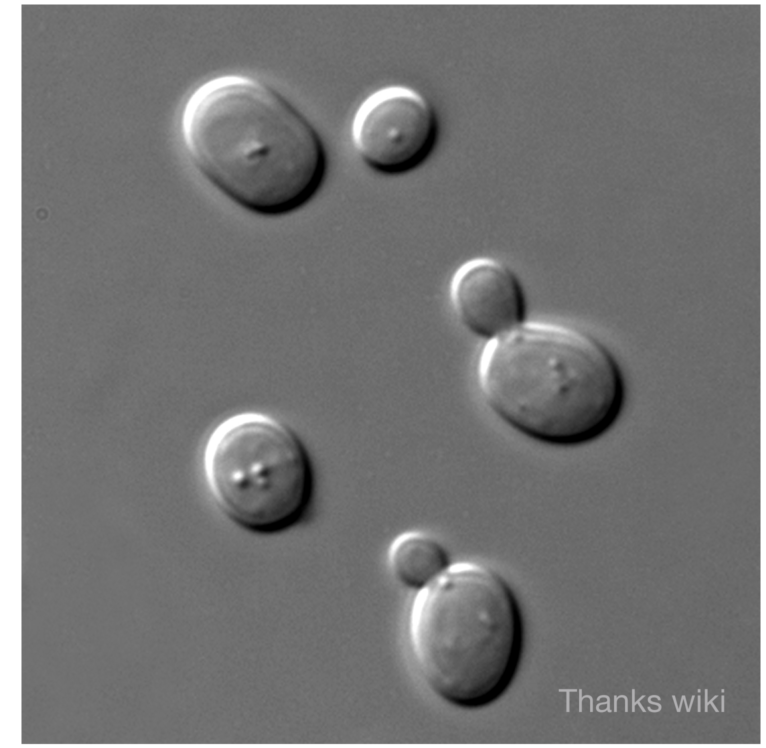
1. Center the data
2. Make sure data vectors are in rows
3. Take SVD of X

or compute eigenvectors of C = X’X

4. Columns of V are the principal components, squared singular values or eigenvalues measure variance along that dimension

Principal Component Analysis (PCA)

- Gene expression example: We have data about the expression levels of a large population of genes recorded in budding yeast
  - 208 genes under 79 experimental conditions
  - We know that these genes fall under 5 functional categories
- We would like to see if we can recover these 5 categories based on the experimental conditions we exposed the yeast to



Exp cond

Gene expression

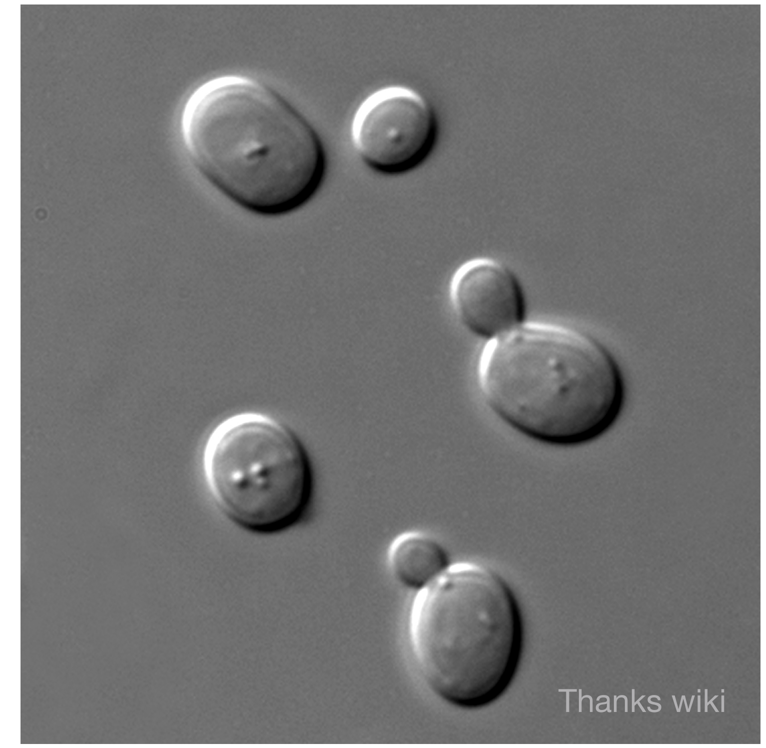
-0.25	0.31	0.07	0.34
0.44	-0.07	0.38	-0.03
0.61	-0.17	0.60	-0.09
0.30	-0.04	0.22	-0.10
-0.29	-0.17	-0.89	0.30

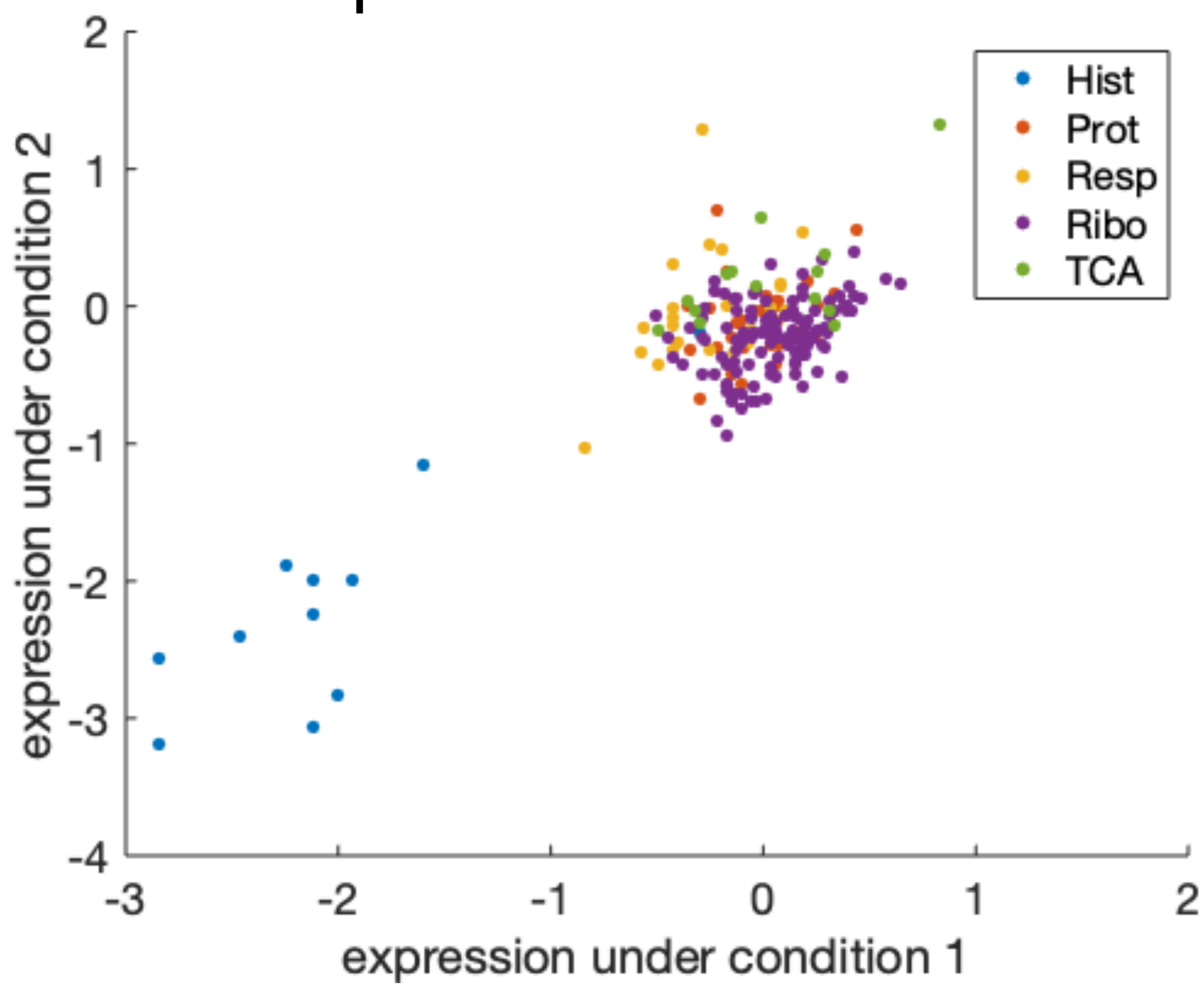
…

⋮

Principal Component Analysis (PCA)

- Gene expression example: We have data about the expression levels of a large population of genes recorded in budding yeast
- Thought: what if we just looked at the first two experimental conditions?





Exp cond

Gene expression

-0.25	0.31	0.07	0.34	
0.44	-0.07	0.38	-0.03	
0.61	-0.17	0.60	-0.09	…
0.30	-0.04	0.22	-0.10	
-0.29	-0.17	-0.89	0.30	

⋮

Principal Component Analysis (PCA)

- Gene expression example: We have data about the expression levels of a large population of genes recorded in budding yeast
- What if we include all experimental conditions?
    - …that seems kind of hard to visualize since that would be a 79 dimensional plot
    - How about looking at the data in a lower dimension?
- Using PCA let's come up with a better way of choosing specific conditions to reduce the 79 dimensions

Exp cond

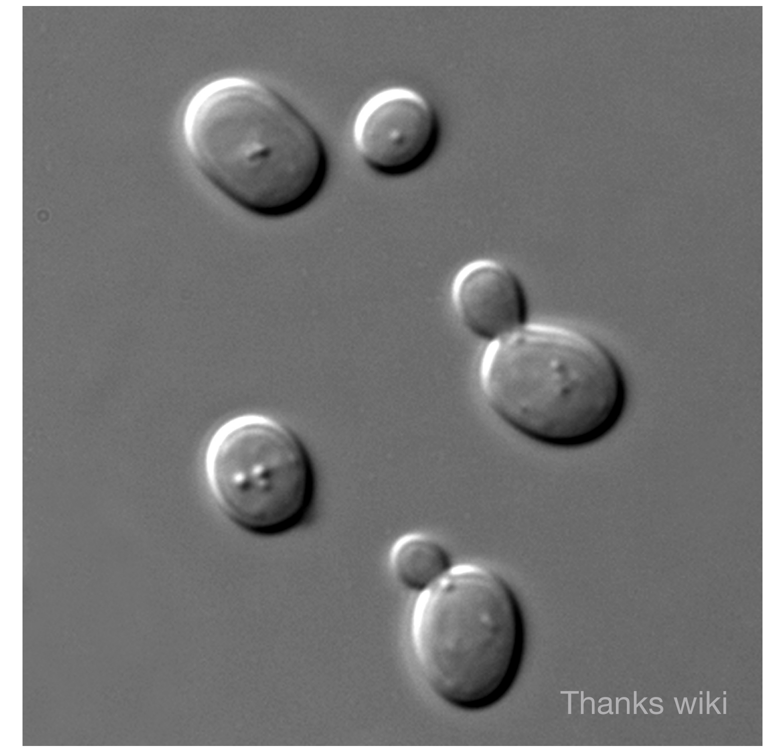
Gene expression

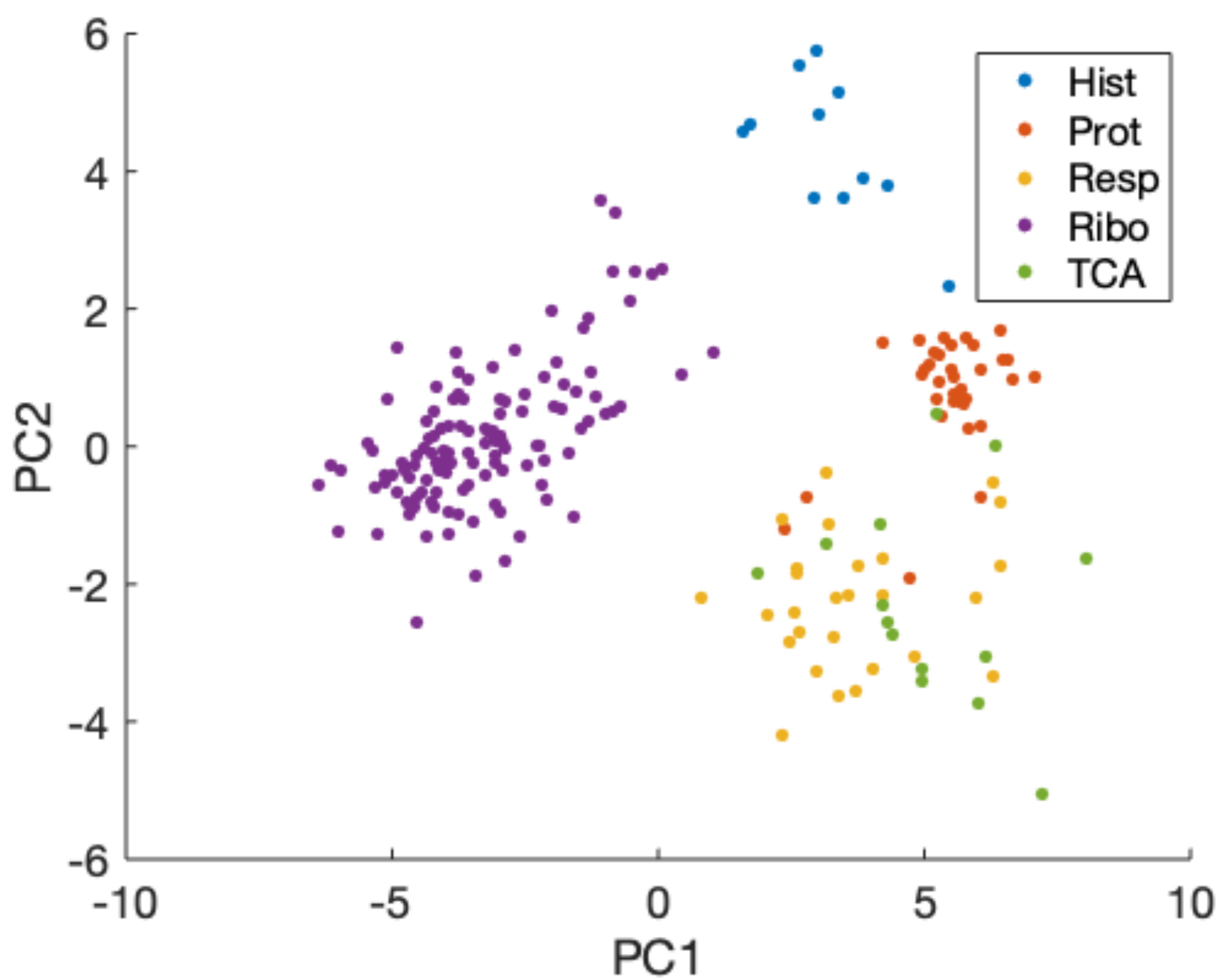
-0.25	0.31	0.07	0.34	
0.44	-0.07	0.38	-0.03	
0.61	-0.17	0.60	-0.09	…
0.30	-0.04	0.22	-0.10	
-0.29	-0.17	-0.89	0.30	

⋮

Principal Component Analysis (PCA)

- Gene expression example: We have data about the expression levels of a large population of genes recorded in budding yeast
- What did we find from PCA?





Exp cond

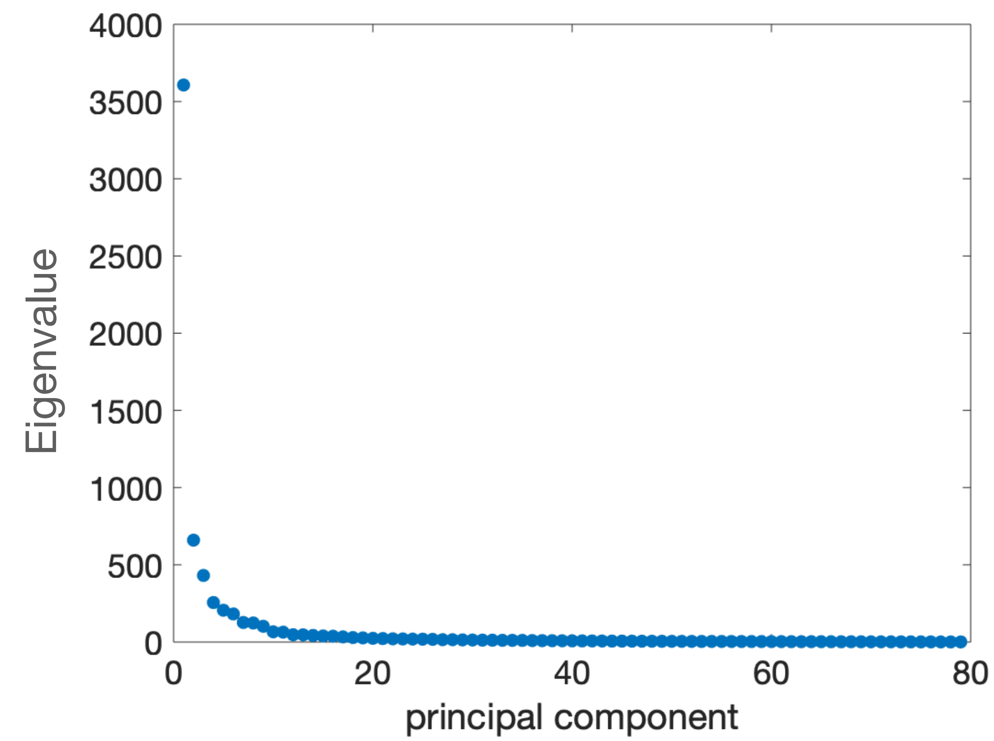
Gene expression

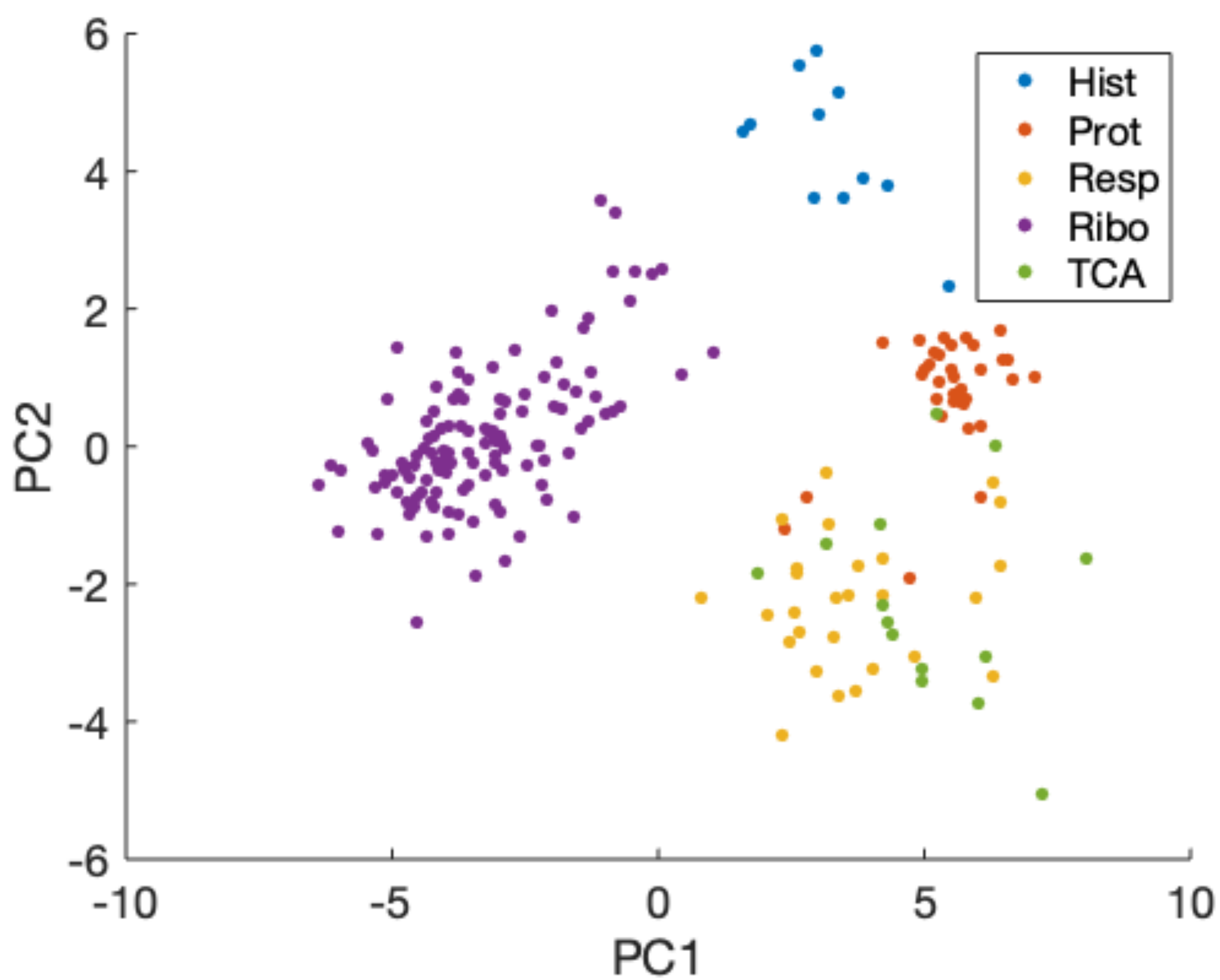
-0.25	0.31	0.07	0.34	
0.44	-0.07	0.38	-0.03	
0.61	-0.17	0.60	-0.09	…
0.30	-0.04	0.22	-0.10	
-0.29	-0.17	-0.89	0.30	

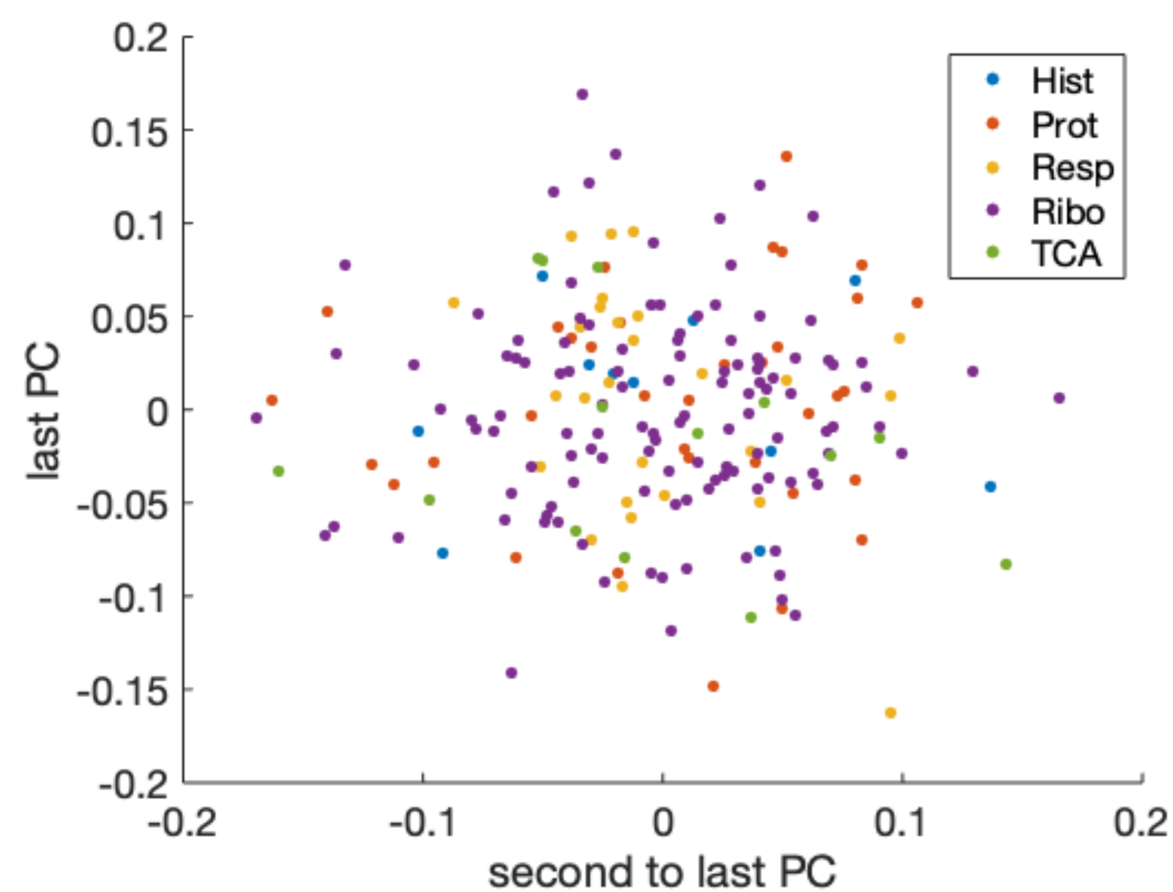
⋮

Principal Component Analysis (PCA)

- Gene expression example: We have data about the expression levels of a large population of genes recorded in budding yeast
- What did we find from PCA?

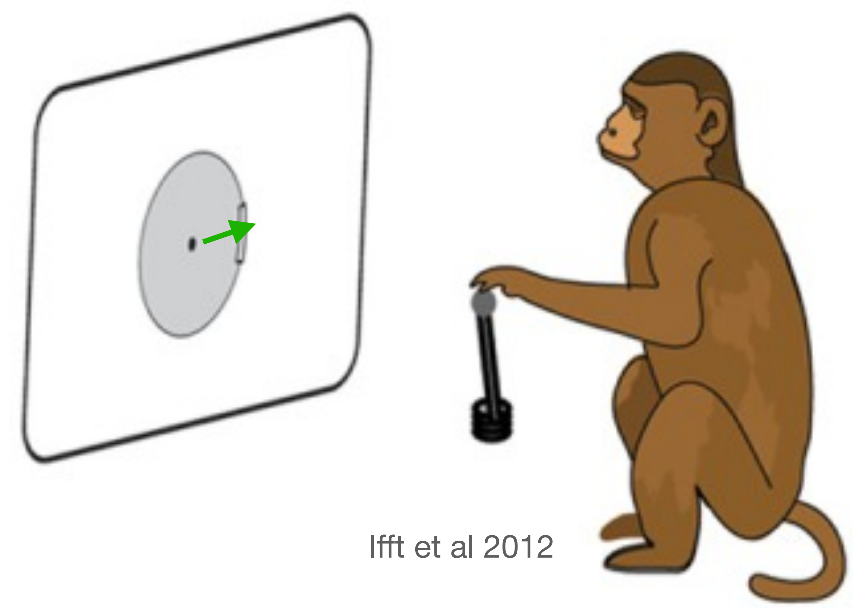






Principal Component Analysis (PCA)

- Macaque recordings example: We have neural signals recorded in macaque motor cortex during a center-out reaching task.
- The data contains 143 neurons recorded during 158 trials, and the direction for each trial
- Your job is to see if you can tell the reach direction from the neural activity



Ifft et al 2012

Coding Exercises:

Dimensionality Reduction via PCA