Searching for Population Structure

Principal Component Analysis and Clustering

Phillip Compeau

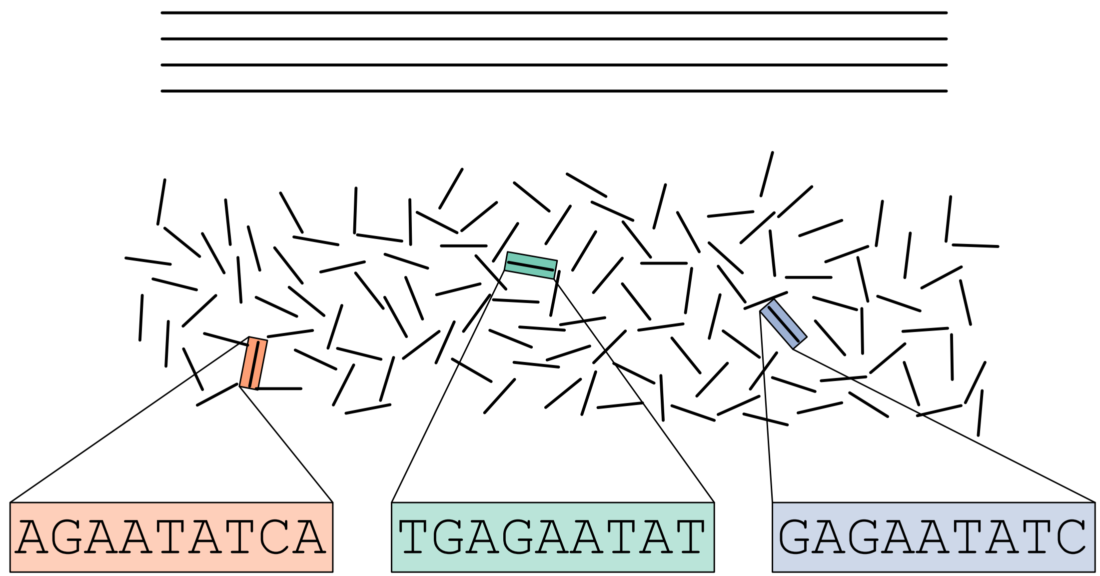
Recall: Mapping Reads against Reference

Multiple identical copies of a genome

Shatter the genome into reads

Sequence the reads

(Lab)



Then, we "map" these reads against a reference human genome (the most commonly used reference is 70% RP11, or "some guy from Buffalo").

Another Aim: Understanding "Population Structure"

Population structure: genetic differences between subpopulations in a population of individuals (i.e., the human species).

Another Aim: Understanding "Population Structure"

Population structure: genetic differences between subpopulations in a population of individuals (i.e., the human species).

Checkpoint: any thoughts on how we could use existing approaches we have learned to find population structure?

Another Aim: Understanding "Population Structure"

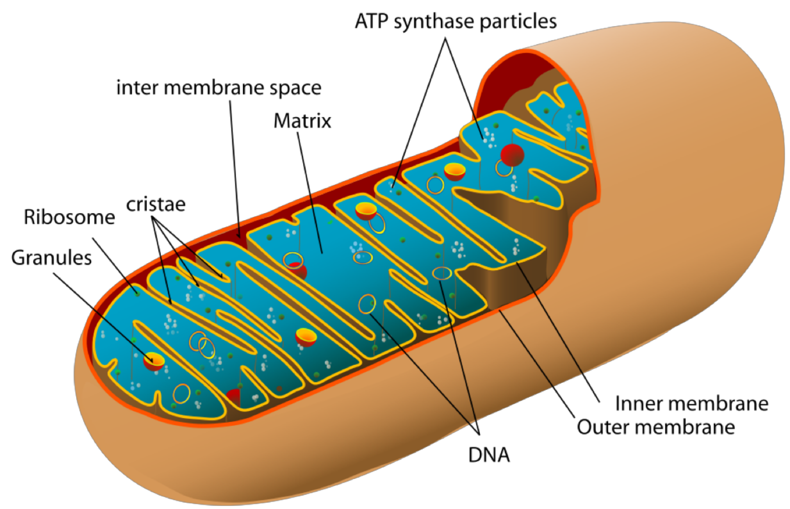
Population structure: genetic differences between subpopulations in a population of individuals (i.e., the human species).

Checkpoint: any thoughts on how we could use existing approaches we have learned to find population structure?

This sounds a lot like evolutionary tree construction.

Mitochondrial Sequencing Reveals Population Structure

Mitochondrial genome: a 16,569 base-pair circular chromosome replicated independently of "nuclear DNA" in mitochondria and inherited maternally.

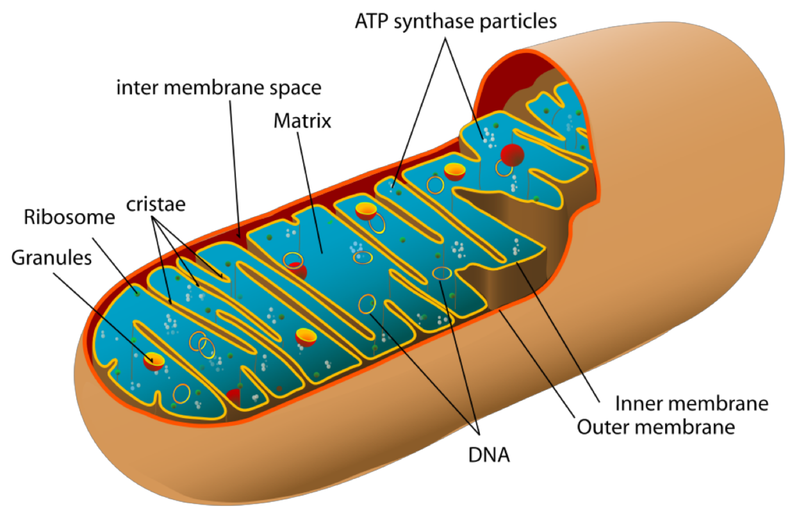


https://commons.wikimedia.org/wiki/Mitochondrion#/media/File:Animal_mitochondrion_diagram_en.svg

Mitochondrial Sequencing Reveals Population Structure

Mitochondrial genome: a 16,569 base-pair circular chromosome replicated independently of "nuclear DNA" in mitochondria and inherited maternally.

Checkpoint: Where do you think that mitochondria came from?

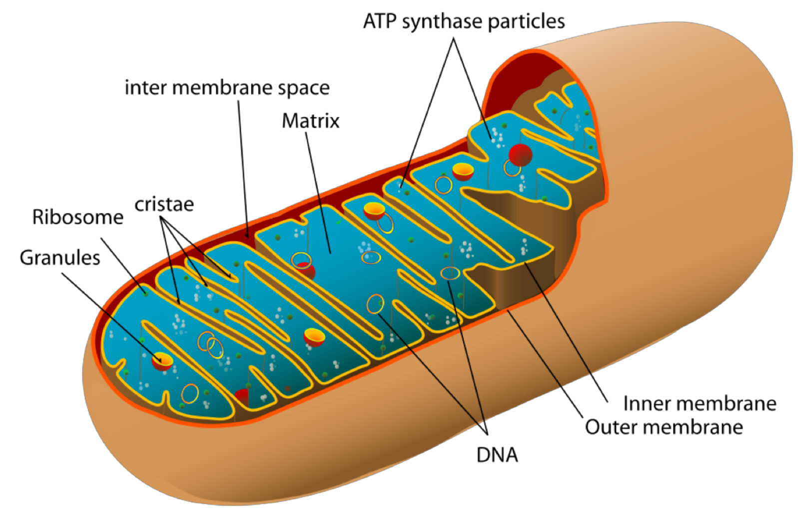


https://commons.wikimedia.org/wiki/Mitochondrion#/media/File:Animal_mitochondrion_diagram_en.svg

Mitochondrial Sequencing Reveals Population Structure

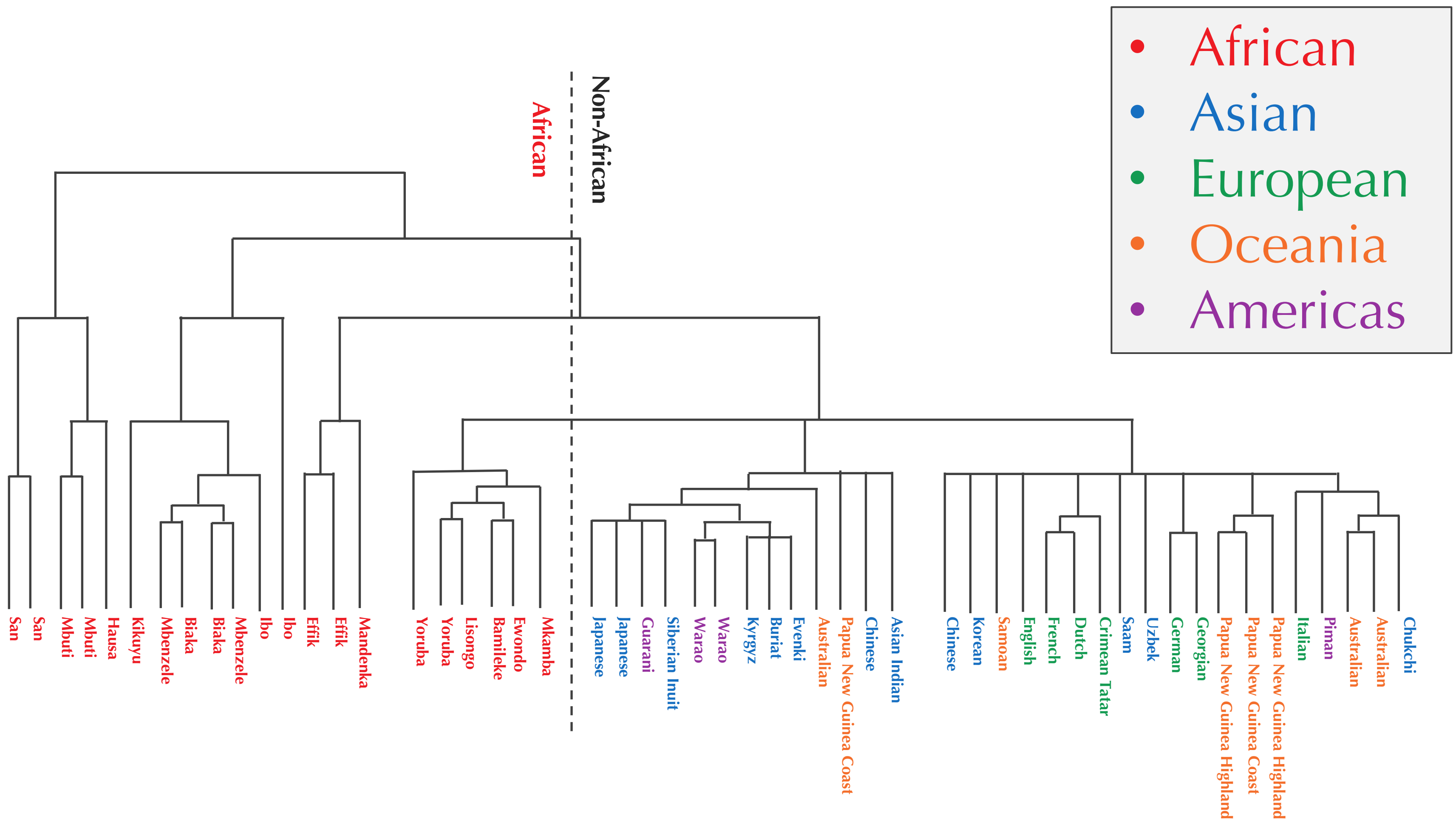
Mitochondrial genome: a 16,569 base-pair circular chromosome replicated independently of "nuclear DNA" in mitochondria and inherited maternally.

Note: "mtDNA" was used in human studies before cheap full genome sequencing because it is abundant in cells and short.

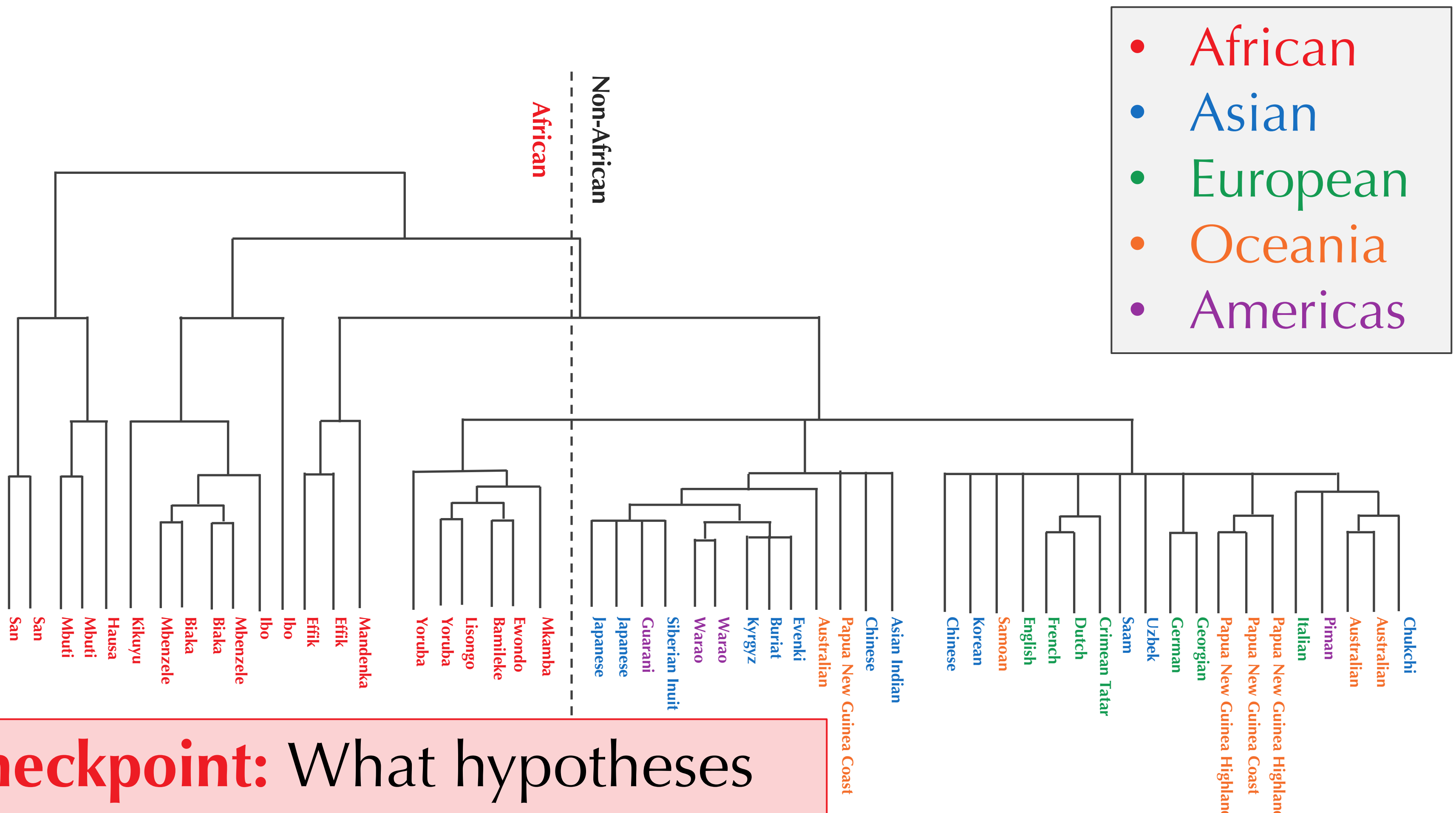


https://commons.wikimedia.org/wiki/Mitochondrion#/media/File:Animal_mitochondrion_diagram_en.svg

Mitochondrial Sequencing Reveals Population Structure

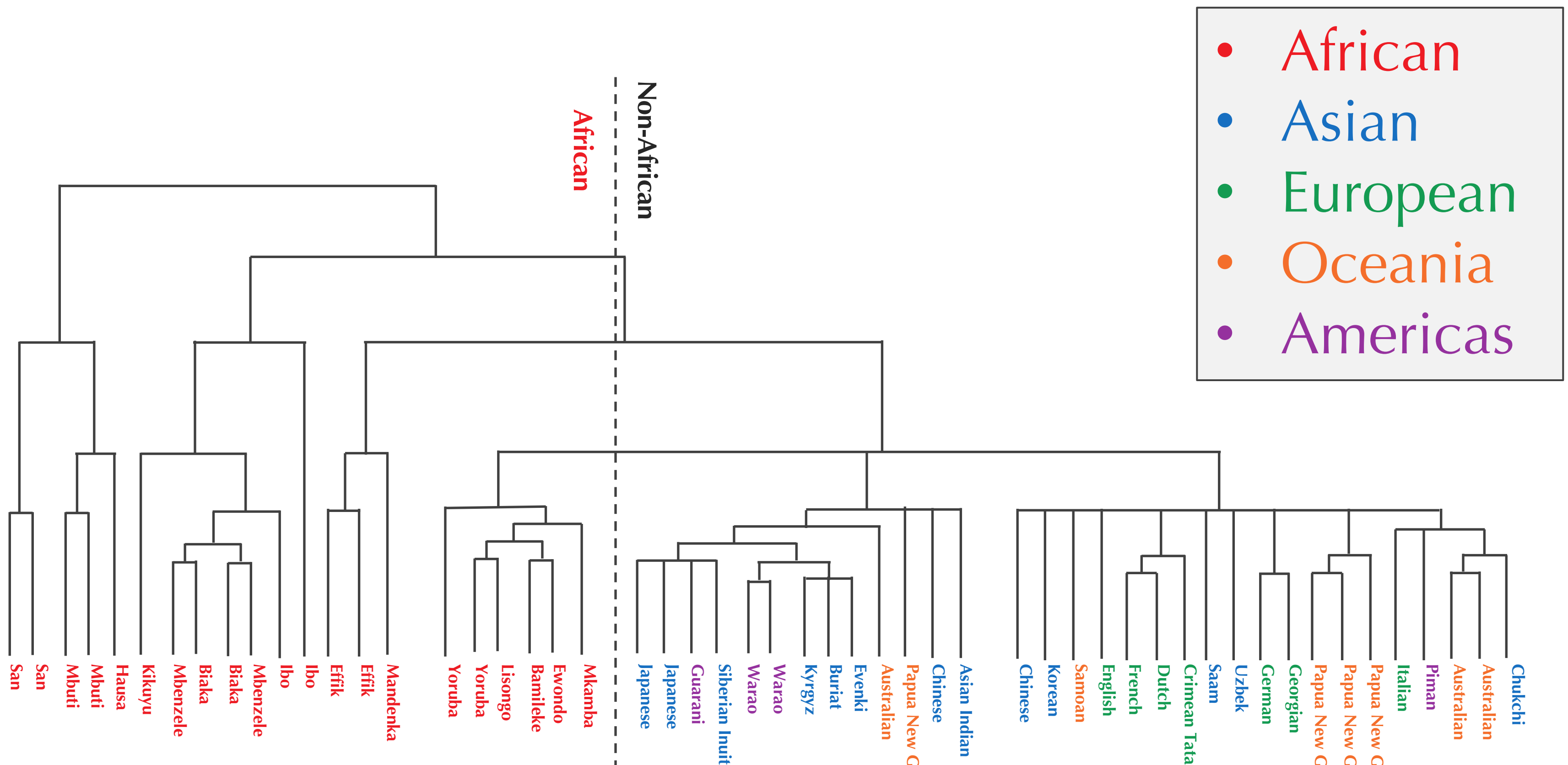


Mitochondrial Sequencing Reveals Population Structure



Checkpoint: What hypotheses would you make from this?

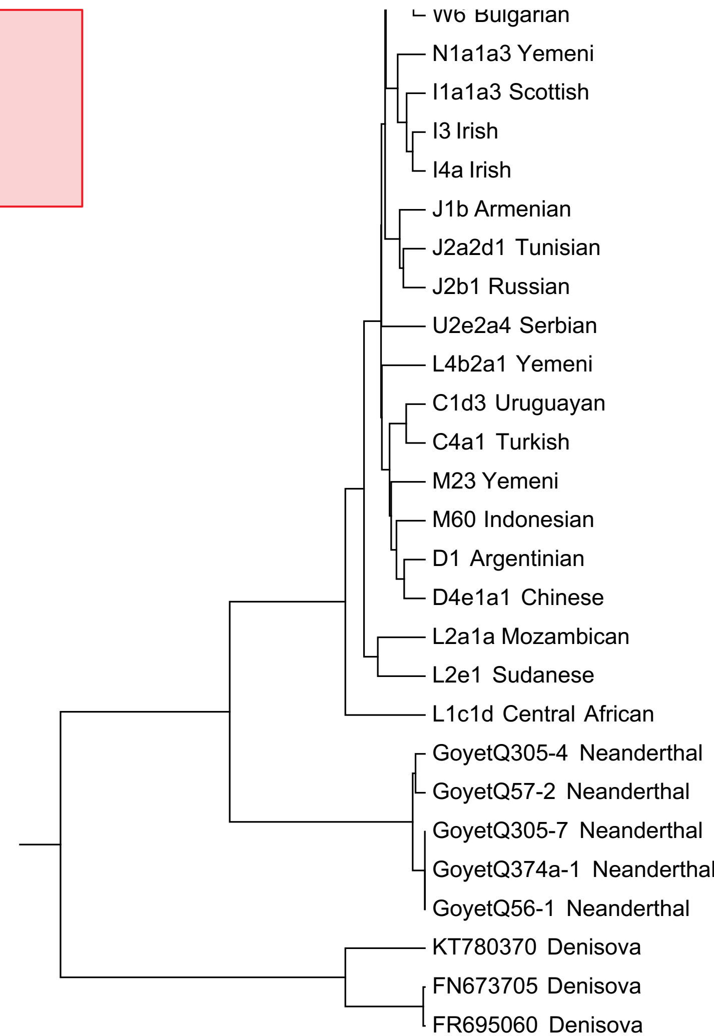
Mitochondrial Sequencing Reveals Population Structure



Out of Africa Hypothesis: All non-Africans are descended from a migration ~70,000 years ago.

Adding Neanderthals/Denisovans to the Mix

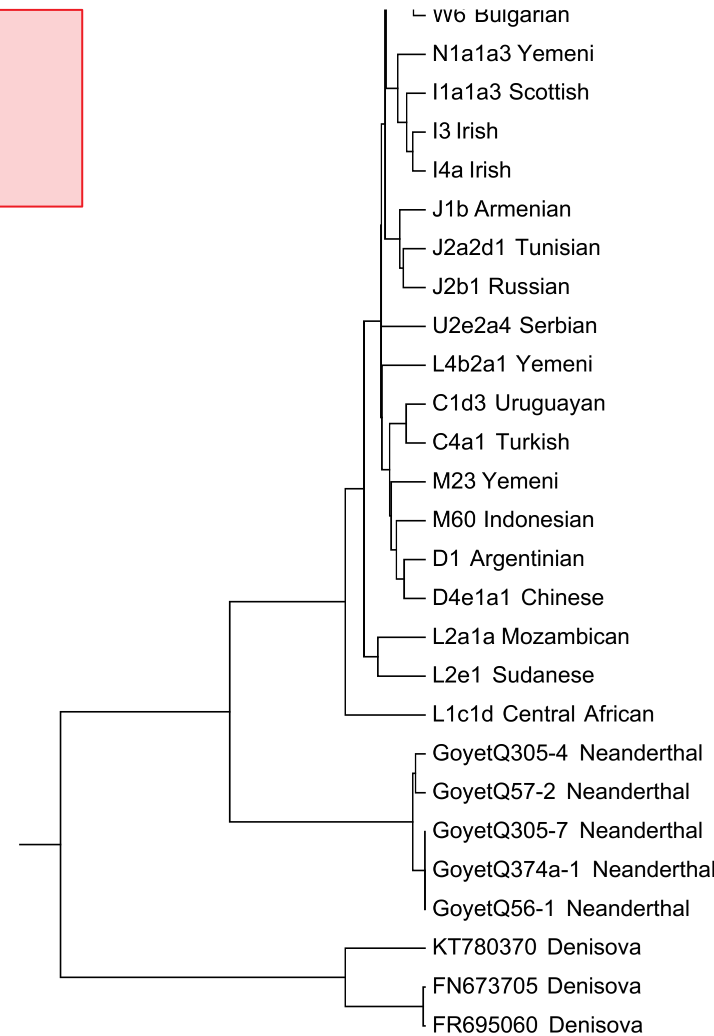
Checkpoint: What hypotheses would you make from this?



Adding Neanderthals/Denisovans to the Mix

Checkpoint: What hypotheses would you make from this?

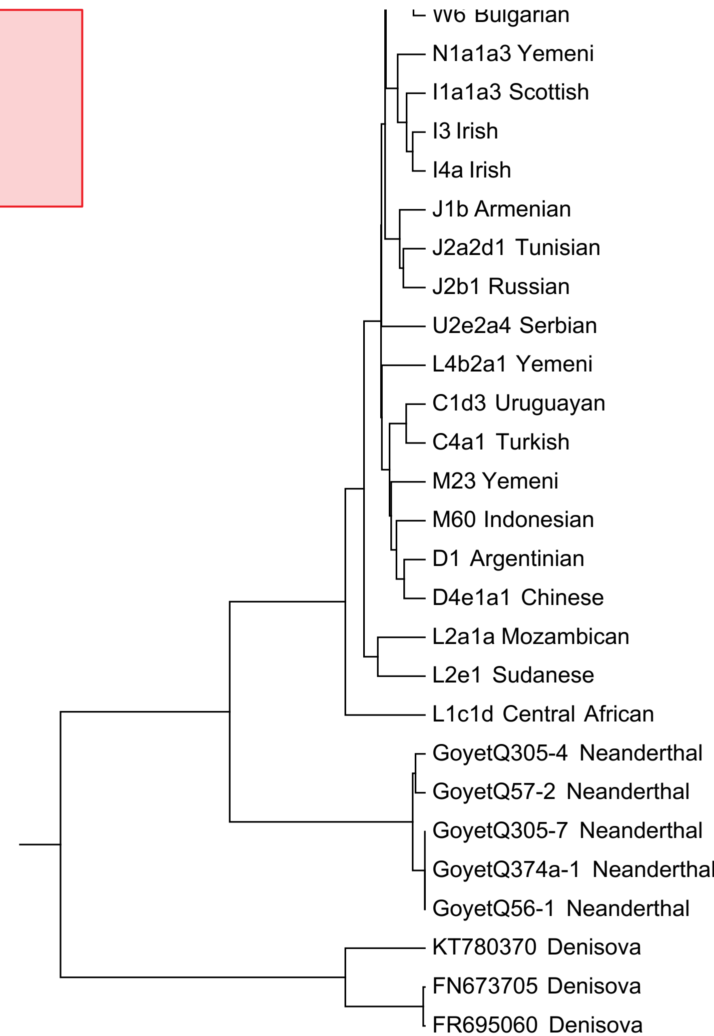
Neanderthals/Denisovans, ancient humans living in Europe/Siberia, seem to be distinct from modern humans.



Adding Neanderthals/Denisovans to the Mix

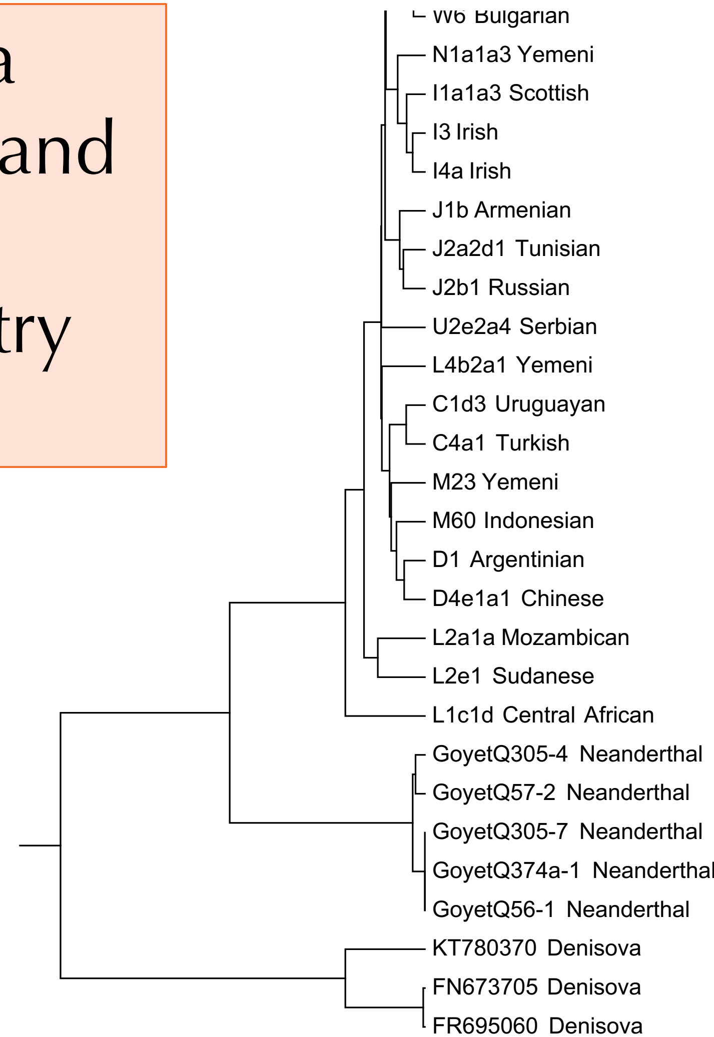
Checkpoint: What hypotheses would you make from this?

Wrong! Europeans may be up to 4% Neanderthal, and Australian aborigines up to 6% Denisovan.



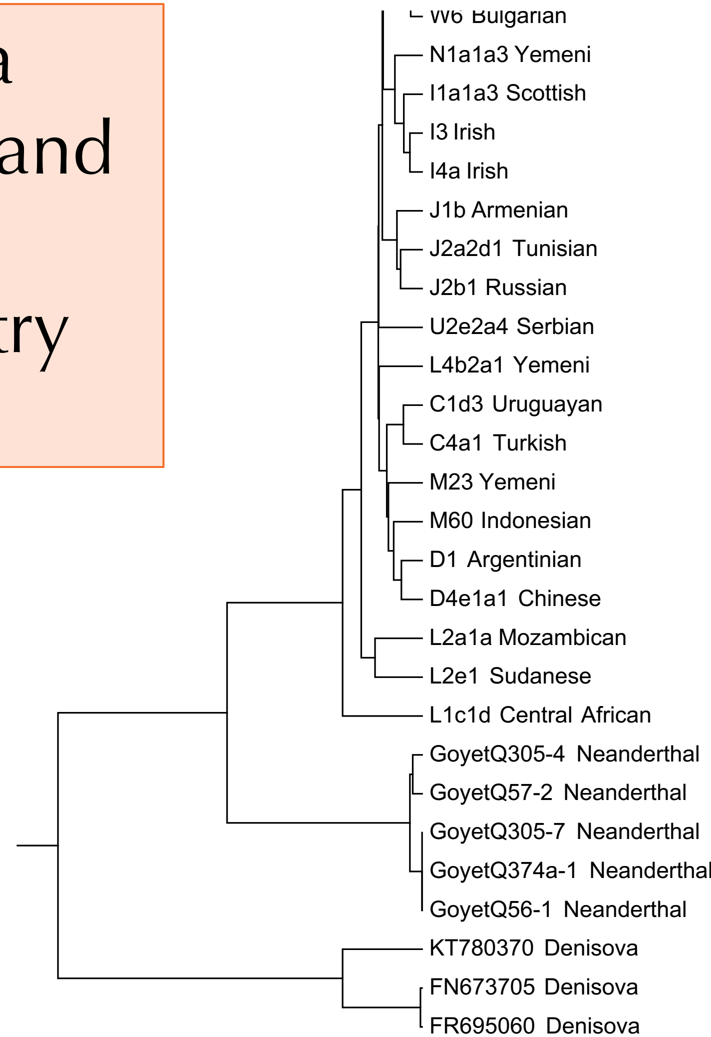
From Strict Population Structure to Admixture

An evolutionary tree gives us a very one-dimensional picture and is only reliable if we sample individuals with known ancestry for many generations.



From Strict Population Structure to Admixture

An evolutionary tree gives us a very one-dimensional picture and is only reliable if we sample individuals with known ancestry for many generations.

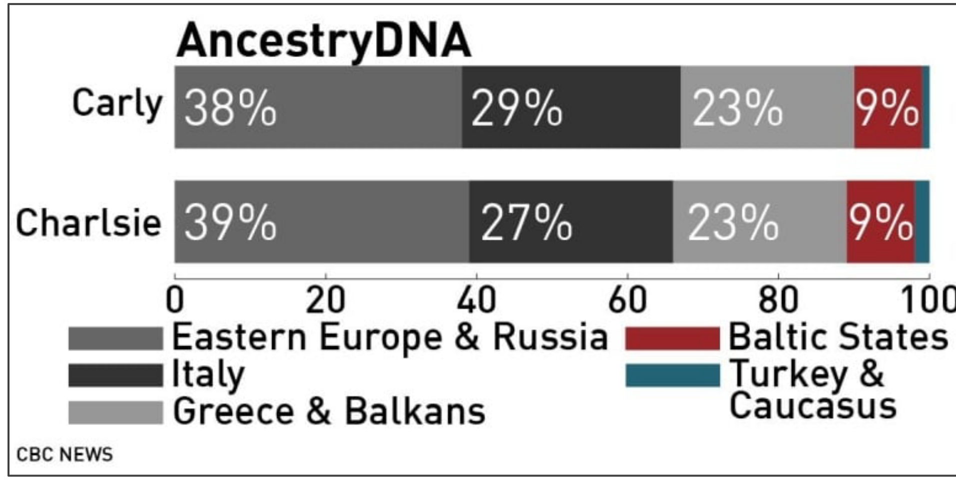


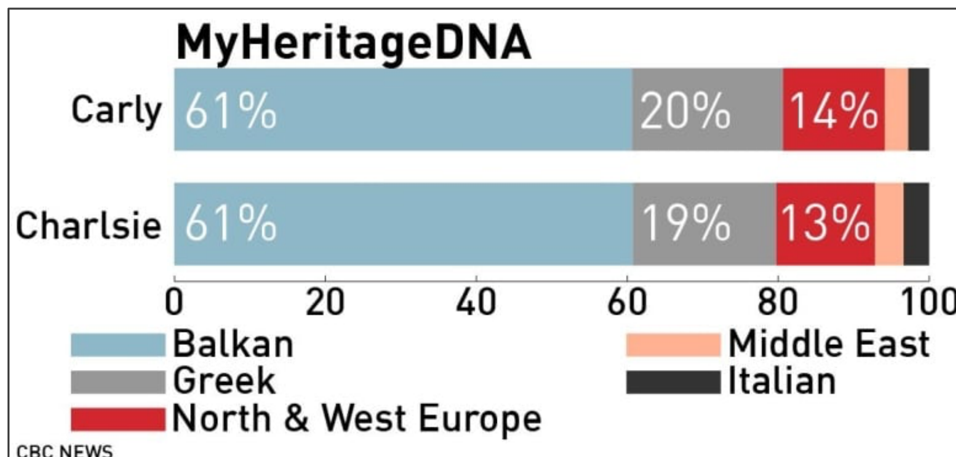
But how can we say that you are *x*% Eastern European, *y*% West African, *z*% Native American, etc.? This is **admixture**.

"Twins get 'Mystifying' [Genotyping] Results"

"*[Genotyping is] kind of a science and an art*" – Paul Maier, population geneticist at FamilyTreeDNA

"*Compromise is the shared hypotenuse of the conjoined triangles of success.*" – Jack Barker, *Silicon Valley*





https://www.cbc.ca/news/technology/dna-ancestry-kits-twins-marketplace-1.4980976

From Genomics to Genotyping

Genotyping: Identifying a collection of genetic **markers** that an individual possesses without obtaining full sequencing information.

From Genomics to Genotyping

Genotyping: Identifying a collection of genetic **markers** that an individual possesses without obtaining full sequencing information.

- **Single-nucleotide polymorphisms (SNPs):** single nucleotide variants present in > 1% of population.
- **Short tandem repeats (STRs):** short number of base pairs repeating a variable number of times consecutively.

From Genomics to Genotyping

Genotyping: Identifying a collection of genetic **markers** that an individual possesses without obtaining full sequencing information.

- **Single-nucleotide polymorphisms (SNPs):** single nucleotide variants present in > 1% of population.
- **Short tandem repeats (STRs):** short number of base pairs repeating a variable number of times consecutively.

Companies will sample 100K to 1 million markers on the order of $100.

Toward a Computational Problem

- **Input:** A collection of n markers for m individuals.
- **Output:** an identification of population structure in a multi-dimensional way that makes it easy for us to visualize admixture.

Toward a Computational Problem

- **Input:** A collection of n markers for m individuals.
- **Output:** an identification of population structure in a multi-dimensional way that makes it easy for us to visualize admixture.

Checkpoint: How could we represent the n markers for a given individual?

Toward a Computational Problem

- **Input:** A collection of *n* markers for *m* individuals.
- **Output:** an identification of population structure in a multi-dimensional way that makes it easy for us to visualize admixture.

Answer: Each individual corresponds to a {0, 1, 2}-valued point (vector) in *n*-dimensional space.

(2, 1, 0, 1, 1, 0, 0, 1, 2, 1, 1, 0, 1, 2, 0, 1)

Toward a Computational Problem

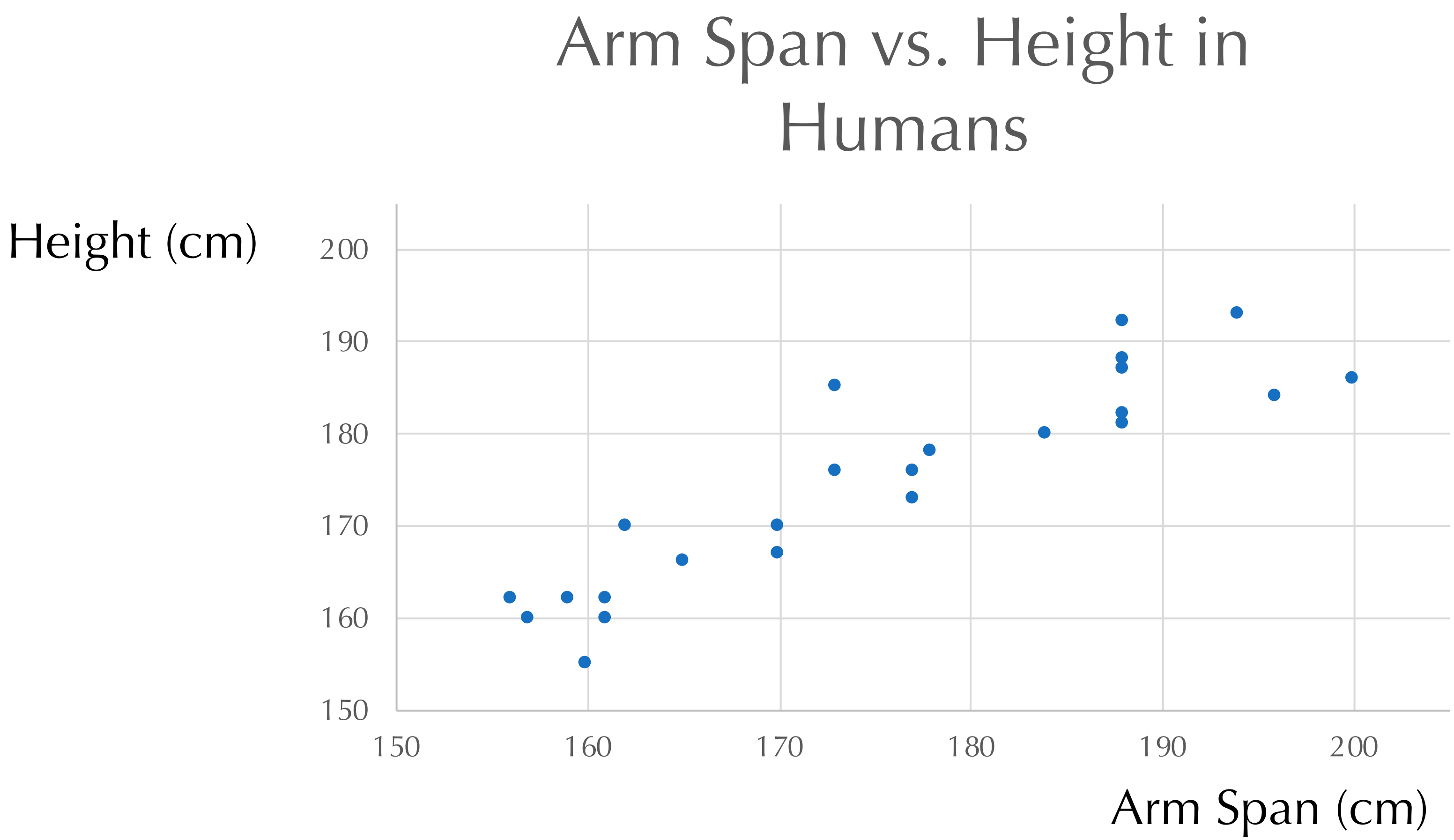
- **Input:** A collection of n markers for m individuals.
- **Output:** an identification of population structure in a multi-dimensional way that makes it easy for us to visualize admixture.

Answer: Each individual corresponds to a {0, 1, 2}-valued point (vector) in n-dimensional space.

(2, 1, 0, 1, 1, 0, 0, 1, **2**, 1, 1, 0, 1, 2, 0, 1)

Number of alleles over two chromosomes for *k*th marker

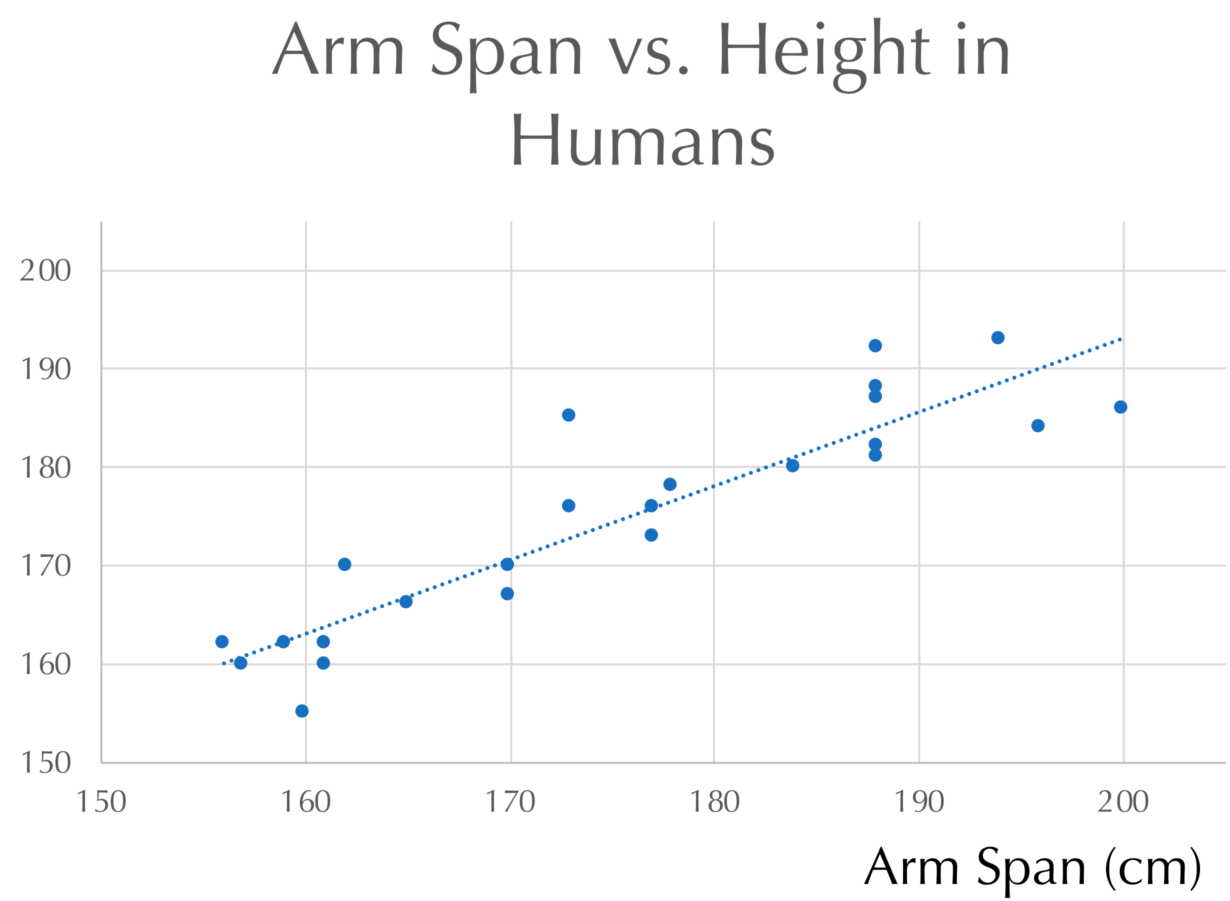
A 2-Dimensional Example



https://www.learner.org/courses/learningmath/data/session7/part_a/further.html

A 2-Dimensional Example

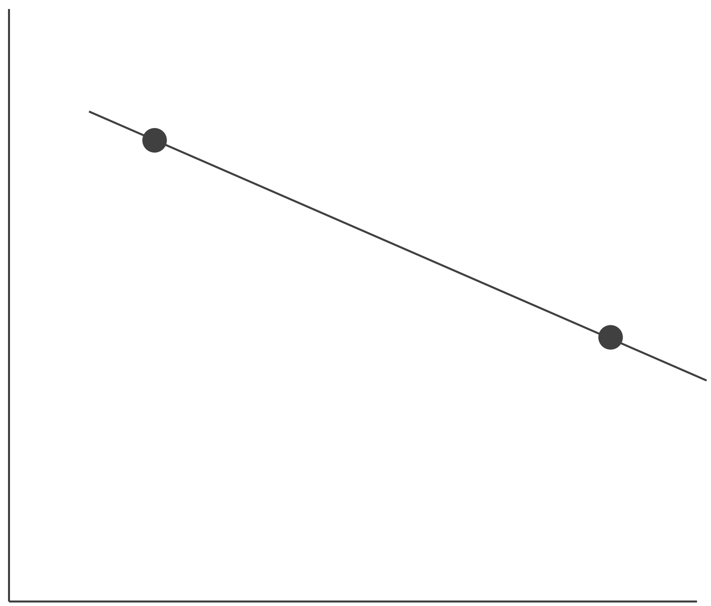
Height (cm)



Note: The line is a 1-D object that does a good job approximating a 2-D dataset.

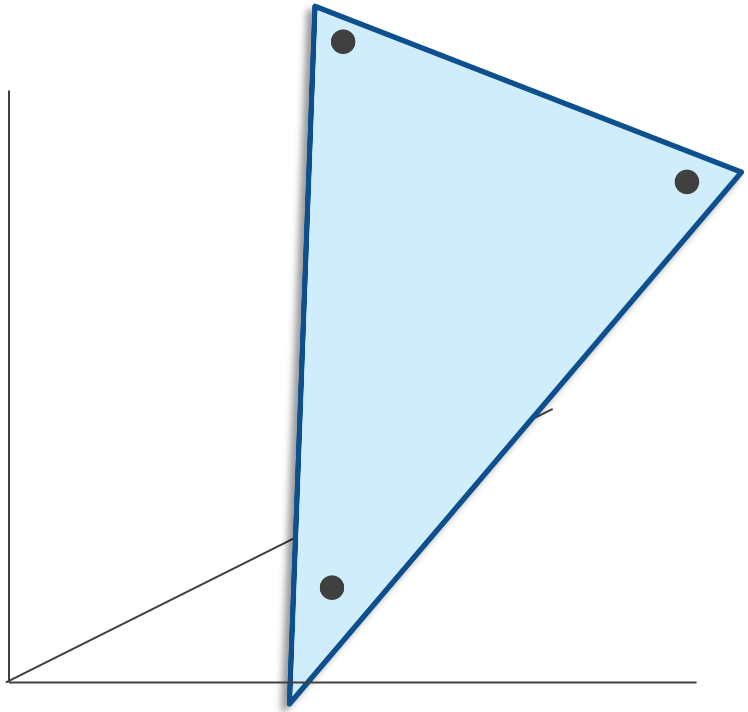
https://www.learner.org/courses/learningmath/data/session7/part_a/further.html

The Need for Dimension Reduction



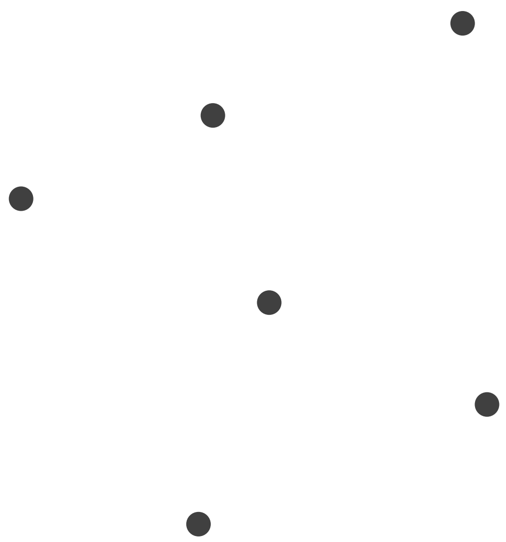
In any dimensional space, I can always find a line that "perfectly explains" two given points.

The Need for Dimension Reduction



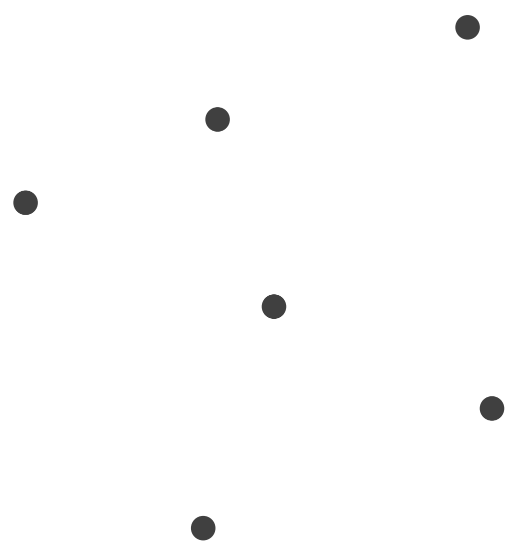
In any dimensional space, I can always find a line or a plane that "perfectly explains" three given points.

The Need for Dimension Reduction



In n dimensional space, I can always find a "hyperplane" of dimension at most $k - 1$ that "perfectly explains" $k < n$ given points.

The Need for Dimension Reduction



In n dimensional space, I can always find a "hyperplane" of dimension at most $k - 1$ that "perfectly explains" $k < n$ given points.

Checkpoint: What will happen if we use 1 million markers for a sample of 100,000 people?

The Need for Dimension Reduction

Curse of dimensionality: The phenomenon that having more dimensions than samples can produce a space so sparse that any "signal" gets washed out.

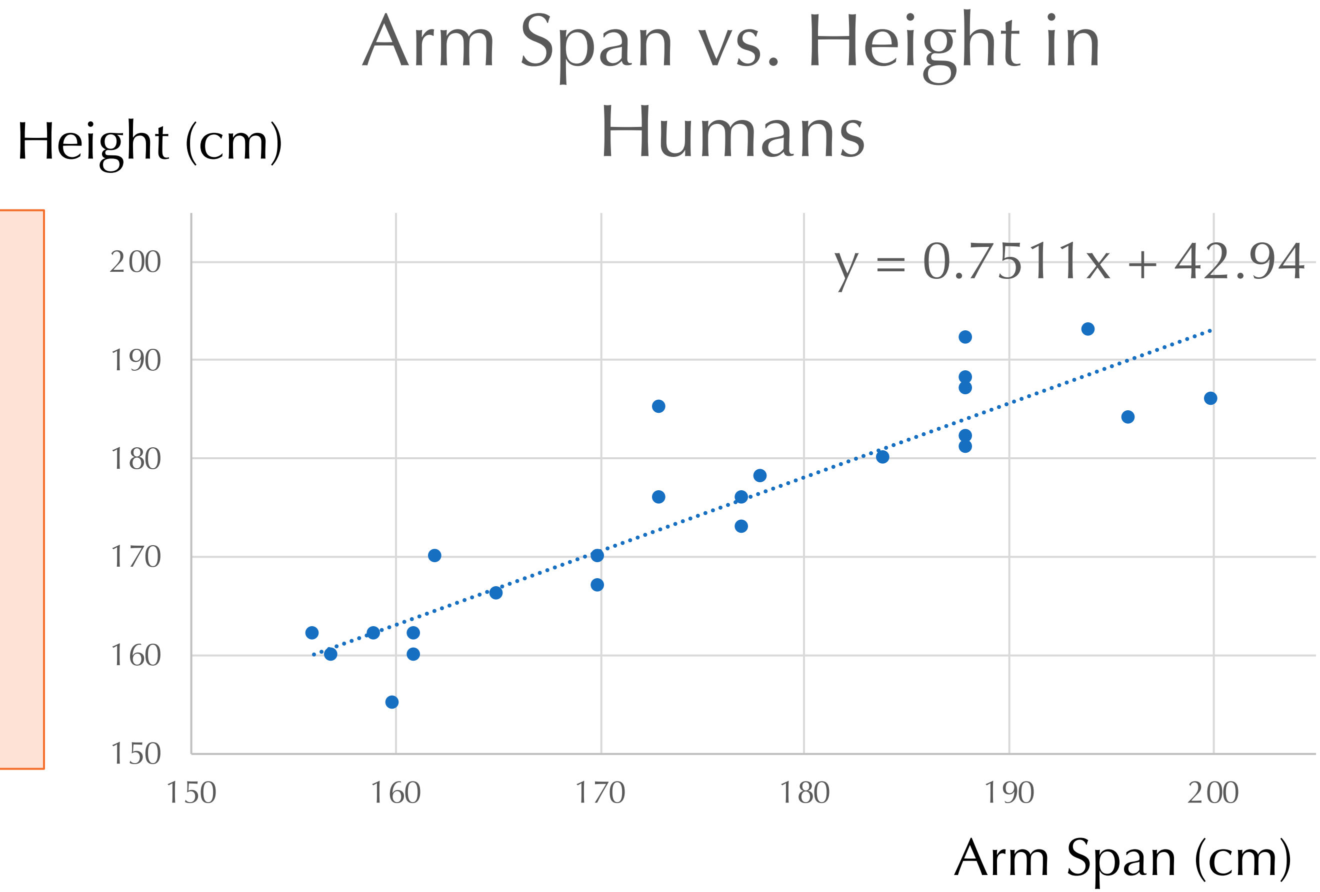
The Need for Dimension Reduction

Curse of dimensionality: The phenomenon that having more dimensions than samples can produce a space so sparse that any "signal" gets washed out.

Dimension reduction: Reducing the number of dimensions of a dataset in order to avoid the "curse" and better visualize its analysis.

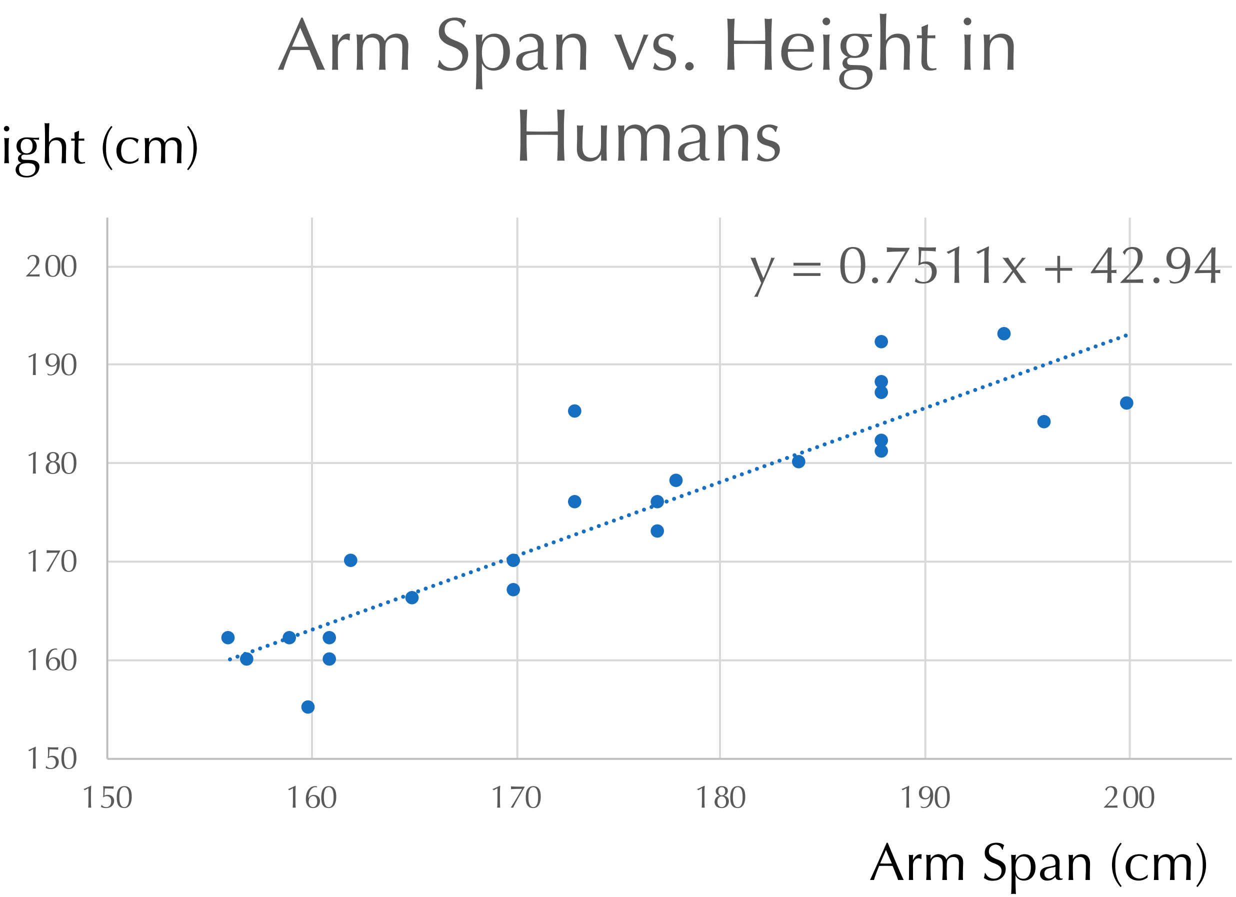
Back to Our Example

Goal: Find the line explaining “as much variance as possible”



https://www.learner.org/courses/learningmath/data/session7/part_a/further.html

Back to Our Example

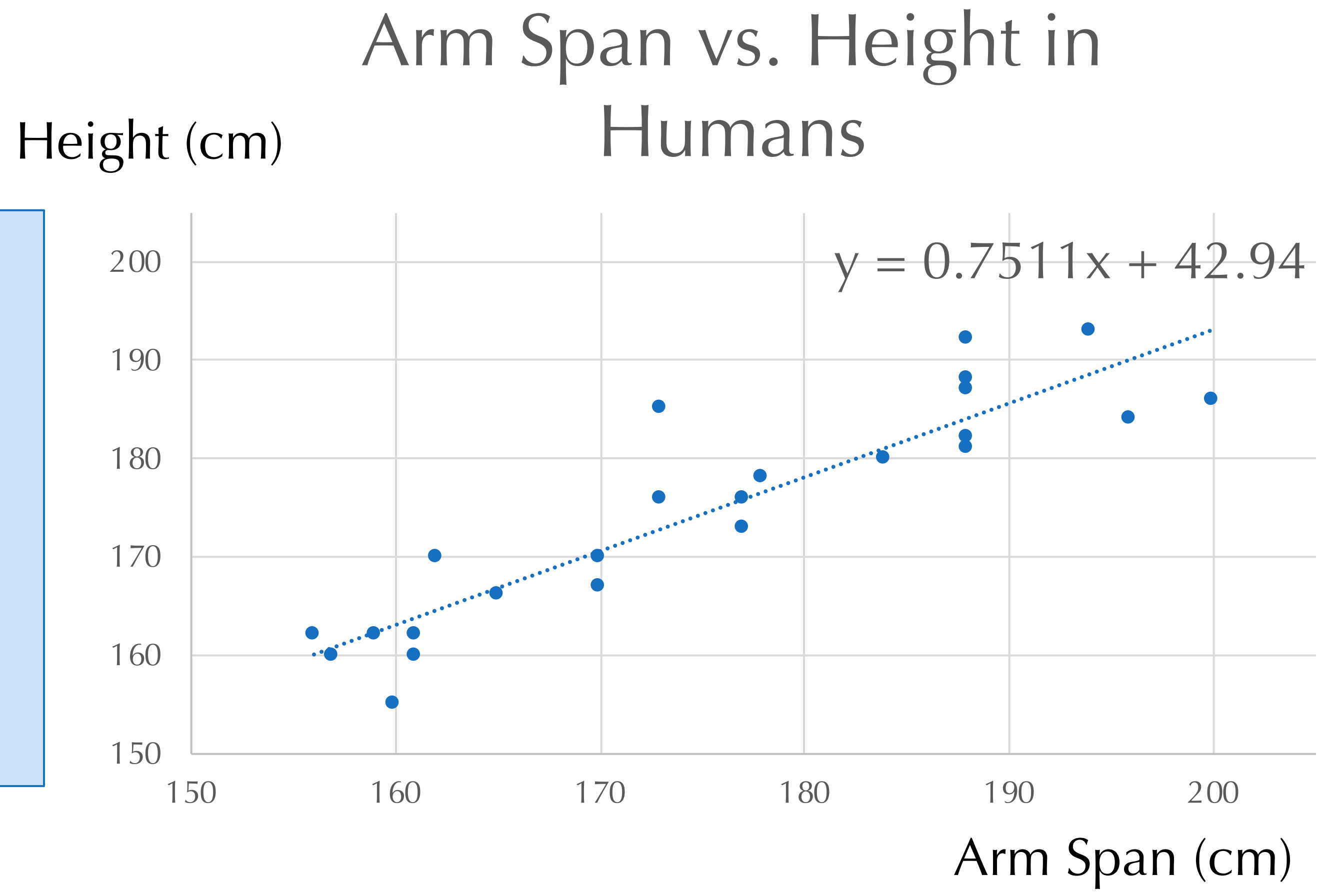


Checkpoint: Where do you think the equation for the line comes from?

https://www.learner.org/courses/learningmath/data/session7/part_a/further.html

Back to Our Example

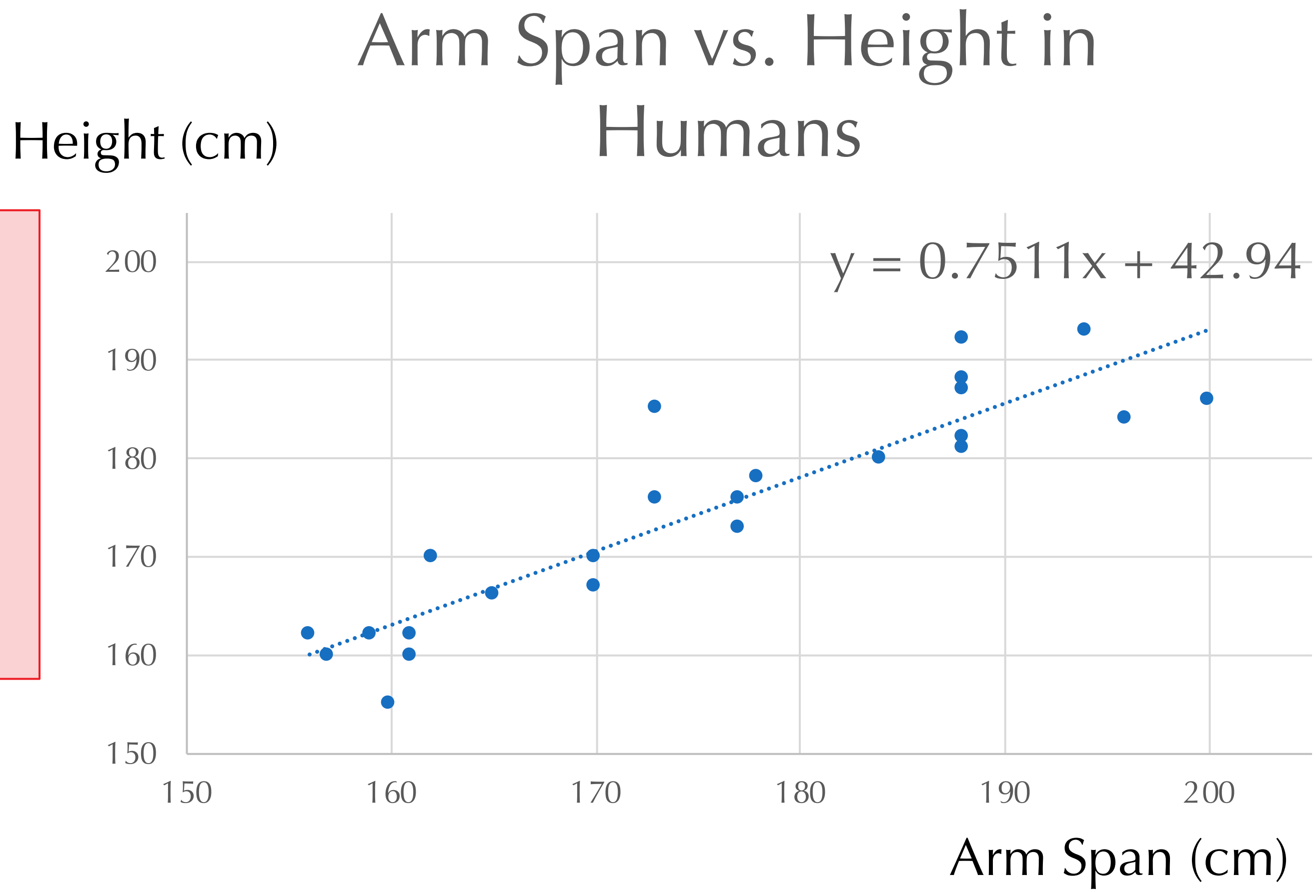
Regression: Minimize the sum of $(y_{observed} - y_{predicted})^2$ over all y.



https://www.learner.org/courses/learningmath/data/session7/part_a/further.html

Back to Our Example

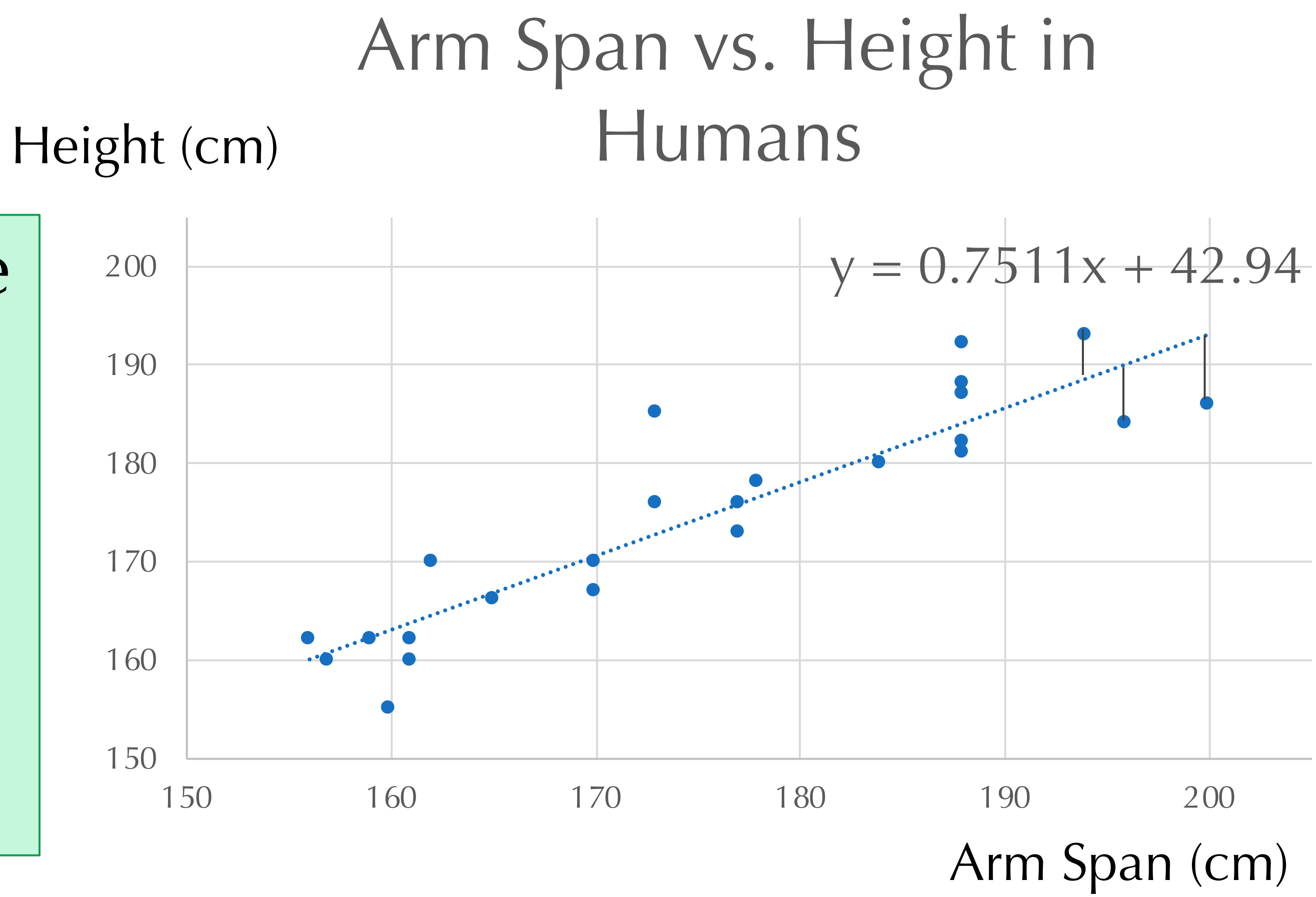
Checkpoint: where are the $(y_{observed} - y_{predicted})^2$ in this plot?



https://www.learner.org/courses/learningmath/data/session7/part_a/further.html

Back to Our Example

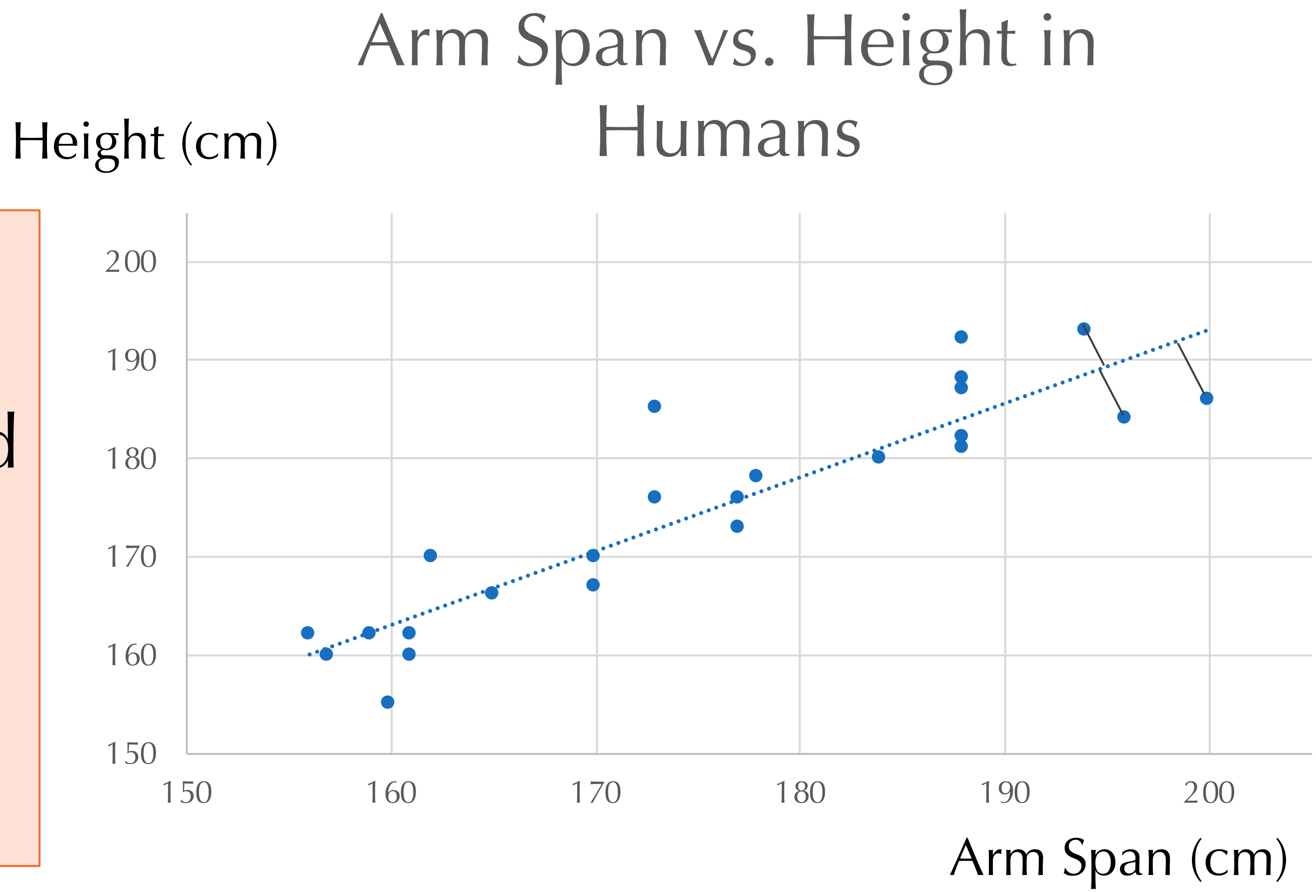
Answer: The (square of) vertical distances from each point to the line.



https://www.learner.org/courses/learningmath/data/session7/part_a/further.html

Back to Our Example

If y isn't a function of x, we should minimize squared distances to line.



https://www.learner.org/courses/learningmath/data/session7/part_a/further.html

Principal Component Analysis

Principal Component Analysis (PCA) Problem

- **Input:** A collection of data points *Data* in n-dimensional space and an integer $d < n$.
- **Output:** the d-dimensional "linear hyperplane" through *Data* minimizing the sum of squared distances from points in *Data* to the hyperplane.

Principal Component Analysis

Principal Component Analysis (PCA) Problem

- **Input:** A collection of data points *Data* in n-dimensional space and an integer $d < n$.
- **Output:** the d-dimensional "linear hyperplane" through *Data* minimizing the sum of squared distances from points in *Data* to the hyperplane.

Checkpoint: In matrix algebra, Principal Component Analysis is called ______________________________.

Principal Component Analysis

Principal Component Analysis (PCA) Problem

- **Input:** A collection of data points *Data* in *n*-dimensional space and an integer $d < n$.
- **Output:** the *d*-dimensional "linear hyperplane" through *Data* minimizing the sum of squared distances from points in *Data* to the hyperplane.

Answer: In matrix algebra, Principal Component Analysis is called "singular value decomposition".

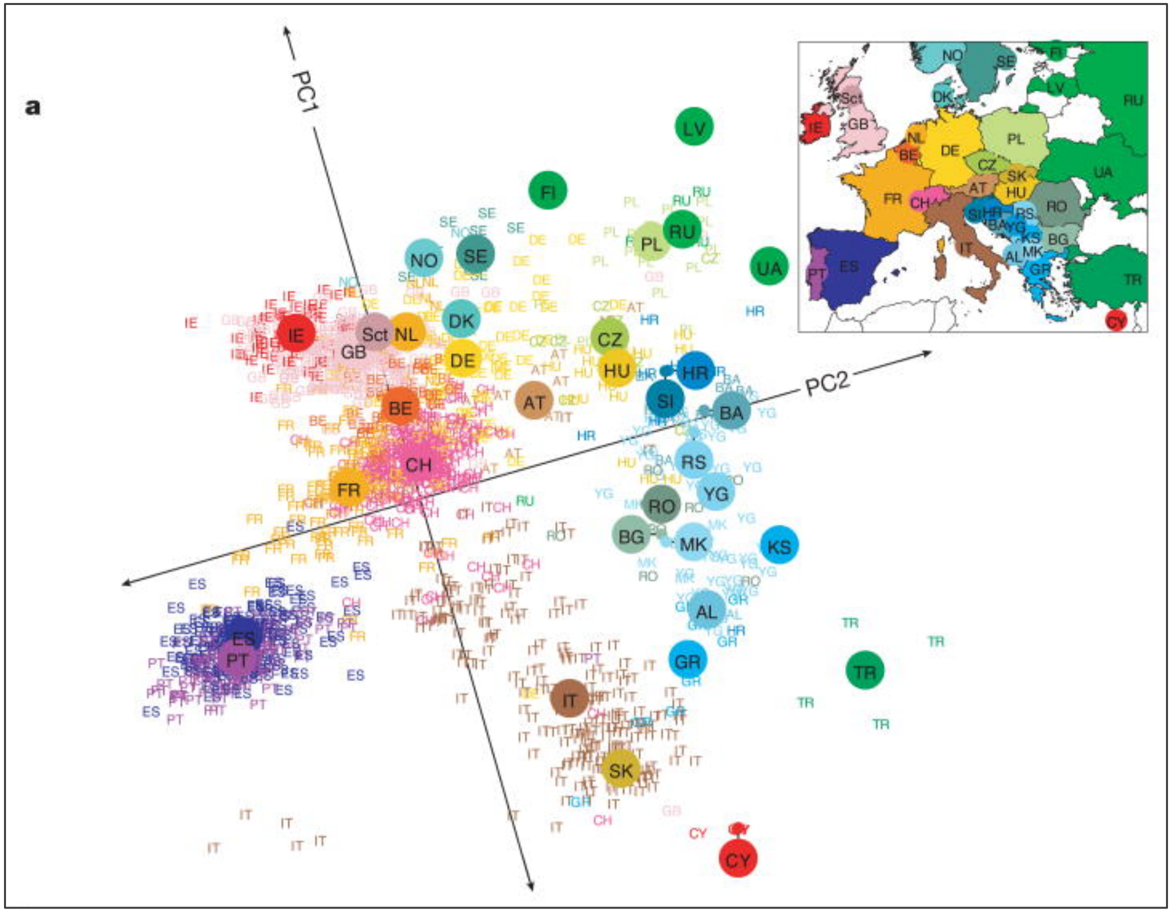
Principal Component Analysis

Principal Component Analysis (PCA) Problem

- **Input:** A collection of data points *Data* in n-dimensional space and an integer $d < n$.
- **Output:** the d-dimensional "linear hyperplane" through *Data* minimizing the sum of squared distances from points in *Data* to the hyperplane.

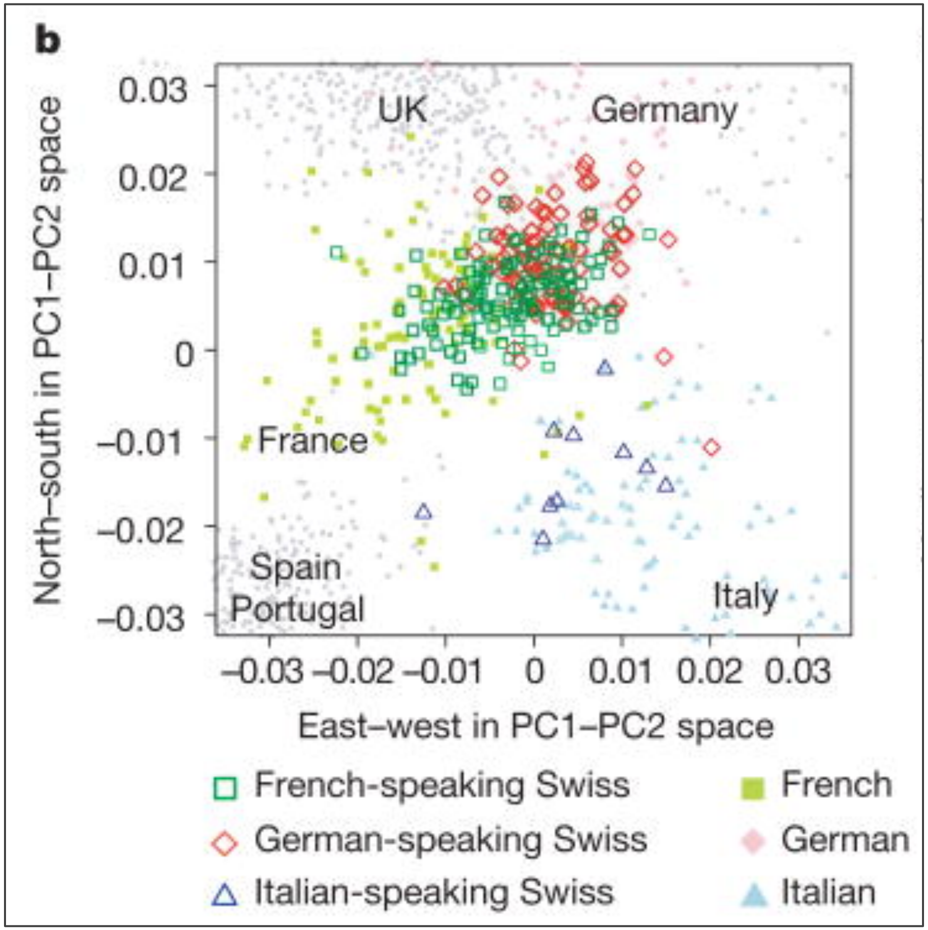
Note: we can then associate each point *Datapoint* with its nearest point *Datapoint′* on the hyperplane and "reduce" the dimension of *Data* to d.

PCA with d = 2 Shows Europe is Inbred



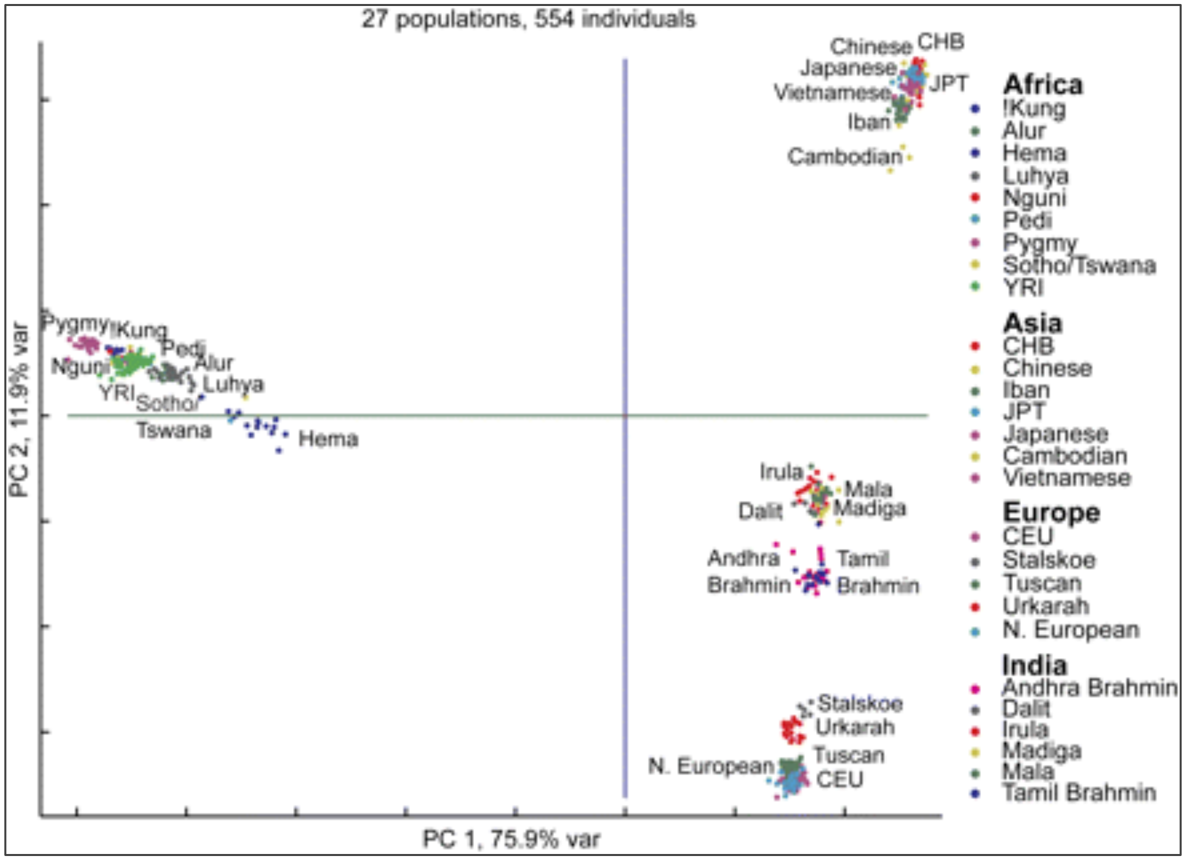
Novembre et al. 2008,
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2735096/

Switzerland's Genes Divide out by Language Spoken



Novembre et al. 2008,
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2735096/

Continental Structure is Visible Too



Xing et al. 2009,
https://genome.cshlp.org/content/19/5/815.full.html

Returning to Our Original Aim

- **Input:** A collection of *n* markers for *m* individuals.
- **Output:** an identification of population structure in a multi-dimensional way that makes it easy for us to visualize admixture.

Returning to Our Original Aim

- **Input:** A collection of *n* markers for *m* individuals.
- **Output:** an identification of population structure in a multi-dimensional way that makes it easy for us to visualize admixture.

Note: dimensionality reduction will help *as an initial step*, but we should address this problem under the assumption that we don't know the ancestry of most or all individuals.

High-Level Overview of Classification

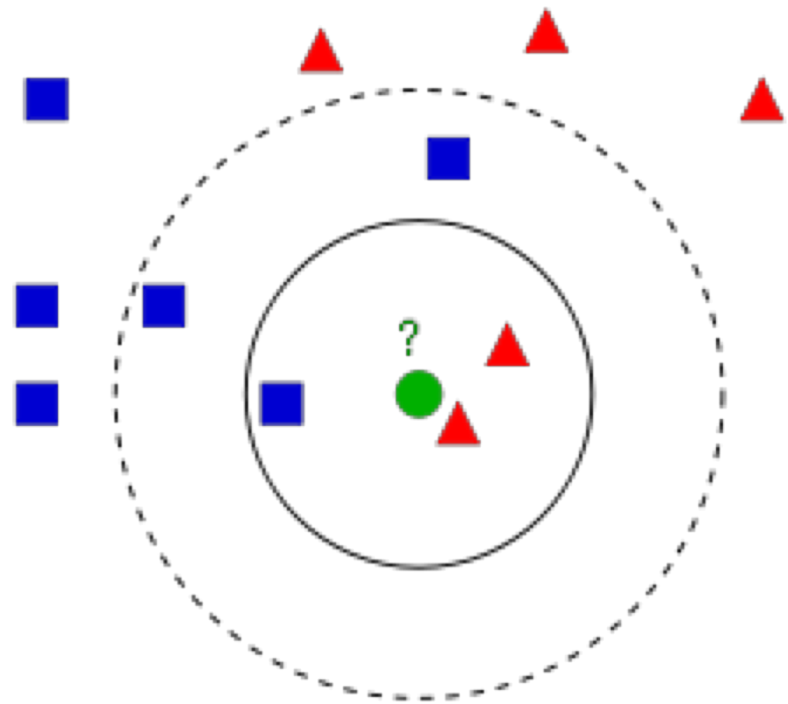
Classification Problem

- **Input:** A collection of data points divided into a **training set** (known ancestry) and a **test set.** (unknown ancestry). Each training data point has a **label** corresponding to its ancestry.
- **Output:** a predictive labeling of all the points in the test set.

k-Nearest Neighbors Algorithm

Say that we have classified training data labeled blue and red, and a new point (green).

Checkpoint: How would you classify the green point? Why?

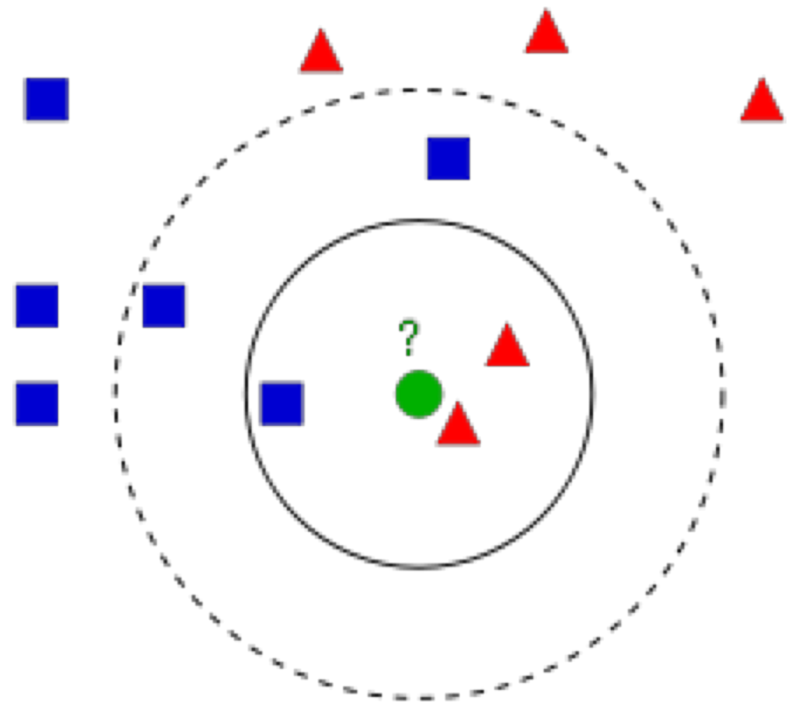


https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#/media/File:KnnClassification.svg

k-Nearest Neighbors Algorithm

Say that we have classified training data labeled blue and red, and a new point (green).

The simplest thing we could do would be to assign this point to be red because a red point is its nearest training point.

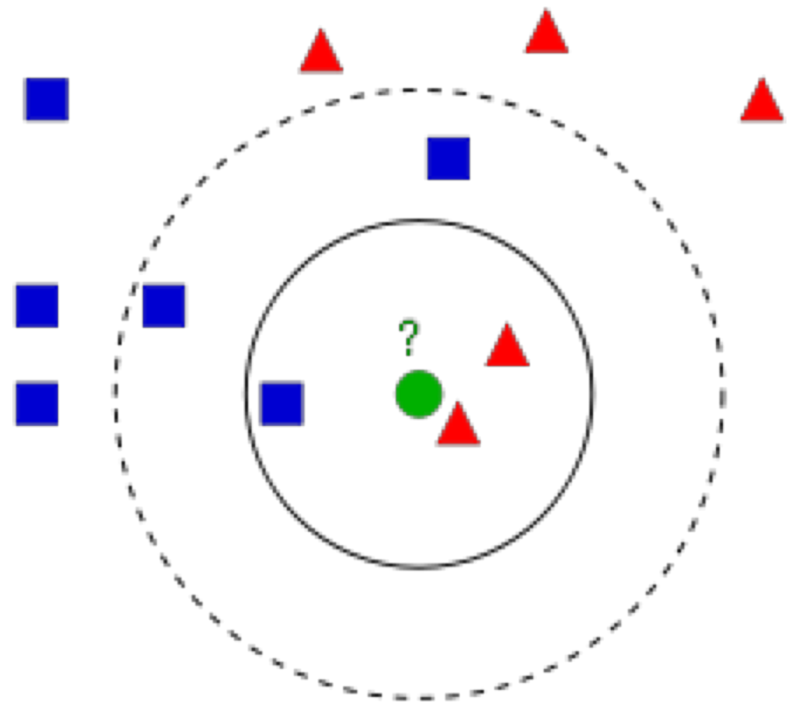


https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#/media/File:KnnClassification.svg

k-Nearest Neighbors Algorithm

Say that we have classified training data labeled blue and red, and a new point (green).

k-Nearest Neighbors: classify the unknown point according to the majority of its *k* nearest neighbors.

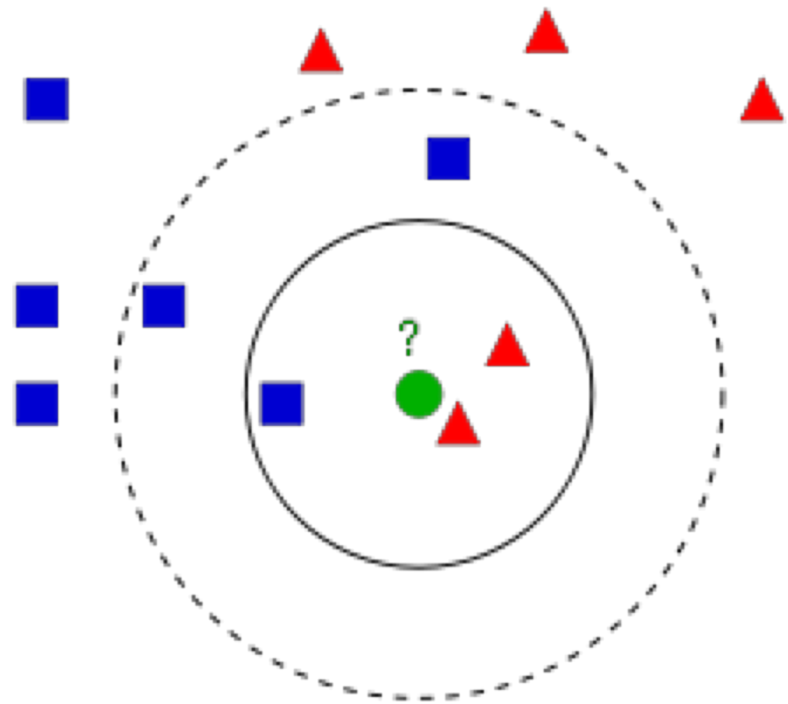


https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#/media/File:KnnClassification.svg

k-Nearest Neighbors Algorithm

$k = 1$: point is labeled red.
$k = 3$: point is labeled red.
$k = 5$: point is labeled blue.

k-Nearest Neighbors: classify the unknown point according to the majority of its k nearest neighbors.



https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#/media/File:KnnClassification.svg

The Problem with Classification

Classification Problem

- **Input:** A collection of data points divided into a **training set** (known ancestry) and a **test set.** (unknown ancestry). Each training data point has a **label** corresponding to its ancestry.
- **Output:** a predictive labeling of all the points in the test set.

The problem with genotyping as a classification problem is that we usually don't have many gold standard training samples compared to the test data.

High-Level Overview of Clustering

Clustering Problem

- **Input:** A collection of (unlabeled) data points in n dimensional space, and an integer k.
- **Output:** An "optimal" assignment of the input points to k "clusters" (labels).

High-Level Overview of Clustering

Clustering Problem

- **Input:** A collection of (unlabeled) data points in *n* dimensional space, and an integer *k*.
- **Output:** An "optimal" assignment of the input points to *k* "clusters" (labels).

Note: Just like the classification problem, this isn't well defined and we get different results depending on how we define "optimal".

k-Means Clustering: A Popular Approach

The **squared error distortion** between m points *Data* and m points *Centers*:

$$Distortion(Data, Centers) =$$

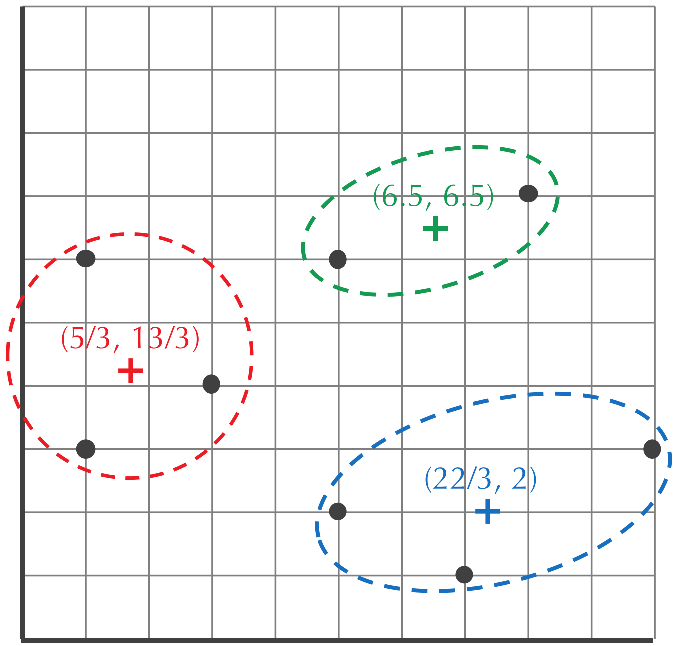
$$\sum_{DataPoint \text{ from } Data} d(DataPoint, Centers)^2/m$$

k-Means Clustering: A Popular Approach

The **squared error distortion** between *m* points *Data* and *m* points *Centers*:

$$Distortion(Data, Centers) = \sum_{DataPoint\text{ from }Data} d(DataPoint, Centers)^2/m$$

Exercise: Compute the squared error distortion of the points and centers (shown as crosses) at right.



k-Means Clustering: A Popular Approach

The **squared error distortion** between *m* points *Data* and *m* points *Centers*:

$$Distortion(Data,\ Centers) =$$

$$\sum_{DataPoint\ \text{from}\ Data} d(DataPoint,\ Centers)^2/m$$

***k*-Means Clustering Problem:**

- **Input:** A set of points *Data* in *n*-dimensional space and an integer *k*.
- **Output:** A set of *k* points *Centers* that minimizes *Distortion*(*Data*,*Centers*) over all choices of *Centers*.

k-Means Clustering: A Popular Approach

The **squared error distortion** between m points *Data* and m points *Centers*:

$$Distortion(Data, Centers) =$$

$$\sum_{DataPoint \text{ from } Data} d(DataPoint, Centers)^2/m$$

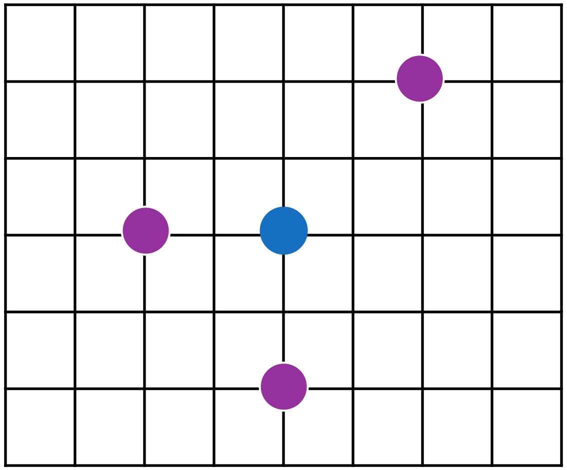
k-Means Clustering Problem:

NP-Hard for $k > 1$

- **Input:** A set of points *Data* in n-dimensional space and an integer k.
- **Output:** A set of k points *Centers* that minimizes *Distortion*(*Data*,*Centers*) over all choices of *Centers*.

Center of Gravity

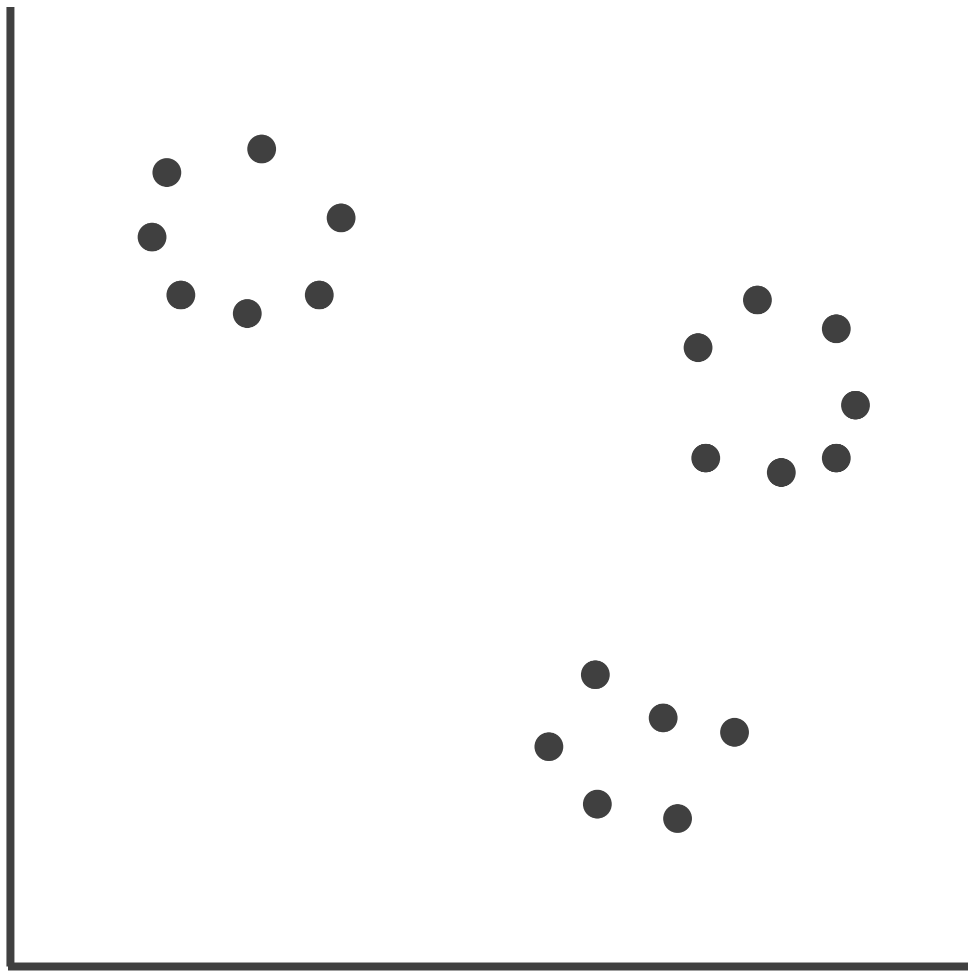
The **center of gravity** of *m* points *Data* is the point whose *i*-th coordinate is the average of the *i*-th coordinates of all points in *Data*.



i-th coordinate of center of gravity = average of the *i*-th coordinates of datapoints:

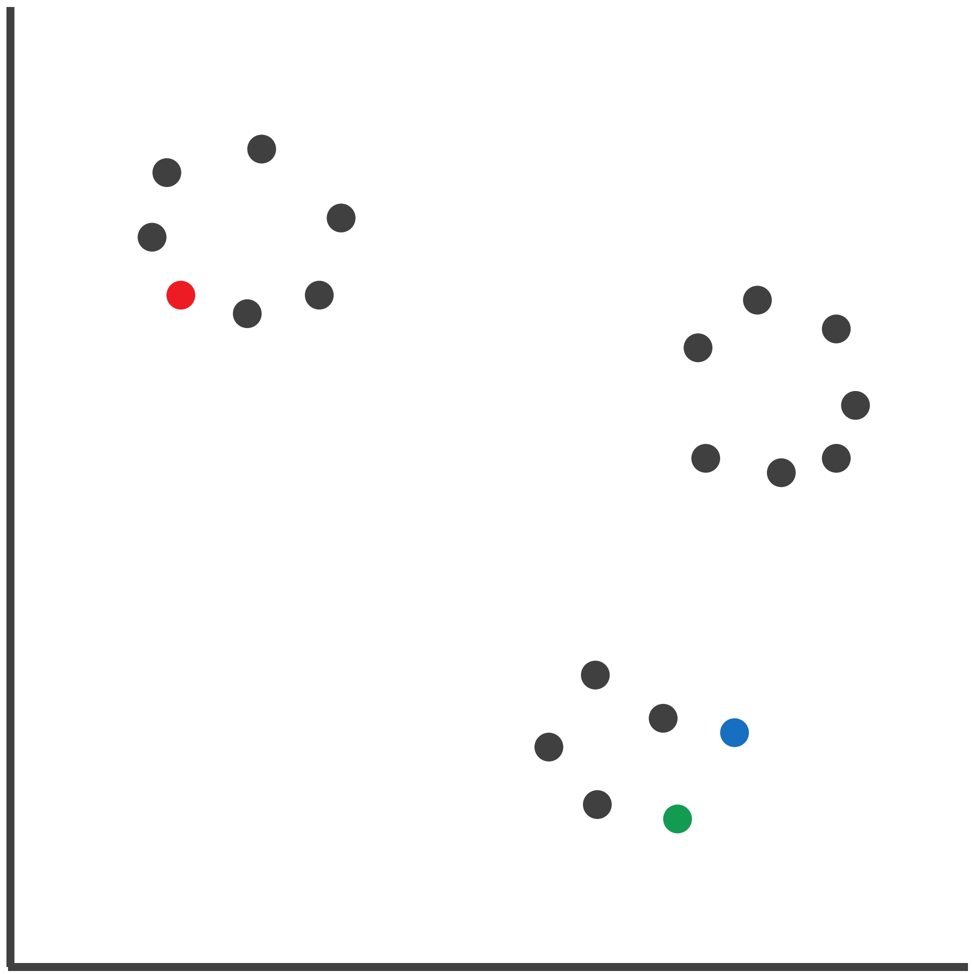
((2+4+6)/3, (3+1+5)/3) = (4, 3)

The Lloyd Algorithm in Action



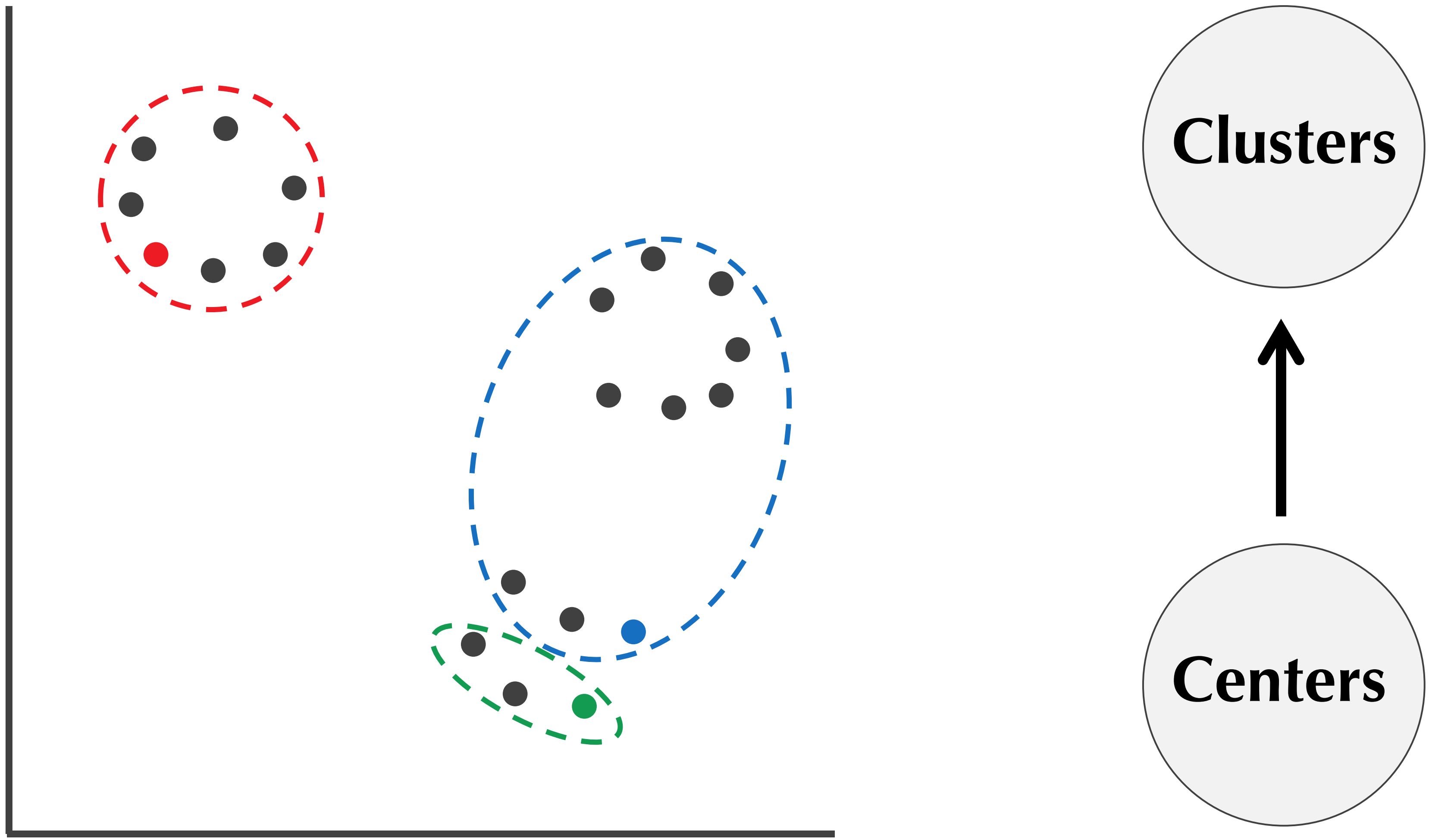
Lloyd algorithm: a clustering heuristic that alternates between updating centers of gravity and assigning points to their nearest centers.

The Lloyd Algorithm in Action



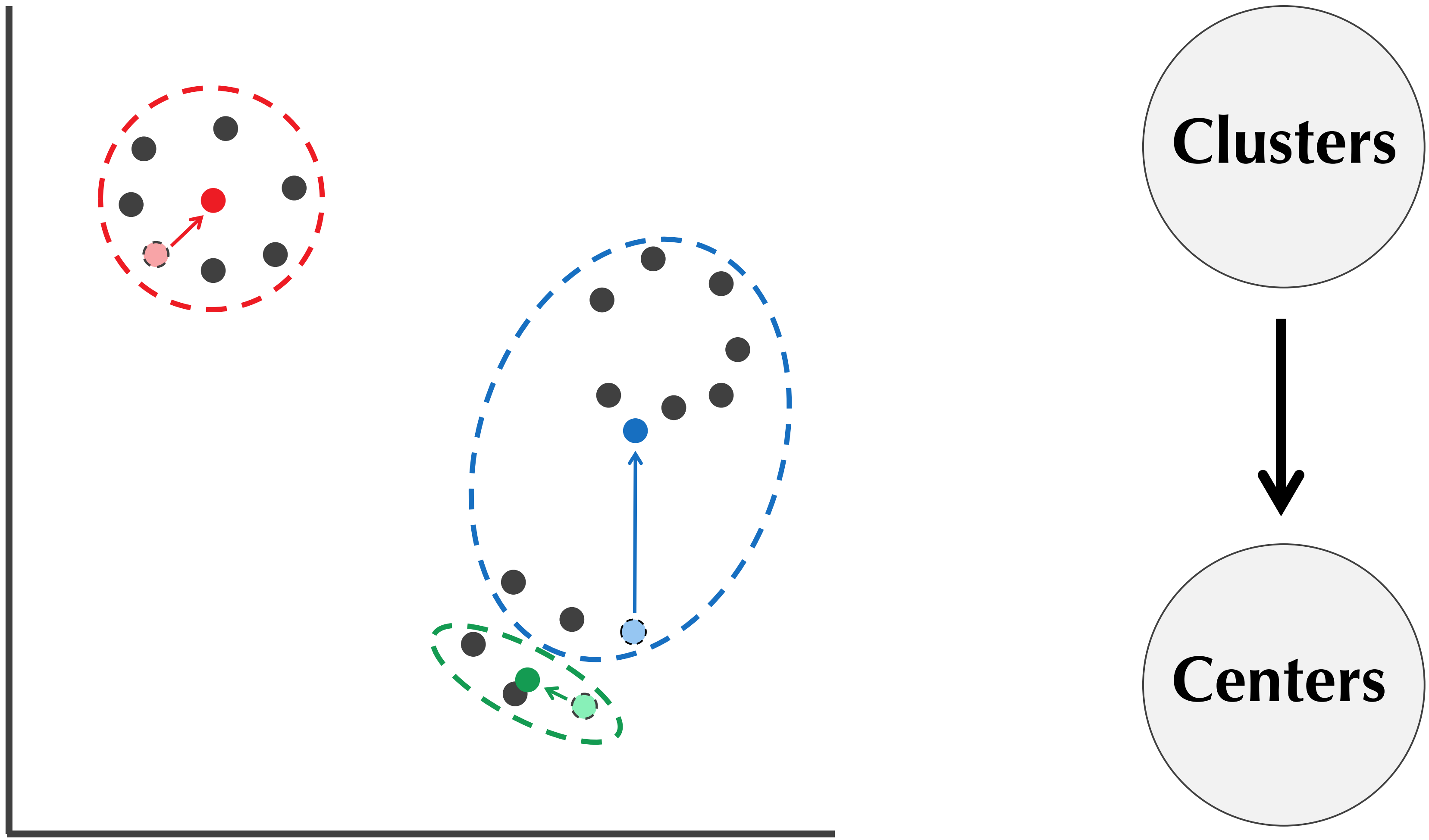
Select *k* arbitrary data points as *Centers*

The Lloyd Algorithm in Action



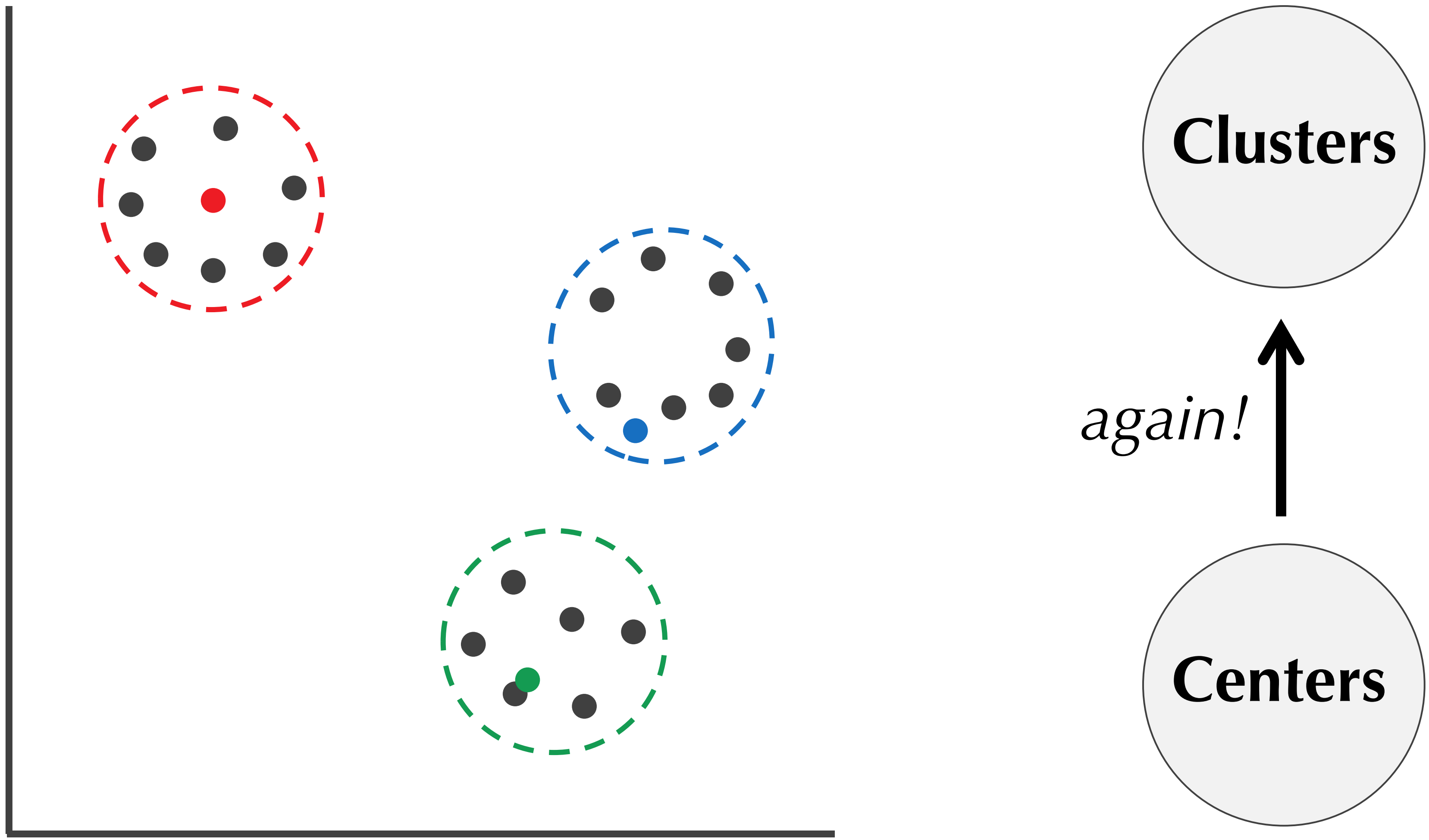
assign each data point to its nearest center

The Lloyd Algorithm in Action



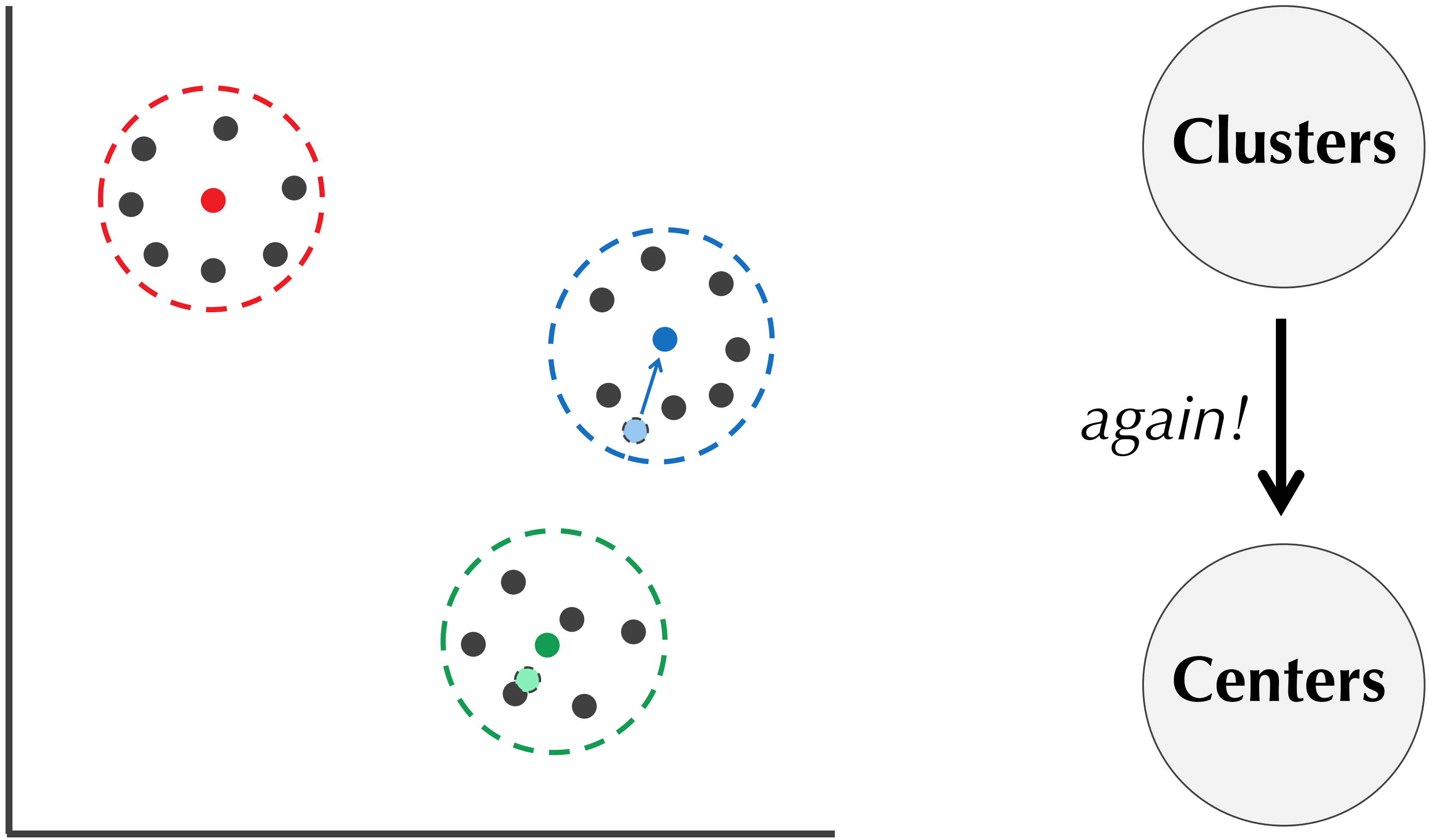
new centers ← clusters' centers of gravity

The Lloyd Algorithm in Action



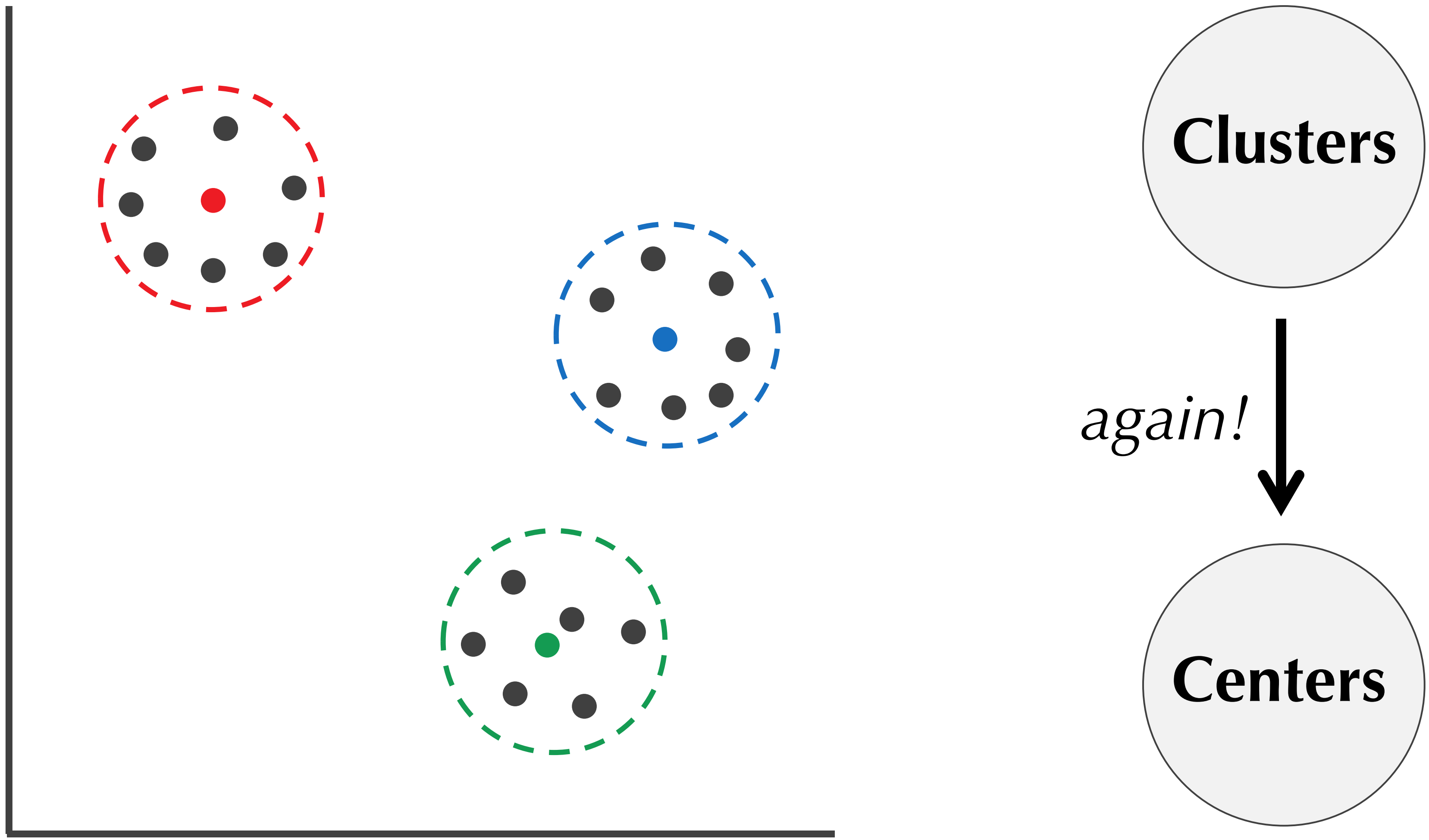
assign each data point to its nearest center

The Lloyd Algorithm in Action



new centers ← clusters' centers of gravity

The Lloyd Algorithm in Action



assign each data point to its nearest center

Lloyd Algorithm in Summary

Select *k* arbitrary data points as *Centers* and then iteratively perform the following steps:

- **Centers to Clusters**: Assign each data point to the cluster corresponding to its nearest center (ties are broken arbitrarily).
- **Clusters to Centers**: After the assignment of data points to *k* clusters, compute new centers as clusters′ center of gravity.

Lloyd Algorithm in Summary

Select *k* arbitrary data points as *Centers* and then iteratively perform the following steps:

- **Centers to Clusters**: Assign each data point to the cluster corresponding to its nearest center (ties are broken arbitrarily).
- **Clusters to Centers**: After the assignment of data points to *k* clusters, compute new centers as clusters' center of gravity.

The algorithm terminates when the centers stop moving (**convergence**).

Lloyd Algorithm in Summary

Select *k* arbitrary data points as *Centers* and then iteratively perform the following steps:

- **Centers to Clusters**: Assign each data point to the cluster corresponding to its nearest center (ties are broken arbitrarily).
- **Clusters to Centers**: After the assignment of data points to *k* clusters, compute new centers as clusters' center of gravity.

Checkpoint: What does the Lloyd algorithm remind you of?

Lloyd Algorithm in Summary

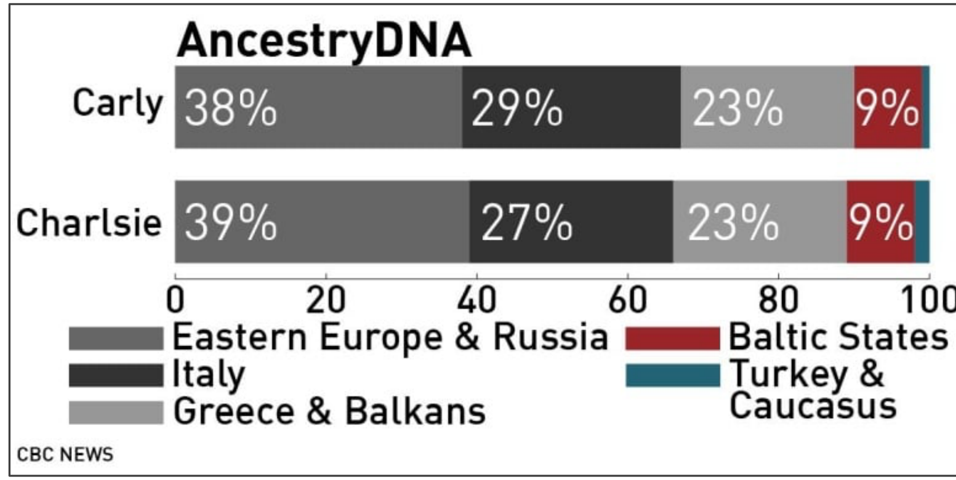
Select k arbitrary data points as *Centers* and then iteratively perform the following steps:

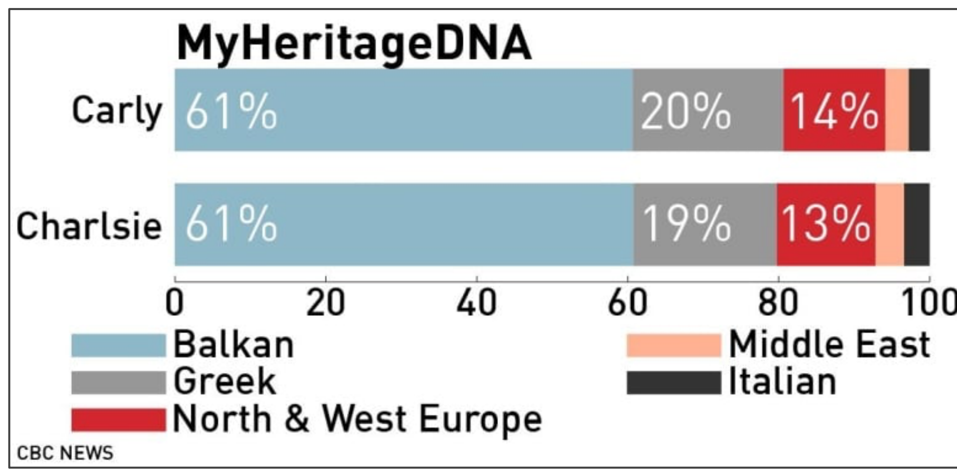
- **Centers to Clusters**: Assign each data point to the cluster corresponding to its nearest center (ties are broken arbitrarily).
- **Clusters to Centers**: After the assignment of data points to k clusters, compute new centers as clusters' center of gravity.

Answer: centers and clusters are both hidden and we try to infer them in stages … just like EM/Gibbs!

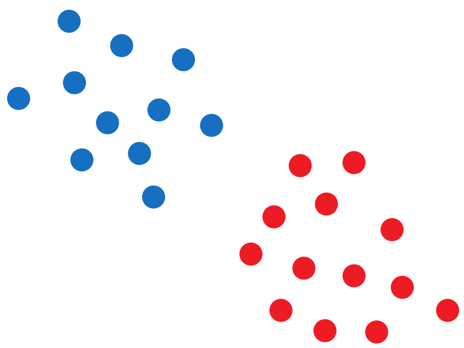
Returning to Admixture

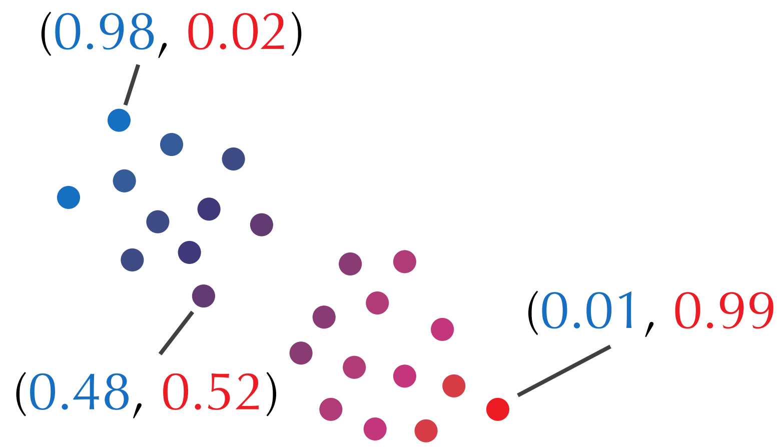
Checkpoint: Clusters give a rigid assignment of individuals to populations. How do you think that we can conclude a collection of percentages for an individual? And why might they differ?





From Hard to Soft Clustering





Hard choices: points are colored red or blue depending on their cluster membership.

Soft choices: points are assigned "red" and "blue" *responsibilities* r_{blue} and r_{red} ($r_{\text{blue}} + r_{\text{red}} = 1$)