Dimension Reduction

Why and How

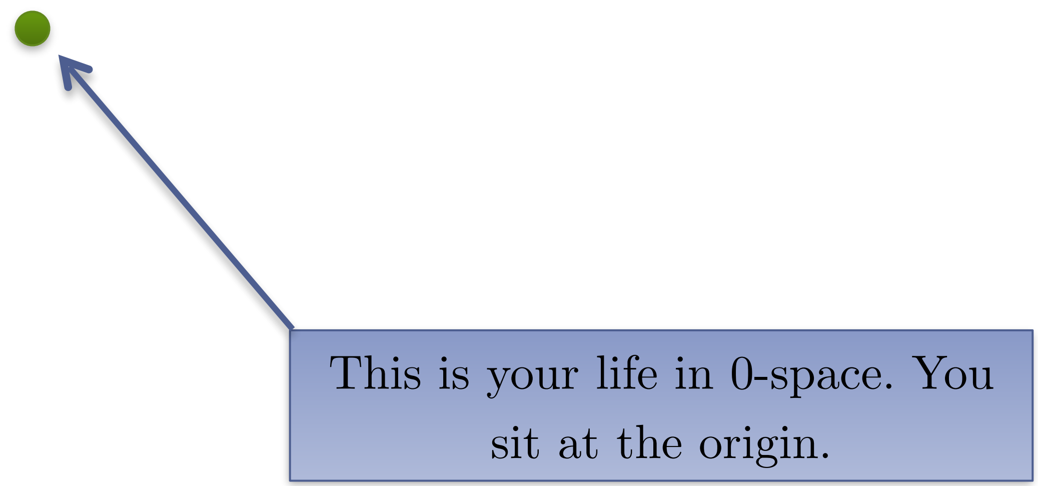
The Curse of Dimensionality

- As the dimensionality (i.e. number of variables) of a space grows, data points become so spread out that the ideas of *distance* and *density* become murky.

- This is simply due to the incredible spacial increase that comes from adding an additional dimension.

Pretend you are a point

(with infinite potential energy)



Pretend you are a point

(with infinite potential energy)

SUDDENLY,
you are given a dimension

Pretend you are a point

(with infinite potential energy)

The JOY of movement!

Pretend you are a point

(with infinite potential energy)

The JOY of movement!

Pretend you are a point

(with infinite potential energy)

The JOY of movement!

Pretend you are a point

(with infinite potential energy)

The JOY of movement!

Pretend you are a point

(with infinite potential energy)

The JOY of movement!

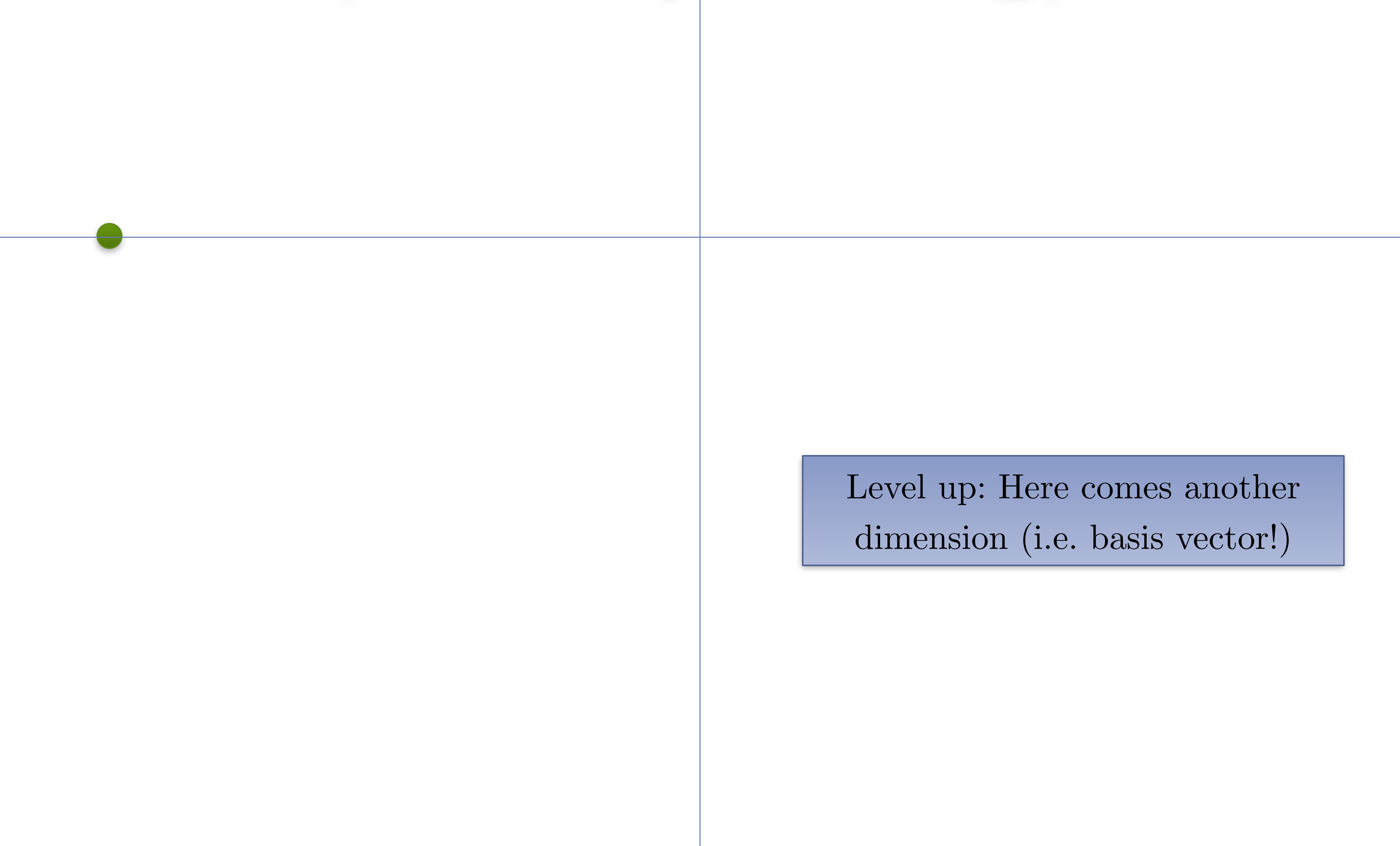
Pretend you are a point

(with infinite potential energy)

Compared to your previous existence, your world seems infinitely more expansive!

Pretend you are a point

(with infinite potential energy)



Pretend you are a point

(with infinite potential energy)

Like....woah....

Pretend you are a point

(with infinite potential energy)

Like....woah....

Pretend you are a point

(with infinite potential energy)

Like....woah....

Pretend you are a point

(with infinite potential energy)

Like....woah....

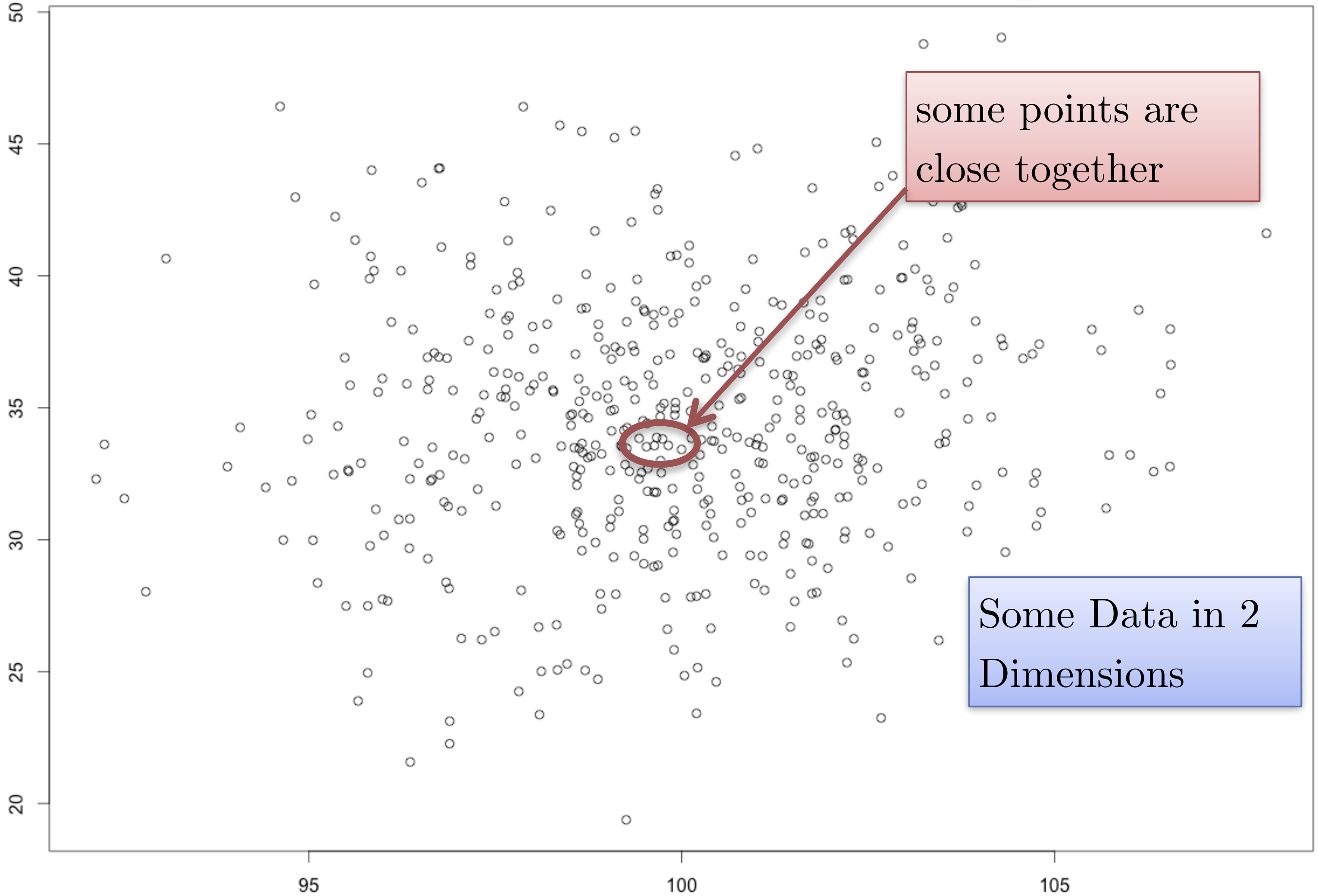
Pretend you are a point

(with infinite potential energy)

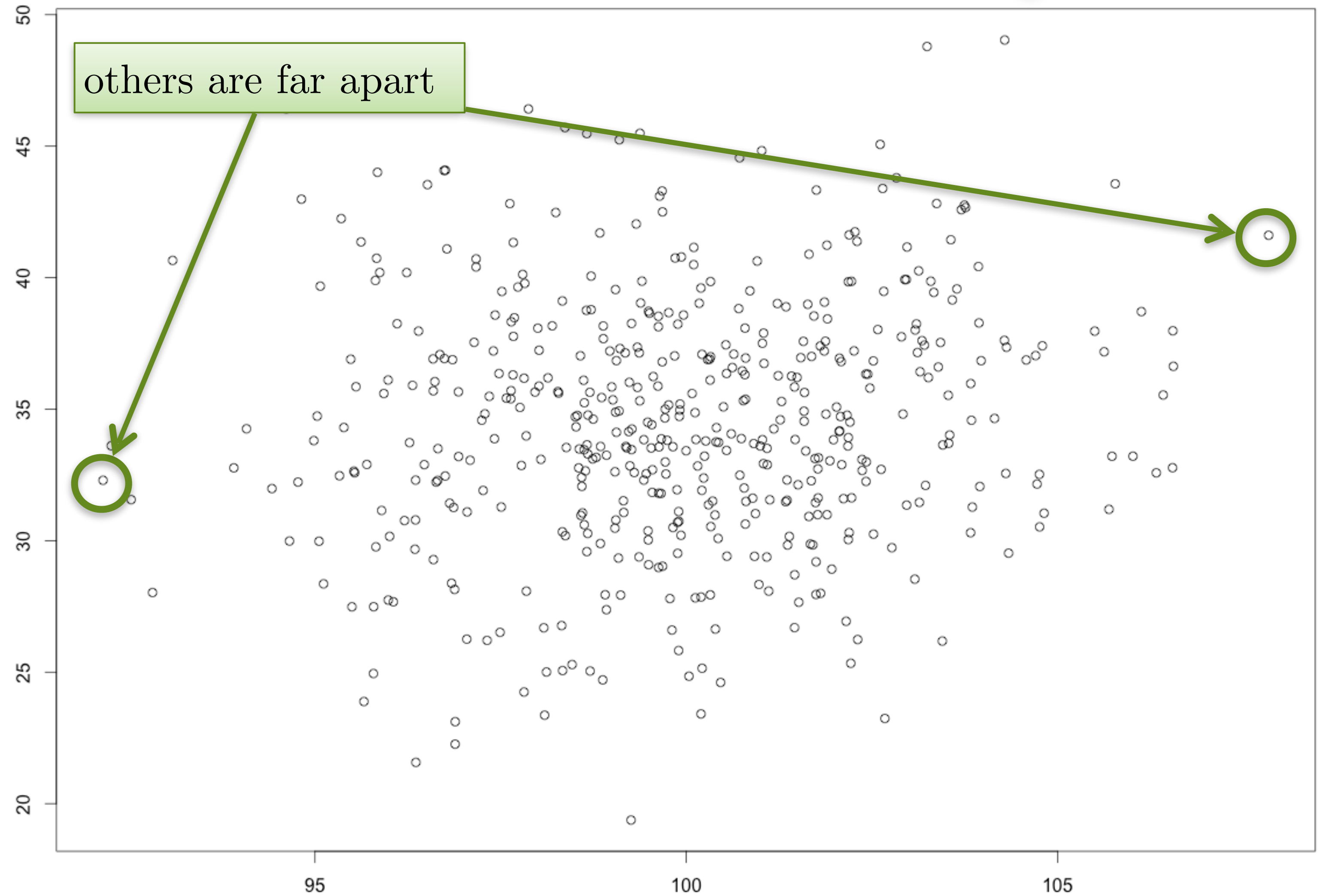
Now imagine a *third* dimension.

The amount of *additional space added* in each dimension actually *seems to get larger*.

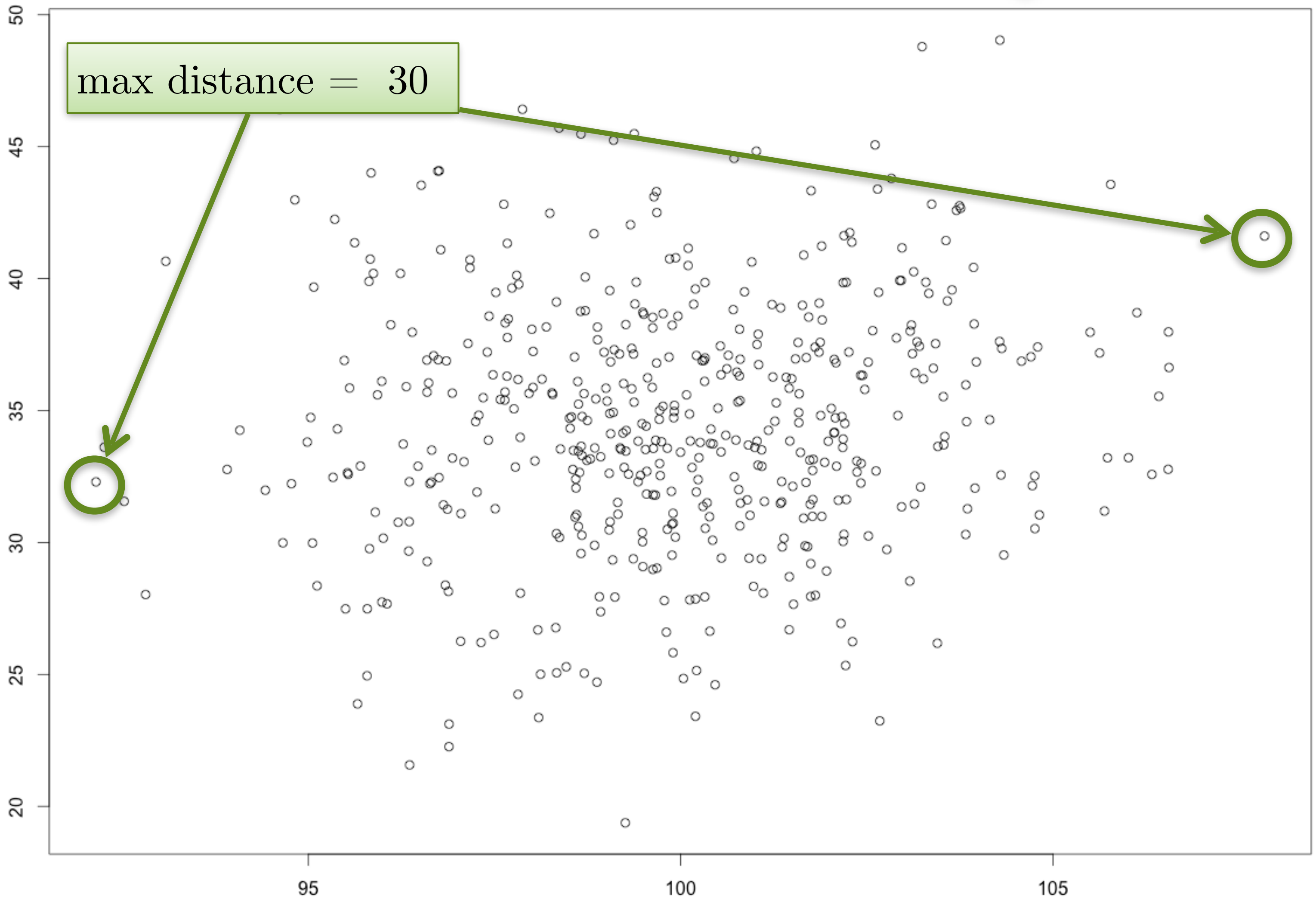
The Curse of Dimensionality



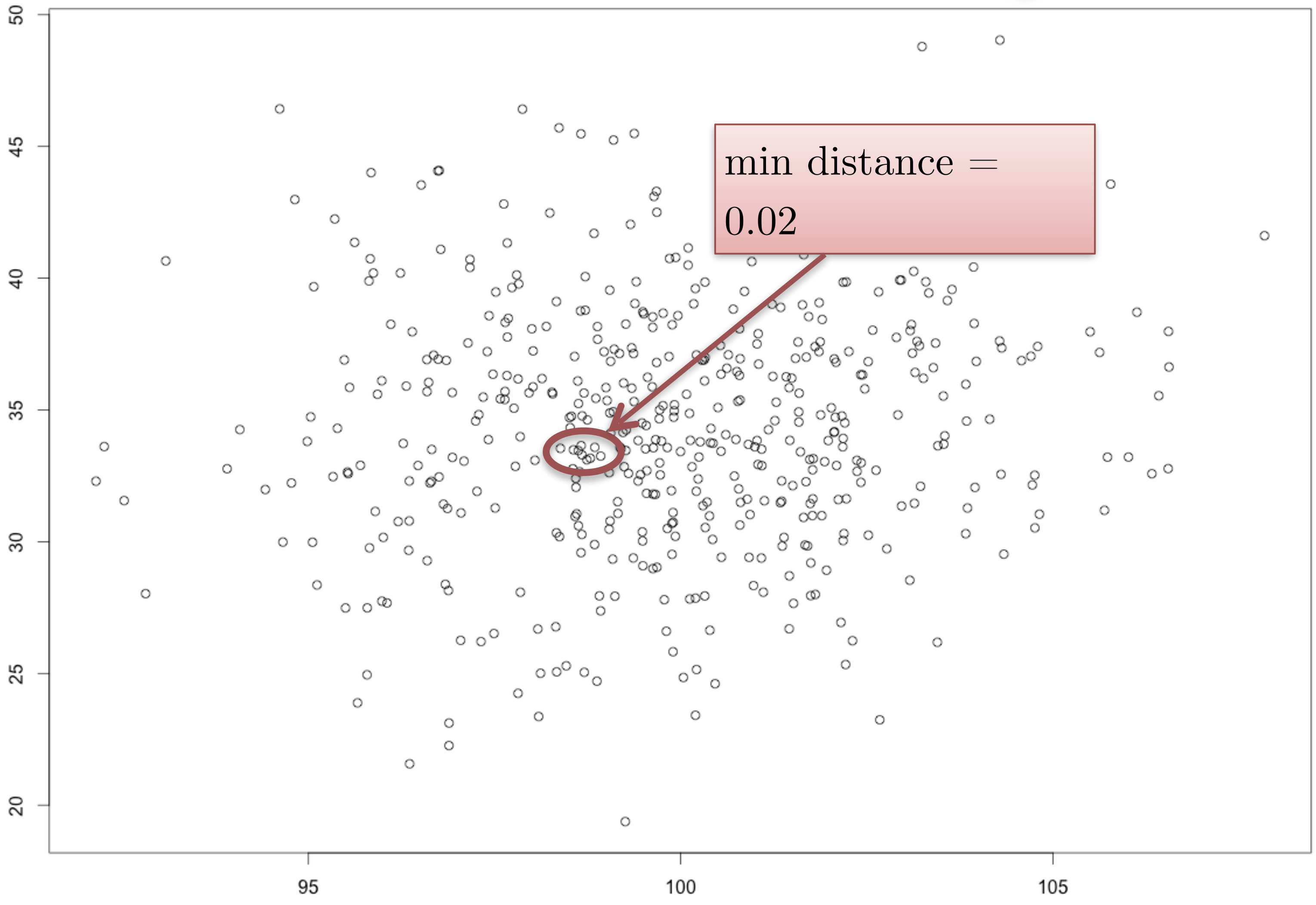
The Curse of Dimensionality



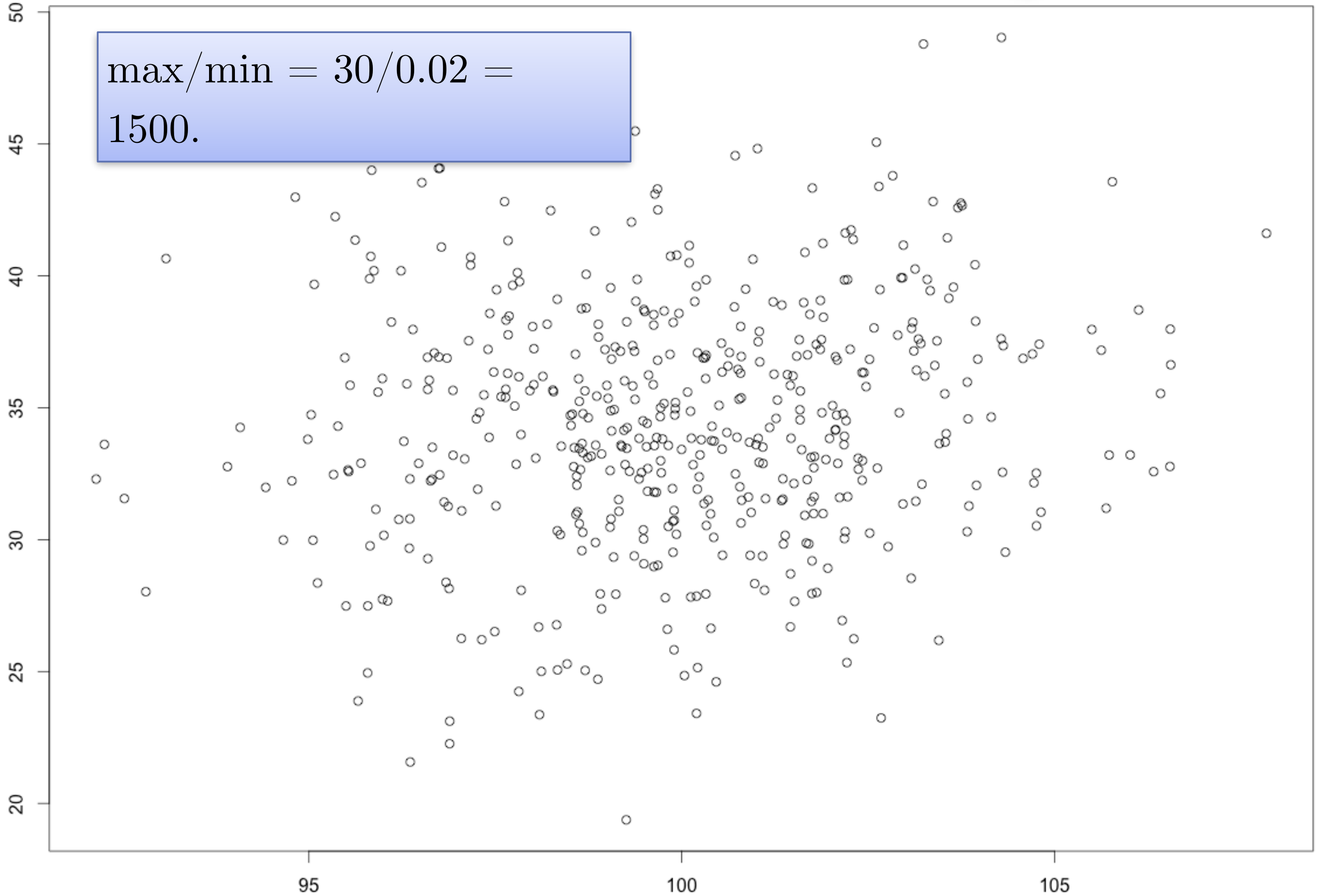
The Curse of Dimensionality



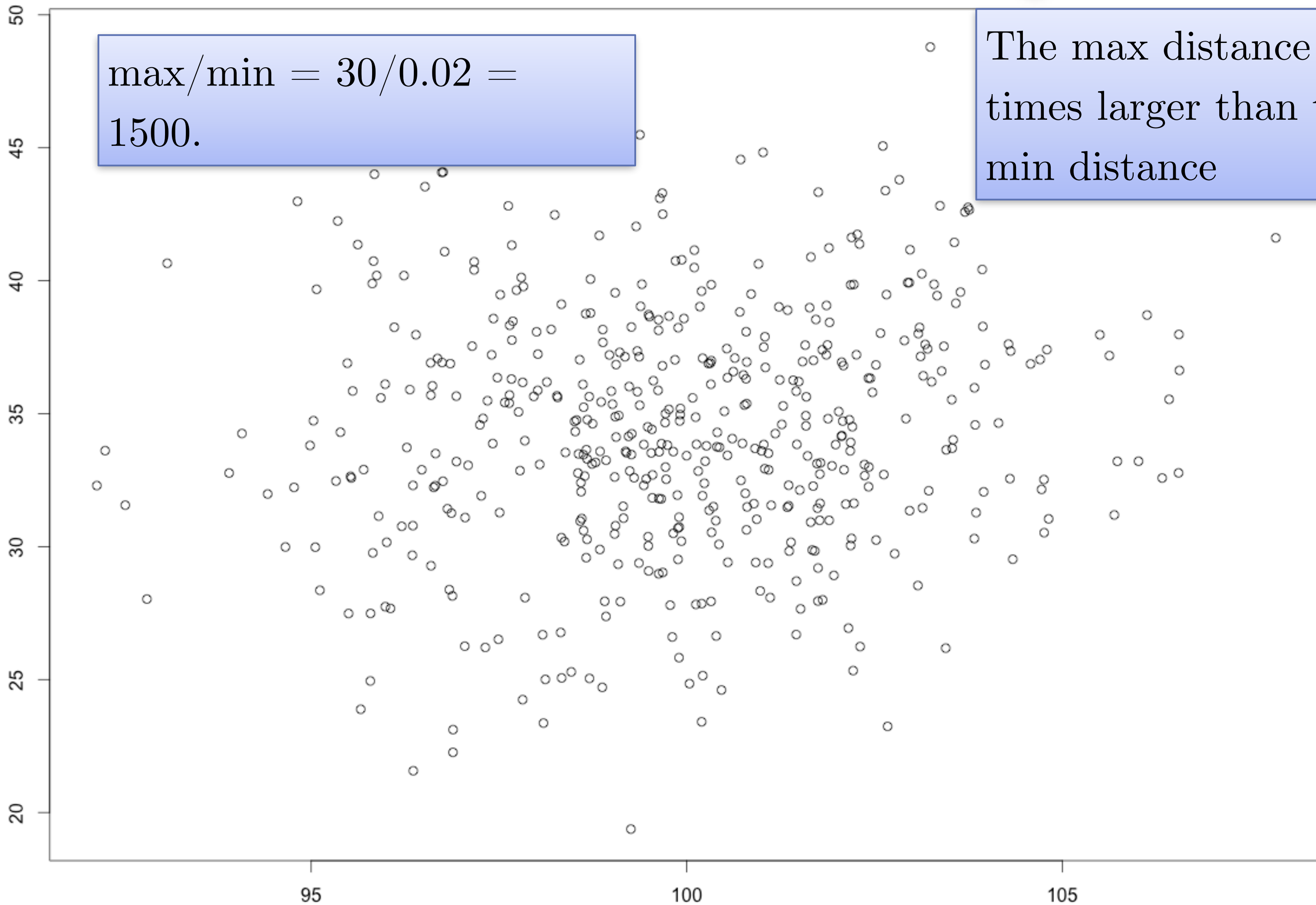
The Curse of Dimensionality



The Curse of Dimensionality



The Curse of Dimensionality



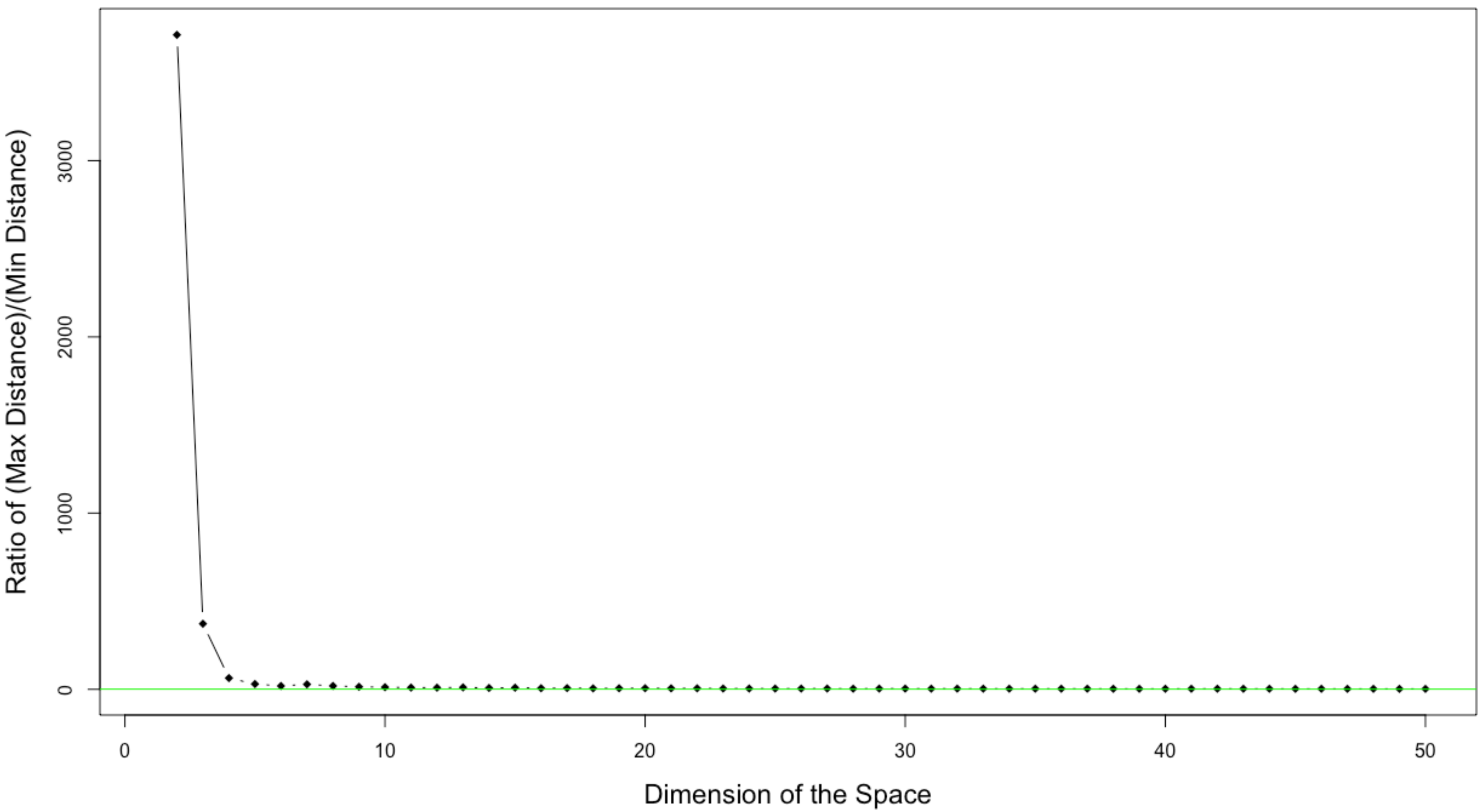
$\max/\min = 30/0.02 = 1500.$

The max distance is 1500 times larger than the min distance

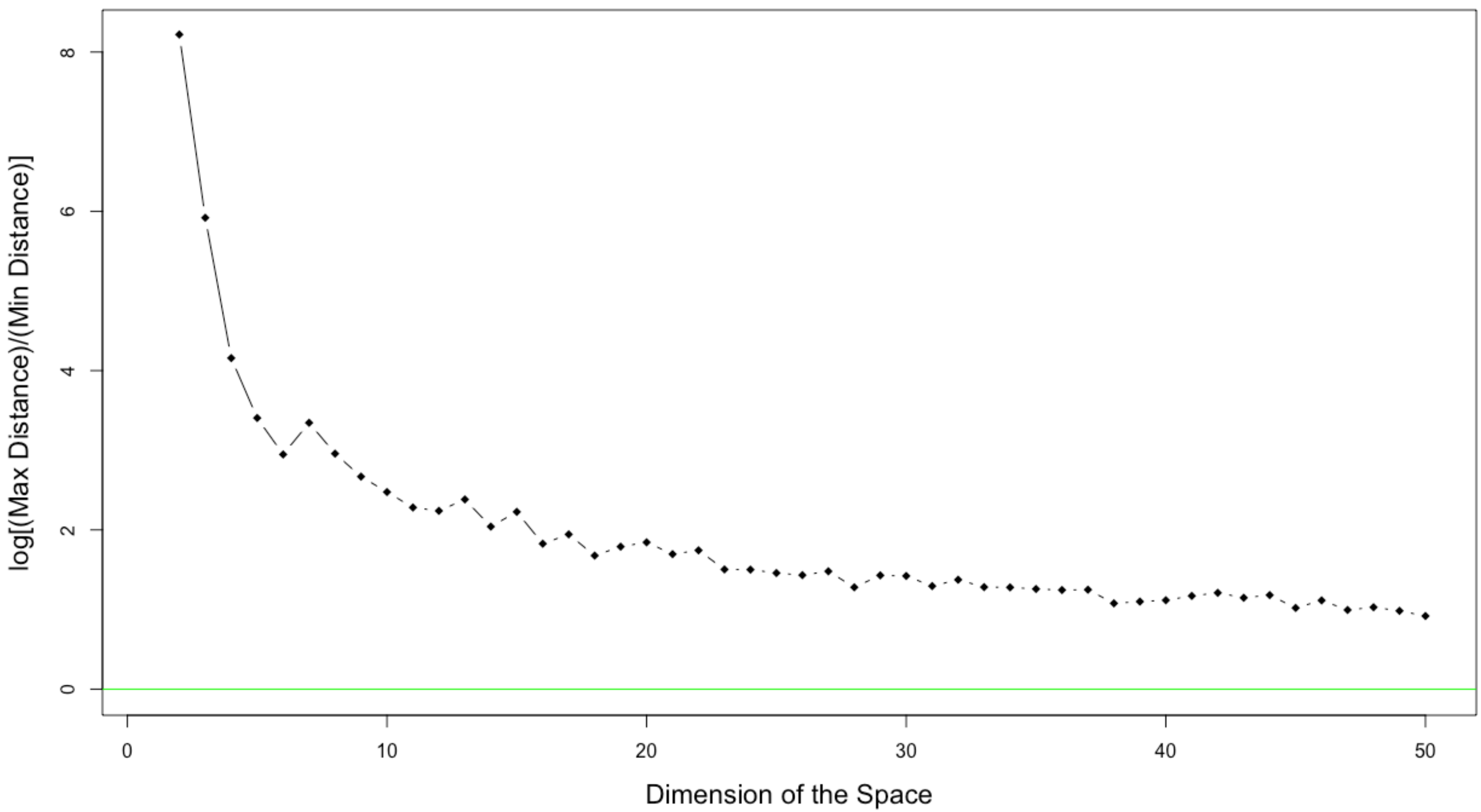
The Curse of Dimensionality

- Now generate those 500 points in $\mathbb{R}^3, \mathbb{R}^4, \ldots, \mathbb{R}^{500}$
- Compute that same metric, the ratio of the maximum distance to the minimum distance
- Observe behavior as the number of dimensions grows...

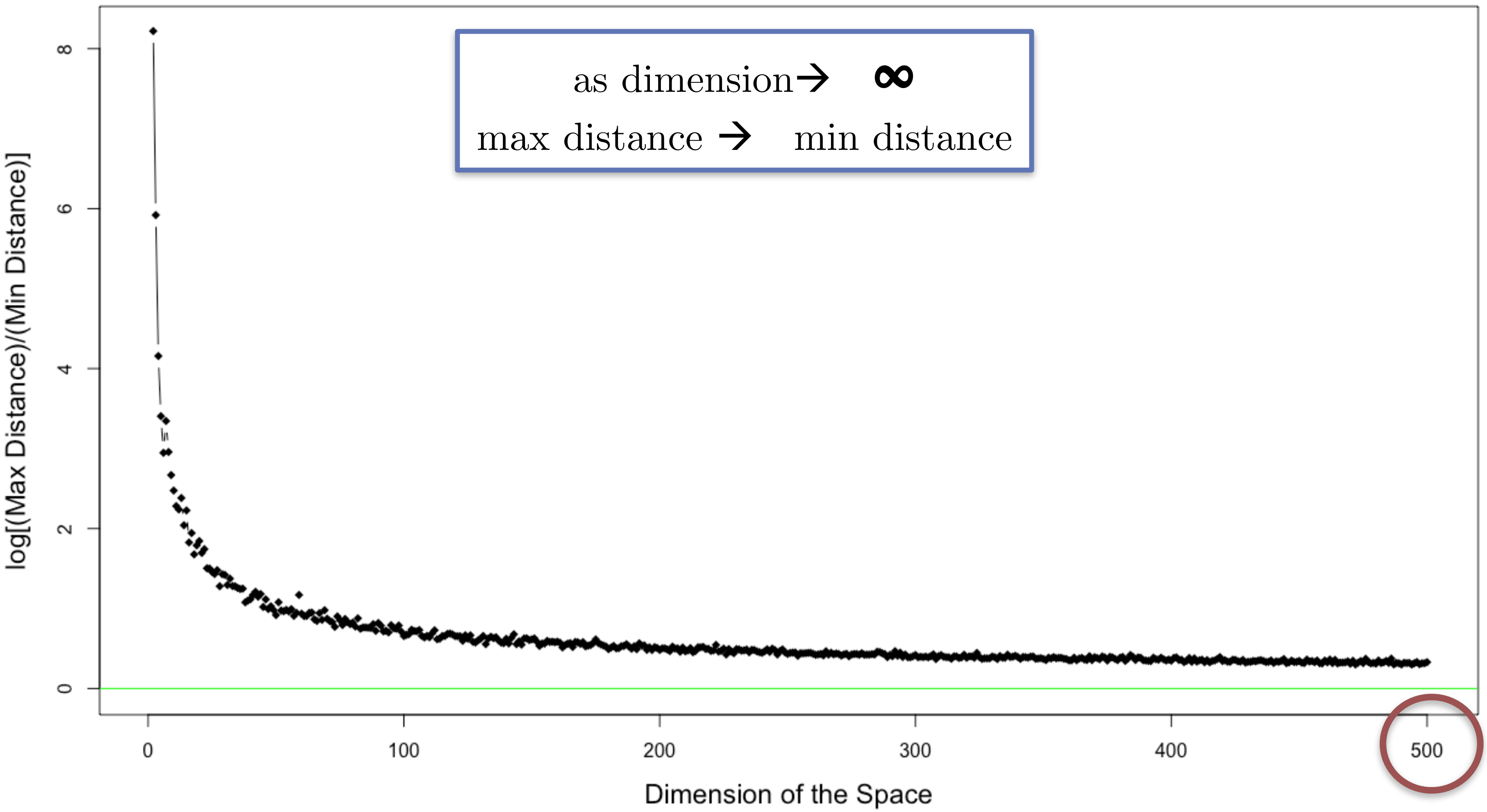
The Curse: Euclidean Distance



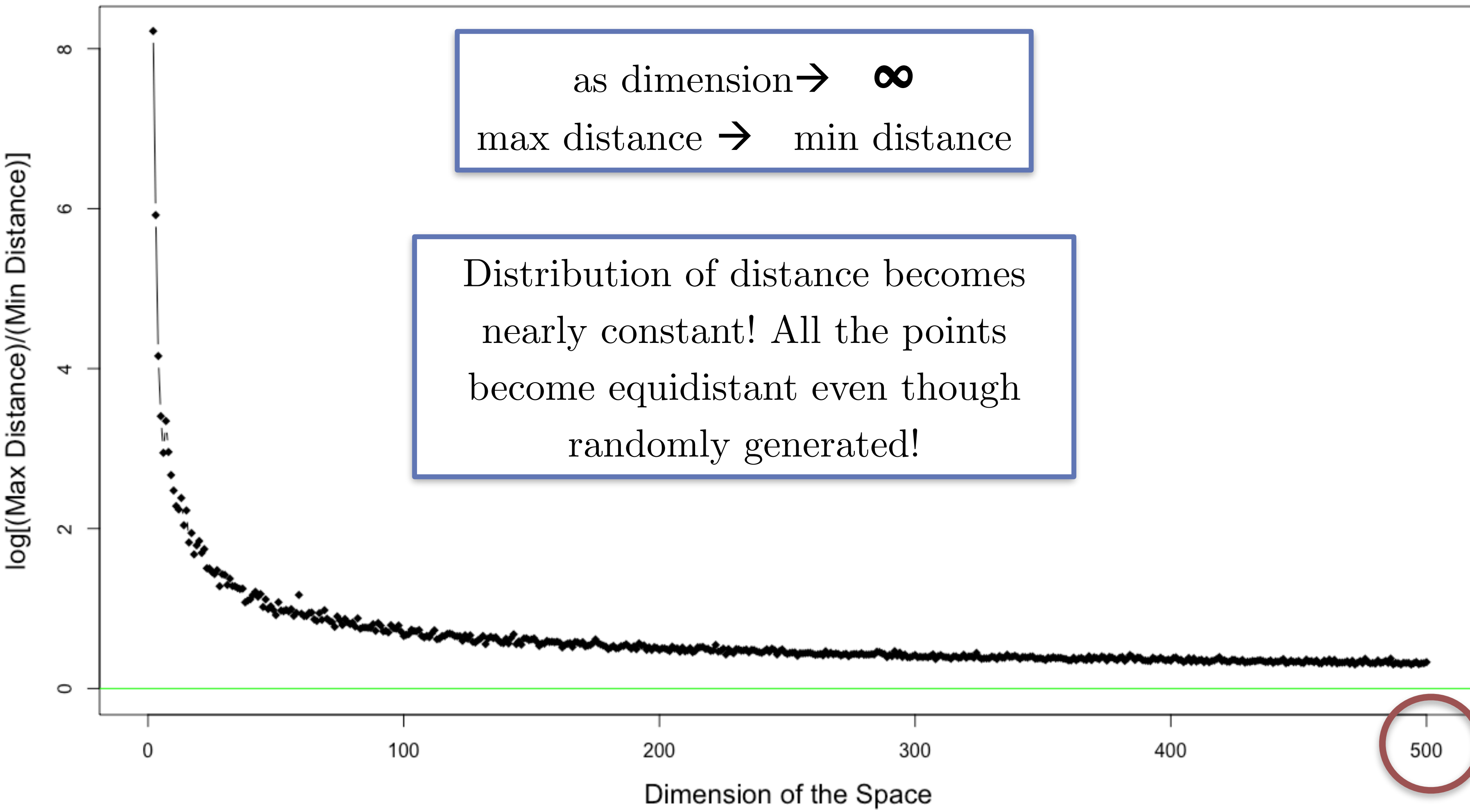
The Curse: Euclidean Distance



The Curse: Euclidean Distance

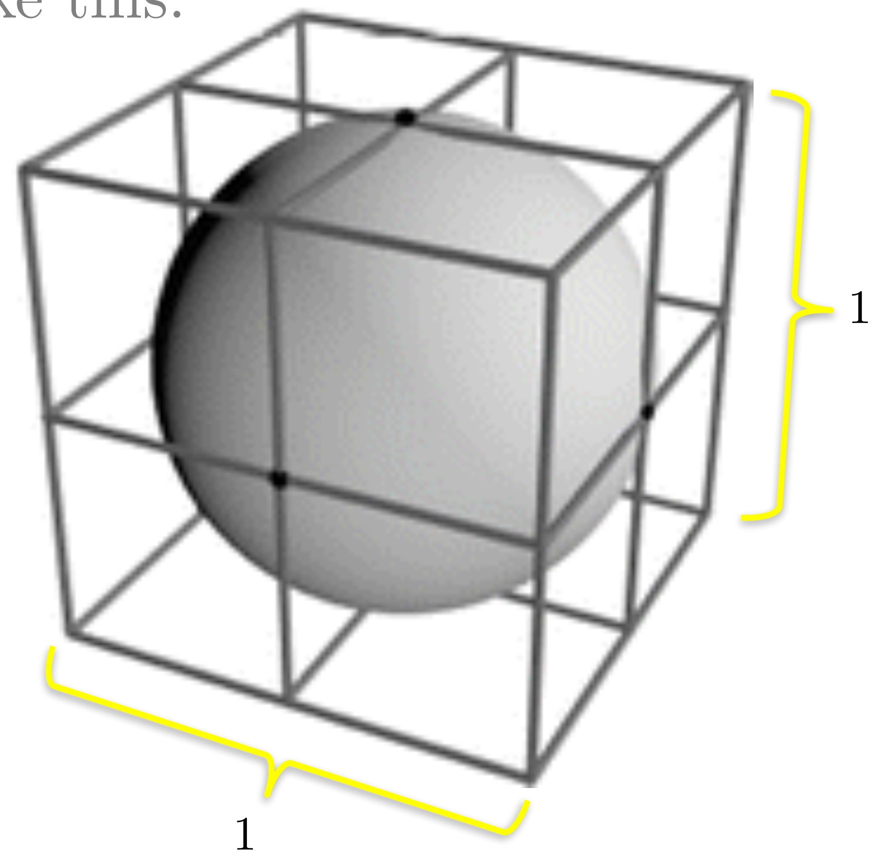


The Curse: Euclidean Distance

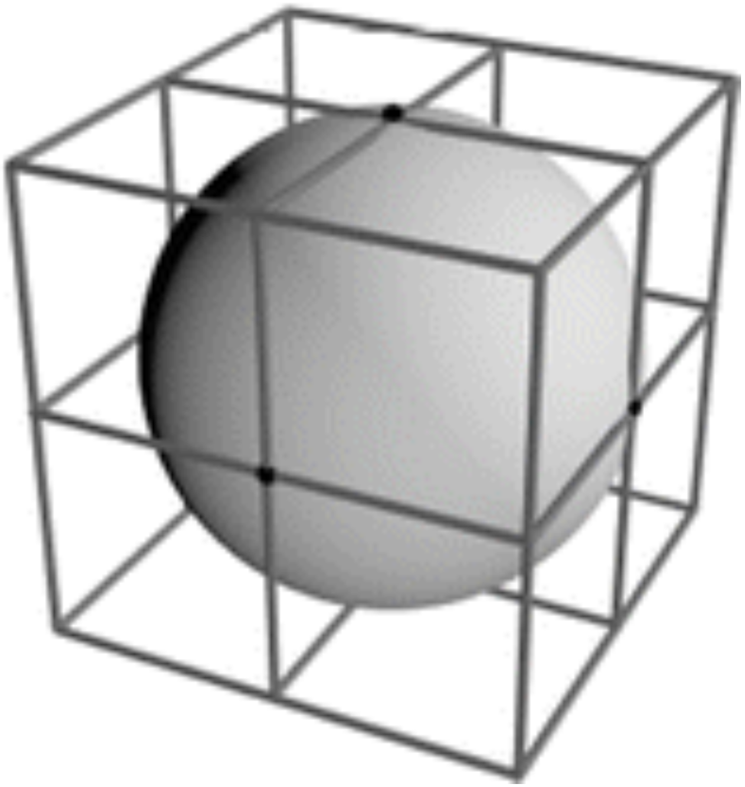


The Curse: Volume of Sphere to Cube

- Imagine a sphere that sits perfectly (inscribed) inside of a cube.
- In 3-dimensions, it looks like this:
- Assume a *unit* cube and unit diameter sphere



The Curse: Volume of Sphere to Cube



Volume of Sphere:

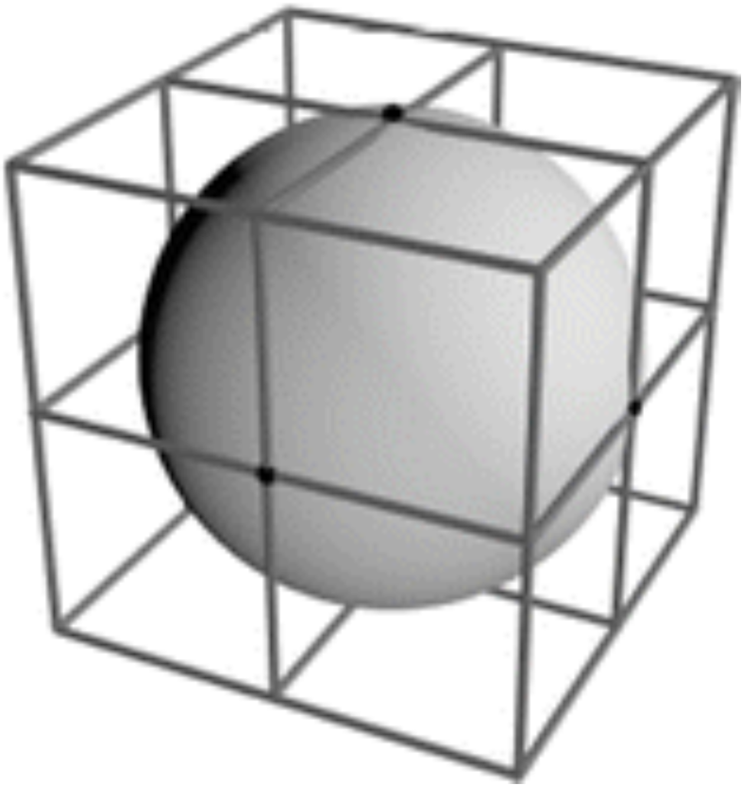
$(4/3)\pi(0.5)^3 \approx 0.52$

Volume of Cube:

1

So the sphere takes up over half of the space.

The Curse: Volume of Sphere to Cube



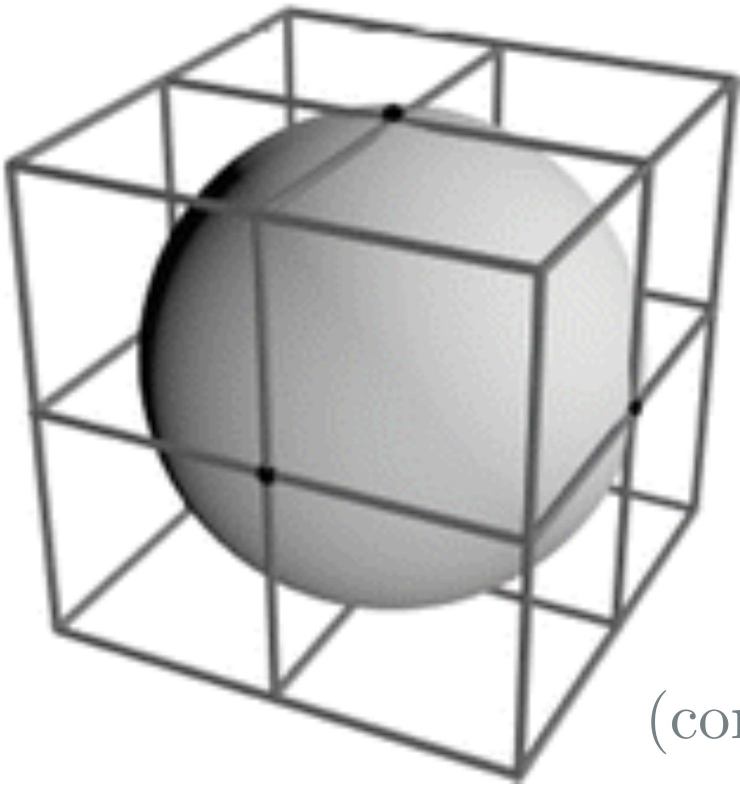
In d-space, the volume of hypersphere:

$$\frac{2r^d\pi^{\frac{d}{2}}}{d\Gamma(\frac{d}{2})}$$

Volume of hypercube:

$$l^d = 1$$

The Curse: Volume of Sphere to Cube



$$\lim_{d\to\infty} \frac{\text{SphereVolume}}{\text{CubeVolume}} = 0$$

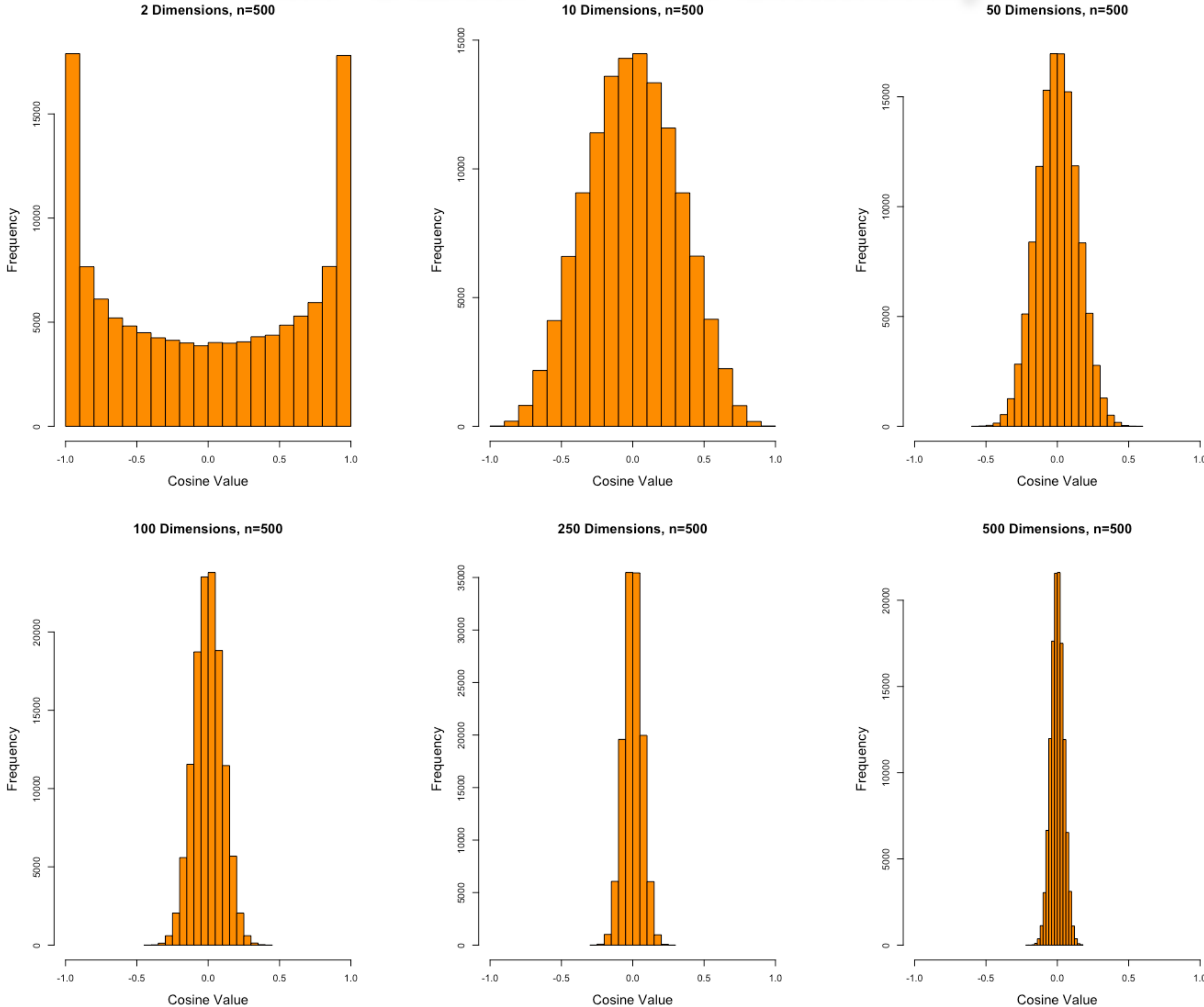
It's as if ***ALL*** of the volume of the hypercube is contained in the corners as the dimension of the space grows large!

(comparatively no volume in the sphere)

The Curse of Dimensionality

- No distance/similarity metric is immune to the vastness of high dimensional space.
- One more. Let's look at the distribution (or lack thereof) of cosine similarity.
- Compute the cosine similarity between each pair of points, observe the distribution as the space grows.

The Curse: Cosine Similarity



When is this a problem?

- *Primarily* **when using algorithms which rely on distance or similarity**
  - Clustering
  - Nearest-neighbor methods
- On any model due to collinearity and a desire for **model simplicity** and **computational efficiency**.
  - Predictive models usually suffer from high variance (overfitting) in high dimensional data
  - Computational load can be *greatly* reduced in many scenarios

What can we do about it?

…

Dimension Reduction

Dimension Reduction Overview

FEATURE SELECTION	FEATURE EXTRACTION
Choose subset of existing features	Create new features
By their relationship to a target (supervised)	Often linear combinations of existing features (PCA, SVD, NMF)
By their distribution/correlation with others (unsupervised)	Often chosen to be uncorrelated

Feature Selection

- Removing features manually
  - Redundant (multicollinearity/VIFs)
  - Irrelevant (Text mining stop words)
  - Poor quality features (>50% missing values)
- Forward/Backward/Stepwise Regression
- Decision Tree
  - Variable Importance Table
  - Can change a little depending on metric
    - Gini/Entropy/Mutual Information/Chi-Square

Feature Extraction: Continuous Variables

- PCA
  - Create a new set of features as linear combinations of your originals
  - These new features are ranked by variance (importance/information)
  - Use the first several PCs in place of original features
- SVD
  - Same as PCA, except the 'variance' interpretation is no longer valid
  - Common for text-mining, since X^TX is related to cosine similarity.
- Factor Analysis
  - The principal components are rotated so that our new features are more interpretable.
  - Occasionally other factor analysis algorithms like maximum likelihood are considered.

Feature Extraction: Continuous Variables

- Discretization/Binning

- While this doesn't reduce the dimensions of your data (it increases them!), it is still a form of feature extraction!

Feature Extraction: Nominal Variables

Encoding variables with numeric values.

Checking Account Balance	
Original Level	New Value
Negative	-100
No checking account	0
Balance is zero	0
0<Balance<200	100
200<Balance<800	500
Balance>800	900
Balance>800 and IncomeDD	1000

Feature Extraction: Nominal Variables

- **Target encoding/Optimal Scaling** with numeric values.
  - In supervised learning, can **let the numeric value** of level=L be **the average target value** of all observations that have level = L

- **Correspondence analysis**
  - Method similar to PCA for categorical data.
  - Uses chi-squared table (contingency table) and chi-squared distance.
  - **Provides coordinates of categorical variables in a lower-dimensional space.**
  - More often used as exploratory method, potentially for binning purposes.