DATA REDUCTION AND TRANSFORMATIONS

DATA COLLECTION AND DATA PROCESSING

LEARNING OBJECTIVES

Familiarity with the following concepts:

- Dimensionality of data
- Curse of Dimensionality
- Feature selection
- Principal Component Analysis (PCA)
- Data transformation
- Scaling
- Discretization

DIMENSIONALITY OF DATA

In data analysis, the **dimension** of the data is the number of variables (or attributes) that are collected in a dataset, represented by the number of columns.

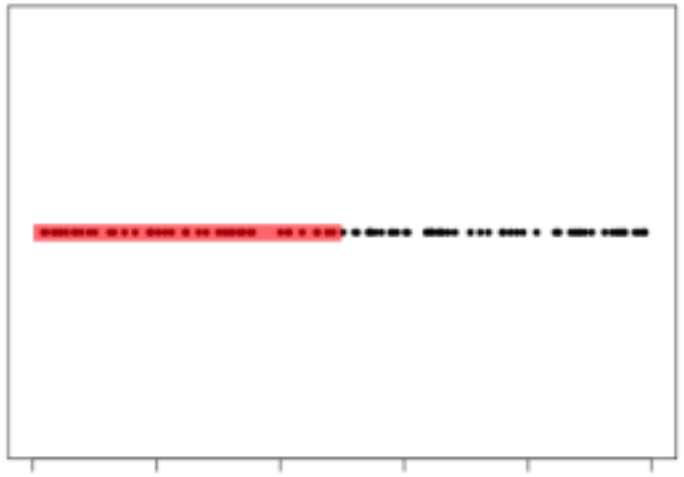
Here the term dimension is an extension of the use of the term to refer to the size of a vector.

We can think of the number of variables used to describe each object (row) as a vector describing that object.

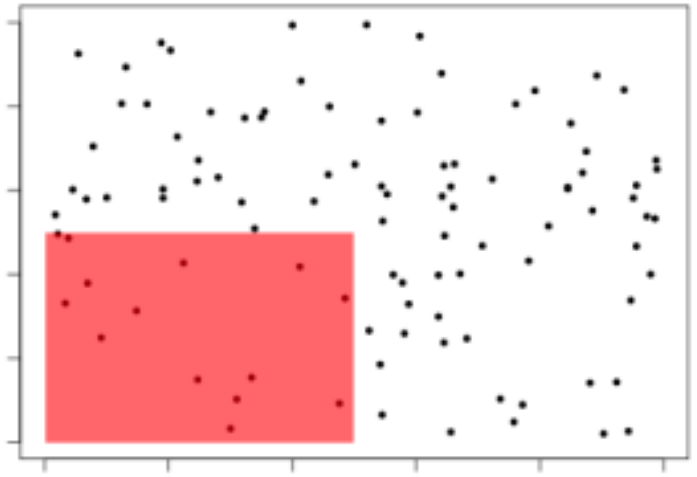
(Note – the term dimension is used differently in business intelligence contexts)

CURSE OF DIMENSIONALITY

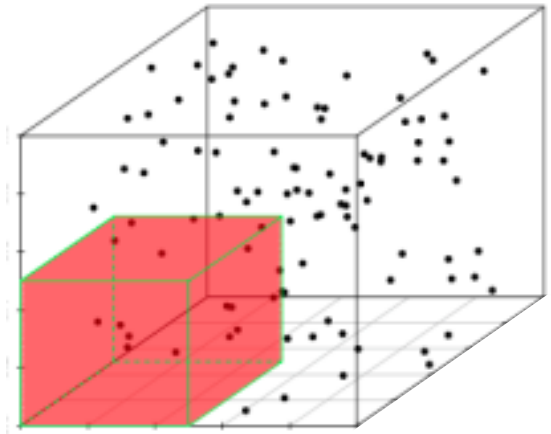
42% of data is captured



14% of data is captured



7% of data is captured



$N = 100$ observations, uniformly distributed on $[0,1]^d, d = 1,2,3.$
% of observations captured by $[0,1/2]^d, d = 1,2,3.$

SAMPLING OBSERVATIONS

Question: does every row of the dataset need to be used?

If rows are selected randomly (with or without replacement), the resulting sample might be **representative** of the entire dataset.

Drawbacks:

- if the signal of interest is rare, sampling might drown it altogether
- if aggregation is happening down the road, sampling will necessarily affect the numbers (passengers vs. flights)
- even simple operations on a large file (finding the # of lines, say) can be taxing on the memory and in terms of computation time – **prior information on the dataset structure can help**

FEATURE SELECTION

Removing **irrelevant** or **redundant** variables is a common data processing task.

Motivations:

- modeling tools do not handle these well (variance inflation due to multicolinearity, etc.)
- dimension reduction (# variables ≫ # observations)

Approaches:

- filter vs. wrapper
- unsupervised vs. supervised

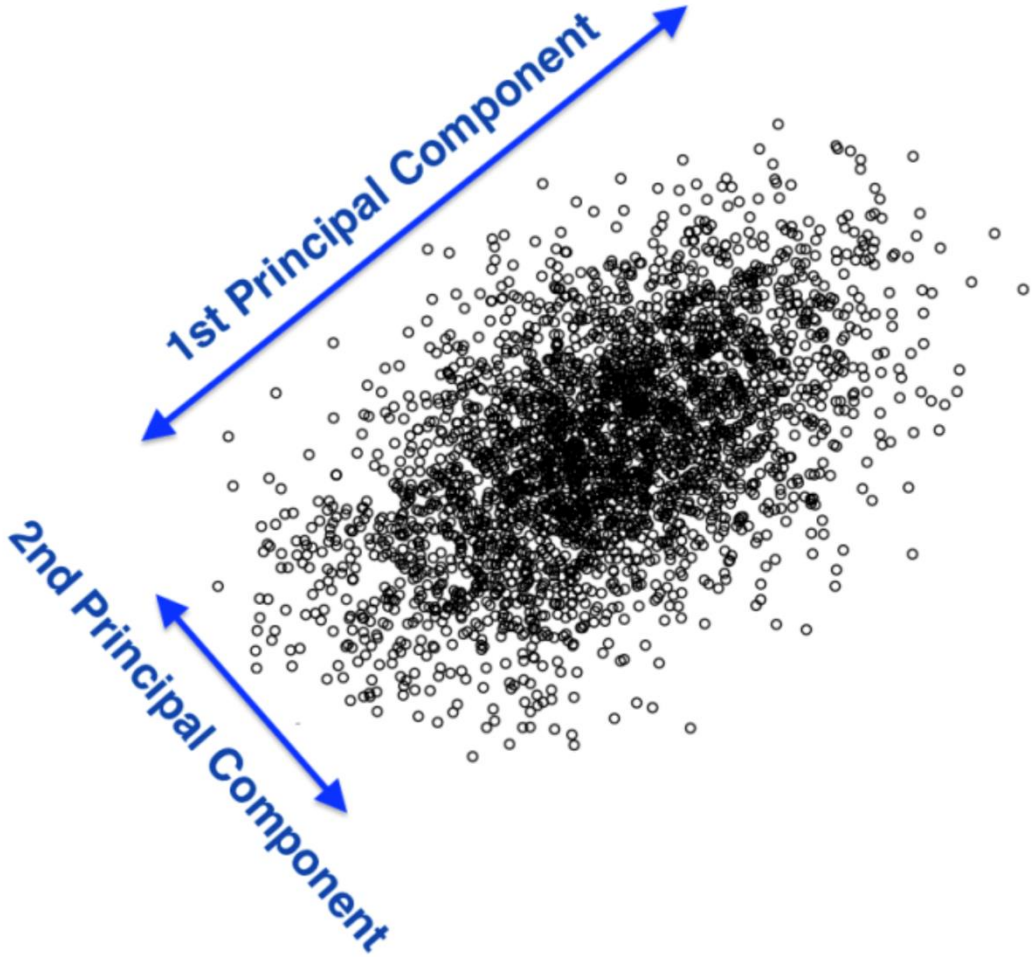
[A. Ng, K. Soo, *Numsense!*]

PRINCIPAL COMPONENT ANALYSIS

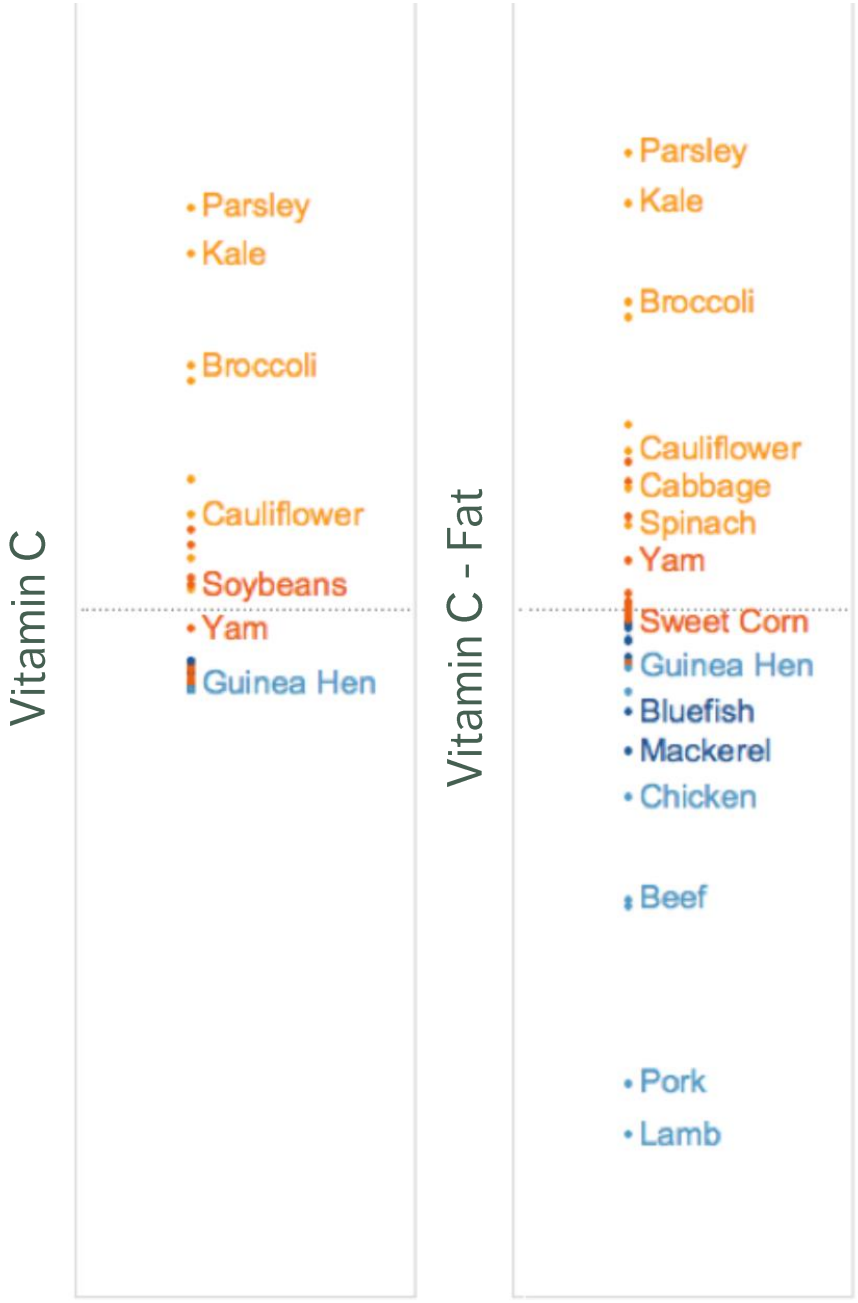
Motivational Example:
Nutritional Content of Food

What is the best way to differentiate food items? Vitamin content, fat, or protein level? A bit of each?

Principal Component Analysis (PCA) can be used to find the combinations of variables along which the data points are **most spread out.**



[A. Ng, K. Soo, *Numsense!*]



DIFFERENTIATION

Vitamin C is present in various levels in fruit and vegetables, but not in meats. It **separates** vegetables from meats, and specific vegetables from one another (to some extent), but the meats are **clumped together** (left).

The situation is reversed for *Fat* levels, so the **combination** of vitamin C and fat **separates** vegetables from meats, and **spreads** vegetables and meats (right).

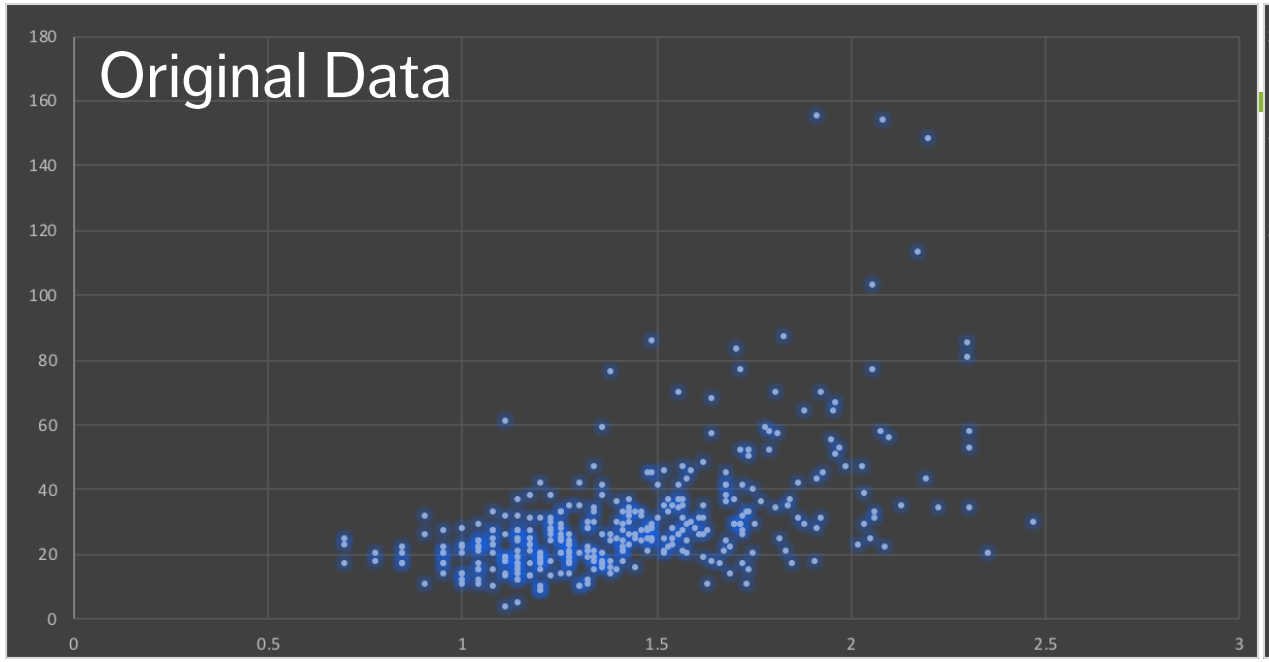
COMMON TRANSFORMATIONS

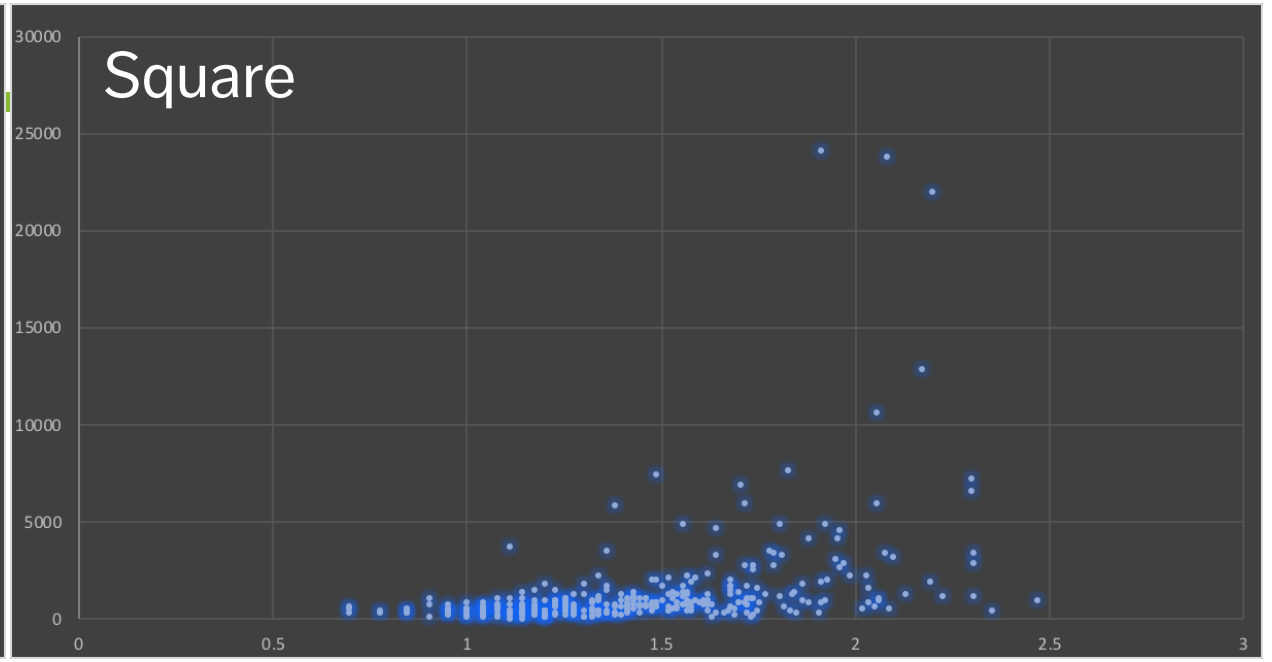
Models sometimes require that certain data assumptions be met (normality of residuals, linearity, etc.).

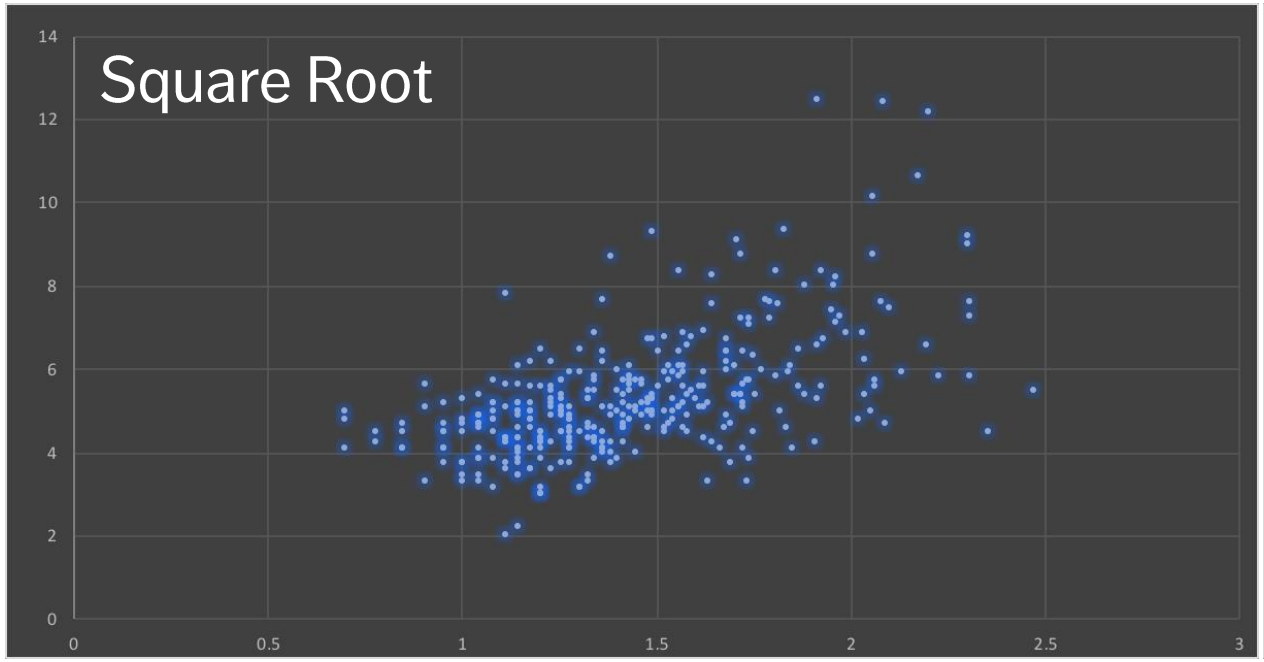
If the raw data does not meet the requirements, we can either

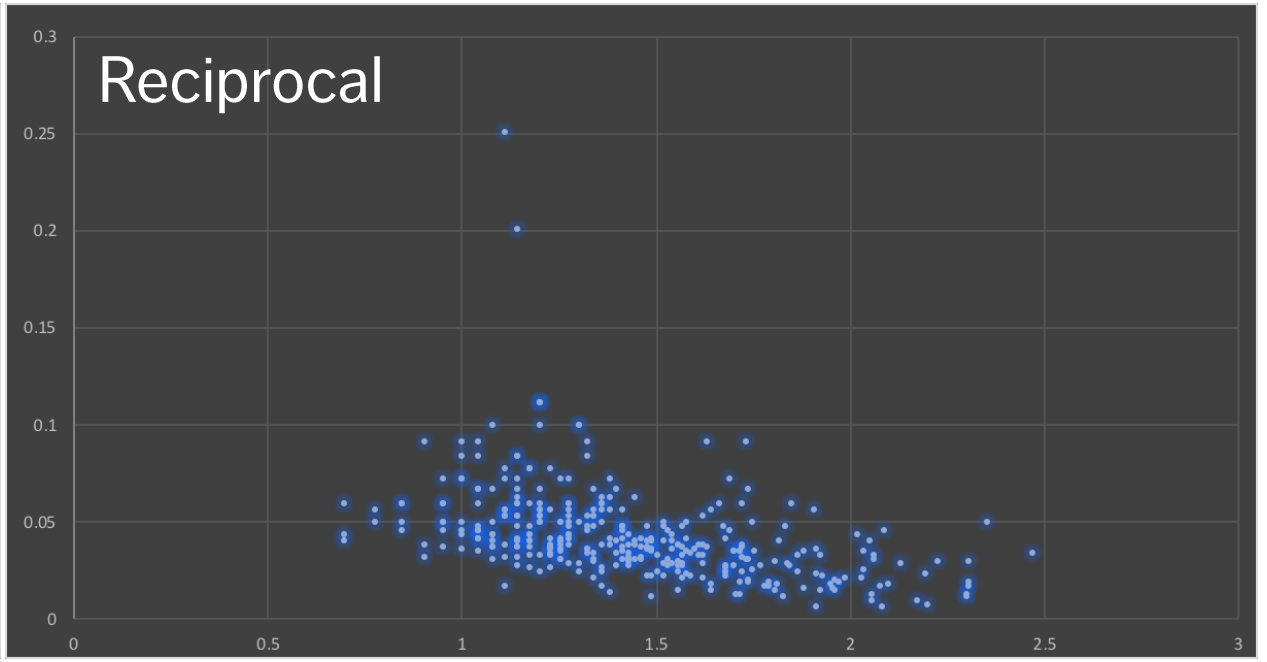
- abandon the model
- attempt to **transform** the data

The second approach requires an inverse transformation to be able to draw conclusions about the original data.


Original Data
180
160
140
120
100
80
60
40
20
0
0
0.5
1
1.5
2
2.5
3


Square
30000
25000
20000
15000
10000
5000
0
0
0.5
1
1.5
2
2.5
3


Square Root
14
12
10
8
6
4
2
0
0
0.5
1
1.5
2
2.5
3


Reciprocal
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.5
1
1.5
2
2.5
3

SCALING

Numeric variables may have different **scales** (weights and heights, for instance).

The variance of a large-range variable is typically greater than that of a small-range variable, introducing a bias (for instance).

Standardization creates a variable with mean 0 and std. dev. 1:

$$Y_i = \frac{X_i - \bar{X}}{s_X}$$

Normalization creates a new variable in the range $[0,1]$: $Y_i = \frac{X_i - \min X}{\max X - \min X}$

DISCRETIZING

To reduce computational complexity, a numeric variable may need to be replaced by an **ordinal** variable (from *height* value to "*short*", "*average*", "*tall*", for instance).

Domain expertise can be used to determine the bins' limits (although that could introduce unconscious bias to the analyses)

In the absence of such expertise, limits can be set so that either

- the bins each contain the same number of observations
- the bins each have the same width
- the performance of some modeling tool is maximized

CREATING VARIABLES

New variables may need to be introduced:

- as **functional relationships** of some subset of available features
- because modeling tool may require **independence of observations**
- because modeling tool may require **independence of features**
- to simplify the analysis by looking at **aggregated summaries** (often used in text analysis)

Time dependencies ⟶ time series analysis

Spatial dependencies ⟶ spatial analysis

Supplemental Material

LOCAL METHODS IN HIGH DIMENSIONS

A model is said to be **local** if it depends solely on the data *near* the input vector (kNN is local, linear regression isn't).

With a **large training set**, we could increase k (in a kNN model, say) and get enough data points to provide a solid approximation to the theoretical boundary.

The **Curse of Dimensionality** (CoD) is the breakdown of this approach in high-dimensional spaces: when the # of features increases, the # of observations required to maintain predictive power also increases... **but at a substantially higher rate**.

MANIFESTATIONS OF COD

Let $x_i \sim U^1(0,1), i = 1, \ldots, N$ be i.i.d.

For any $z \in [0,1]$ and $\varepsilon > 0$ such that

$$I_1(z;\varepsilon) = \left[z - \frac{\varepsilon}{2}, z + \frac{\varepsilon}{2}\right] \subseteq [0,1],$$

we expect $\left|I_1(z;\varepsilon) \cap \{x_i\}_{i=1}^N\right| \approx \varepsilon \cdot N$

In other words, a subset whose edge is ε percent of the original set in $\mathbb{R}$ contains ε percent of the observations.

MANIFESTATIONS OF COD

Let $x_i \sim U^2(0,1), i = 1, \ldots, N$ be i.i.d.

For any $z \in [0,1]^2$ and $\varepsilon > 0$ such that

$$I_2(z;\varepsilon) = \left[z_1 - \frac{\varepsilon}{2}, z_1 + \frac{\varepsilon}{2}\right] \times \left[z_2 - \frac{\varepsilon}{2}, z_2 + \frac{\varepsilon}{2}\right] \subseteq [0,1]^2,$$

we expect $\left|I_2(z;\varepsilon) \cap \{x_i\}_{i=1}^N\right| \approx \varepsilon^2 \cdot N$

In other words, a subset whose edge is ε percent of the original set in $\mathbb{R}^2$ contains ε^2 percent of the observations.

MANIFESTATIONS OF COD

Let $x_i \sim U^p(0,1), i = 1, \ldots, N$ be i.i.d.

For any $z \in [0,1]^p$ and $\varepsilon > 0$ such that

$$I_p(z;\varepsilon) = \prod_{j=1}^{p} \left[z_j - \frac{\varepsilon}{2}, z_j + \frac{\varepsilon}{2} \right] \subseteq [0,1]^p,$$

we expect $\left| I_p(z;\varepsilon) \cap \{x_i\}_{i=1}^N \right| \approx \varepsilon^p \cdot N$

In other words, a subset whose edge is ε percent of an original set in $\mathbb{R}^p$ contains ε^p percent of the observations.

MANIFESTATIONS OF COD

To capture r percent of the observations uniformly distributed in a unit p-hypercube, we need a hyper-subset with edge

$$\varepsilon_p(r) = r^{1/p}.$$

For instance, for $r = 33\%$, we need a subset with edge

- $\varepsilon_1(1/3) \approx 0.33$ in $\mathbb{R}$
- $\varepsilon_2(1/3) \approx 0.58$ in $\mathbb{R}^2$
- $\varepsilon_{10}(1/3) \approx 0.90$ in $\mathbb{R}^{10}$

Locality is lost!

SUPERVISED FILTER METHODS

Correlation between a feature X and a target variable Y:

$$\rho_{X,Y} = \frac{\sum_{i=1}^{N}(y_i - \bar{y})\,(x_i - \bar{x})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2} \cdot \sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}}$$

Features which are highly correlated with the target variable are retained, but this approach is limited if the relationship to the target variable is **non-linear**.

SUPERVISED FILTER METHODS

Mutual Information of nominal target Y from nominal feature X:

$$I(Y;X) = H(Y) - H(Y|X)$$

where the **entropy** and **conditioned class entropy** are given by

$$H(Y) = -\sum_c P(Y=c) \cdot \log P(Y=c)$$ (v, c represent levels of X, Y).

$$H(Y|X) = -\sum_{v,c} P(X=v, Y=c) \cdot \log \frac{P(X=v,Y=c)}{P(X=v)}$$

$I(Y;X)$ measures the amount of information that can be obtained about Y by knowing X.

[Convention: $0 \times \log(0) = 0$]

LASSO AND VARIANTS

Stepwise Selection is a form of *Occam's Razor*: at each step, a new feature is considered for **inclusion** or **removal** from the current features set based on some criterion (F-test, t-test, etc.).

Limitations:

- tests are biased, since they are based on the same data.
- adjusted R^2 only takes into account the number of features in the final fit, and not the d.f. that have been used in the entire model.
- if cross-validation is used, stepwise selection has to be repeated for each sub-model (that's not usually done).
- classic example of p-hacking (results without hypothesis).

LASSO AND VARIANTS

In what follows, we assume that we have N **centered** and **scaled** $x_i = \left(x_{1,i}, \cdots, x_{p,i}\right)^T$ and a target observation y_i.

Let $\hat{\beta}_{LS,j} = [(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}]_j$ be the j^{th} OLS coefficient and set a threshold $\lambda > 0$, whose value depends on the training dataset.

In general, there are **no restrictions** on the values taken by the coefficients $\hat{\beta}_{LS,j}$ – larger magnitudes imply that corresponding features **play an important role** in predicting the target.

LASSO AND VARIANTS

Ridge regression is a method to **regularize** the regression coefficients (the effect is to shrink the coefficient values)

The problem consists in solving

$$\arg\min_{\boldsymbol{\beta}}\{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+N\lambda\|\boldsymbol{\beta}\|_2^2\},$$

which yields ridge coefficients

$$\hat{\beta}_{RR,j}=\frac{\hat{\beta}_{LS,j}}{1+N\lambda}$$

LASSO AND VARIANTS

Regression with best subset selection is a method that sets some regression coefficients to 0 (potentially).

The problem consists in solving

$$\arg\min_{\boldsymbol{\beta}}\{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+N\lambda\|\boldsymbol{\beta}\|_0\},\ \text{where}\ \|\boldsymbol{\beta}\|_0=\sum_j \text{sign}(|\beta_j|),$$

which yields coefficients

$$\hat{\beta}_{BS,j}=\begin{cases}0 & \text{if } |\hat{\beta}_{LS,j}|<\sqrt{N\lambda}\\ \hat{\beta}_{LS,j} & \text{if } |\hat{\beta}_{LS,j}|\geq\sqrt{N\lambda}\end{cases}$$

LASSO AND VARIANTS

Least Absolute Shrinkage and Selection Operator (LASSO) is a regression method for **feature selection** and **regularization**.

The problem consists in solving

$$\arg\min_{\boldsymbol{\beta}}\{\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+N\lambda\|\boldsymbol{\beta}\|_1\}$$

which yields lasso coefficients

$$\hat{\beta}_{L,j}=\hat{\beta}_{LS,j}\cdot\max\left(0,1-\frac{N\lambda}{|\hat{\beta}_{LS,j}|}\right)$$

LASSO AND VARIANTS

LASSO combines the properties of **ridge regression** (shrinkage) and **best subset selection** (feature selection).

Ridge regression can be viewed as linear regression with prior **normal distributions** assigned to the coefficients; these are **Laplace distributions** in lasso regression.

Lasso selects **at most** $\max\{p, N\}$ features, and usually selects no more than one feature in a group of highly correlated variables.

Extensions: elastic nets; group, fused and adaptive lassos; bridge regression

[A. Ng, K. Soo, *Numsense!*]

PRINCIPAL COMPONENTS

Presence of nutrients appears to be correlated among food items.

In the (small) sample consisting of Lamb, Pork, Kale, and Parsley, *Fat* and *Protein* levels seem in step, as do *Fiber* and *Vitamin C*.

In a larger dataset, the correlations are $r = 0.56$ and $r = 0.57$.

How much could 2 variables explain?

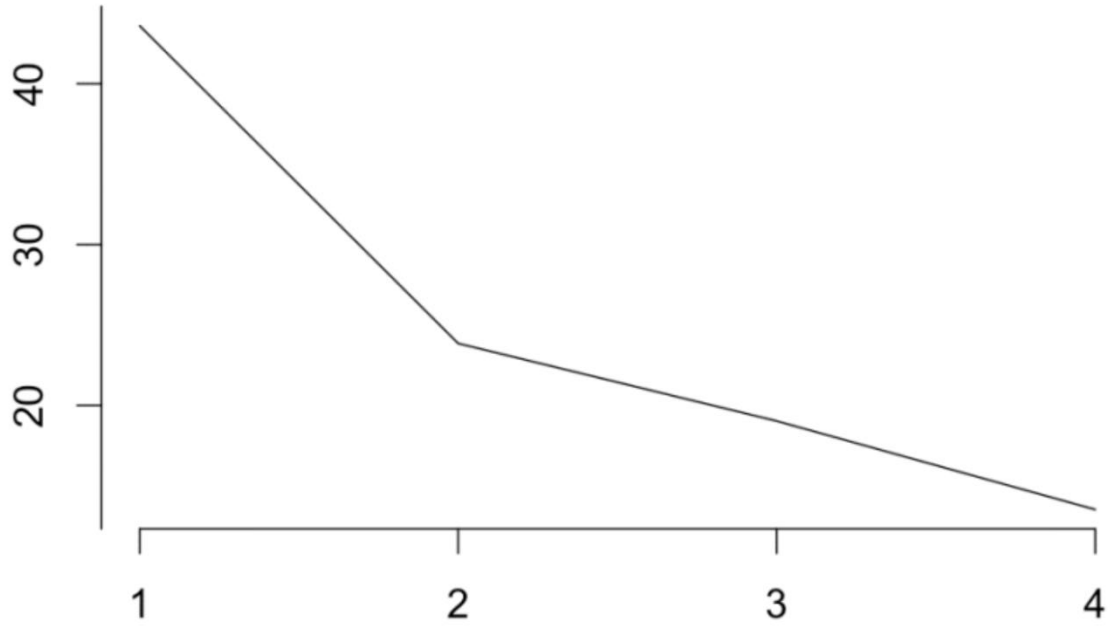
RETAINING PRINCIPAL COMPONENTS

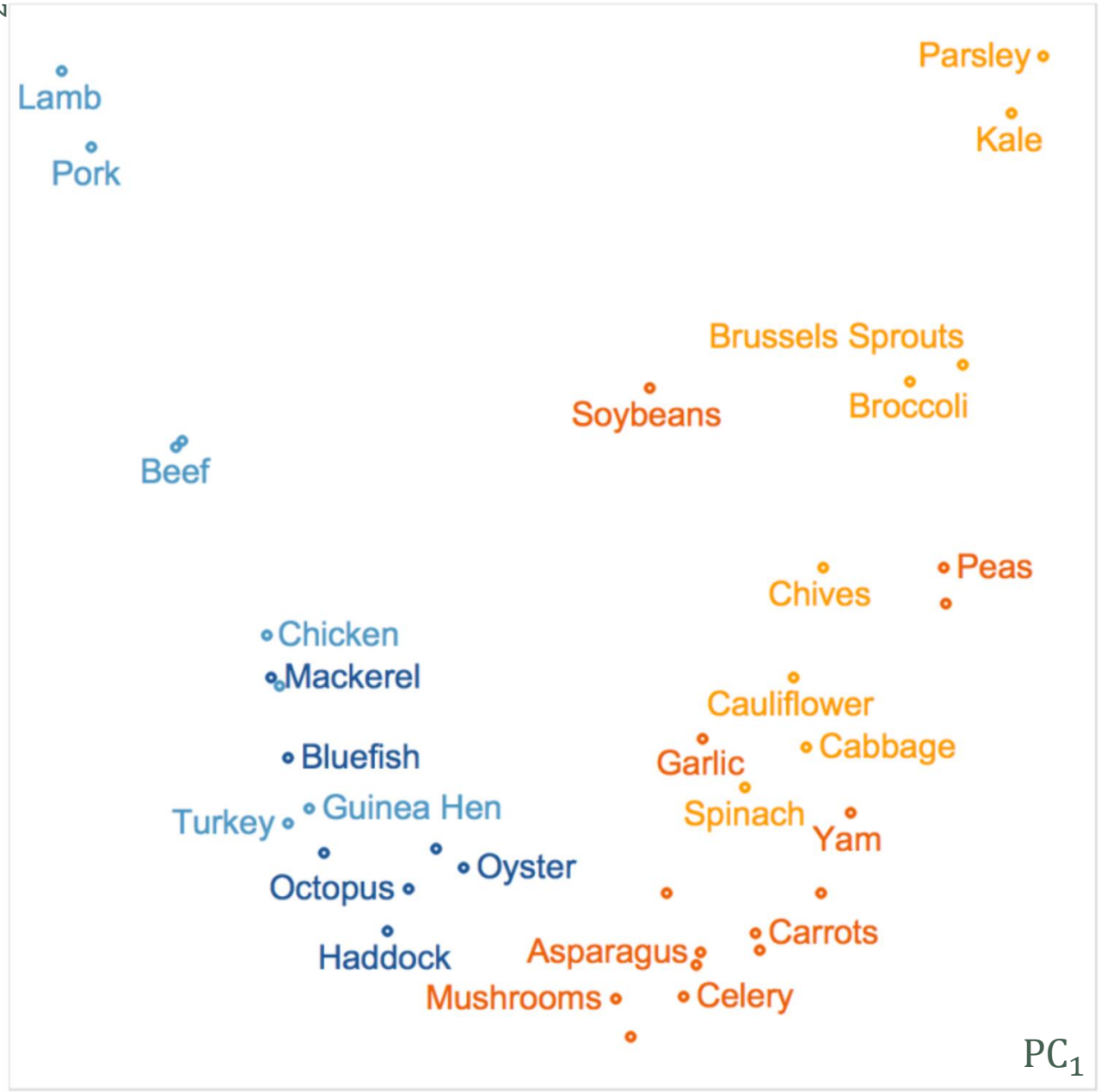
The **proportion of the spread** in the data which can be explained by each principal component is shown in the scree plot.

How many PCs are retained in the analysis?

- keep the PCs where the cumulative proportion is below some threshold
- keep the PCs leading to a kink

Here, 2 PCs ≈ 68% of the spread.





DIFFERENTIAION (REPRISE)

PC_1 differentiates vegetables from meats; PC_2 differentiates two **sub-categories** within these:

- Meats are concentrated on the left (low PC_1 values).
- Vegetables are concentrated on the right (high PC_1 values).
- Seafood have lower *Fat* content (low PC_2 values) and are concentrated at the bottom.
- Non-leafy veggies have lower *Vitamin C* content (low PC_2 values) and are also bunched at the bottom.

PCA IN THEORY

PCA attempts to fit a p**-ellipsoid** to centered and scaled* data. Ellipsoid axes represent the principal components of the data. Small axes are components along which the variance is "small"; removing these component leads to a "small" loss of information.

Procedure:

1. Centre and scale the data: matrix $\boldsymbol{X}$
2. Compute the data's covariance matrix $\boldsymbol{K} = \boldsymbol{X}^T \boldsymbol{X}$
3. Compute $\boldsymbol{K}$'s eigenvalues $\boldsymbol{\Lambda}$ and orthonormal eigenvector matrix $\boldsymbol{W}$
4. Each eigenvector $\boldsymbol{w}$ represents an axis, whose variance is given by the associated eigenvalue $\boldsymbol{\lambda}$

PCA IN THEORY

The eigenvectors $\boldsymbol{w}$ are also called the **loadings**.

Typically, the eigenvalues are ordered in **decreasing** sequence, so that the first loading corresponds to the largest axis.

$\boldsymbol{K}$ positive semi-definite $\Longrightarrow$ eigenvalues $\lambda = s^2$ are positive; s is a singular value of $\boldsymbol{X}$ (i.e. a diagonal entry of $\boldsymbol{\Sigma}$ in the **singular value decomposition** $\boldsymbol{X} = \boldsymbol{U\Sigma W}^T$).

The PCA decomposition of $\boldsymbol{X}$ is $\boldsymbol{T} = \boldsymbol{XW} = \boldsymbol{U\Sigma}$.

PCA IN THEORY

The link between the PCs and the eigenvectors can be made explicit:

- the **first** principal component is the loading which maximizes the variance of the first column of $\boldsymbol{T}$

$$\boldsymbol{w}^1 = \arg\max_{\|\boldsymbol{w}\|=1}\{\mathrm{Var}(\boldsymbol{t}^1)\}$$

- but $\boldsymbol{T}$ is centered so the variance is simply

$$\boldsymbol{w}^1 = \arg\max_{\|\boldsymbol{w}\|=1}\left\{t_{1,1}^2 + \cdots + t_{1,N}^2\right\}$$

- using the PC decomposition of $\boldsymbol{X}$, this becomes

$$\boldsymbol{w}^1 = \arg\max_{\|\boldsymbol{w}\|=1}\{\langle \boldsymbol{x}_1|\boldsymbol{w}\rangle^2 + \cdots + \langle \boldsymbol{x}_N|\boldsymbol{w}\rangle^2\} = \arg\max_{\|\boldsymbol{w}\|=1}\{\|\boldsymbol{X}\boldsymbol{w}\|^2\}$$

PCA IN THEORY

- which by definition of the norm is $\boldsymbol{w}^1 = \arg\max_{\|\boldsymbol{w}\|=1}\{\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}\}$
- since $\|\boldsymbol{w}\| = 1$, the loading also satisfies

$$\boldsymbol{w}^1 = \arg\max_{\|\boldsymbol{w}\|=1}\left\{\frac{\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}}\right\}$$

- using Lagrange multipliers, it can be shown that the critical points of $\frac{\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}}$ are exactly the eigenvectors of $\boldsymbol{K} = \boldsymbol{X}^T\boldsymbol{X}$ (of which there are p)
- if $\boldsymbol{w}$ (unit) and $\lambda^* \geq 0$ are such that $\boldsymbol{K}\boldsymbol{w}=\lambda\boldsymbol{w}$, then

$$\frac{\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}} = \frac{\boldsymbol{w}^T\boldsymbol{K}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}} = \frac{\boldsymbol{w}^T\lambda\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}} = \lambda\frac{\boldsymbol{w}^T\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}} = \lambda$$

PCA NOTES

The loading that explains the most variance along a single axis is the eigenvector of the empirical covariance matrix corresponding to the largest eigenvalue, and that variance is proportional to the eigenvalue.

The process is repeated, yielding **orthonormal** principal components PC_1, ... PC_r, where $r = \mathrm{rank}(\boldsymbol{X})$.

GENERALIZATIONS

Nonlinear PCA-like methods attempt to find **principal manifolds**.

- self-organizing maps
- auto-encoders
- curvilinear component analysis
- manifold sculpting

Rather than reducing the dimensionality, we may **expand** it with kernel PCA (this is equivalent to replacing the usual inner product by more exotic objects).

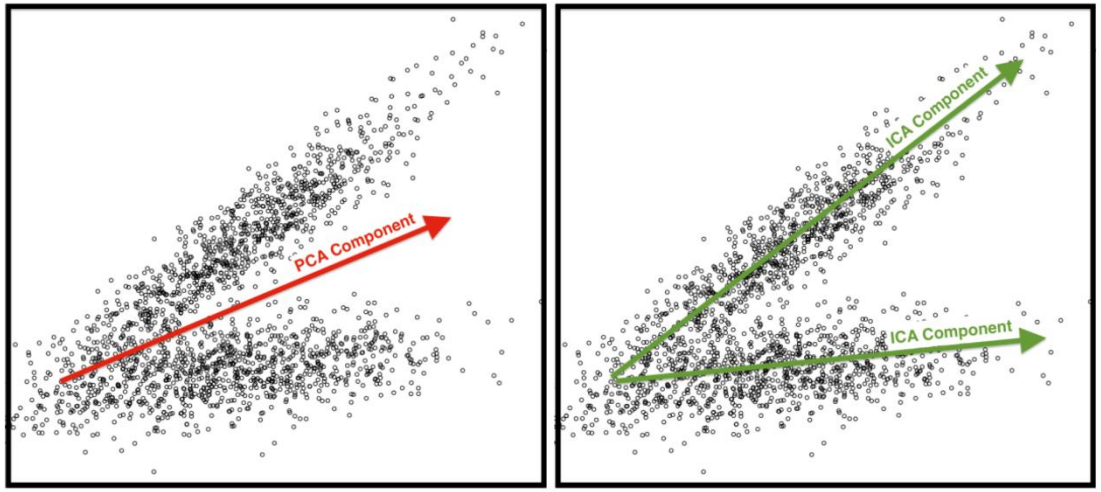
LIMITATIONS

PCA is dependent on scaling (not unique).

With no prior domain expertise, interpreting PCs may be difficult.

Assumptions are **not always met**

- important structures and spread are correlated (i.e. counting pancakes)
- PCs are orthogonal (what about ICA?)
- change of basis framework (i.e. Ferris wheel tracking)



Sensitive to outliers.

HIGH DIMENSIONALITY AND 'BIG' DATA

Datasets can be "big" in a variety of ways:

- too large for the **hardware** to handle (cannot be stored or accessed properly due to # of observations, # of features, or the overall size)
- dimensions can go against specific **modeling assumptions** (# of features ≫ # observations)

Examples:

- Multiple sensors recording 100+ observations per second in a large geographical area over a long time period = **very big dataset**.
- In a corpus' *Term Document Matrix* (cols = terms, rows = documents), the number of terms is usually substantially higher than the number of documents, leading to **excessively sparse data**.

FEATURE SELECTION METHODS

Filter methods inspect each variable individually and score them according to some **importance metric**.

The less relevant features (i.e. importance score below some set threshold) are then removed.

Wrapper methods seek feature subsets for which the evaluation criterion used by the eventual analytical method is "optimized".

The process is **iterative**, and typically computationally intensive: candidate subsets are used in the analysis until one produces an acceptable evaluation metric for the analysis.

FEATURE SELECTION METHODS

Unsupervised methods determine the importance of a feature based only on its values.

Supervised methods evaluate each feature's importance by studying the relationship with a **target feature** (correlation, etc.)

Wrapper methods are usually supervised.

Unsupervised filter methods: removing constant variables, ID-like variables (different on all observations), features with low variability, etc.

SUPERVISED FILTER METHODS

Correlation between a feature X and a target variable Y:

$$\rho_{X,Y} = \frac{\sum_{i=1}^{N}(y_i - \bar{y})\,(x_i - \bar{x})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2} \cdot \sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}}$$

Features which are highly correlated with the target variable are retained, but this approach is limited if the relationship to the target variable is **non-linear**.

OTHER SUPERVISED METRICS

Classification Tasks

- Gain Ratio
- Inf Gain
- Gini
- MDL, etc.

Regression Tasks

- MSE of Mean
- MAE of Mean
- Relief (evaluates features simultaneously), etc.

COMMON TRANSFORMATIONS

In the regression context, transformations are **monotonic:**

- logarithmic
- square root, inverse, power: W^k
- exponential
- Box-Cox, etc.

Transformations on X may achieve linearity, but usually at some price (correlations are not preserved, for instance). Transformations on Y can help with non-normality and unequal variance of error terms.

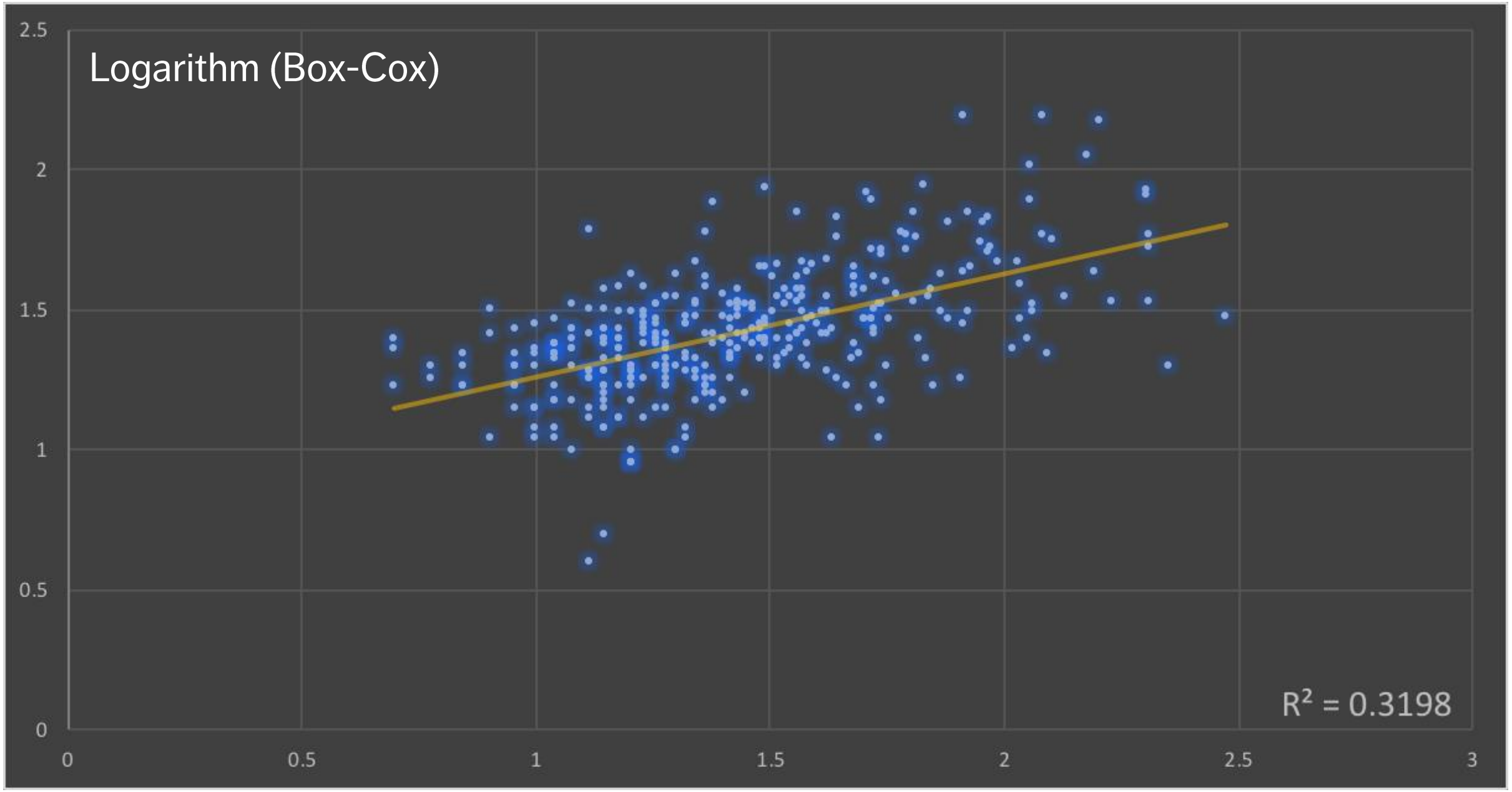
BOX-COX TRANSFORMATION

Assume the usual model $Y_j = \sum_i \beta_i X_{j,i} + \varepsilon_j$ with either

- skewed residuals
- not-constant variance
- non-linear trend

The **Box-Cox transformation** $Y_j \mapsto Y_j'(\lambda)$ suggests a choice: select λ which maximizes the corresponding log-likelihood

$$Y_j'(\lambda) = \begin{cases} \mathrm{gm}(\boldsymbol{Y}) \times \ln(Y_j), & \lambda = 0 \\ \lambda^{-1}\mathrm{gm}(\boldsymbol{Y})^{1-\lambda} \times ({Y_j}^{\lambda} - 1), & \lambda \neq 0 \end{cases}$$


Logarithm (Box-Cox)
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
R² = 0.3198

BOX-COX TRANSFORMATION

The procedure provides a **guide** to select a transformation.

Theoretical rationales may exist for a particular choice of λ.

Residual analysis is still required to ensure that the choice was appropriate.

The resulting parameters have the least squares property only with respect to the transformed data points.