Visualization

LAST LECTURE **chart literacy**

1. anatomy of a plot
2. scale theory
3. scale perception
4. making comparisons
5. atomic plots

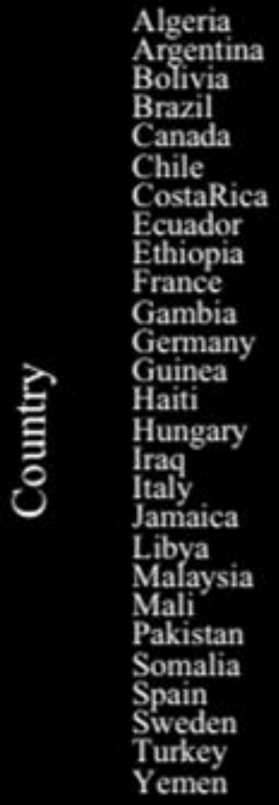
the "grammar" of plots

TODAY **embedding**

6. unsupervised learning
7. dimension reduction / PCA
8. self-supervised learning / tSNE
9. content scales

plots for data exploration

Nominal: no comparison is meaningful

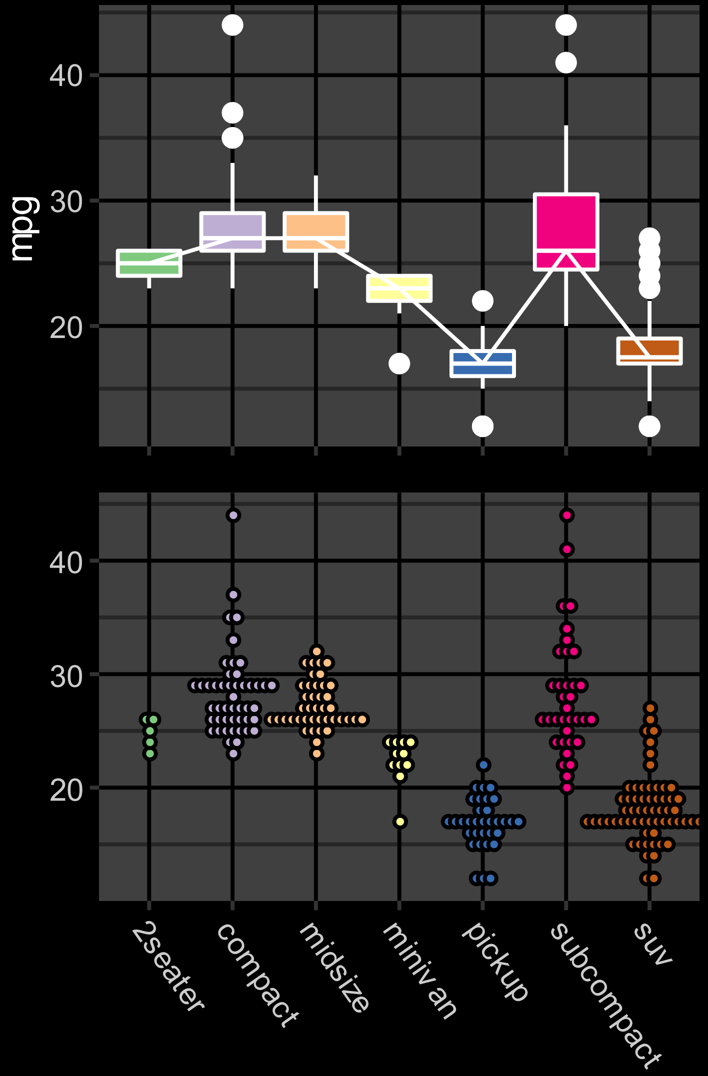


This is dumb.

How can I say “no comparison is meaningful” — and at the same time render onto a y scale?

6. Supervised versus unsupervised plotting

Some plots are associated with supervised learning, some with unsupervised.



Supervised learning

- What is the conditional distribution of $Y|X$?
- How do I build a predictor / classifier?

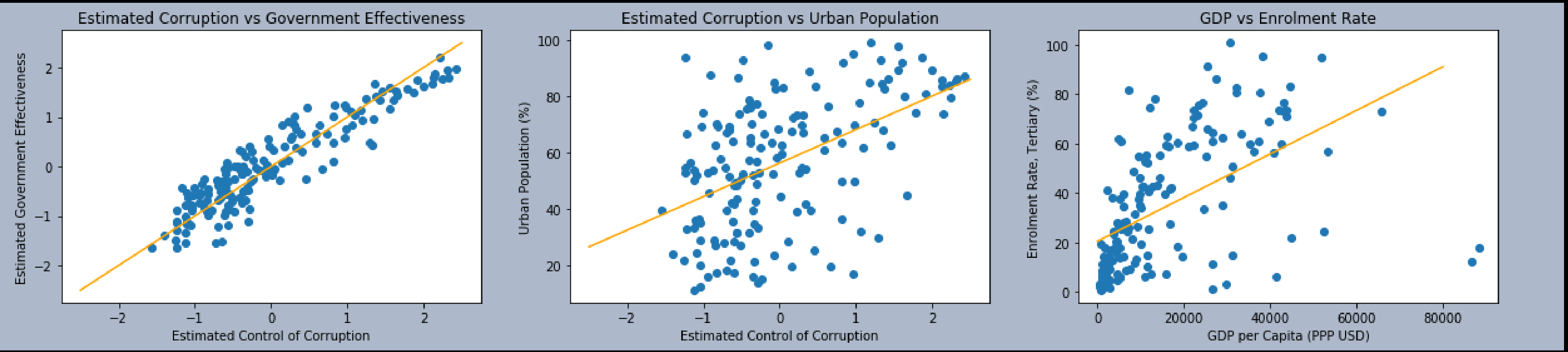
Generative (unsupervised) learning

- What is the joint distribution of (X, Y)?
- How do I generate new random datapoints?

Who decides which variable is to be the predictor and which is to be the response?

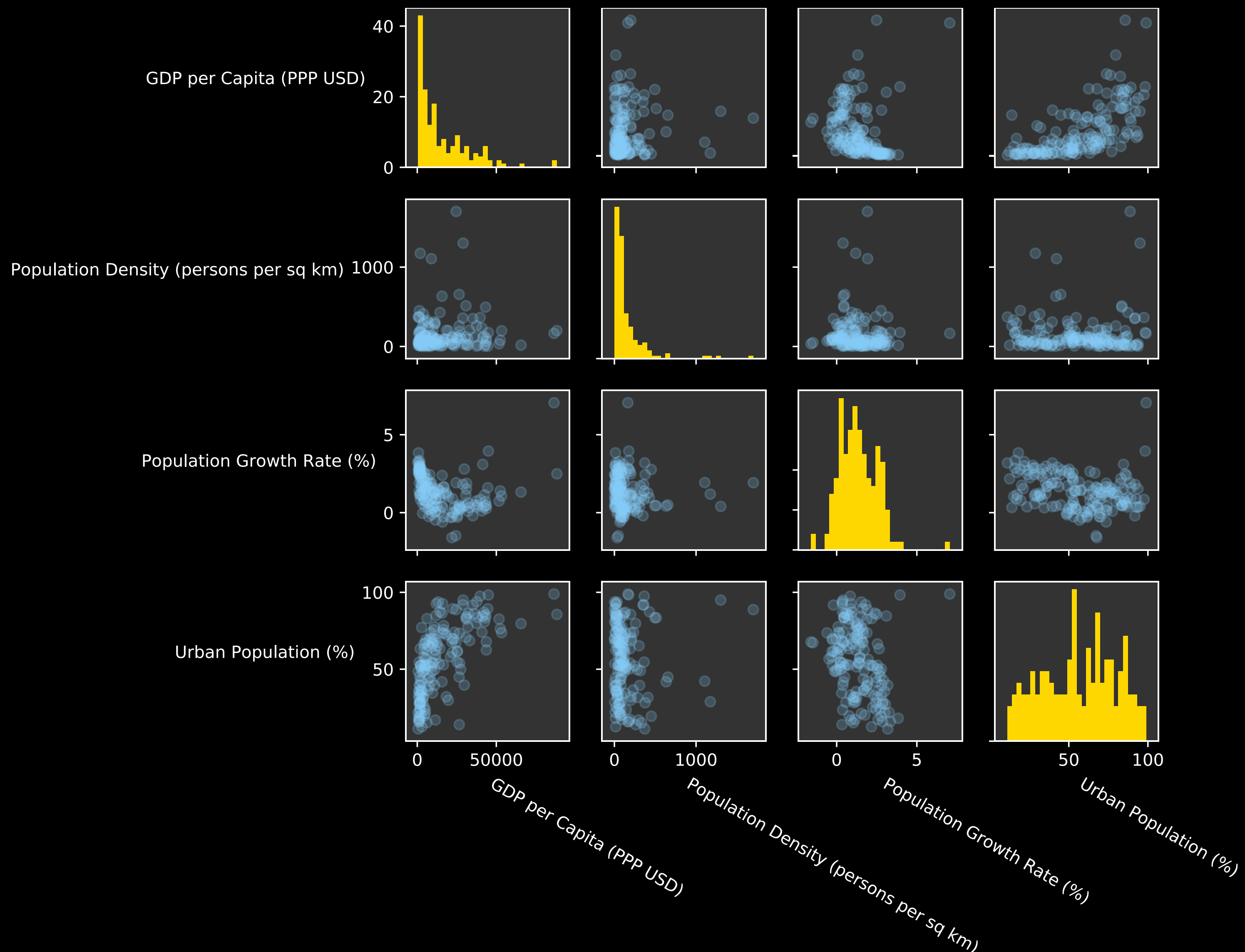
Data set: World Bank country / demographic data

Country Name	GDP per Capita (PPP USD)	Population Density (persons per sq km)	Population Growth Rate (%)	Urban Population (%)	Life Expectancy at Birth (avg years)
Afghanistan	1560.67	44.62	2.44	23.86	60.07
Albania	9403.43	115.11	0.26	54.45	77.16
Algeria	8515.35	15.86	1.89	73.71	70.75

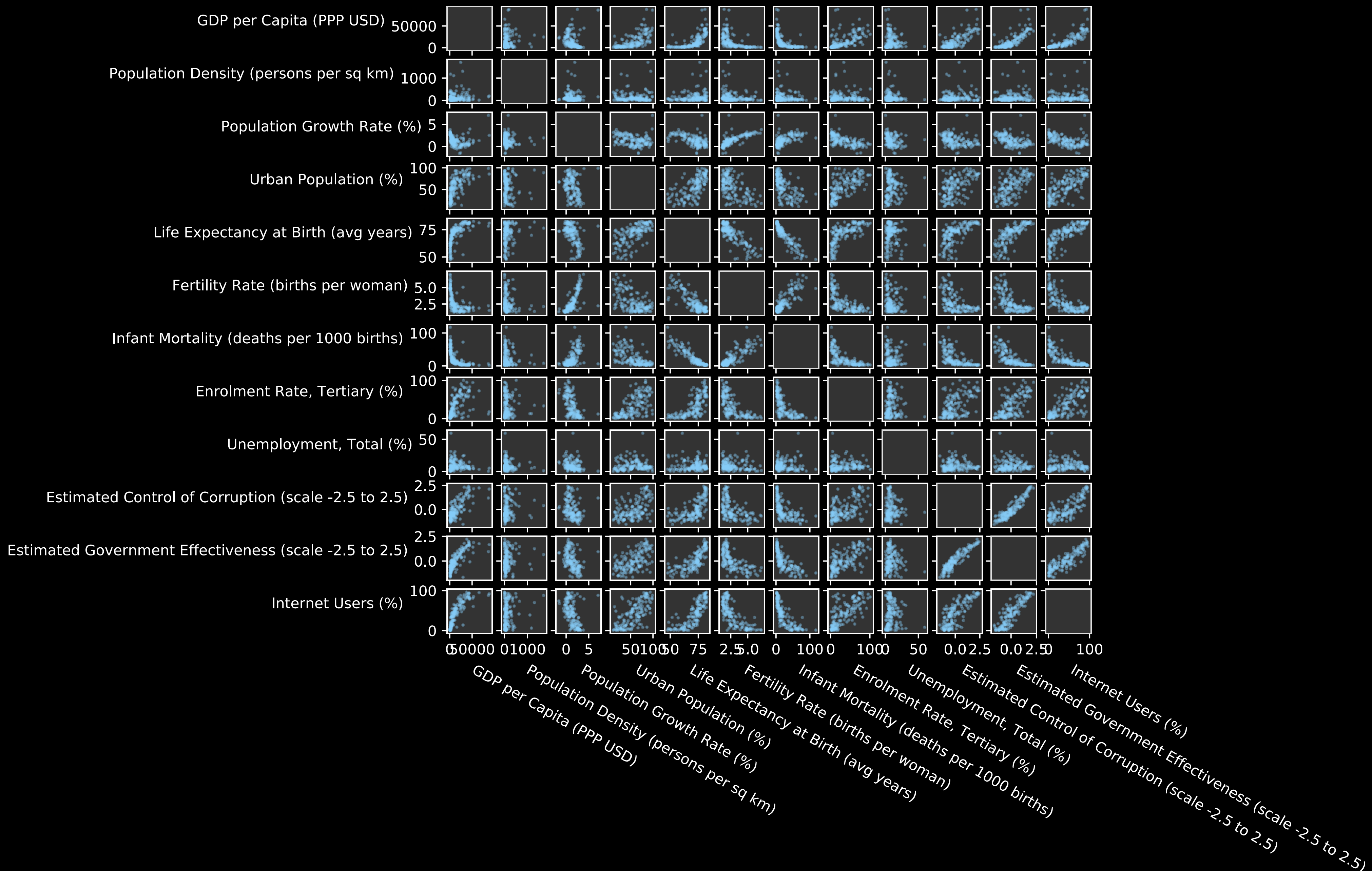


Some of the linear models from Lecture 2

A splom (scatter plot matrix) shows all pairs of variables. If you don't have a definite prediction task in mind, this is a good starting point.



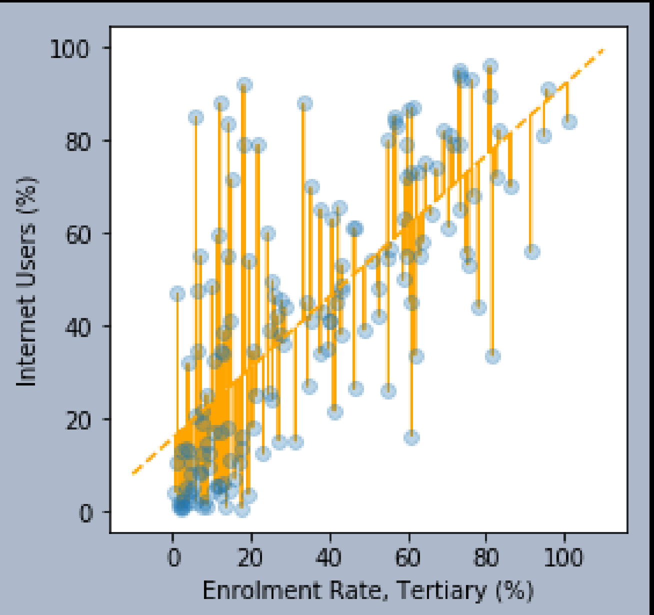
A splom is unwieldy for 12 variables. And what if we had 1200 or 12000?

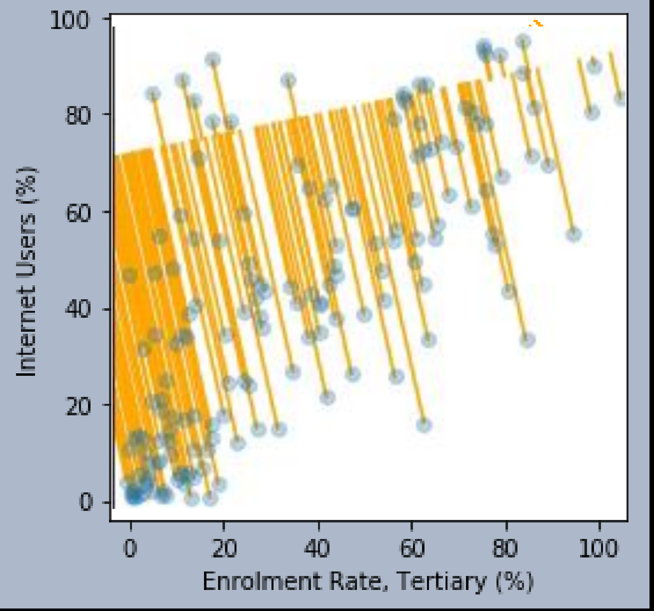


7. Dimension reduction with PCA

PCA is a tool that takes in high-dimensional data and compresses it lossily into fewer dimensions.

7. Dimension reduction with PCA





In linear regression, we model the data by

$$y_i = \alpha\, x_i + \beta + \varepsilon_i$$

and choose the parameters α and β to minimize

$$L(\alpha, \beta) = \frac{1}{2} \sum_{i=1}^{n} \varepsilon_i^2$$

compress 2d (x, y)

into 1d (λ)

In PCA, we model the data by

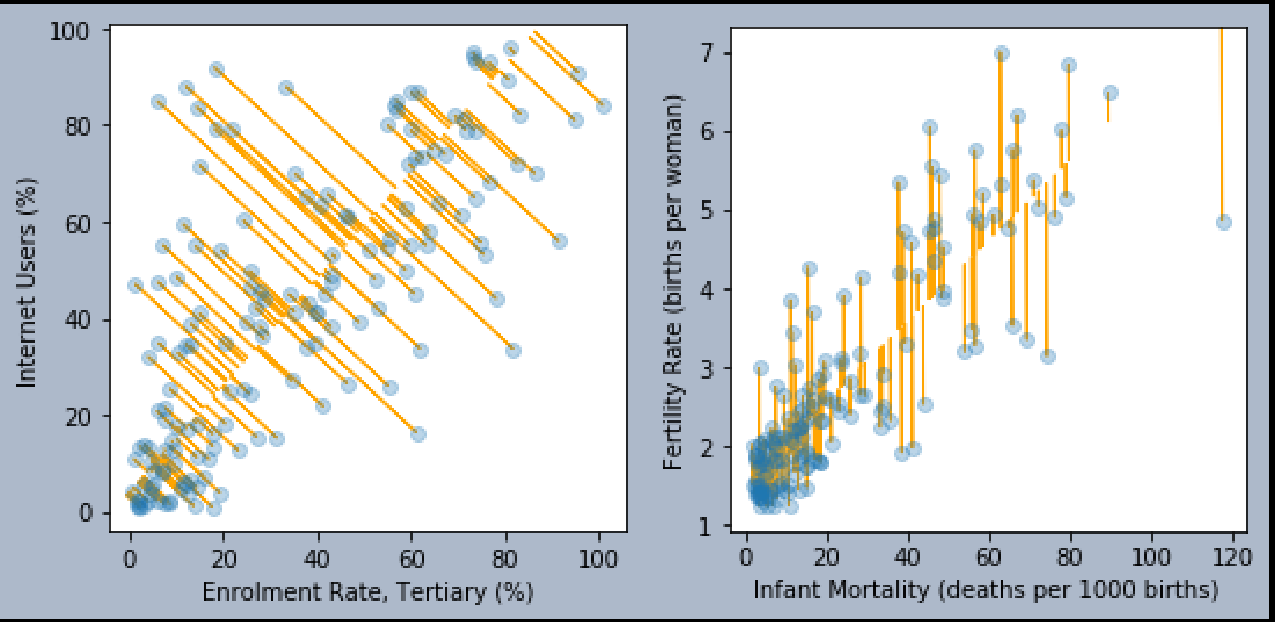
$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix} + \lambda_i \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} + \begin{pmatrix} \varepsilon_{x,i} \\ \varepsilon_{y,i} \end{pmatrix}$$

and choose the parameters μ and δ to minimize

$$L(\mu, \delta) = \frac{1}{2} \sum_{i=1}^{n} \left\| \begin{pmatrix} \varepsilon_{x,i} \\ \varepsilon_{y,i} \end{pmatrix} \right\|^2$$

7. Dimension reduction with PCA

Why do these two PCA plots, on two different data features, come out so different?



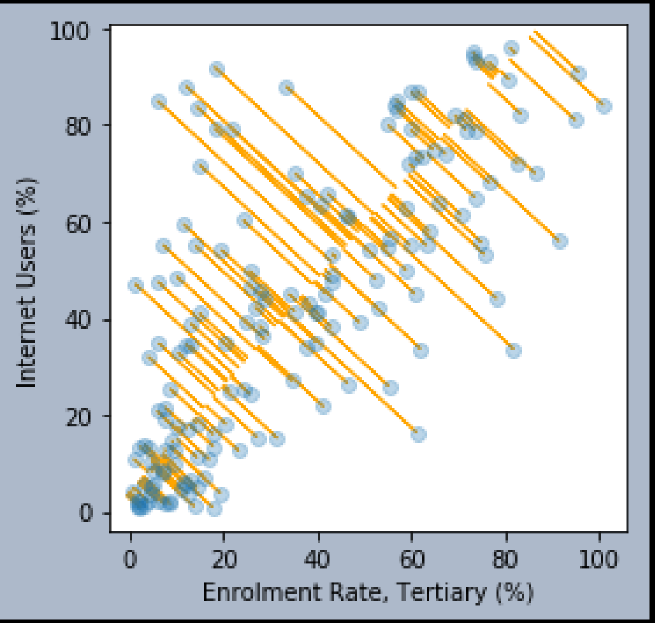
Because we're minimizing

$$\frac{1}{2}\sum_{i=1}^{n}(\varepsilon_{x,i} + \varepsilon_{y,i})^2$$

and the x and y units are so different.

Solution: scale the x and y columns so they have the same standard deviation, before doing the fit.

7. Dimension reduction with PCA



- PCA is a tool that takes in high-dimensional data and compresses it lossily into fewer dimensions.
- This avoids an arbitrary choice of predictor vs response variables.
- We've shown it for 2d → 1d compression but it also works for arbitrary dimensions $N \rightarrow K$.

How to run PCA

We model N-dimensional data by picking a K-dimensional subspace, and representing each point by its projection onto that subspace,

$$\vec{x}_i = \vec{\mu} + \sum_{k=1}^{K} \lambda_{k,i}\, \vec{\delta}_k + \vec{\varepsilon}_i$$

We pick the subspace so as to minimize the mean square error $\sum \|\vec{\varepsilon}_i\|^2$.

Magic linear algebra trick:

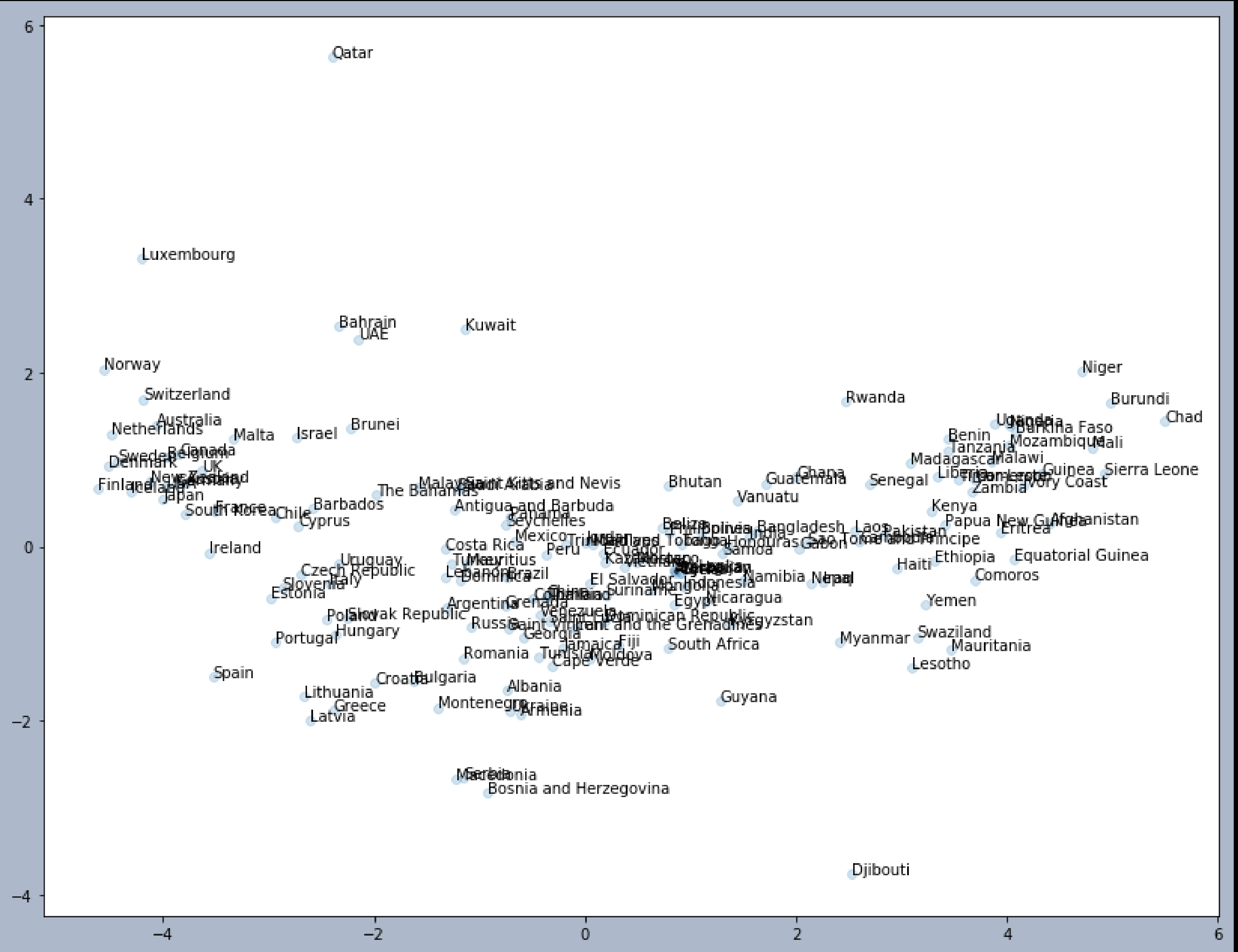
- the optimal subspaces are nested, $S_1 \subset S_2 \subset \cdots \subset S_N$
- so the sensible choice is to let the components $\vec{\delta}_k$ be an ordered basis,

  $S_1 = \mathrm{span}(\vec{\delta}_1)$

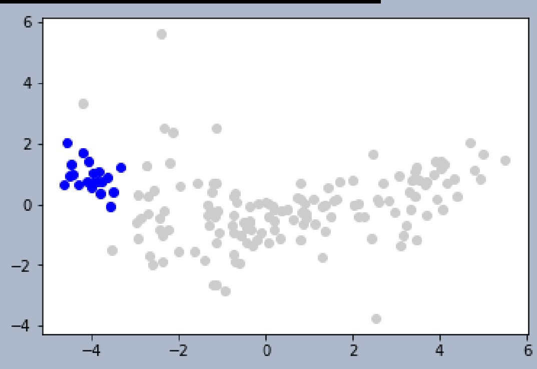
  $S_2 = \mathrm{span}(\vec{\delta}_1, \vec{\delta}_2)$ …

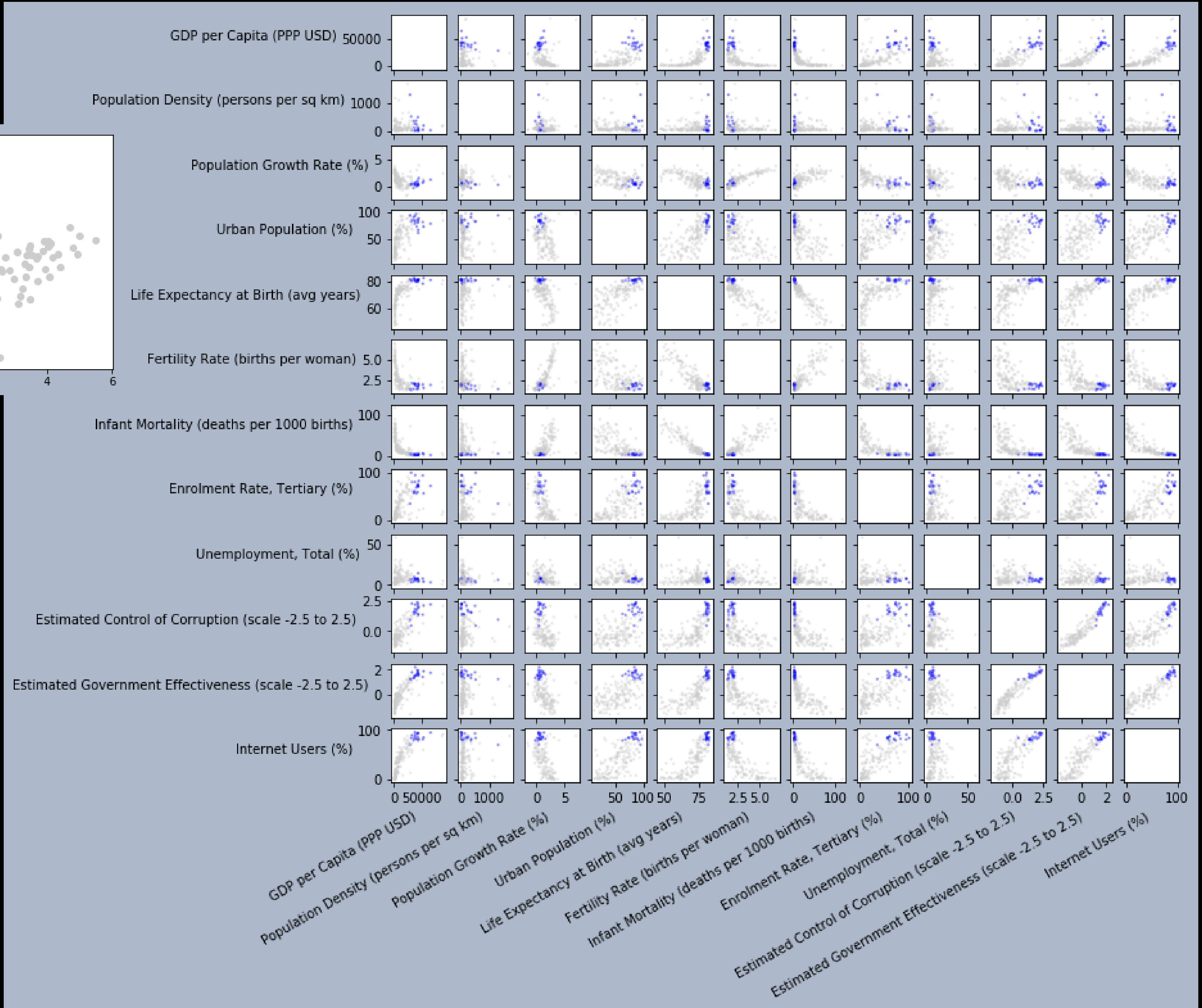
```
X = countries[features].values
pca = sklearn.decomposition.PCA()
pca_result = pca.fit_transform(X)

μ = pca.mean_
pred = μ + np.zeros_like(pca_result)
for k in range(L):       # L = number of PCA components to use
    λk = pca_result[:,k]
    δk = pca.components_[k]
    pred = pred + λk.reshape((-1,1)) * δk.reshape((1,-1))
```

Run PCA, then plot the first two components $(\lambda_{1,i}, \lambda_{2,i})$ to get a nice visualization.



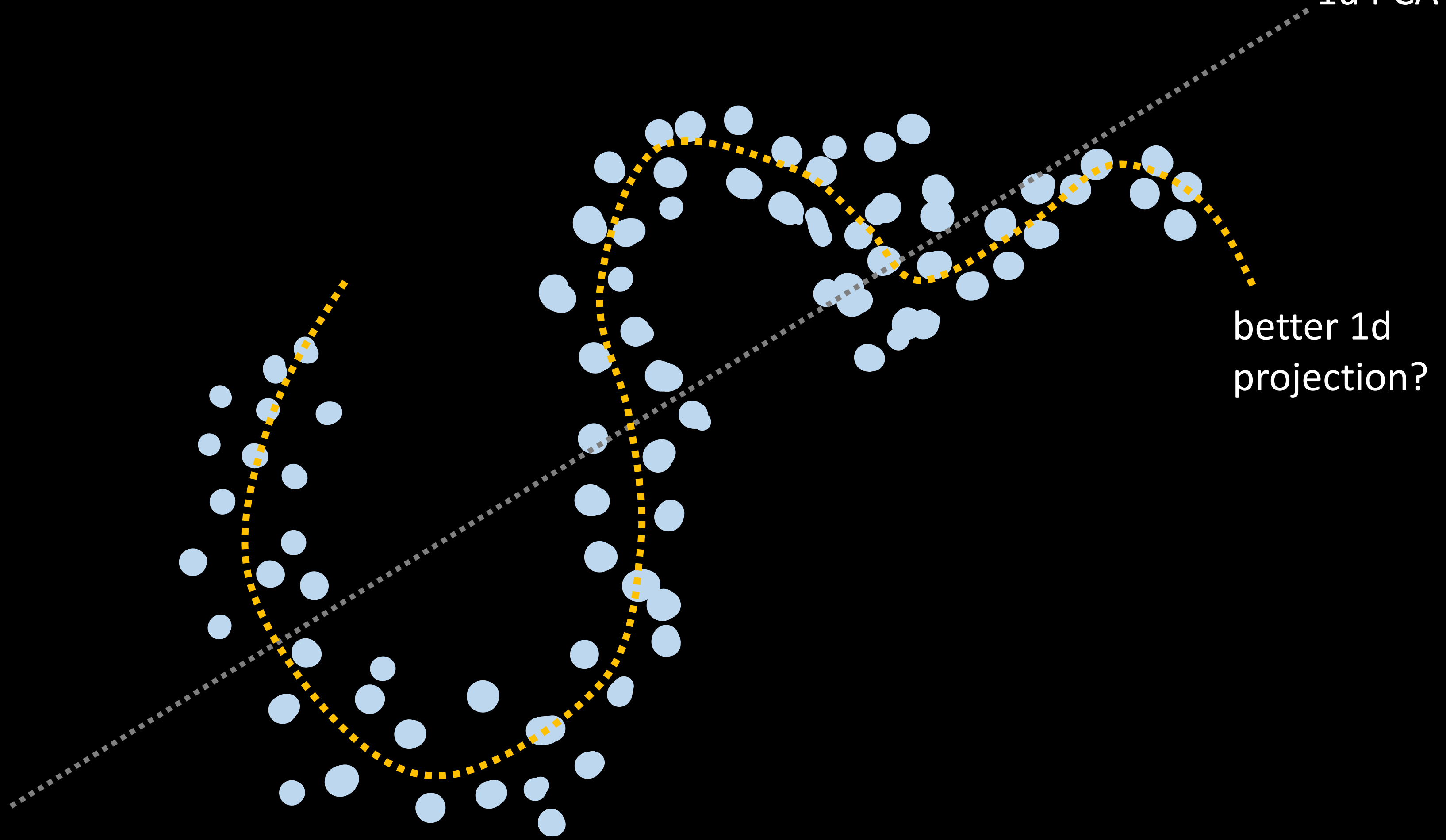
Nearby points in the PCA diagram should represent records with similar features.





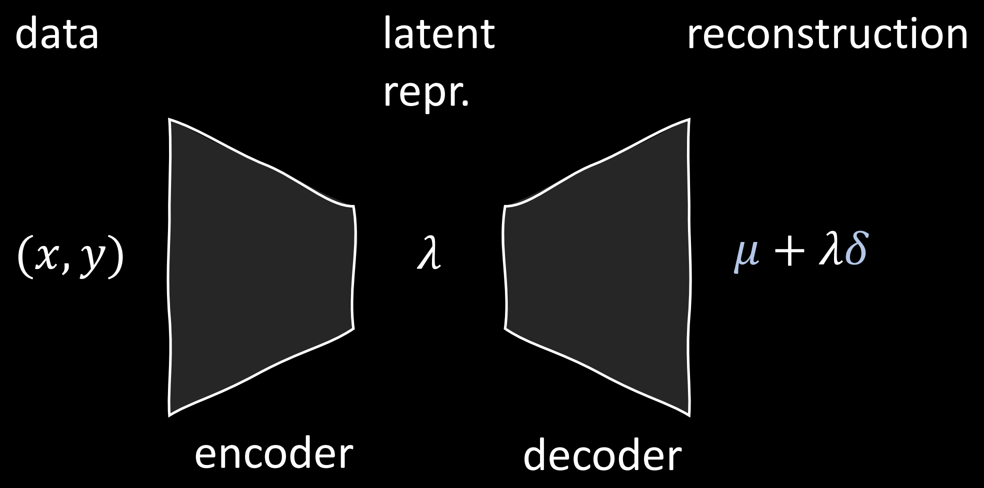
8. Self-supervised learning / embedding

What if the data isn't linear?



To understand this, we'll first look at a different interpretation of PCA.

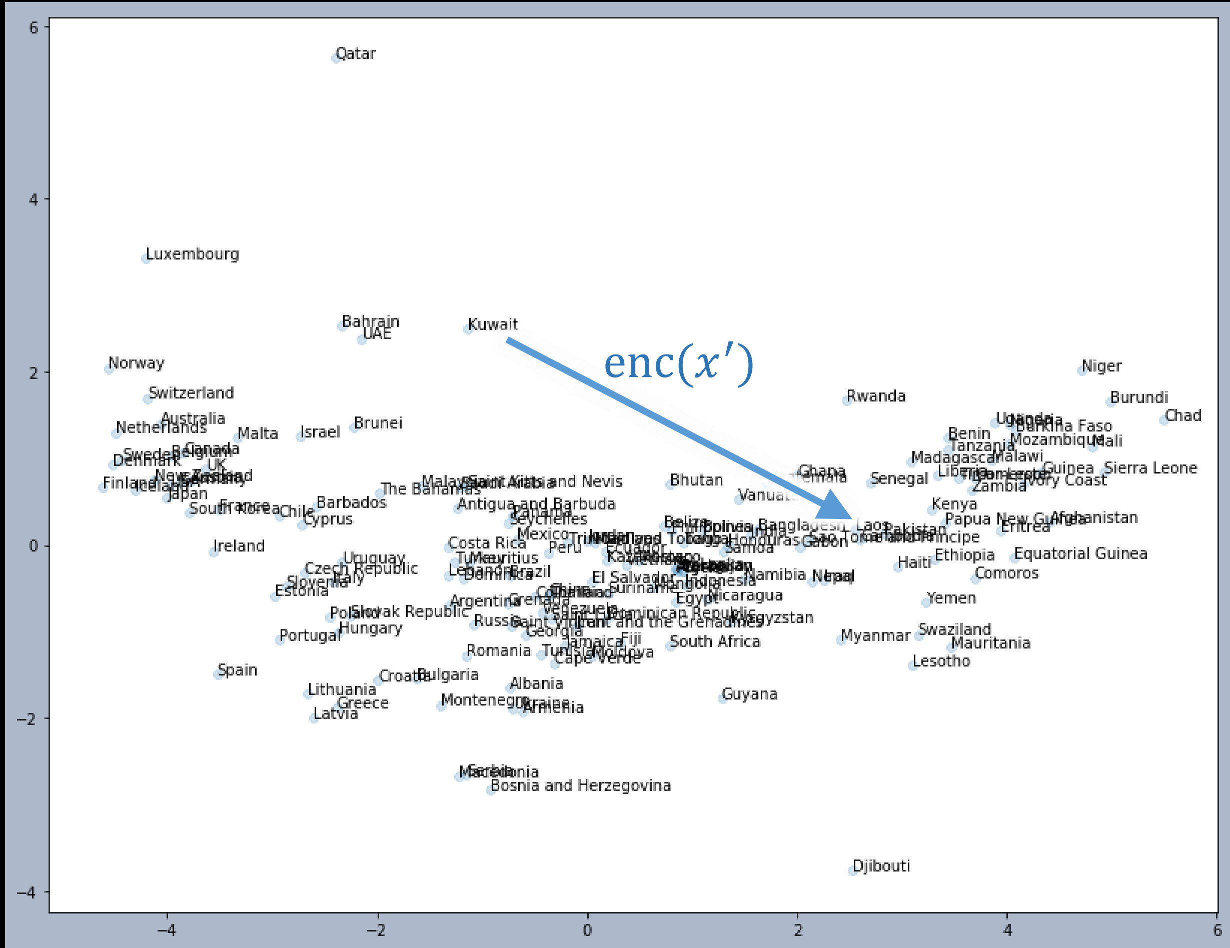
Self-supervised learning



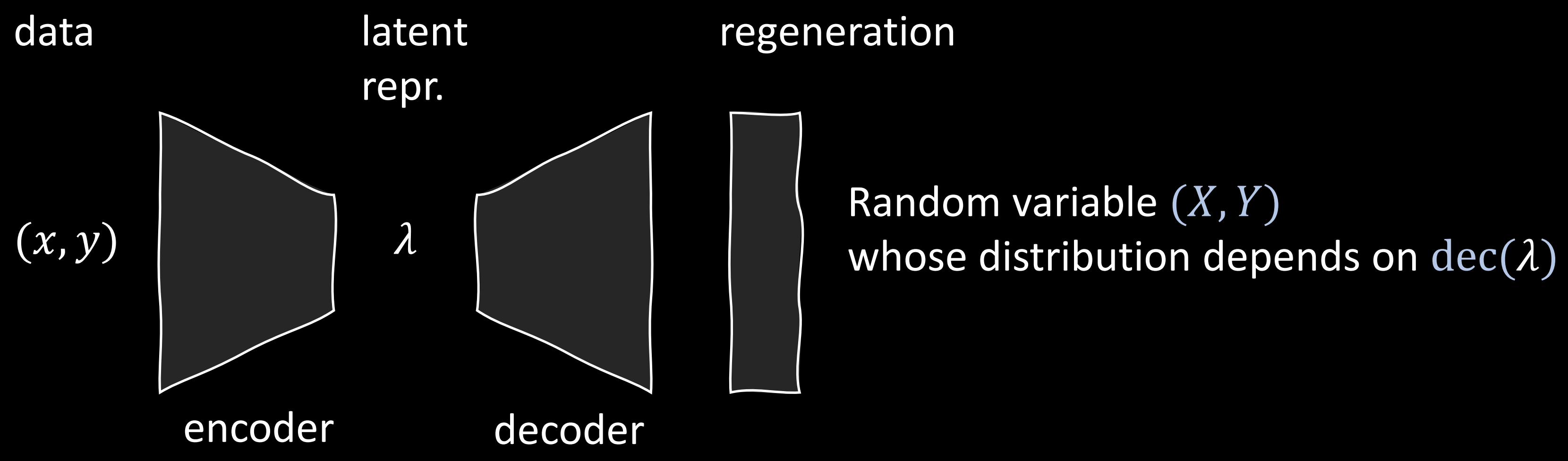
- In the training phase, we train $\mathrm{enc}(\cdot|\mu, \delta)$ and $\mathrm{dec}(\cdot \mid \mu, \delta)$
- After training, we have enc and dec which can be applied to new data

Self-supervised learning is “learning a coordinate system from the data”

- Don’t think of PCA as “an algorithm that lossily compresses my dataset”
- Think of it as “learning a coordinate system, based on my training data”
- We can ask, for example:
  if a datapoint x' has one of its features varied, how does $\mathrm{enc}(x')$ vary?



Probabilistic self-supervised learning

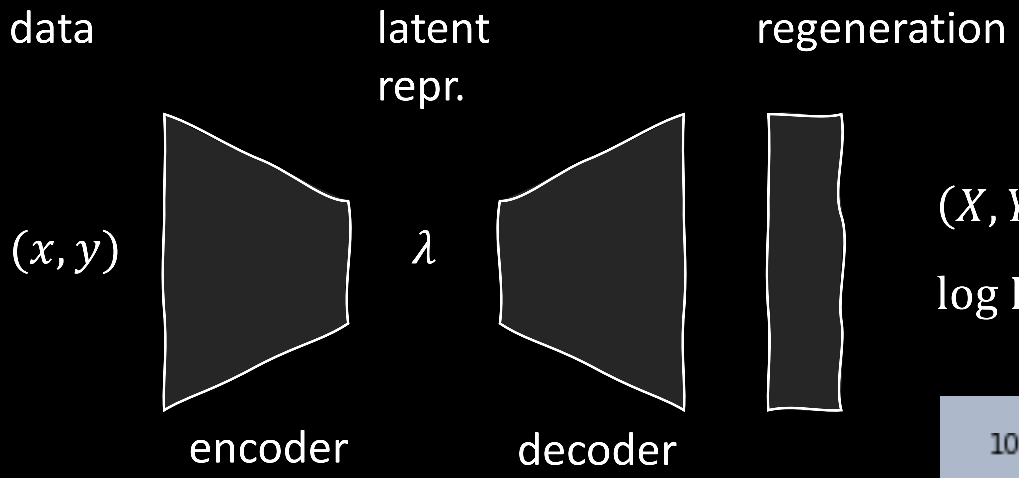


- Training requires an objective function

- In standard probabilistic supervised machine learning,
  - let x be the predictor and y the label
  - we decide on a probability distribution $Y|x, \theta$
  - treating each label y_i as a sample from $Y|x_i, \theta$
    we optimize the parameters θ to maximize the log likelihood

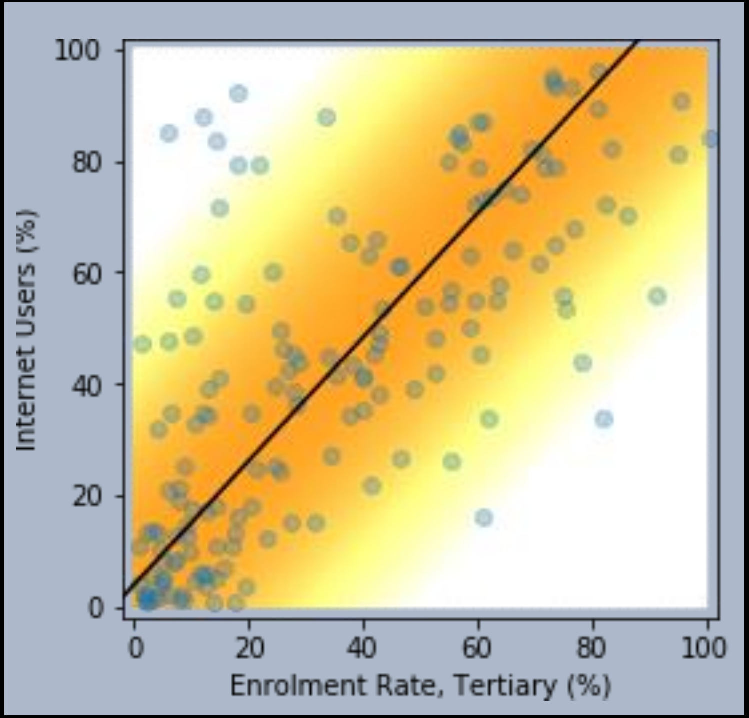
- Self-supervised learning is like this, except that we treat (X, Y) as random,
  and let it be predicted by (x, y), via the encoded form $\lambda = \text{enc}(x, y)$

PCA, seen as probabilistic self-supervised learning

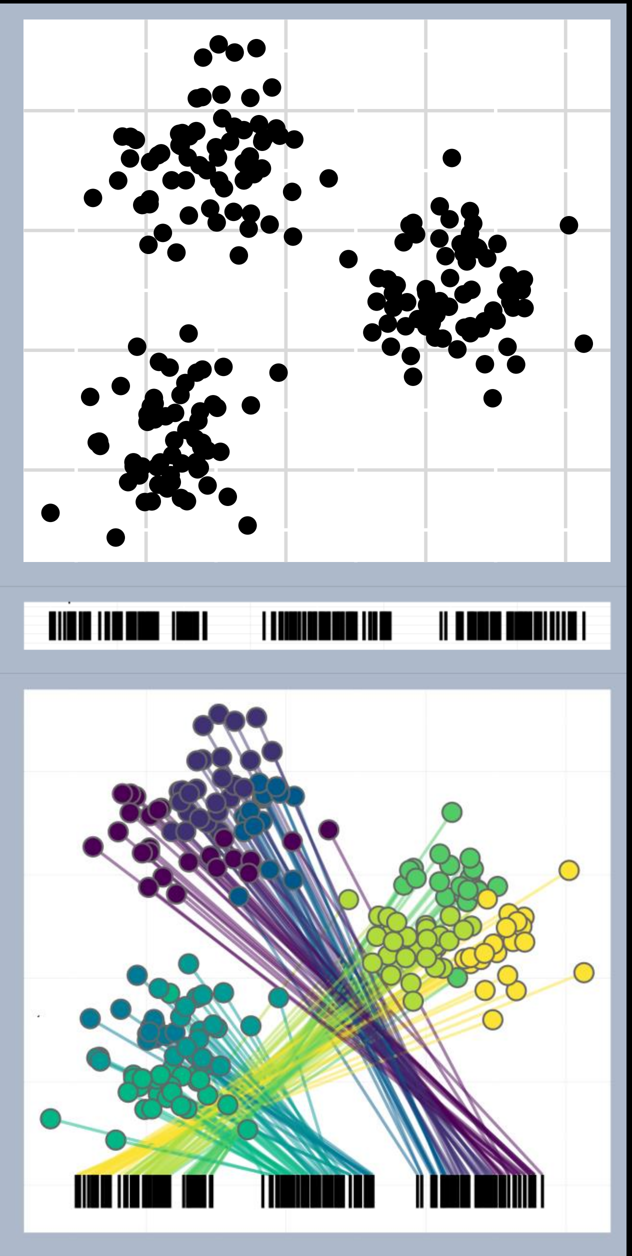


$$(X, Y) \sim N(\mu + \lambda\delta, \sigma^2 I)$$

$$\log \Pr_{X,Y}(x, y) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|(x, y) - (\mu + \lambda\delta)\|^2$$



tSNE is another probabilistic self-supervised learning method



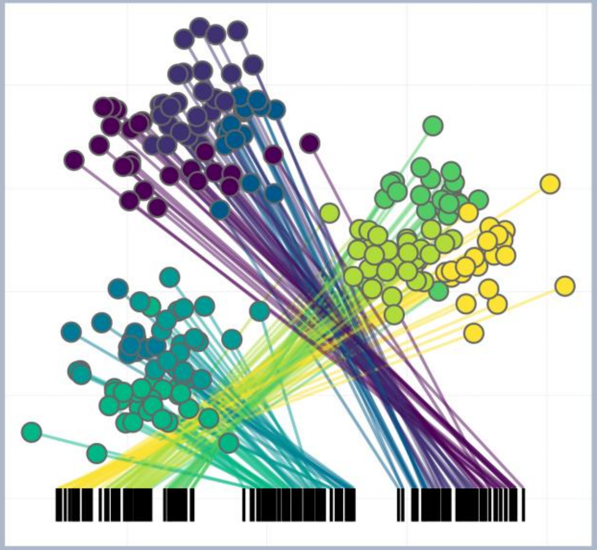
the data
(two dimensions)

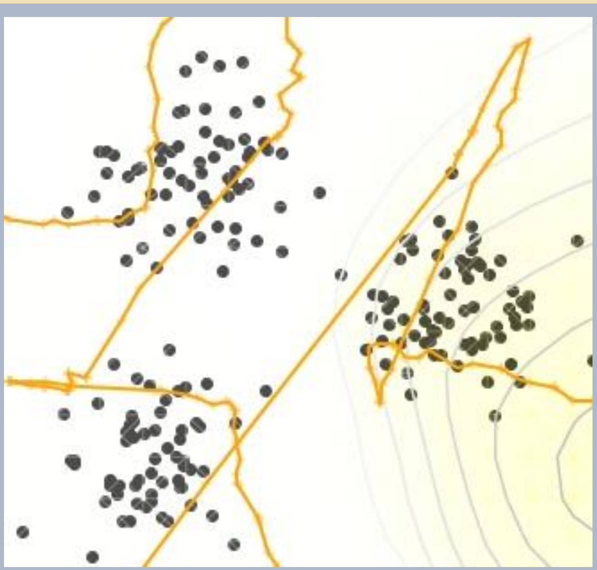
latent space
(using one dimension)

encoding

tSNE embedding

PCA embedding

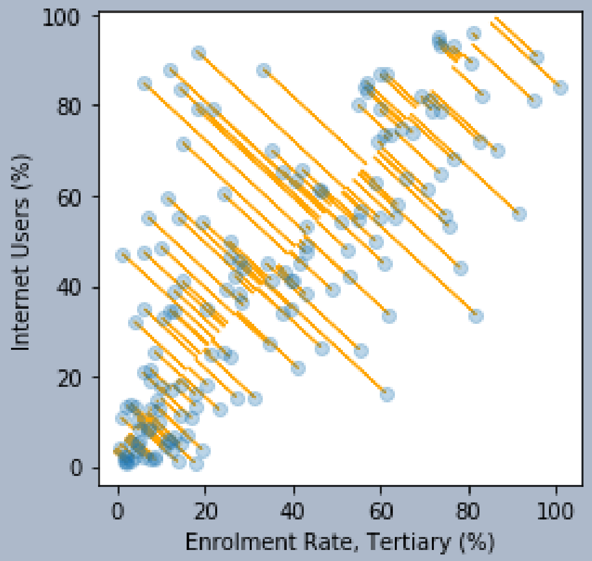


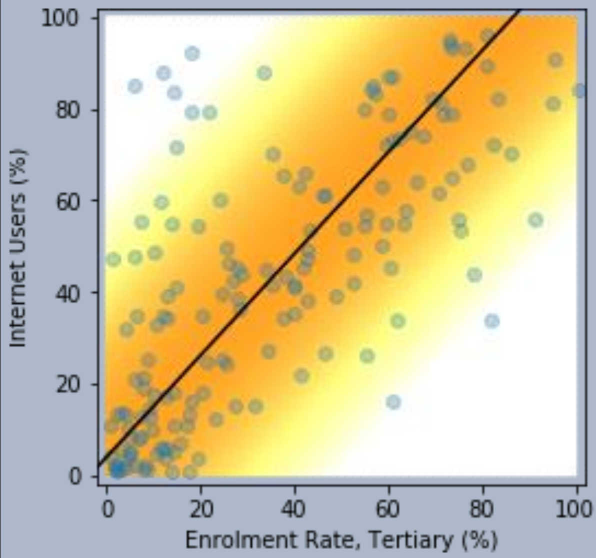


encoding

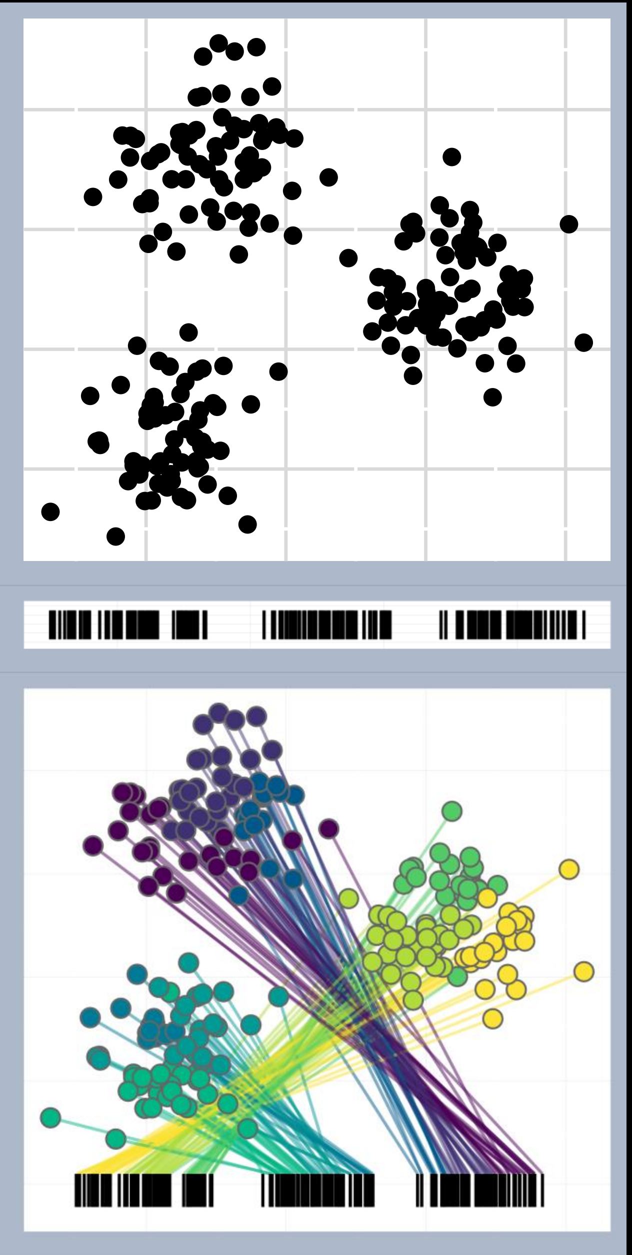
latent space
(using one dimension)

decoding





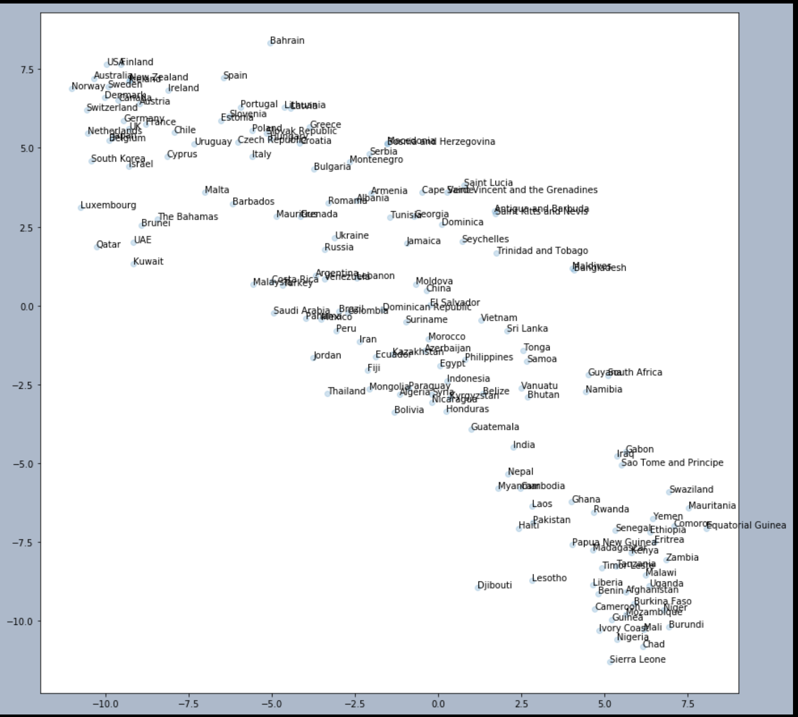
The training mechanism for tSNE



1. Each datapoint $\vec{x}_i$ is to be encoded to a k-dimensional latent variable λ_i
2. Each $\vec{x}_i$ has a certain set of nearest-P neighbours in the data space. Likewise each λ_i has a certain set of nearest-P neighbours in the latent space.

   We'd like these two sets of neighbours to be as similar as possible, across all datapoints.
   (The exact method isn't quite this, but it's close.)
3. We find the optimal encoding, using gradient descent.
4. The hyperparameter P is the *perplexity.* It affects the fit.

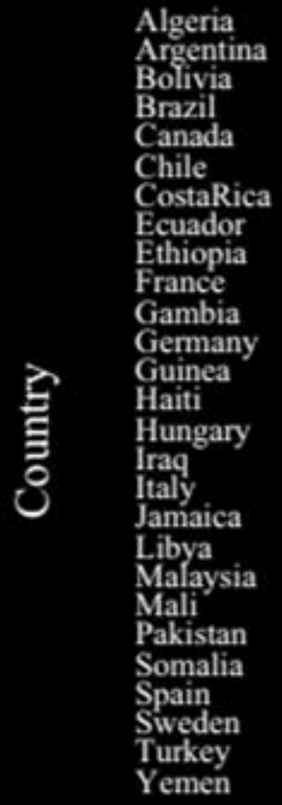
How to run tSNE



```python
X = countries[features].values
# scale the columns, so they have the same variance
for k in range(len(features)):
    X[:,k] = X[:,k] / np.std(X[:,k])

# you have to declare up front how many
# dimensions you want to reduce to
tsne = sklearn.manifold.TSNE(n_components=2)
tsne_results = tsne.fit_transform(X)

p1,p2 = tsne_results[:,0], tsne_results[:,1]
plt.scatter(p1, p2, alpha=.2)
```

9. Content scales

Nominal: no comparison is meaningful



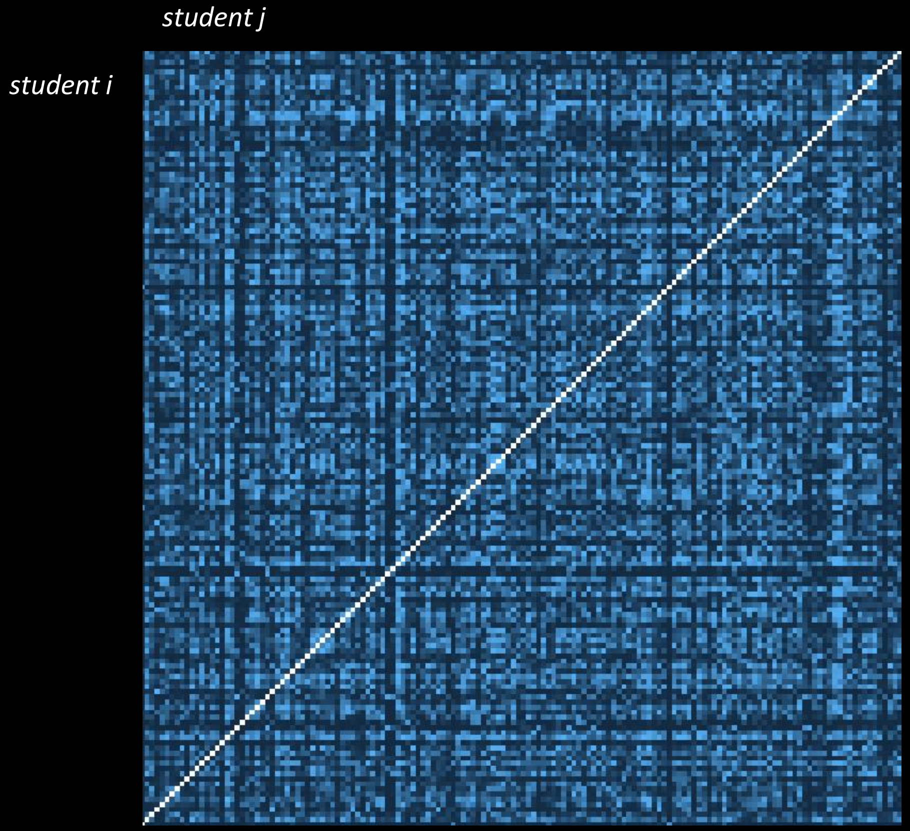
This is dumb.

How can I say “no comparison is meaningful” — and at the same time render onto a y scale?

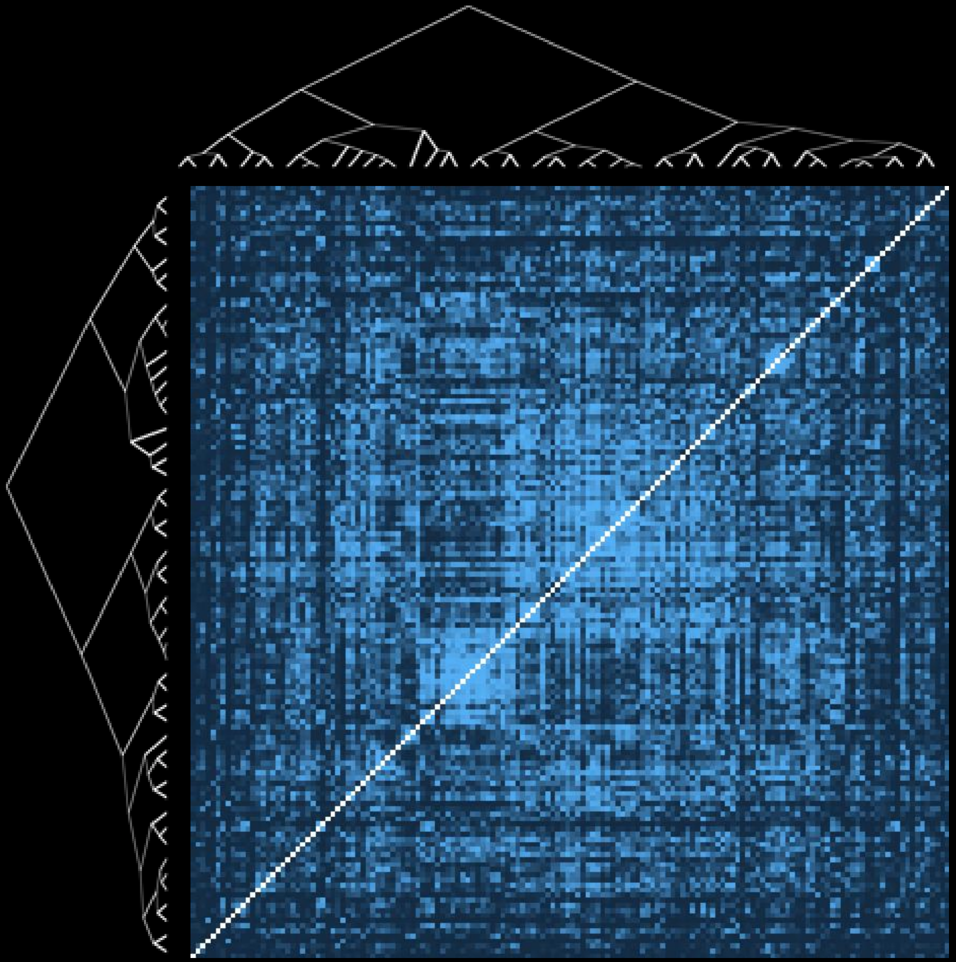
Actually, this scale is an embedding of countries into a 1d latent space, based on proximity in the data domain.

Content scales

- We've seen how to create a content scale, i.e. an embedding, for high-dimensional numerical data.
- When we have nominal or ordinal data, we should pick an embedding that's a useful reflection of the content we're interested in.



Students ordered alphabetically

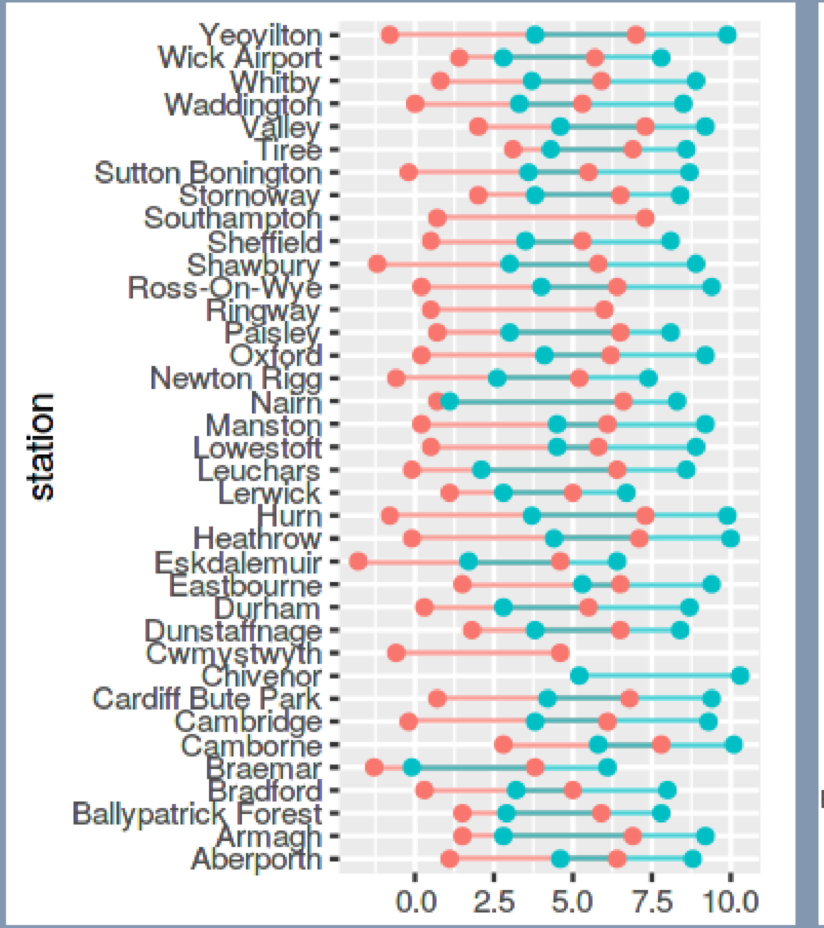


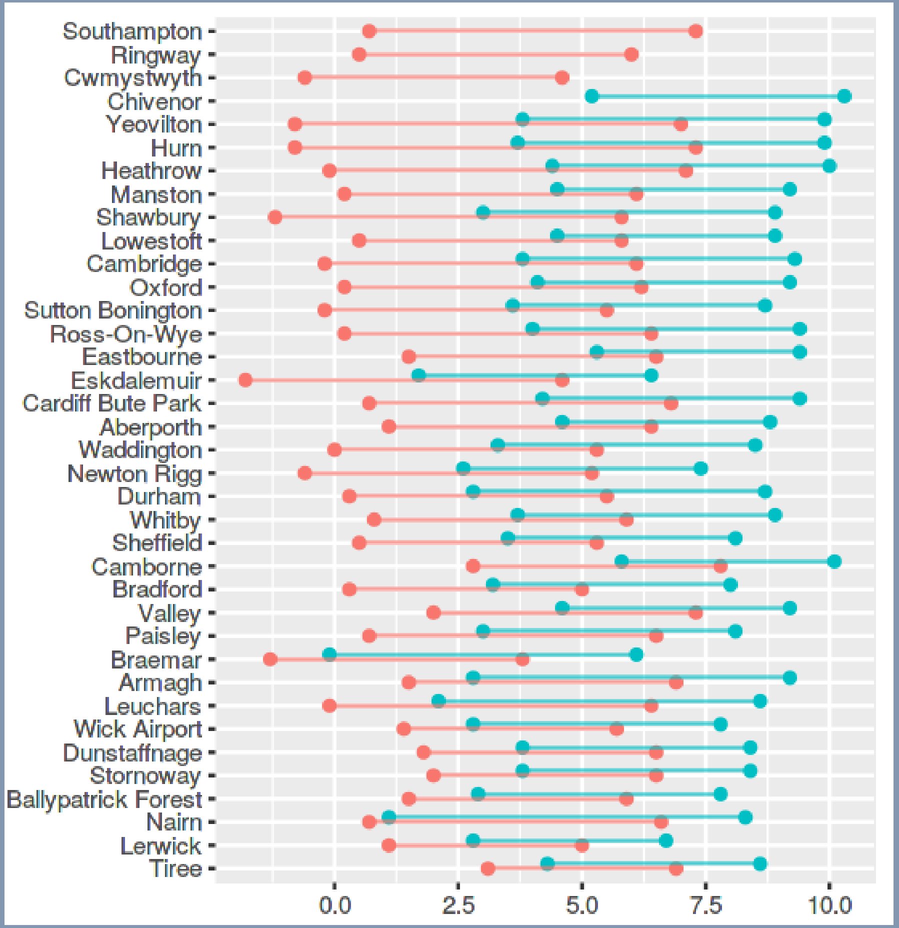
Students ordered by clustering —reveals cliques of cheaters

Similarity between student coursework submissions

- 🟦 high similarity between work by *i* and *j*
- 🟦
- 🟦
- 🟦 low similarity

Content scales





min and max February temperature at stations across the UK

● 1981

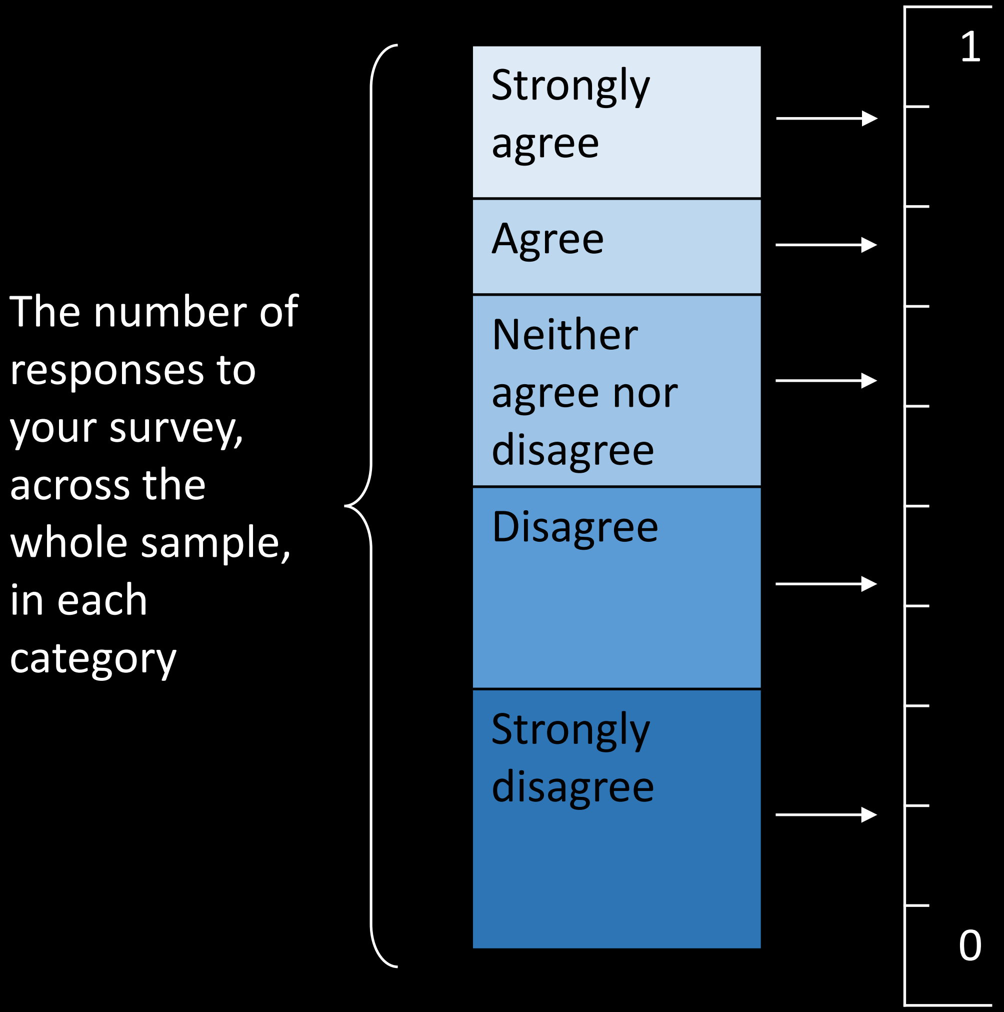
● 2017

A good scale, if the content I'm interested in is the name of the station (for easy lookup)

A good scale, if the content I'm interested in is temperature increase from 1981 to 2017

Embedding an ordinal scale, such as Likert items

Here's a handy trick for getting useful numbers out of ordinal data. It's not deep and rigorous, but it is helpful.

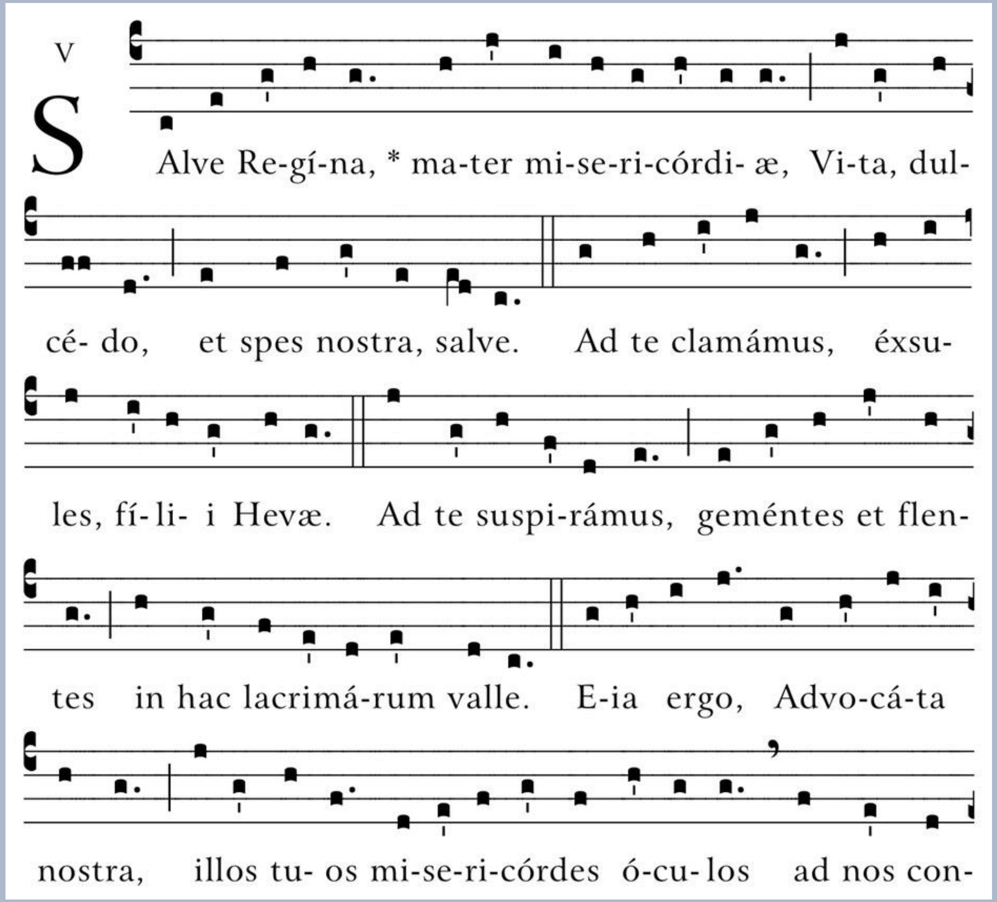
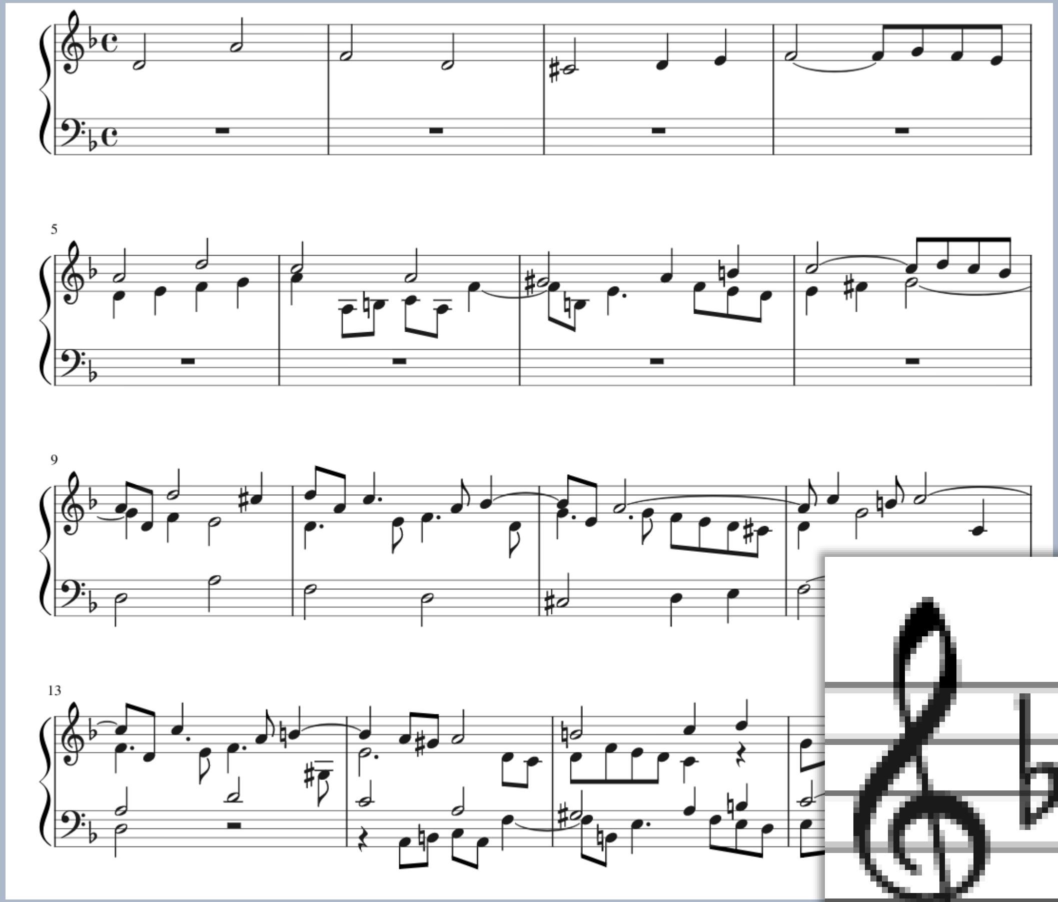

The number of responses to your survey, across the whole sample, in each category
Strongly agree
Agree
Neither agree nor disagree
Disagree
Strongly disagree
1
0

Map each ordinal answer to a number in the range [0,1], based on its quantile within the whole sample.

This trick lets you compute average responses, run linear regression, etc.

Content versus metrical scales

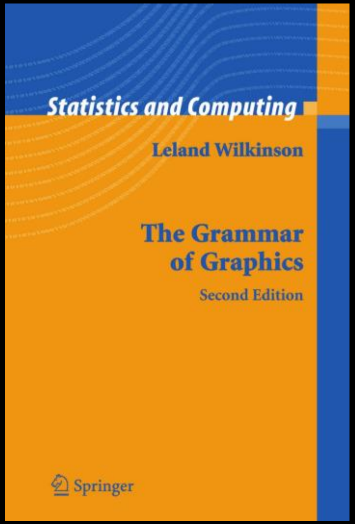
- In a remarkable period from 1250–1350, we started to use metrical scales, systematically. This changed how we see the world. *The measure of reality: quantification and western society, 1250–1600,* by Alfred W. Crosby, 1997.
- Computer power is letting us go back to useful content scales / embeddings.

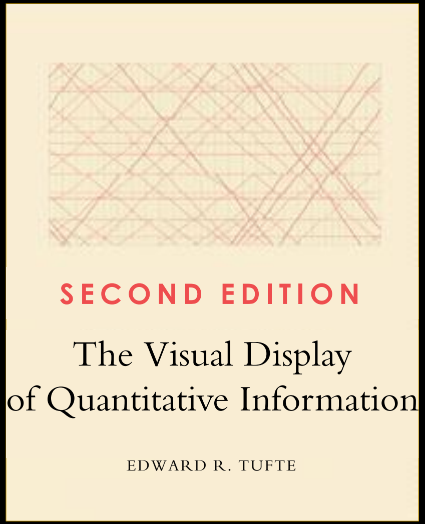

V
SAlve Re-gí-na, * ma-ter mi-se-ri-córdi- æ, Vi-ta, dul-
cé- do, et spes nostra, salve. Ad te clamámus, éxsu-
les, fí-li- i Hevæ. Ad te suspi-rámus, geméntes et flen-
tes in hac lacrimá-rum valle. E-ia ergo, Advo-cá-ta
nostra, illos tu- os mi-se-ri-córdes ó-cu-los ad nos con-


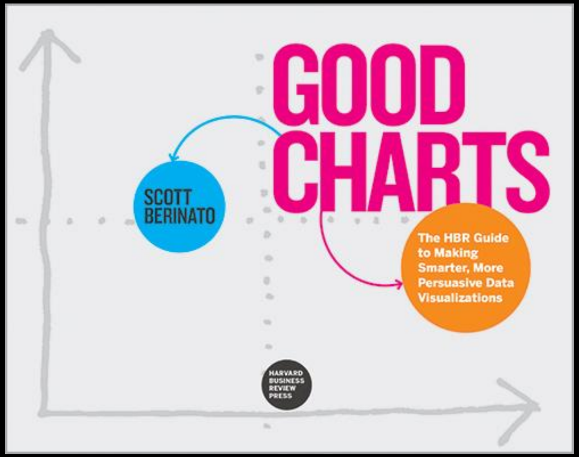
We have studied the grammar of graphics.

Grammar doesn't tell you how to create great charts. But it does give you tools to think systematically about your charts.
You also need • style • the skill to tell a story • good software libraries.

rhetoric = grammar + style + reason / arrangement


Statistics and Computing
Leland Wilkinson
The Grammar of Graphics
Second Edition
Springer


SECOND EDITION
The Visual Display of Quantitative Information
EDWARD R. TUFTE


GOOD CHARTS
SCOTT BERINATO
The HBR Guide to Making Smarter, More Persuasive Data Visualizations
HARVARD BUSINESS REVIEW PRESS

R + ggplot2

Javascript + D3

Vega Lite

and many many badly conceived libraries ...

Practical 4

- Practical 4 will be released tonight
- It’s mainly about PCA and tSNE
- Practical session 26 November 3–4pm