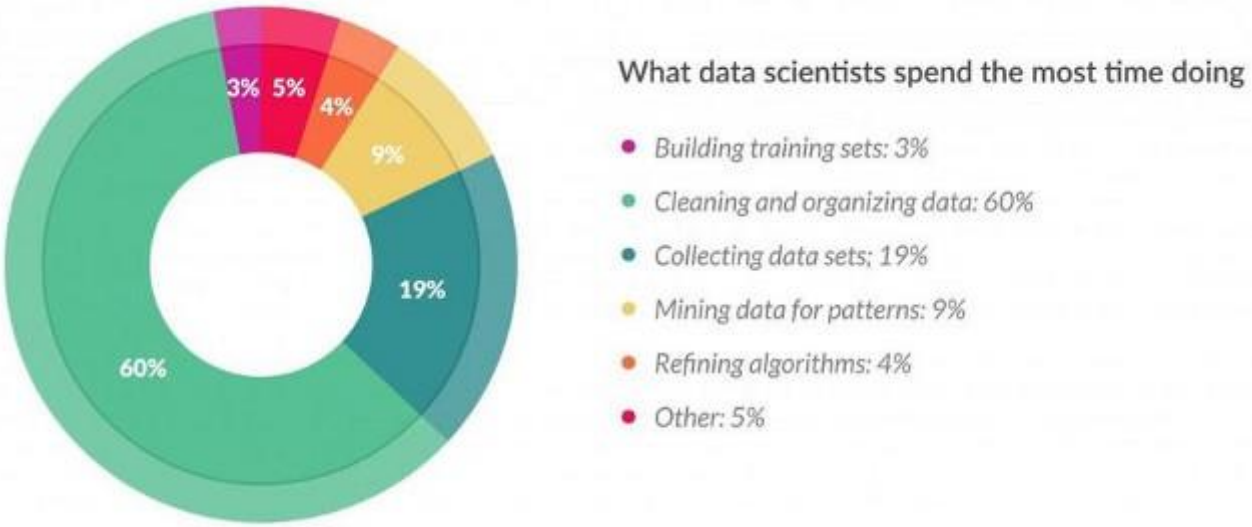
Feature Engineering

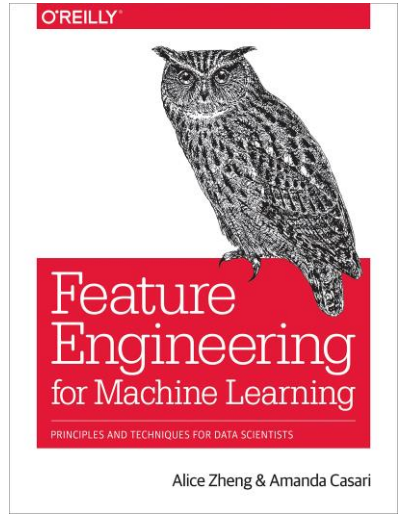
(important & underestimated)

Fundamentals of ML for Econs

Manipulating Data is “the” JOB



- Source: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/



“More data beats clever algorithms,
But better data beats more data”

Peter Norvig

Feature Engineering (is an ART!)

- Garbage in -> garbage out
- Feature engineering can significantly affect the results, it's a "make-it-or-break-it" step

- Domain knowledge often very useful

- **Feature** ~ variable & a transformation of a variable

Feature “Creation”

- **MANUAL** (machines still need us, humans…)
- **AUTOMATIC** – learners can do that (deep learning) but then the “network architecture” is a human task…

- Unless you are creating a new estimation technique, **feature engineering is the MOST CREATIVE PART** of the applied machine learning

Data Wrangling

- Data transformation & Feature engineering are not the same, but related

- **Data transformation** and cleaning, labelling, reshaping, reordering, removing spaces, dots, etc.

- **Feature engineering** – thinking about the meaning of the data, applying valuable domain knowledge…

Some common tasks

- Scaling & normalization
- Encoding of categorical variables
  ("one-hot-encoding" i.e. dummy variables)
- Handling of missing data
- Thinking about outliers
- Transformations (e.g. Box-Cox, logs…)
- Filtering (wavelets, specialized filters, HP/BP filter,…)
- Binning and aggregating

Work Organization -- **Pipelines**

- Use the software that can run algorithms you need **AND** where you feel comfortable wrangling the data…

- It's important to be able to run the pipeline (routine) on the input data, as it'll be done for cross-validation or bootstrapping to avoid "*information leakage*"

- For example:
  - In R, dplyr etc. are tools to help with dataframes etc.
  - In Python, "pandas" will help you to massage the dataframes
  - **Stata** is good too, if you know it…

Work Organization -- **Pipelines**

- Load_data(x,y) -> clean_data(x,y) -> create_features(x,y)

- In cross validation [or bootstraps]

Some Examples

- You can provide “income” and “loan” as separate variables, but also provide “loan-to-income”, wealth to income, …
- You can provide “distance to max loan amount”,…

loan	income	LTI
1230002	195432	**6.29**
340000	54234	**6.27**
430000	31456	**13.67**
3650000	513256	**7.11**

Categorical data

- One-hot encoding is common (in econ, we call it dummy variables)

- As usual, avoid perfect collinearity… ☺

pple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Categorical data

- Very high cardinality can lead to super sparse data
- Pair & aggregate to reduce number of categories

- With “variable selection” methods, make sure you know what is selected!

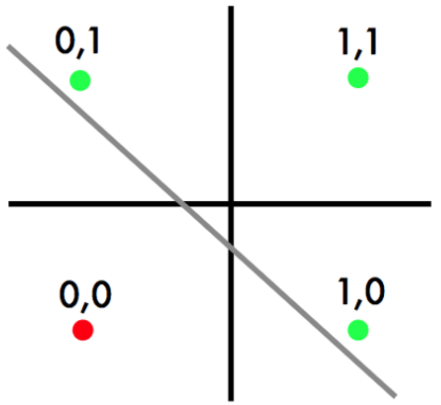
- You drop a variable? Are you dropping “day of week” or just “Mondays”. (Group lasso, categorical flags for trees, etc.)

Features of Categorical Data &

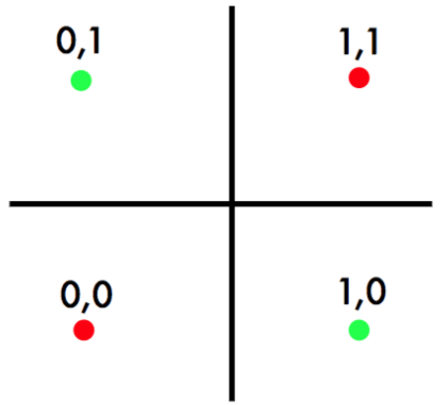
- Helps to achieve linear separability…
- Solving the “XOR” problem is hard for lots of methods.
- It’s NOT linearly separable.

X1	X2	Outcome	
0	0	0	(red)
0	1	1	(green)
1	0	1	(green)
1	1	0	(red)

The XO



OR



XOR

Features of Categorical Data &

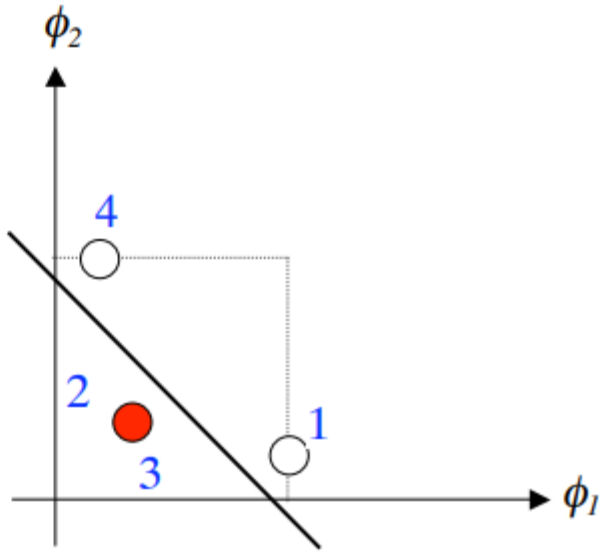
- **Radial basis function** (RBF) (kernel tricks) can help with XOR, and other stuff…

$$\phi_1(\mathbf{x}) = \exp\left(-\|\mathbf{x} - \boldsymbol{\mu}_1\|^2\right) \quad \text{with} \quad \boldsymbol{\mu}_1 = (0,0)$$

$$\phi_2(\mathbf{x}) = \exp\left(-\|\mathbf{x} - \boldsymbol{\mu}_2\|^2\right) \quad \text{with} \quad \boldsymbol{\mu}_2 = (1,1)$$

NOTE: these PHI's will essentially work as "hidden nodes" in neural networks!!!

p	x_1	x_2	ϕ_1	ϕ_2
1	0	0	1.0000	0.1353
2	0	1	0.3678	0.3678
3	1	0	0.3678	0.3678
4	1	1	0.1353	1.0000

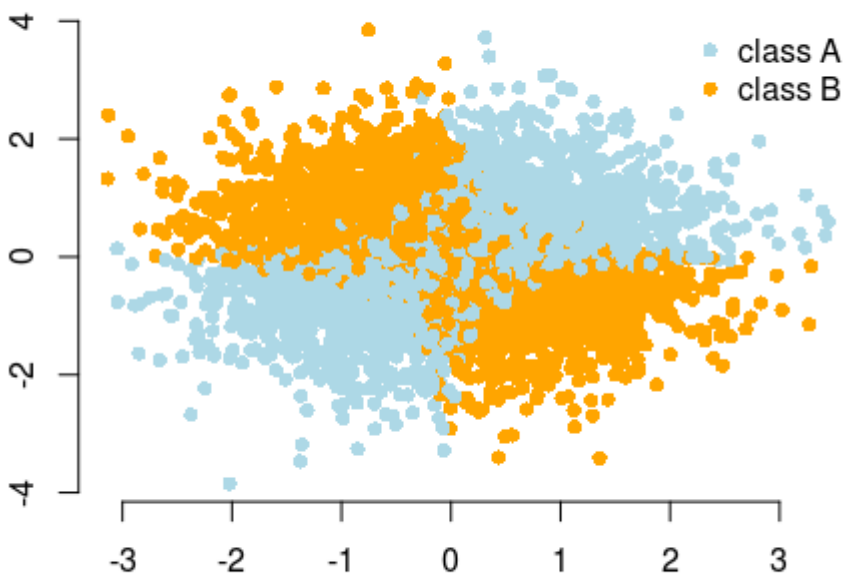


Categorical Data (interaction)

- Depending on the data encoding, creating a new feature helps to linearly separate the data

- Given the SIGNS of the data points on X and Y axis (and the associated category), a variable “Z = X*Y” essentially classifies the problem!

- Z > 0 => BLUE ☺



Aim for “linearity”

Cover’s Theorem:

“A complex pattern classification problem cast in a high dimensional space non-linearly is more likely to be linearly separable than in a low dimensional space”.

And we know that once we have linear separable patterns, the classification problem is easy to solve.

NaN / NA

- **Missing variables?**
  - Missing at random? Imputation may work… How? [max/min/median]. EM algorithm?
  - Missing income for mortgage/credit card? N/A = predictor of default
  - …

  - It's ok to give missing data their own "category". Income: high/low/missing

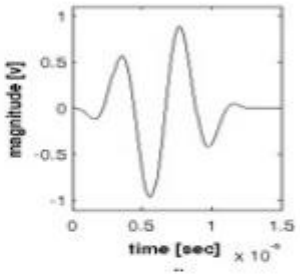
Find the right “coordinate space”

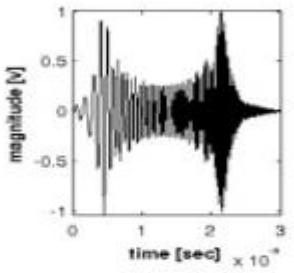
- Find a space where things look “**SIMILAR**” or interpretable

- Things finite in time domain are infinite in freq. domain & vice versa..

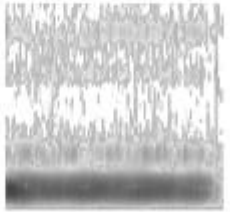
Sound 1 Sound 2

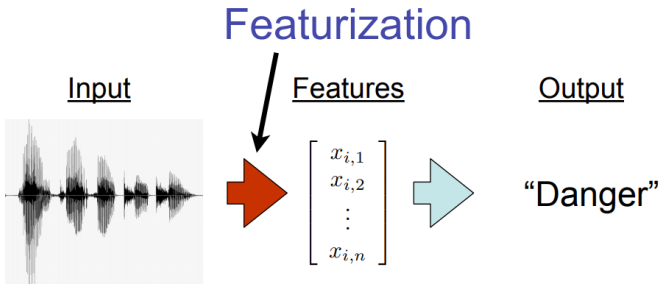
Time domain:

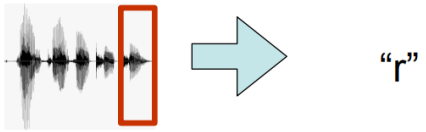




Frequency domain:

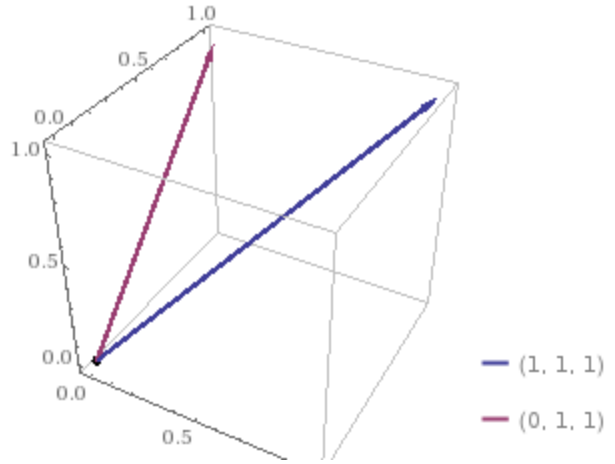


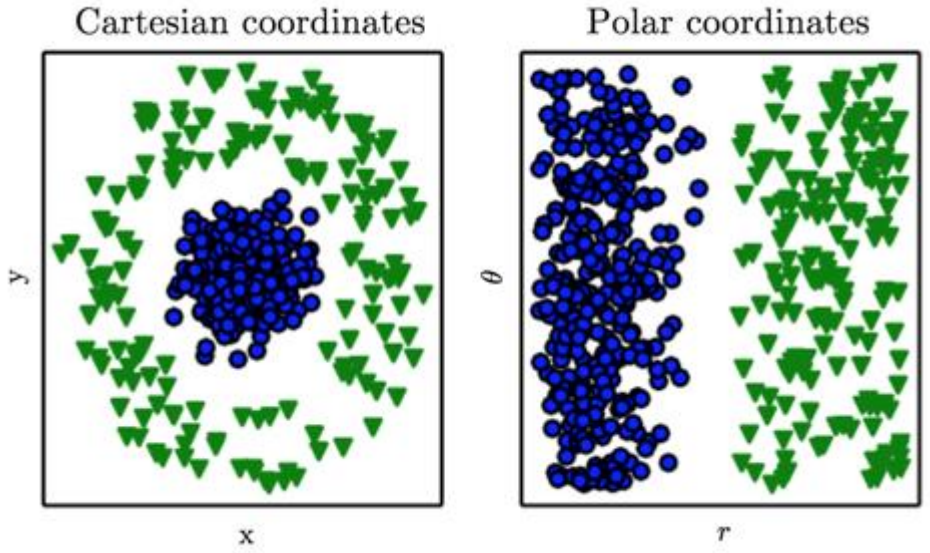




Find the right “coordinate space”

- Define appropriate “distances” and errors
- Text similarity? ANGLES between vectors
- More in “Deep learning” discussion on convolutions & filters…

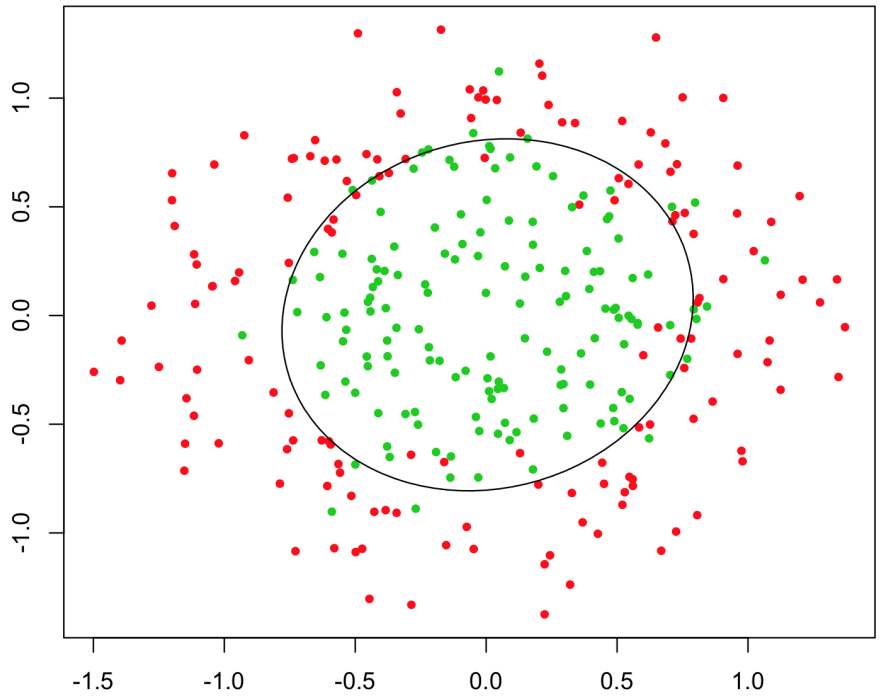




Find the right “coordinate space”

- With y = a0 + a1*x1 + a2*x2 no way…

- With y ~ a0 + a1*x1^2 + a2*x2^2 you can classify this



Dimensionality Reduction

- Project data into another space (usually smaller)

- Use principal components of the problem…
- SVD is the queen of linear algebra
- Deviation from the PCA prediction, etc.

- PCA regression – helps with too many inputs, for instance

Some Examples

- Predicting default? Debt-to-GDP (or debt-to-income) can be a slow-moving variable…
- Would sorting all subjects by debt/GDP in every period help, would the subjects in the distribution tails be more vulnerable…? Deviation from the median? Etc

Some Examples

- Customers spend different amounts in your store… $30, $2000, $14, …
- Categorize by percentiles and put into “bins”: top 1/3, mid 1/3, low 1/3 …?

- **Missing variables?**
  - Missing at random? Imputation may work… How? [max/min/median]. EM algorithm?
  - Missing income for mortgage/credit card? N/A = predictor of default
  - …

FILTERING

FILTERING

- Specialized domain-specific filters (image recognition, median filters, wavelets..)

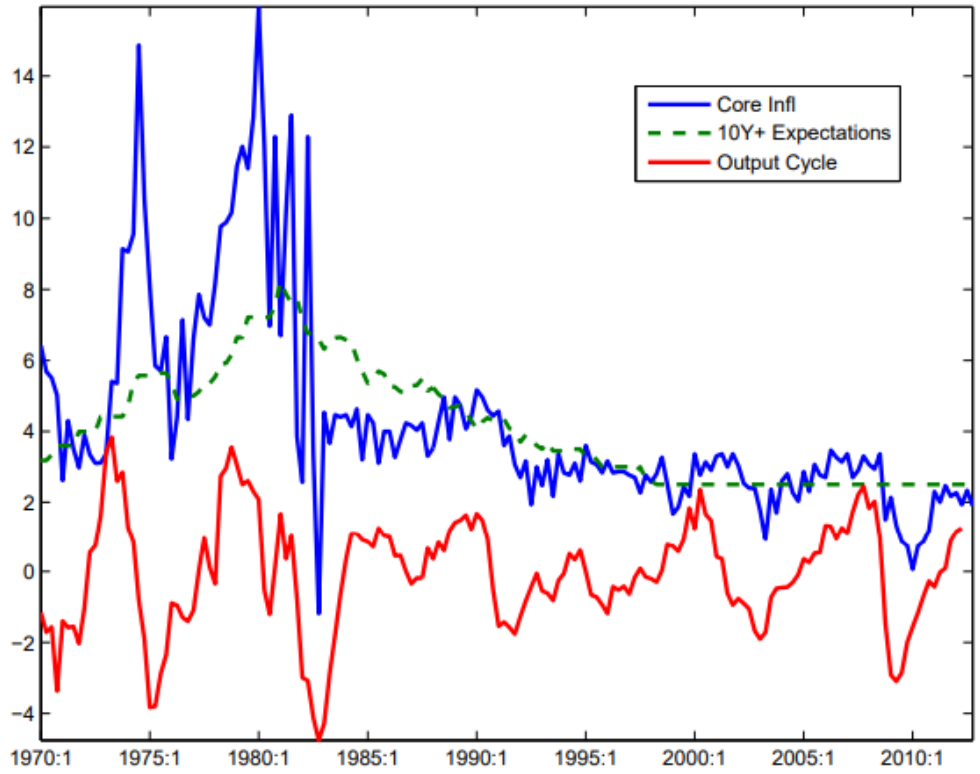
- Hodrick-Prescott/Lesser filter, band-pass filters, etc.

Some Examples

FILTERING

- Trend & cycle time series models

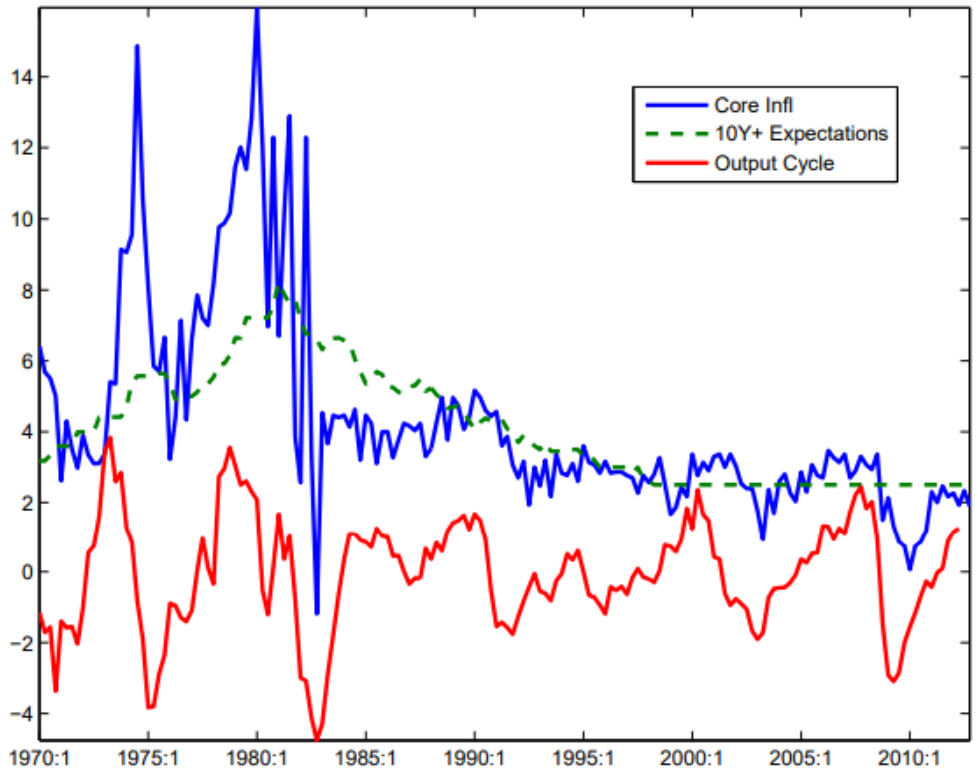
U.S. Inflation vs. Output Cycle

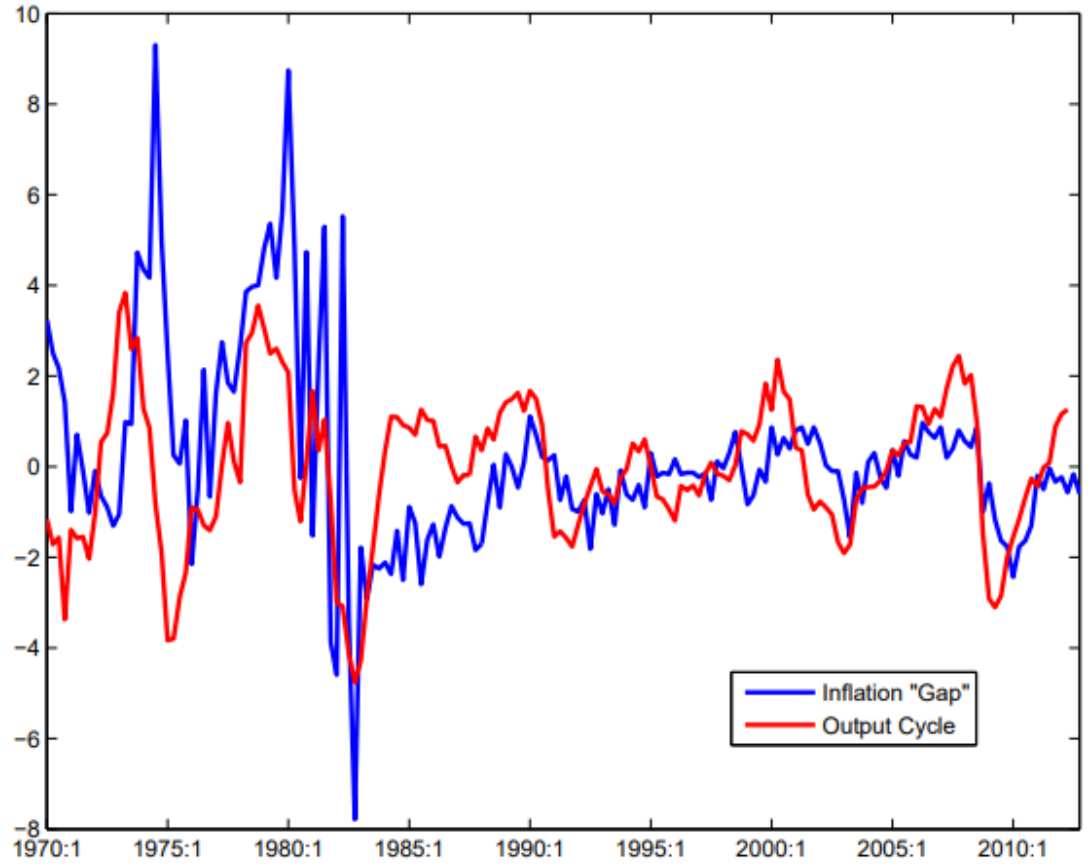


Phillips Curve – Dead or Alive?

FILTERING

U.S. Inflation vs. Output Cycle





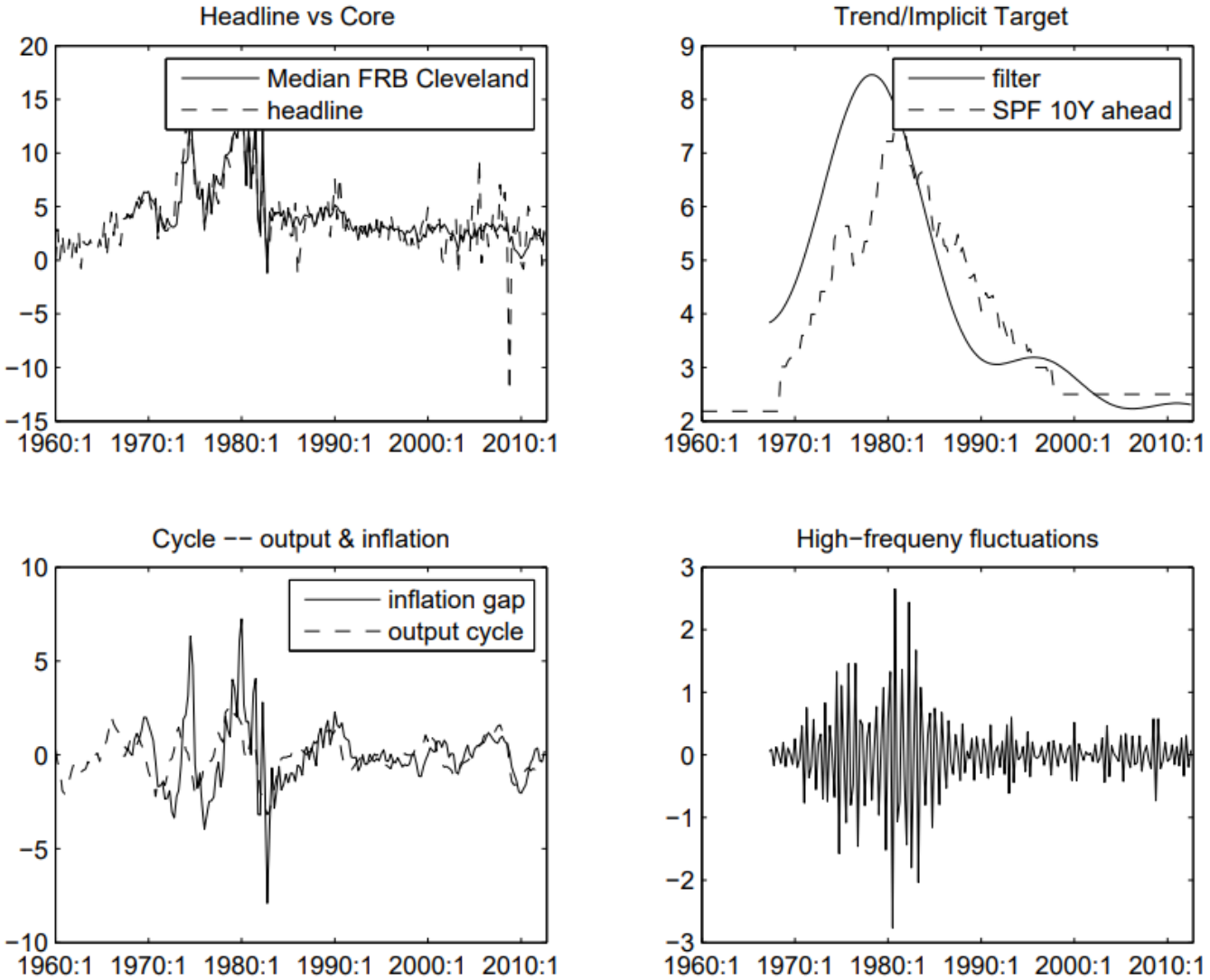
Some Examples

- **FILTERING**

- Median infl. Vs headline
- Subtract infl. Target or LR infl. Expectations…

- Cyclical frequency of output vs. convergence

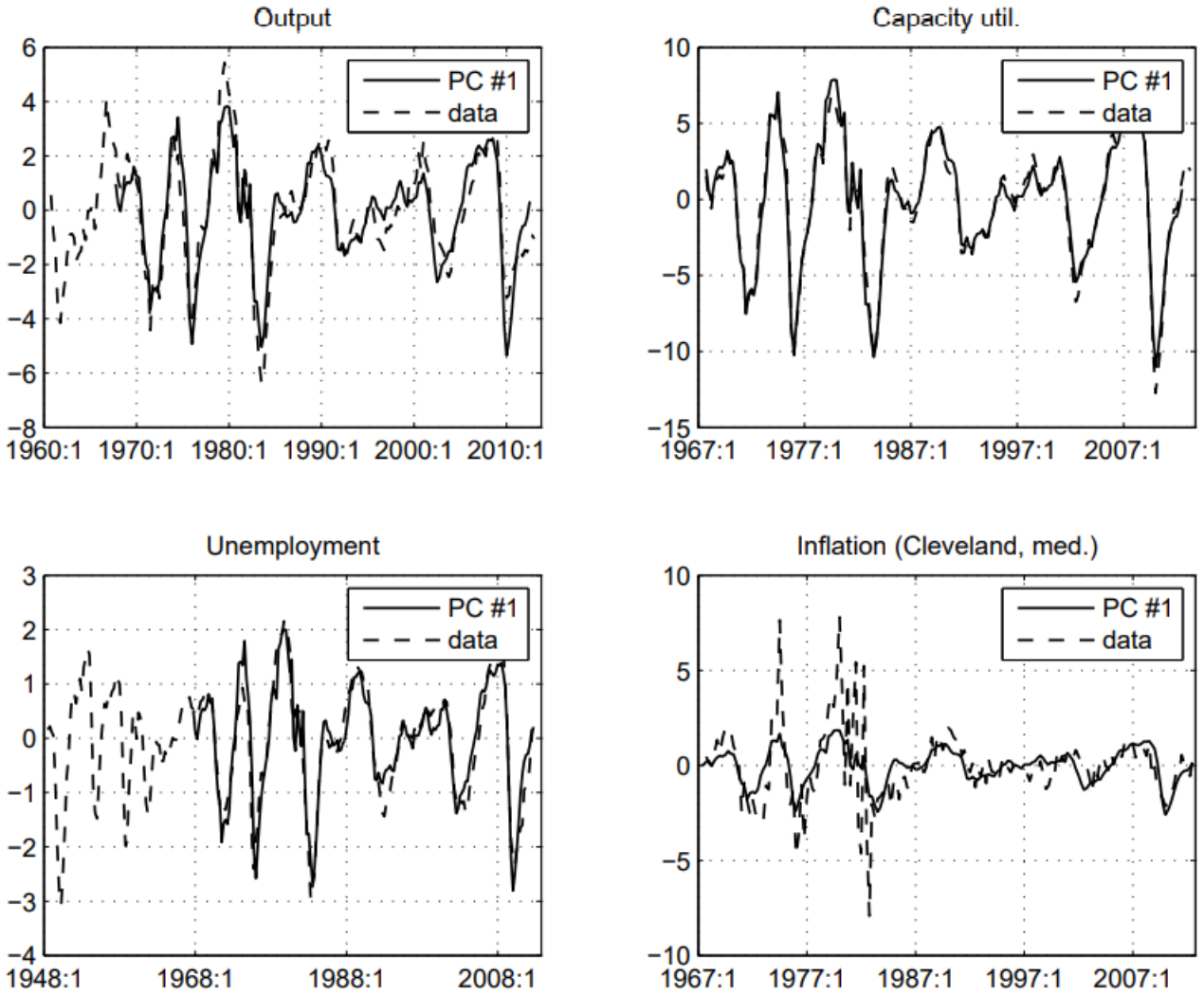
Figure 5: Decomposition of inflation



Some Examples

- **FILTERING**

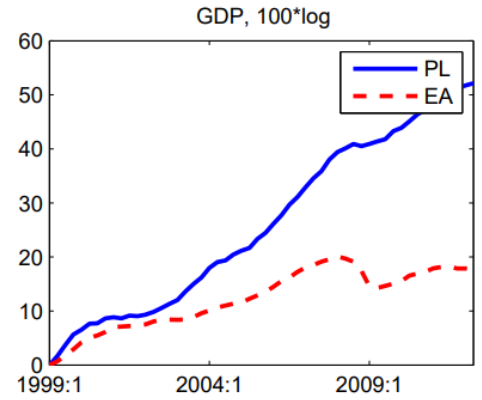
Figure 6: Principal component analysis

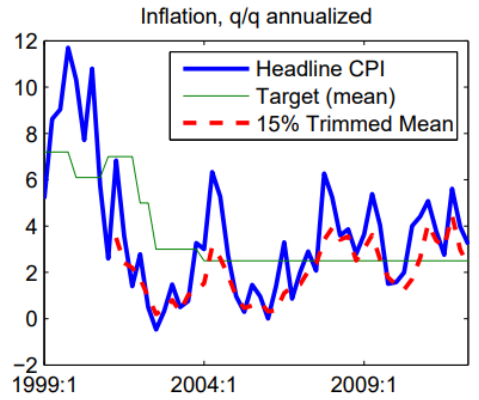


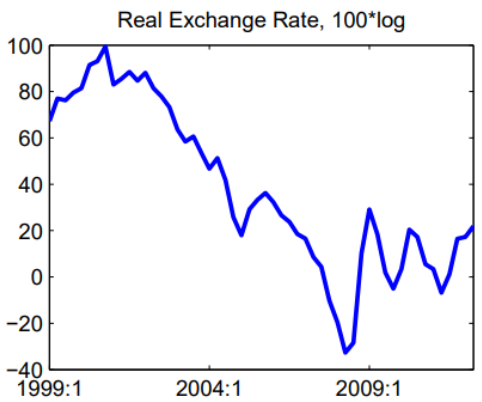
Convergence vs Business Cycle

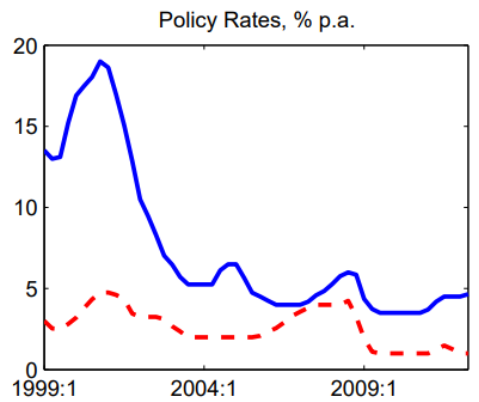
- Poland and Germany have strong cyclical co-movement… BUT Germany is growing at much slower pace…

- Intentional disinflation,
  so nominal rates & inflation co-move at low freqs… (price puzzle ☺☺)









Geospatial data?

- Make sure you understand units…
- Convert GPS coordinates to distances, angles, etc.

- Have you heard of "**Haversine Formula**"?
- …radius matters, India is large! etc.

Distance

This uses the '**haversine**' formula to calculate the great-circle distance between two points – that is, the shortest distance over the earth's surface – giving an 'as-the-crow-flies' distance between the points (ignoring any hills they fly over, of course!).

Haversine formula:

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$$

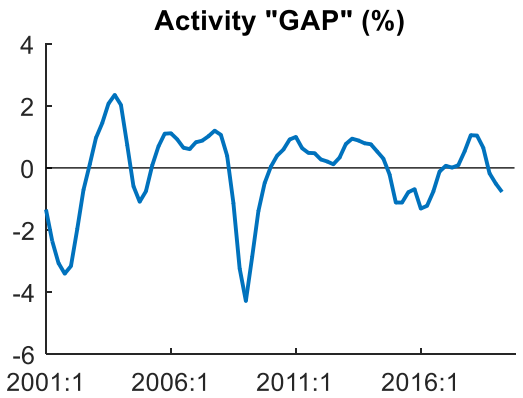
$$d = R \cdot c$$

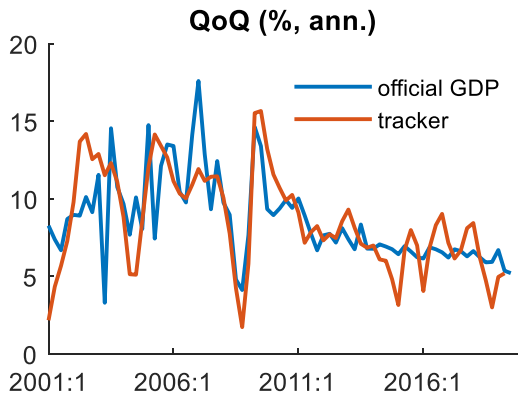
where φ *is latitude,* λ *is longitude,* R *is earth's radius (mean radius = 6,371km); note that angles need to be in radians to pass to trig functions!*

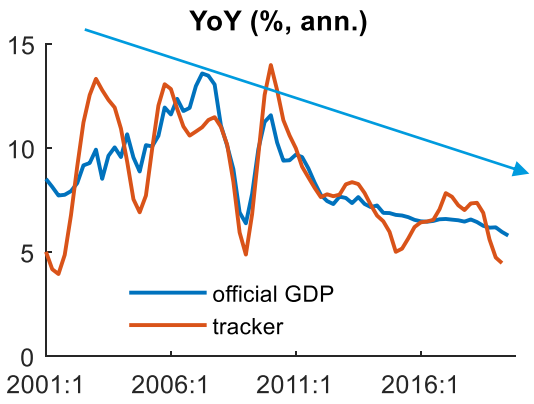
Forecasting Growth? Assessing risk?

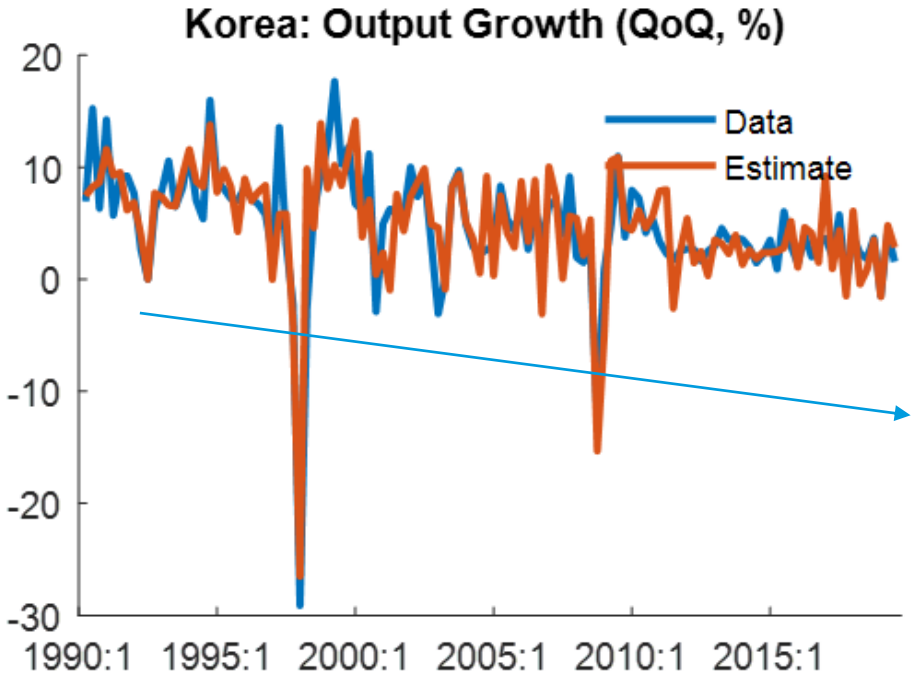
- Growth of many countries (or firms…) is not stable…
- China has been growing ~ 10% a year, now decelerating to "only" ☺ 5 % a year… You cannot just feed the data as is!
- It's unlikely it will go back to growing 10% a year again, given the stage of economic convergence… => a naïve model will predict reversion to 10% growth!!

Predicting Growth? Assessing Risk?









Pipelines & Leakage

- For out-of-sample testing, it's crucial to avoid "**LEAKAGE**"
- For instance, if you do HP the series for the whole sample before ANY analysis, the real-time performance can be very different…
- Two-sided filters transport "future" to the past data, etc. Knowing feature can cost you lots of **$$$$$**

!! You have to understand your problem !!