



**HAL**  
open science

## Data management in precision farming

Pierre Blavy

► **To cite this version:**

Pierre Blavy. Data management in precision farming. Data management in precision farming (Data management in precision farming), 2017. hal-02788369

**HAL Id: hal-02788369**

**<https://hal.inrae.fr/hal-02788369>**

Submitted on 5 Jun 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

# Data management in precision farming

Pierre Blavy, INRA

12 december 2017

# Adequation between question and data



- What can I do with existing data? (meta analysis)
- What's the question ?
  - ⇒ Will my data answer it (write the method+stats first)

# Adequation between question and data

- Practical decision

- ▶ Real time (Importantn for large farms)
  - ★ On farm alerts (vêlage, chaleurs, ...)
- ▶ Long term
  - ★ Herd management tools
  - ★ Planning asistants

- V.S. Research question

- ▶ Impact of feeding on production/environment
- ▶ Gentic selection on food efficiency?
- ▶ Animal behaviour/welfare?

# Costs v.s. benefits



Plonk et Replonk

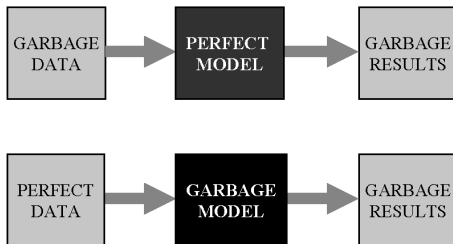
⇒ Cost v.s. benefits tradeoff

- ▶ money
- ▶ time
- ▶ human factors

# Get consistent data

## MODEL CALCULATIONS

”Garbage In-garbage Out” Paradigm



- Farm proof systems (machine robustness)
- Know what you measure
  - ▶ What's the data (definition, format, unit)
  - ▶ In which context it's measured / usable ?

# Organize data

- Use databases (SQL is your friend)
  - ▶ SQL <https://www.w3schools.com/sql/>
  - ▶ SQLite in R <http://cpc.cx/iZj>
- Maximize automation (program boring repetitive stuff)
  - ▶ With R <http://cpc.cx/kQJ>
  - ▶ With python <http://cpc.cx/kQK>
  - ▶ Others softwares are OK
- Write documentation
  - ▶ Someday, it will save someone
  - ▶ Who may be you.

# Watch your data

- Make plots

- ▶ R <http://cpc.cx/kQL>
- ▶ R+ggplot2 <http://cpc.cx/kQM>
- ▶ R+shiny <https://shiny.rstudio.com/>

- Descriptive stats

- ▶ mean, median, in, max...
- ▶ proportion of missing values
- ▶ R summary

- PCA...



# Principal Component Analysis



minimal variance



maximal variance

Sadly not my little sister

- PCA maximize the projected variance
  - ▶ Orthogonal axis
  - ▶ R, factominer <http://cpc.cx/kQQ>
- PCA is a good tool for
  - ▶ Summarizing data (check inertia)
  - ▶ Finding outliers (check individual weight)
  - ▶ Constructing synthetic variables (check axis contribution)
- PCA is NOT a tool for
  - ▶ Accurate correlations  $\Rightarrow$  watch the correlation matrix
  - ▶ Testing hypothesis  $\Rightarrow$  statistical test

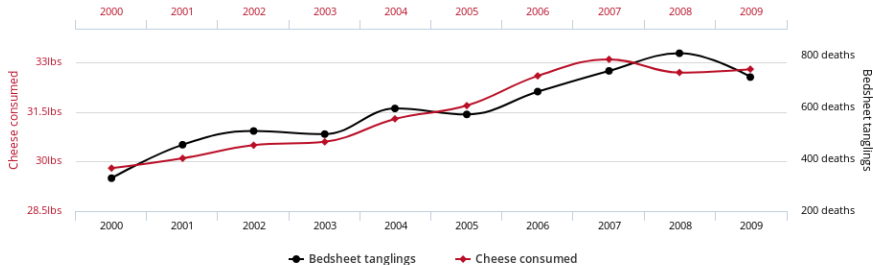
# Data preprocessing

## Per capita cheese consumption

correlates with

## Number of people who died by becoming tangled in their bedsheets

Correlation: 94.71% ( $r=0.947091$ )



Data sources: U.S. Department of Agriculture and Centers for Disease Control & Prevention

<http://www.tylervigen.com/spurious-correlations>

# Data preprocessing



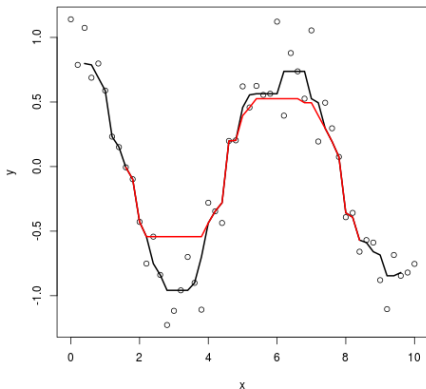
Calvin and Hobbes, Bill Watterson

- Remove irrelevant variables
  - ▶ Remove unrelated variables (spurious correlation)
    - ★ Crucial on datasets with many variables
  - ▶ Remove a posteriori variables
    - ★  $\text{crash survival day7} = \text{alcohol} + \text{crash speed} + \text{reeducation} - \text{duration}$
- Remove irrelevant data
  - ▶ Out of scope data (ex : holstein model, montbeliarde data)
  - ▶ Outliers (PCA, boxplots, ...)
    - ★ Wrong data (ex broken sensor)  $\Rightarrow$  discard
    - ★ Rare events (ex mastitis)  $\Rightarrow$  kept or not?

# Data preprocessing

- Data smoothing
- OK when we expect few changes within neighbors (time, space)
- Methods
  - ▶ kernel methods (ex : rowing average)
  - ▶ rowing median
  - ▶ splines
  - ▶ ...

# Data preprocessing



median filter, black 5 points, red 17 points

- Smoothing strength is a compromise.
- That depends on your question/model.
- An on your data variability.



milky way, with increasing size gaussian kernel



milky way, with increasing size gaussian kernel



milky way, with increasing size gaussian kernel





milky way, with increasing size gaussian kernel

# Aggregate variables

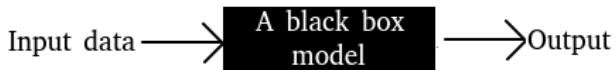
## When?

- The same thing is measured multiple times
  - ▶ Ex : captor1 and captor2
  - ▶ Rare gold standard v.s. frequent low quality measurements
- Strong correlation between variables
  - ▶ Ex behavioural traits

## How?

- Choose a variable
- Do a variable "mix" (ex linear combination)
- PCA : synthetic variables (a linear combination)
  - ▶ Don't overinterpret PCA synthetic variables

# Is my model OK or garbage?



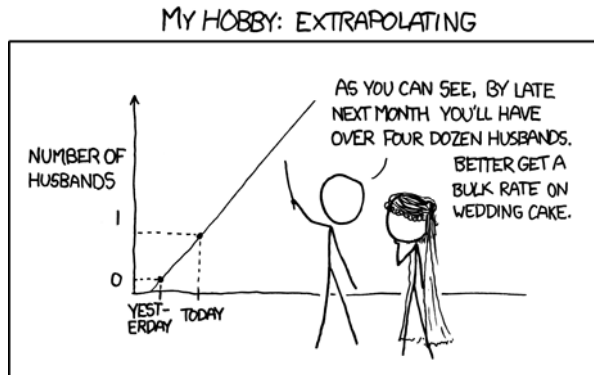
- A model can be anything

- ▶ Linear model <http://cpc.cx/kQE> , <http://cpc.cx/kQF>
- ▶ Non linear model <http://cpc.cx/kQG>
- ▶ A classifier <http://cpc.cx/kQR>
- ▶ ...
- ▶ The devil himself in a unicorn disguise

⇒ Garbage in, garbage out

- We assume that input data is ok, is the model garbage?

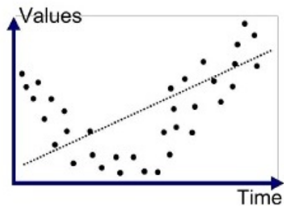
# Typical evil models : extrapolation



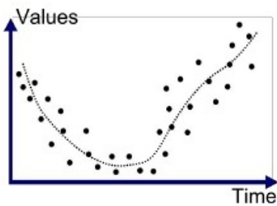
xkcd.com, Randall Munroe, CC-BY-Cn2.5

- Your training and testing data must cover your practical use cases

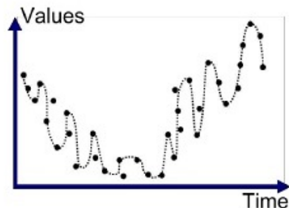
# Typical evil models : Overfitting



Underfitted



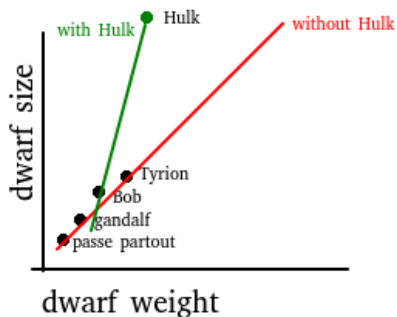
Good Fit/Robust



Overfitted

- Many parameters, few data  $\Rightarrow$  Check degree of freedom

# Typical evil models : outliers



- Outlier with heavy weight  $\Rightarrow$  Check for outliers
- Check your question
  - ▶ General model : keep Hulk, rename axis
  - ▶ Dwarf model : discard Hulk

**WARNING : evil models exists in unexpected cases**  
 $\Rightarrow$  **ALWAYS TEST YOUR MODELS**

# Test your models

	detect mine	detect rock		detect mine	detect rock
mine	9	1		mine	5
rock	50	50		rock	10

Two mines detectors, what's the best one?

- All mistakes are NOT equivalent
  - ▶ Be careful, some methods (like area under ROC curve) say so
  - ▶ When possible use a real-life criterion (human lives, money, ...)
  - ▶ When not, define a reasonable criterion **apriori**.

# Test your models

- What the difference between model prediction and model output
  - ▶ Plot observed v.s. predicted
  - ▶ Model fitting (ex sum of square)
  - ▶ Cross with other knowledge source (common sense, litterature, meta analysis ...)
- Model testing
  - ▶ Use an **independant** dataset
  - ▶ Test the model on it, according to your criterion
  - ▶ See bootstrap like methods.



# Conclusion

