Desiging a Cross-paradigm Modeling Framework

Ben Klemens

ben.klemens@census.gov

14 May 2013

What is a statistical model?

- Stats undergrad: an ordinary least squares regression. What else is there?

What is a statistical model?

- Stats undergrad: an ordinary least squares regression. What else is there?
- Bayesian: a sequence of distributions.
  $P(X, Y, Z) = P(X|Y, Z) \cdot P(Y|Z) \cdot P(Z)$.

What is a statistical model?

- Stats undergrad: an ordinary least squares regression. What else is there?
- Bayesian: a sequence of distributions. $P(X, Y, Z) = P(X|Y, Z) \cdot P(Y|Z) \cdot P(Z)$.
- Engineer: a flowchart explicitly describing the elements of a system.
- Agent-based modeler (ABMer): a collection of agents and rules for their interaction.

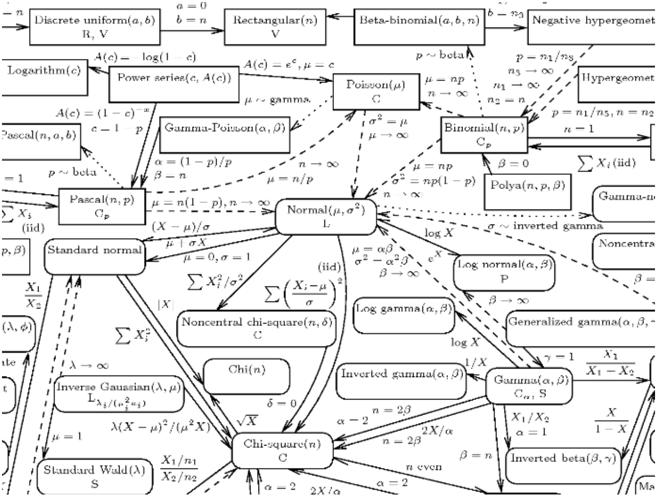
What is a statistical model?

- Stats undergrad: an ordinary least squares regression. What else is there?
- Bayesian: a sequence of distributions. $P(X, Y, Z) = P(X|Y, Z) \cdot P(Y|Z) \cdot P(Z)$.
- Engineer: a flowchart explicitly describing the elements of a system.
- Agent-based modeler (ABMer): a collection of agents and rules for their interaction.
- Empiricist: an observed distribution of occurrences.

The Outline

- Motivation: why a standard model framework?
- Definition: Models as bundles of functions
- Some examples
- Filling in the blanks
  - Quick prototyping: you give me a likelihood function or an RNG; I'll test hypothesis about the model parameters.
- Transformation and combination operations
  - Both with pencil/paper and keyboard: a standard vocabulary for descriptive modeling
- A final example

Transforming a model produces a new model



http://www.math.wm.edu/~leemis/chart/UDR/UDR.html

Why do mathematicians formally define terms?

- If I use correctly-defined transformations on correctly defined atoms, I am guaranteed that the results are coherent.
- I can often determine what is *not* valid by inspection.
- Define transformations
  - Hierarchical models: the output from a set of child models feed data to a parent model.
  - Bayesian models: the output from a prior is used as a parameter set for the likelihood.
  - Structural equation models, causal models: simple models linked together to form a complex larger model.
- Modern computing technology
  - Formal definition maps immediately to structures and functions
- World peace
  - Monoids: [($\mathbb{N}$, $+$), (finite strings, concatenation), (models, cross)]
  - There are commonalities across seemingly un-common genres.

The computing slide

What structure is provided on top of FORTRAN '77?

- Some really are FORTRAN '77 with a pretty interface.
- Some provide tools for one genre only. [Can't use R for ABM; can't use Repast for regression.]
- Even some unifications are still only for small subsets of models: S's GLM model notation; King's Zelig, also for GLMs; BUGS/JAGS/R-BUGS for distributions + GLMs;
- Church, BLOG, Lisp-Stat: broad, unstructured frameworks

The computing slide

What structure is provided on top of FORTRAN '77?

- Some really are FORTRAN '77 with a pretty interface.
- Some provide tools for one genre only. [Can't use R for ABM; can't use Repast for regression.]
- Even some unifications are still only for small subsets of models: S's GLM model notation; King's Zelig, also for GLMs; BUGS/JAGS/R-BUGS for distributions + GLMs;
- Church, BLOG, Lisp-Stat: broad, unstructured frameworks

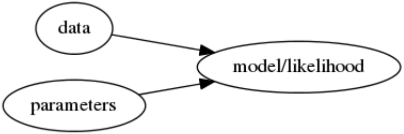
Apophenia, a C library

- This talk will not be an Apophenia tutorial or sales pitch. See `http://apophenia.info` .
- It will only have one slide with actual code.
- Please, implement this on your favorite platform, standalone or via front-end for Apophenia.

Definition

A model intermediates between data, parameters, and likelihoods

- data+parameters input: likelihood, or integrate to CDF



- data input: estimate parameters

- parameter input: draw random data, estimate most likely data

Notation

- $\mathbb{D}$: Data space. Anything required by the model; 'private' to the model unless otherwise noted. $\leq$ is defined. [sample space]
- $\mathbb{P}$: Parameter space. Similarly model-specific. [state space]
- $\mathbb{M}$: Model space. The set of bundles of ML-consistent functions as per the next slide.

A bundle of functions

A model is an internally consistent bundle of functions to intermediate between data, parameters, and likelihoods:

- Likelihood: $(\mathbb{D}, \mathbb{P}) \to \mathbb{R}^+$.
  - Integrates to a finite value; always nonnegative.
  - In some cases, better described as an 'objective function'. Doesn't have to integrate to one.
- Estimation: $\mathbb{D} \to \mathbb{P}$
  - ML-consistency: L(**d**, **p**) is maximized by $\mathbf{p} = \textsc{Est}(\mathbf{d})$.
- RNG: $\mathbb{P}$ (and uniform prng) $\to \mathbb{D}$.
  - Likelihood of draw $\mathbf{d} = \mathrm{RNG}(\mathbf{p}) \propto L(\mathbf{d}, \mathbf{p})$.
- CDF: $(\mathbb{D}, \mathbb{P}) \to [0, 1]$.
  Proportion of random draws $\mathrm{RNG}(\mathbf{p}) \leq \mathbf{d} \to \mathrm{CDF}(\mathbf{d}, \mathbf{p})$.

Alternatives to ML-consistency?

We could replace the consistency rule for $\textsc{Est}$ using other consistency rules:

- KL-minimizing consistency
- Mean-squared-error minimizing
- Entropy-maximizing consistency
- Moments of $\textsc{Est}(\mathbf{d})$ match moments of $\mathbf{d}$.

Alternatives to ML-consistency?

We could replace the consistency rule for $\textsc{Est}$ using other consistency rules:

- KL-minimizing consistency
- Mean-squared-error minimizing
- Entropy-maximizing consistency
- Moments of $\textsc{Est}(\mathbf{d})$ match moments of $\mathbf{d}$.

But composing a entropy-maximizing model with a MoM model doesn't always make sense, so I stick to one genre here.

Three examples

The Normal example

- Likelihood: $\mathcal{N}(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-(x-\mu)^2/2\sigma^2)$ or $\ln \mathcal{N}(x, \mu, \sigma^2) = (-(x-\mu)^2/2\sigma^2) - \ln(2\pi\sigma^2)/2$.
- Estimation: $\hat{\mu} =$ mean of D; $\hat{\sigma} = \sum(d - \hat{\mu})^2/n$.
- RNG: See Devroye (1986).
- CDF: `gsl_cdf_gaussian_P(d-mu, sd)` (or see `erf`).

The Discrete distribution (probability mass function, PMF)

A list of data items d_i, $i = 1 \ldots N$, with weights w_i.

- $\mathbb{D}$: $\mathbb{R}$, categories, ...
- $\mathbb{P}$: $\{\emptyset, \mathbb{D}^1, \mathbb{D}^2, \ldots, \mathbb{D}^N\}$
- Estimation: Copy input data $\rightarrow$ parameters.
- Likelihood: If any elements in new data set $\mathbf{x} \in \mathbb{D}$ are not $\in d$, zero. Else, product of matched weights.
- RNG: draw from d_is weighted by weights.
- CDF: sort d_is, sum weights.

OLS

As given in the textbooks, not a consistent model by the defn here.

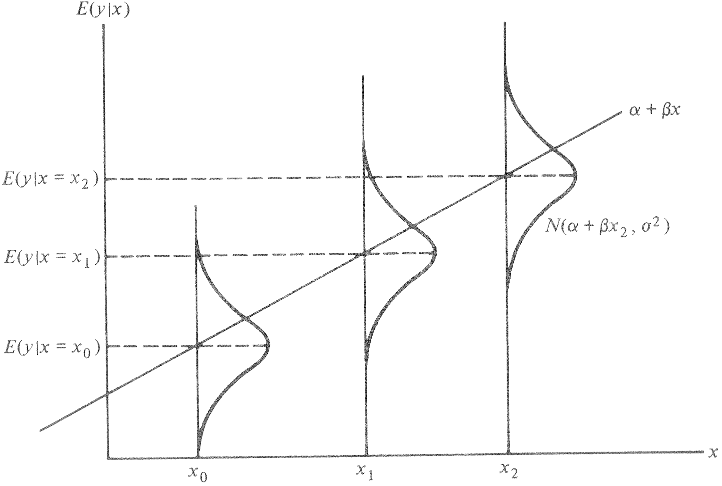


FIGURE 5.3 The classical regression model.

[Greene, *Econometric Analysis*, 2^{nd} ed., p 144]

OLS

Undergrad stats consists of picking $\mathbb{D}$: should it be
{BMI, age, sex, hours exercise/day},
{BMI, age, sex, age$\times$(is female), hours exercise/day},
{BMI, age, sqrt(hours exercise/day)}, . . . ?

OLS

Undergrad stats consists of picking $\mathbb{D}$: should it be
{BMI, age, sex, hours exercise/day},
{BMI, age, sex, age×(is female), hours exercise/day},
{BMI, age, sqrt(hours exercise/day)}, ...?

Given $\mathbb{D}$, starts as expected, but we hit a difficulty with RNG.

- $\mathbb{D}$: as above, K variables.
- $\mathbb{P}$: $\boldsymbol{\beta} \in \mathbb{R}^K$
- Estimation: $\boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{Y})$
- RNG: First, draw $\mathbf{X}$ from a PMF built from the input data; then draw ϵ from a $\mathcal{N}(0, \sigma)$; then $\mathbf{Y} = \mathbf{X}'\beta + \epsilon$.
- Likelihood: $(\mathbf{Y} - \beta\mathbf{X}) \sim \mathcal{N}(0, \sigma)$ (if $\mathbf{X} \in \mathbf{D}$); see Normal model.

With a standard interface, build standard procedures

- Testing: use the CDF or parameter models (and their CDFs).
- Bootstrapping, Jackknifing, Cook's distance: requires only estimation.
- MLE methods: as above, require only log likelihood; may use the score
- ML imputation: also requires only likelihood
- Tea: an R package for survey processing
- K-L divergence: use CDF or likelihoods; RNG can help if you want to do importance sampling

A simulation example

Just a likelihood

I only wrote down a likelihood function, $P(\mathbf{D}, \mathbf{P})$.

- Score (dlog likelihood): numeric deltas.
- Estimation: Use Maximum likelihood estimation.
  - All MLE algorithms repeatedly sample from the likelihood. Some use the score.
- RNG: ARMS (Gilks 1995)
- CDF: make random draws, count the percent up to a given point

Just an RNG

I only wrote down a likelihood function, $P(\mathbf{D}, \mathbf{P})$.

- Likelihood: make a million draws, write down a PMF using those draws.
- Estimation: give a likelihood, use prior slide.
- CDF: make random draws, count the percent up to a given point

A network simulation (just an RNG)

Agents have randomly drawn individual positions, match based on proximity.

- Fix $\sigma = 1$.
- For each of N agents,
  - Draw N preferences (p_i) from a $\mathcal{N}(0, \sigma)$.
- For each pair of agents,
  - Link with probability $1/(1 + |p_i - p_j|)$.
- Count up links, report the sorted list of link counts for each agent.

The two most popular outputs

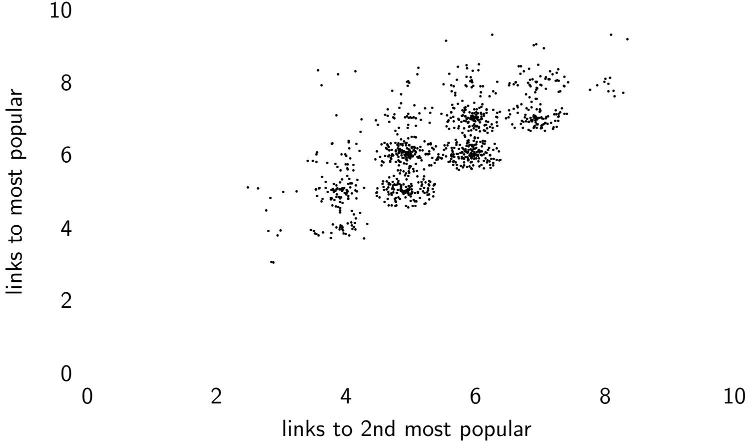


Figure: A distribution of the number of links to the two highest-ranked members of a ten-person network (w/jitter).

Our RNG defined a full model

We can calculate the other elements of the model from the RNG (memoize, use PMF).

- H: the most popular agent has ≤ 4 links.
- $\mathrm{CDF}_{NS}([4, 10, \ldots, 10], \emptyset) \approx 0.0533$.

Our RNG defined a full model

We can calculate the other elements of the model from the RNG (memoize, use PMF).

- H: the most popular agent has $\leq$ 4 links.
- $\mathrm{CDF}_{NS}([4, 10, \ldots, 10], \emptyset) \approx 0.0533$.

This was so easy to do, more people might start doing it.

Transforming models to produce other models: $\mathbb{M} \rightarrow \mathbb{M}$

The basic procedure

- Modify each element of the bundle
- Use defaults if needed
- Check the ML-consistency rules

Fixed parameters

Start with a $\mathcal{N}(\mu, \sigma)$; produce a $\mathcal{N}(\mu, 1)$.

- $\mathbb{D}$: No change
- $\mathbb{P}$: New space is reduced from original space
- Likelihood: Fix the parameter,use the base model's likelihood
- Estimation: MLE.
- RNG: Use the base model's.
- CDF: Use the base model's.

More transformations

- Fixed parameters
- Constrained data
- Constrained parameters
- Jacobian transformations
- Smoothing (e.g., cubic splines, moving average)
- Kernel density (using another model as the kernel)

Joining models: $(\mathbb{M}, \mathbb{M}) \rightarrow \mathbb{M}$

Stacking uncorrelated distributions

You need a Normal/Inverse Wishart prior for your Bayesian updating?

- $\mathbb{D}$: Two options: $\mathbb{D}_1 \times \mathbb{D}_2$ (if $\mathbb{D}_1 = \mathbb{D}_2$, one could send the same data to both models.)
- $\mathbb{P}$: $\mathbb{P}_1 \times \mathbb{P}_2$
- Likelihood: $L_1(\mathbf{d}_1, \mathbf{p}_1) \cdot L_2(\mathbf{d}_2, \mathbf{p}_2)$
- Estimation: Independent estimations.
- RNG: $(\mathrm{RNG}_1(\mathbf{p}_1), \mathrm{RNG}_2(\mathbf{p}_2))$
- CDF: use the default.

Easy to extend to three or more models.

Output from model 1 $\Rightarrow$ input to model 2

- Four options:
  - $\mathbb{P}_{\text{out}} = \mathbb{D}_{\text{in}}$
  - $\mathbb{P}_{\text{out}} = \mathbb{P}_{\text{in}}$
  - $\mathbb{D}_{\text{out}} = \mathbb{D}_{\text{in}}$
  - $\mathbb{D}_{\text{out}} = \mathbb{P}_{\text{in}}$
- Aggregate model is a model in its own right, with its own $\mathbb{P}$ and $\mathbb{D}$ (but either may be $\emptyset$).

Parameter composition ($\mathbb{D}_{\rm out} = \mathbb{P}_{\rm in}$)

Instead of setting $\mathbf{p}_2$ to a fixed value, draw $\mathbf{p}_2$ from another distribution.

- $\mathrm{RNG}_1 : \mathbb{P}_1 \rightarrow \mathbb{D}_1$
- $LL_2 : (\mathbb{D}_2, \mathbb{P}_2) \rightarrow \mathbb{R}$
- These are composable iff $\mathbb{D}_1 \equiv \mathbb{P}_2$. Then:
- $LL_2 : (\mathbb{D}_2, \mathrm{RNG}_1(\mathbb{P}_1)) \rightarrow \mathbb{R}$

Parameter composition ($\mathbb{D}_{\text{out}} = \mathbb{P}_{\text{in}}$)

Instead of setting $\mathbf{p}_2$ to a fixed value, draw $\mathbf{p}_2$ from another distribution.

- $\mathrm{RNG}_1 : \mathbb{P}_1 \to \mathbb{D}_1$
- $LL_2 : (\mathbb{D}_2, \mathbb{P}_2) \to \mathbb{R}$
- These are composable iff $\mathbb{D}_1 \equiv \mathbb{P}_2$. Then:
- $LL_2 : (\mathbb{D}_2, \mathrm{RNG}_1(\mathbb{P}_1)) \to \mathbb{R}$

We like to call M_1 *the prior* and M_2 *the likelihood*.

Parameter composition ($\mathbb{D}_{\text{out}} = \mathbb{P}_{\text{in}}$)

Instead of setting $\mathbf{p}_2$ to a fixed value, draw $\mathbf{p}_2$ from another distribution.

- $\mathrm{RNG}_1 : \mathbb{P}_1 \to \mathbb{D}_1$
- $LL_2 : (\mathbb{D}_2, \mathbb{P}_2) \to \mathbb{R}$
- These are composable iff $\mathbb{D}_1 \equiv \mathbb{P}_2$. Then:
- $LL_2 : (\mathbb{D}_2, \mathrm{RNG}_1(\mathbb{P}_1)) \to \mathbb{R}$

We like to call M_1 *the prior* and M_2 *the likelihood*.

- If M_1 and M_2 are on the conjugate table, then the combination model is a closed-form distribution.
- Else, use Gibbs sampling to produce a PMF model.

Data composition: multilevel modeling

- Do regressions for each classroom, producing params $\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^n$.
- Then do a cluster analaysis on the $\boldsymbol{\beta}$s.
- I.e., use $\mathbb{P}_{\text{out}}$ as $\mathbb{D}_{\text{in}}$.

Data composition: evaluating the simulation

Continuing the example of the network model, which outputs a data set.
A link distribution has some well-known distributions: Zipf, exponential, ...

- $\mathrm{RNG}_1 : \mathbb{P}_1 \to \mathbb{D}_1$
- $\mathrm{L}_2 : (\mathbb{P}_2, \mathbb{D}_2) \to \mathbb{R}^+$
- Compose to produce $\mathrm{L}(\mathbf{p}_2, \mathrm{RNG}_1(\mathbf{p}_1))$.

Filling in the form:

- $\mathbb{D}$: $\emptyset$
- $\mathbb{P}$: λ
- Likelihood: $\mathrm{L}_2(\emptyset, \mathbf{p}_2)$
- Estimation: (Stochastic) MLE.
- RNG, CDF: $\mathbb{D} = \emptyset$.

Data composition: use

- Above, we found the most likely λ from the simulation/evaluation model.
- Better: begin with a prior distribution on λ and use the sim/eval model to update to a posterior distribution on λ.

OK, here's some code.

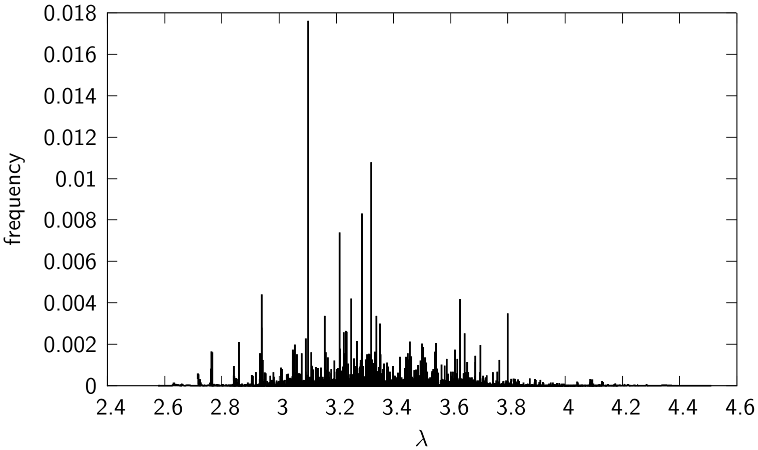
```
#include <apop.h>
#include "network_sim.c"

int main(){
   gsl_rng *r = apop_rng_alloc(1234);
   apop_model *comp = apop_model_dcompose(&network_sim,
                                          &apop_exponential, r);

   Apop_model_add_group(comp, apop_mle, .method=APOP_SIMAN);
   apop_model *estimated = apop_estimate(NULL, *comp);
   apop_model_print(estimated);

   apop_model *norm = apop_model_set_parameters(apop_normal, 3.5, .25);
   apop_model *post = apop_update(.prior=norm, .likelihood=comp);
   apop_data_pmf_compress(post->data);
   apop_data_sort(post->data);
   apop_model_print(post);
}
```

Output



0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
frequency
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
λ

A story problem

The dinner party

- Two types come to my 8PM dinner party:
  - Tries to be on time, but hits a sequence of 10-minute delays, each with probability λ.
  - Shoots for 8:30, gets there on time $\pm\epsilon$.
- Nobody shows up early.

The lateness model

$M_{\text{mix}} =$
mix(
    $\text{Jacobian}_{1/\lambda}($
        $\text{Exponential}(\lambda)$
    ),
    truncate(
        $\text{Normal}(\mu, \sigma)$
)
)

For the aggregate model:

- $\mathbb{P} = (\lambda, \mu, \sigma)$
- $\mathbb{D} = \mathbb{R}^+$ (arrival times)

Don't forget priors

$M_{\text{prior}} =$
stack(
truncate(
Normal(μ_1, σ_1)
),
Normal(μ_2, σ_2),
Invert(
Wishart(Σ)
)
)

For M_{prior}:

- $\mathbb{P} = (\mu_1, \sigma_2, \mu_1, \sigma_2, \Sigma)$
- $\mathbb{D} = (\lambda, \mu, \sigma)$ (AKA $\mathbb{P}_{\text{Mix}}$)

The whole thing

$M_{\mathrm{arrival}} = \text{DP-compose}(M_{\mathrm{prior}}, M_{\mathrm{mix}})$
For M_{arrival}:

- $\mathbb{P}_{\mathrm{arrival}} = (\mu_1, \sigma_1, \mu_2, \sigma_2, \Sigma)$
- $\mathbb{D} = \mathbb{R}^+$ (arrival times)

The whole thing, written out

```
M_arrival = DP-compose(
          stack(
              truncate(
                  Normal(μ1, σ1)
              ),
              Normal(μ2, σ2),
              Invert(
                  Wishart(Σ)
              )
          ),
          mix(
              Jacobian_{1/λ}(
                  Exponential(λ)
              ),
              truncate(
                  Normal(μ, σ)
              )
          )
      )
```

Using the model

Reduced to a nonparametric PMF:

- Fix $\mathbb{P}_{\mathrm{arrival}}$ and find a posterior PMF of arrival times (Bayesian updating).
  - Then, do data-space evaluations, e.g. K-L divergence(M_{arrival}, PMF (data)).

Using the model

Reduced to a nonparametric PMF:

- Fix $\mathbb{P}_{\text{arrival}}$ and find a posterior PMF of arrival times (Bayesian updating).
  - Then, do data-space evaluations, e.g. K-L divergence(M_{arrival}, PMF (data)).

As a parameterized model:

- Use observed arrival times to find the optimum in $\mathbb{P}_{\text{arrival}}$.
  - Then, do parameter-based tests.

Using the model

Reduced to a nonparametric PMF:

- Fix $\mathbb{P}_{\text{arrival}}$ and find a posterior PMF of arrival times (Bayesian updating).
  - Then, do data-space evaluations, e.g. K-L divergence(M_{arrival}, PMF (data)).

As a parameterized model:

- Use observed arrival times to find the optimum in $\mathbb{P}_{\text{arrival}}$.
  - Then, do parameter-based tests.

The conclusion slide

We can formally define a model as a bundle of functions that are internally consistent.

It's a simple definition, but it lets us:

- Apply standard tools to simulations, ML models, ...
- Implement complex models using simple components (both at the keyboard and AFK).
- Describe disparate statistical situations in a consistent manner.
    - Clarify inconsistencies and reveal new applications of old tools.
    - Try methods from different genres of modeling.