Bayesian Workflow

"How to structure the process of your analysis to maximise [sic] the odds that you build useful models."

-Jim Savage

Sean Talts
Core Stan Developer

Bayesian Workflow

Scope out your problem

What inputs and outputs can help you learn? What relationships can you see by eye?

Specify likelihood & priors

Use knowledge of the problem to construct a generative model and shape the scope of the parameters

Check the model with fake data

Generate data, fit model, and evaluate fit as a sanity check

Fit the model to real data

To recover parameters

Check diagnostics

Algorithms should come with diagnostics that let you know when they're not working

Graph fit estimates

Understand your inferences

Check predictive posterior

Perform PPCs to understand predictions

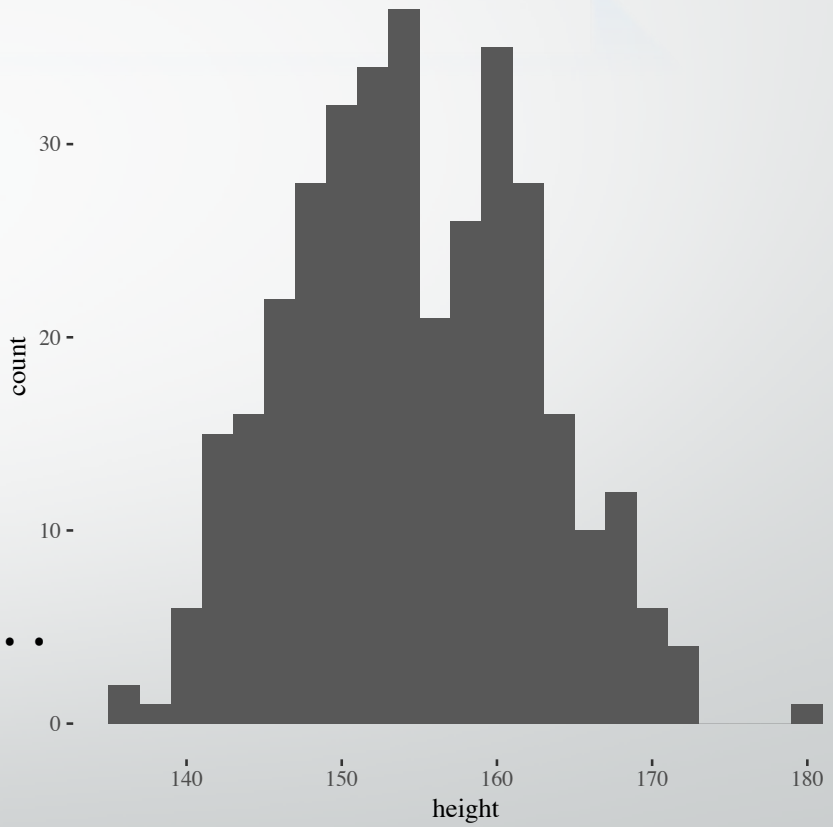
Compare models

Iterate on model design, choose a model

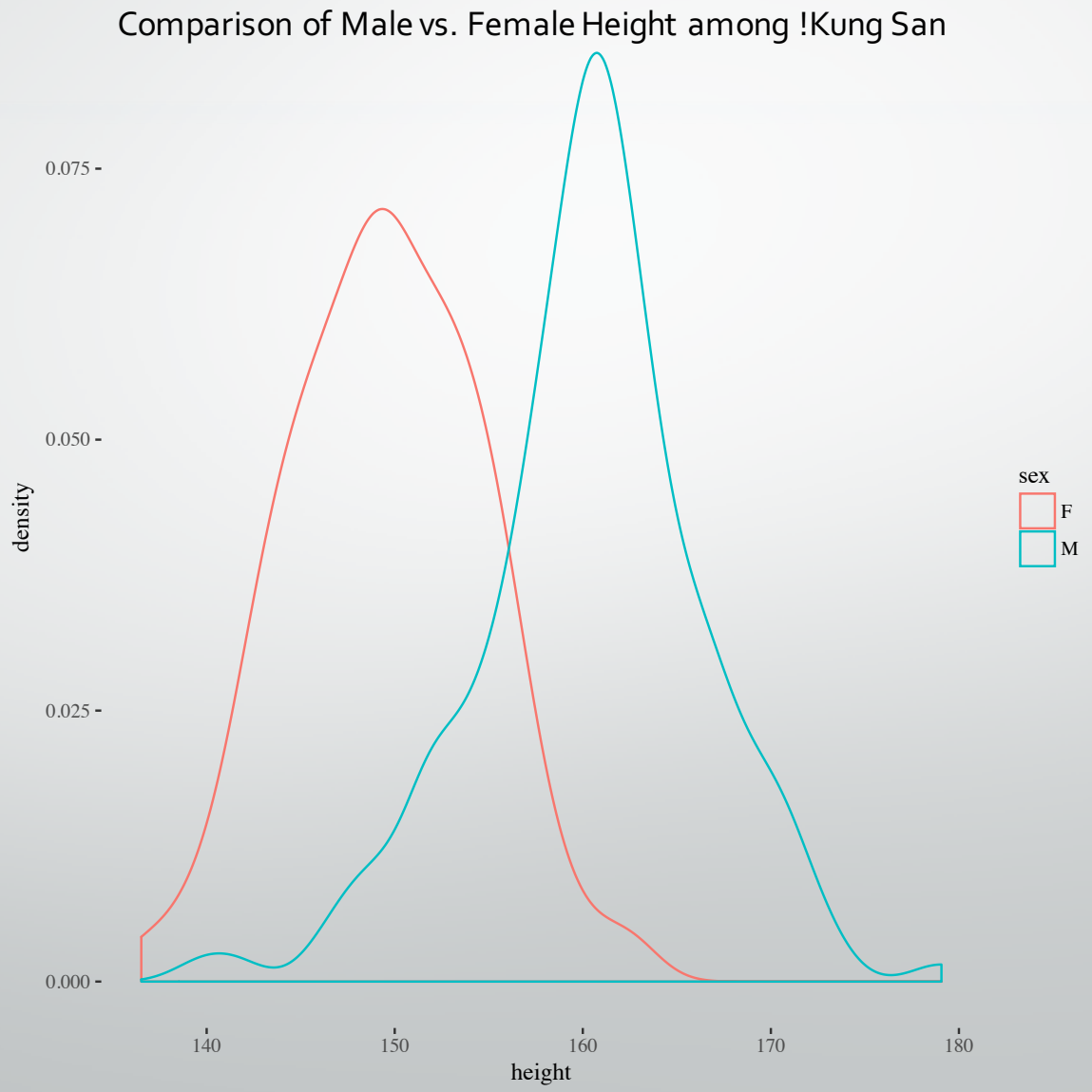
Problem | Model | Fake Data | Fit | Diagnostics | Graph fit | PPCs | Compare

- Height data collected in the 1960s on the !Kung San foraging population
- We want to better understand the population
- Explore how the measurements we have are related to each other

```
'data.frame': 352 obs. of 4 variables:
 $ height: num 152 140 137 157 145 ...
 $ weight: num 47.8 36.5 31.9 53 41.3 ...
 $ age : num 63 63 65 41 51 35 32 27 19 54 ...
 $ male : int 1 0 0 1 0 1 0 1 0 1 ...
```

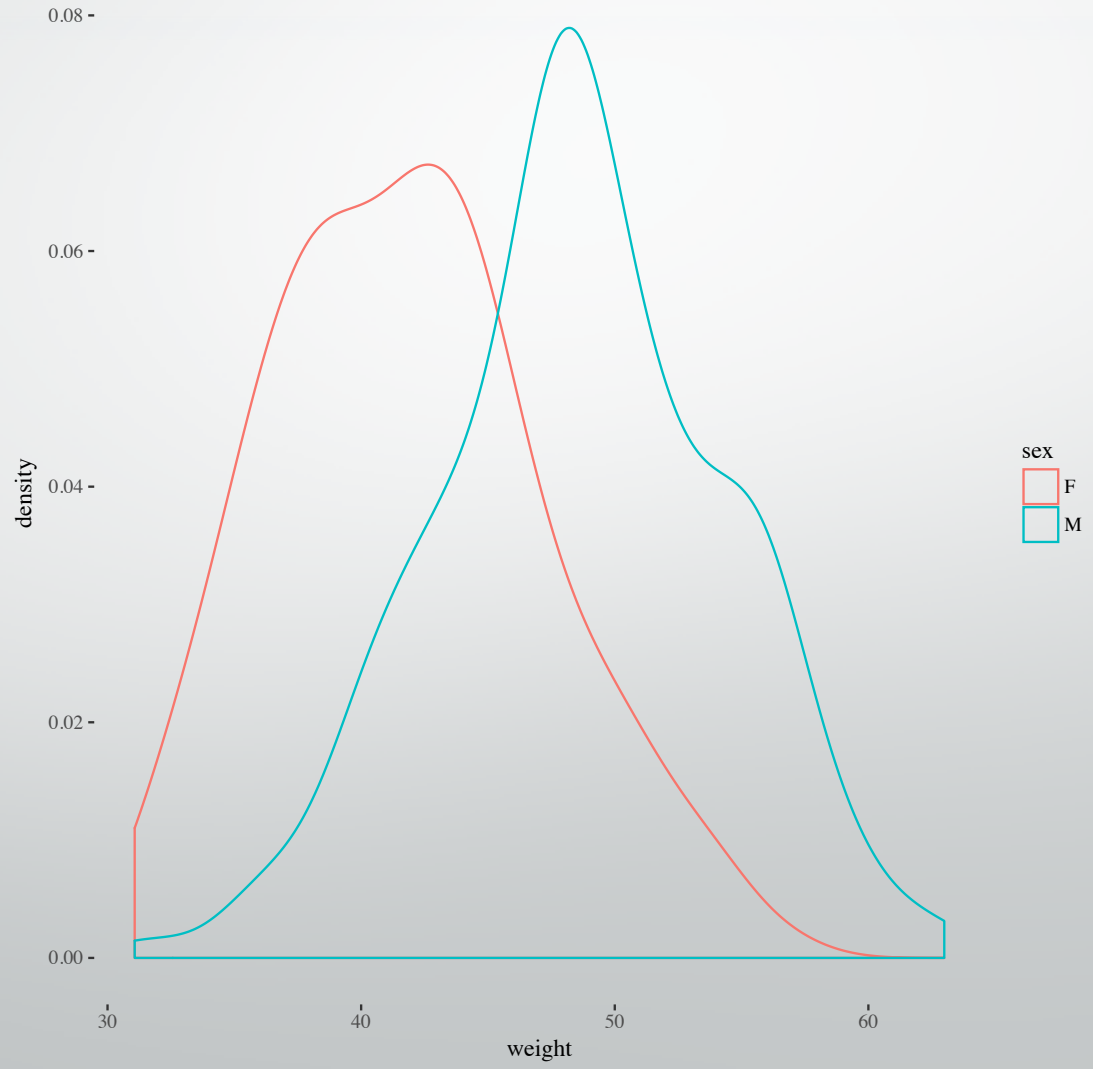


Problem | Model | Fake Data | Fit | Diagnostics | Graph fit | PPCs | Compare

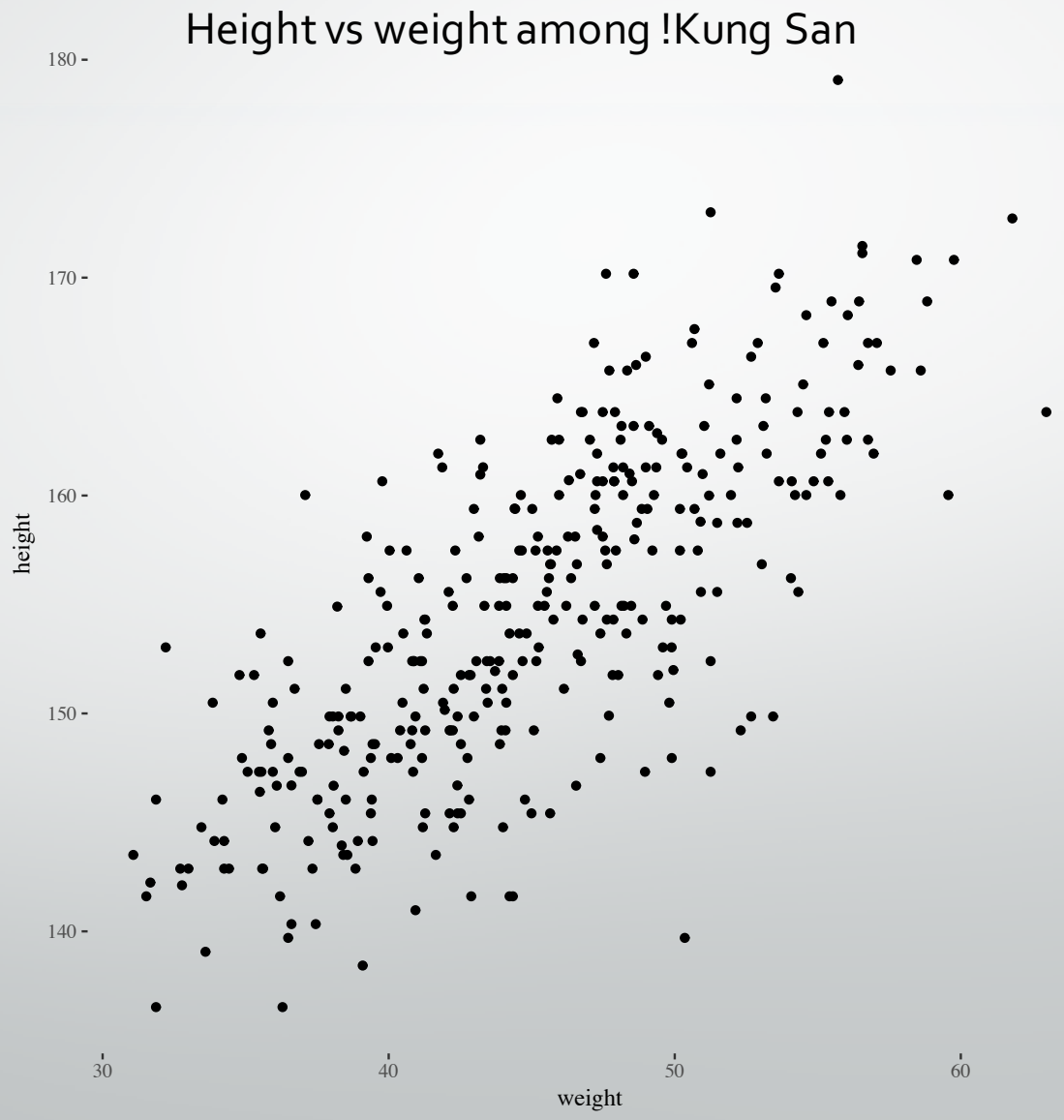


Problem | Model | Fake Data | Fit | Diagnostics | Graph fit | PPCs | Compare

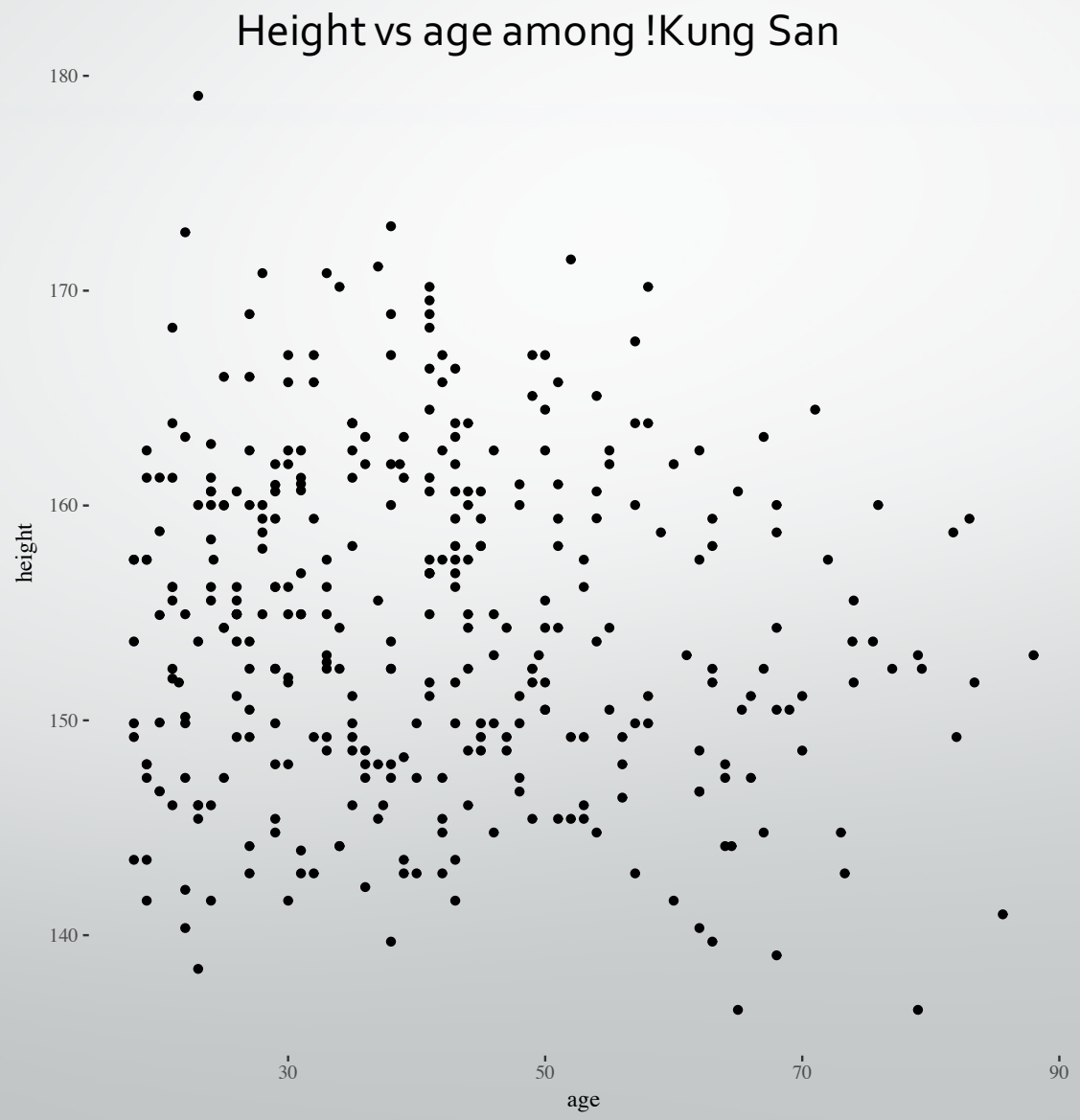
Comparison of Male vs. Female Weight among !Kung San



Problem | Model | Fake Data | Fit | Diagnostics | Graph fit | PPCs | Compare



Problem | Model | Fake Data | Fit | Diagnostics | Graph fit | PPCs | Compare



Model height as a function of weight and sex:

```
data {
  int num_people;
  vector<lower=0>[num_people] weights;
  vector<lower=0> heights[num_people];
}
```

Construct mathematical model you think generated the data:

$$\text{height} \sim \mathcal{N}(\beta * \text{weight} + \alpha, \sigma)$$

In Stan, we'd now write the parameters of this model:

```
parameters {
  real beta;
  real alpha;
  real<lower=0> sigma;
}
```

Construct mathematical model you think generated the data:

$$\text{height} \sim \mathcal{N}(\beta * \text{weight} + \alpha, \sigma)$$

And then write the likelihood:

```
model {
  heights ~ normal(beta * weights + alpha, sigma);
}
```

$$\text{height} \sim \mathcal{N}(\beta * \text{weight} + \alpha, \sigma)$$

Think about reasonable priors for your parameters:

- Beta measures the association between weight and height, in cm/kg
- Alpha is the intercept, or average height for someone with no weight (not a particularly useful number on its own)
- Sigma is the measurement variation for the population

In Stan:

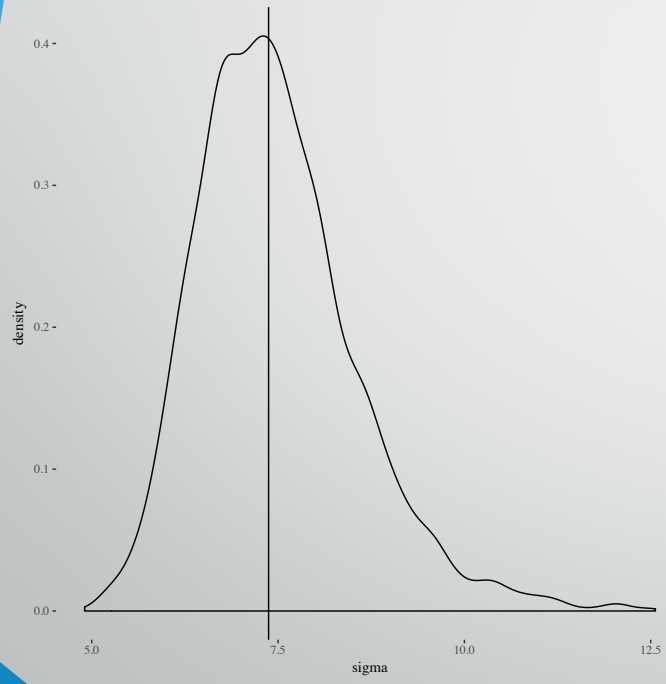
```
model {
  heights ~ normal(beta * weights + alpha, sigma);
  beta ~ normal(0, 10); // cm/kg
  alpha ~ normal(50, 50); // avg cm for 0 kg
  sigma ~ normal(0, 5); // variation from average
}
```

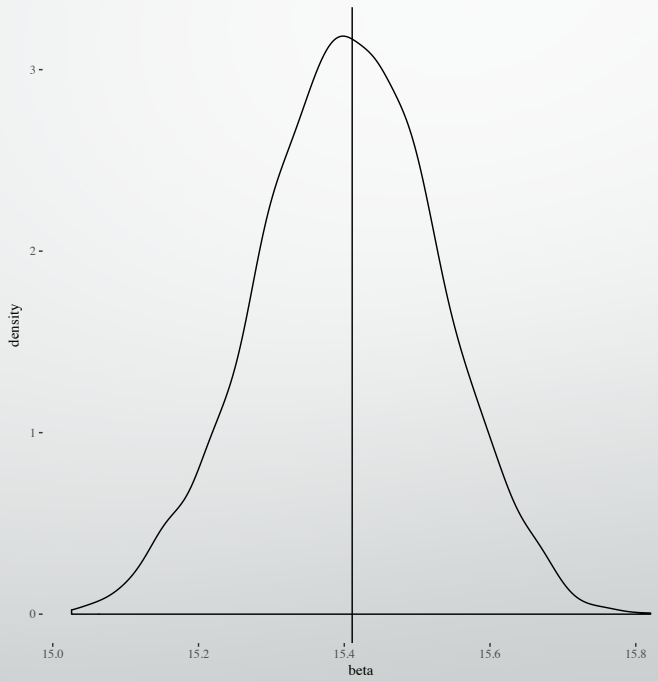
Sanity check:

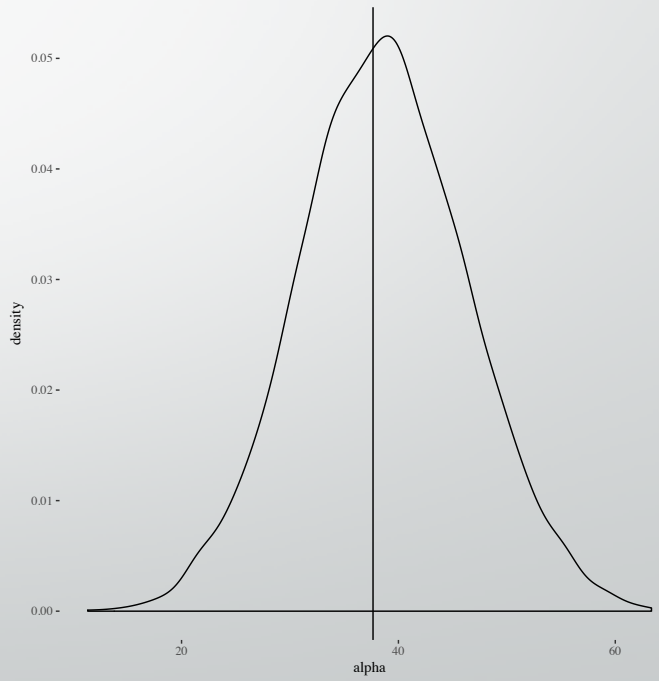
1. Draw parameter values from priors
2. Generate data based on those parameter values
3. Fit model to generated data
4. Check fit is reasonable

```
generated quantities {
  real<lower=0> heights[N];
  real beta = normal_rng(0, 10);
  real alpha = normal_rng(50, 50);
  real sigma = fabs(normal_rng(0, 5));
  for (n in 1:N)
    heights[n] = normal_rng(beta * weights[n] + alpha, sigma);
}
```

Problem | Model | **Fake Data** | Fit | Diagnostics | Graph fit | PPCs | Compare





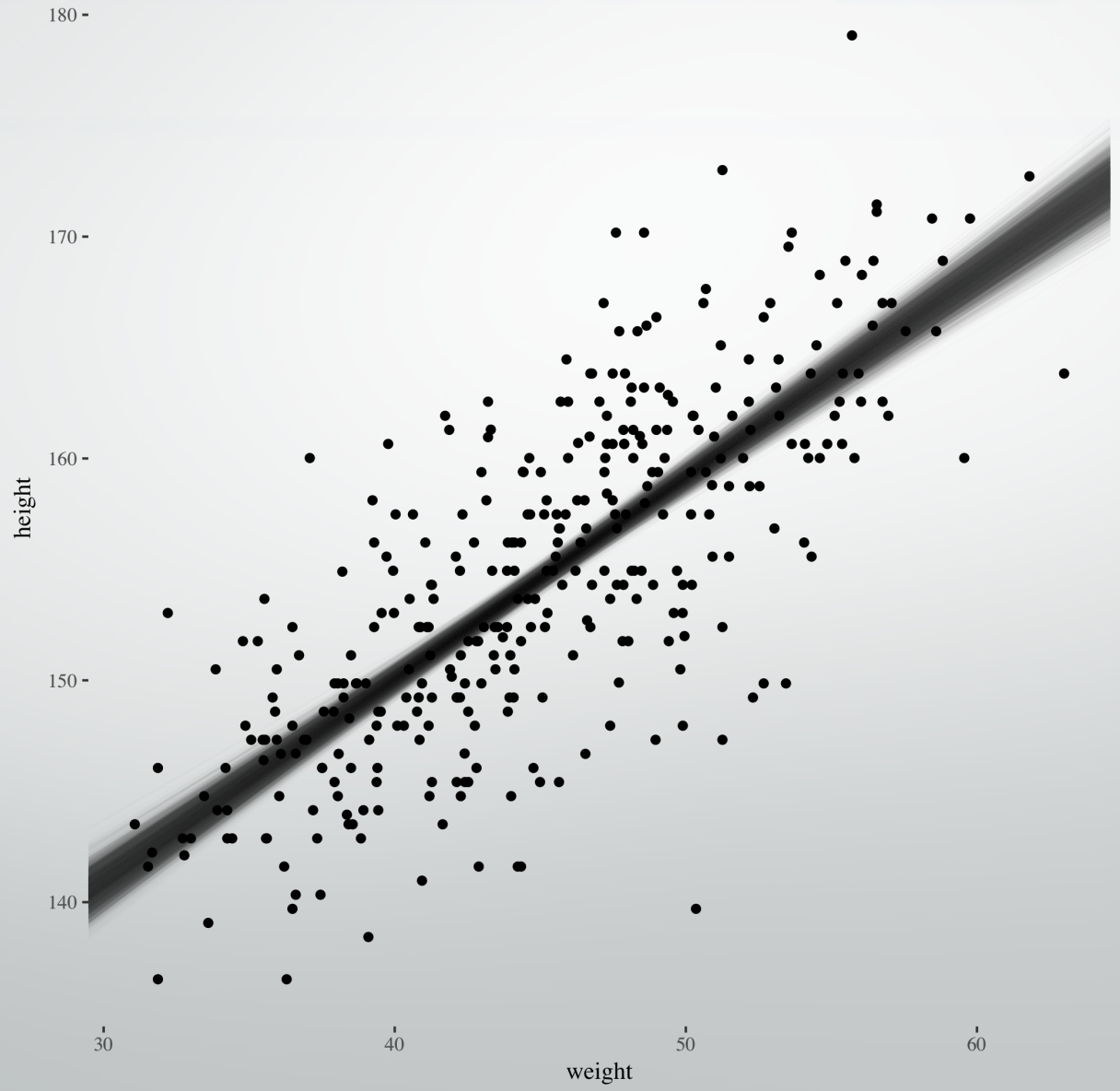


```
fit.real = sampling(heights_workflow, list(N = nrow(hdata),
  heights=hdata$height, weights=hdata$weight))
```

```
Inference for Stan model: heights_workflow.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.
       mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alpha 38.60    0.20 7.69 23.66 33.37 38.53 43.73 54.10  1427    1
beta  15.41    0.00 0.12 15.16 15.32 15.41 15.49 15.64  1432    1
sigma  7.48    0.03 1.07  5.80  6.73  7.35  8.06  9.98  1248    1
```

```
Warning messages:
1: There were 13 divergent transitions after warmup. Increasing
adapt_delta above 0.8 may help. See http://mc-
stan.org/misc/warnings.html#divergent-transitions-after-warmup
2: There were 70 transitions after warmup that exceeded the
maximum treedepth. Increase max_treedepth above 10. See http://mc-
stan.org/misc/warnings.html#maximum-treedepth-exceeded
3: There were 4 chains where the estimated Bayesian Fraction of
Missing Information was low. See http://mc-
stan.org/misc/warnings.html#bfmi-low
4: Examine the pairs() plot to diagnose sampling problems
```

Problem | Model | Fake Data | Fit | Diagnostics | **Graph fit** | PPCs | Compare



For each sample from parameter space, generate some fake data:

```
generated quantities {
  real h_ppc[N];
  for (n in 1:N)
    h_ppc[n] = normal_rng(beta * weights[n] + alpha, sigma);
}
```

Problem → Model → Fake Data → Fit → Diagnostics → Graph fit → **PPCs** → Compare

Save & Close

SHINYSTAN | DIAGNOSE | ESTIMATE | EXPLORE | MORE

NUTS (plots) | HMC/NUTS (stats) | $\hat{R}, n_{eff}, se_{mean}$ | Autocorrelation | PPcheck

Graphical posterior predictive checks

Experimental feature

Shiny Stan

Select data

Plots

- Distribution of observed data vs replications
- Distributions of test statistics
- Scatterplots
- Histograms of residuals

About

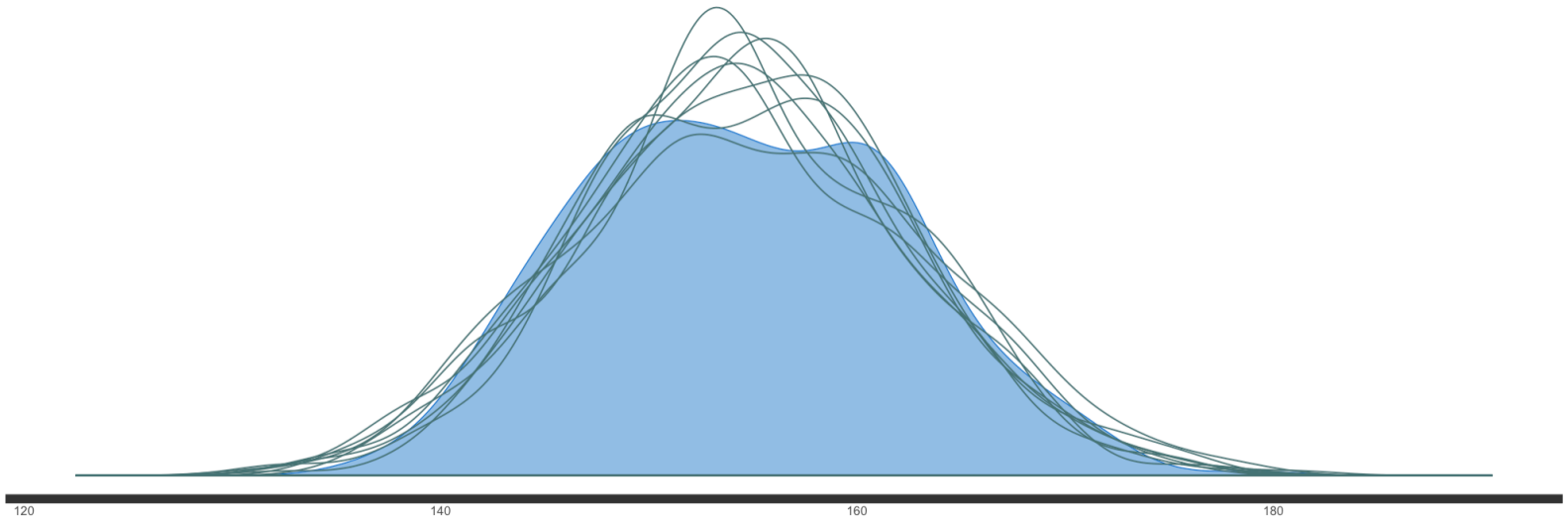
- About graphical posterior predictive checking
- Tutorial

Distributions of observed data and a random sample of replications

Show different replications

Histograms | Densities

Separate | Overlay



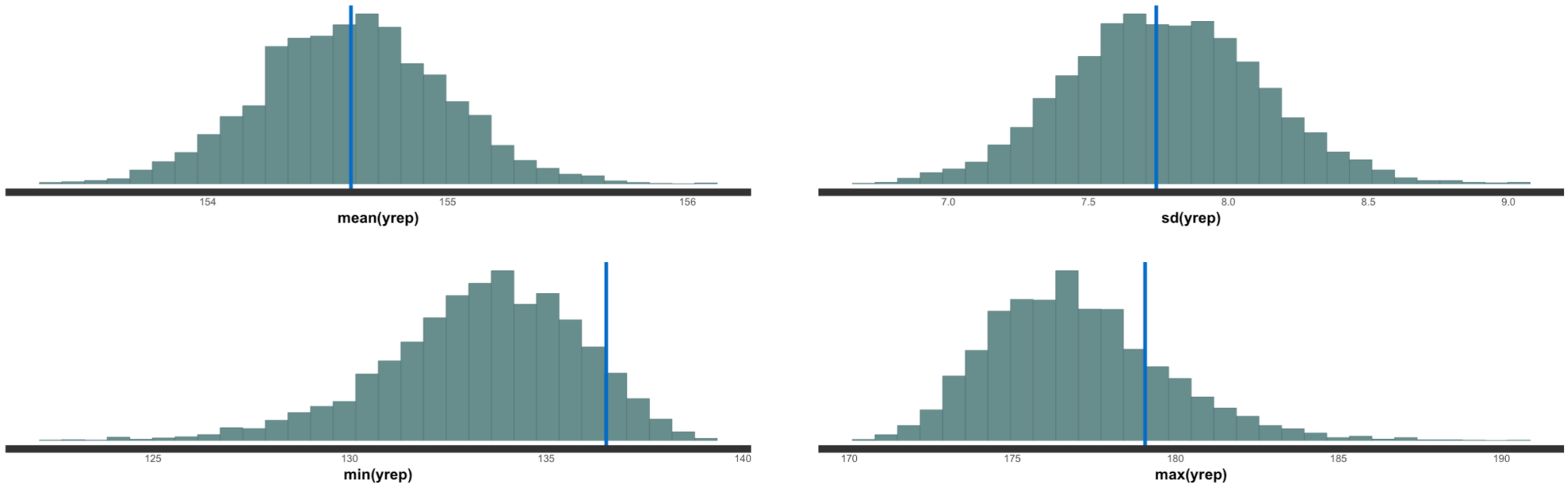
Problem | Model | Fake Data | Fit | Diagnostics | Graph fit | **PPCs** | Compare

Shiny Stan

Distributions of test statistics $T(y^{rep})$

The blue lines show $T(y)$, the value of the statistic computed from the observed data.

Histograms Densities

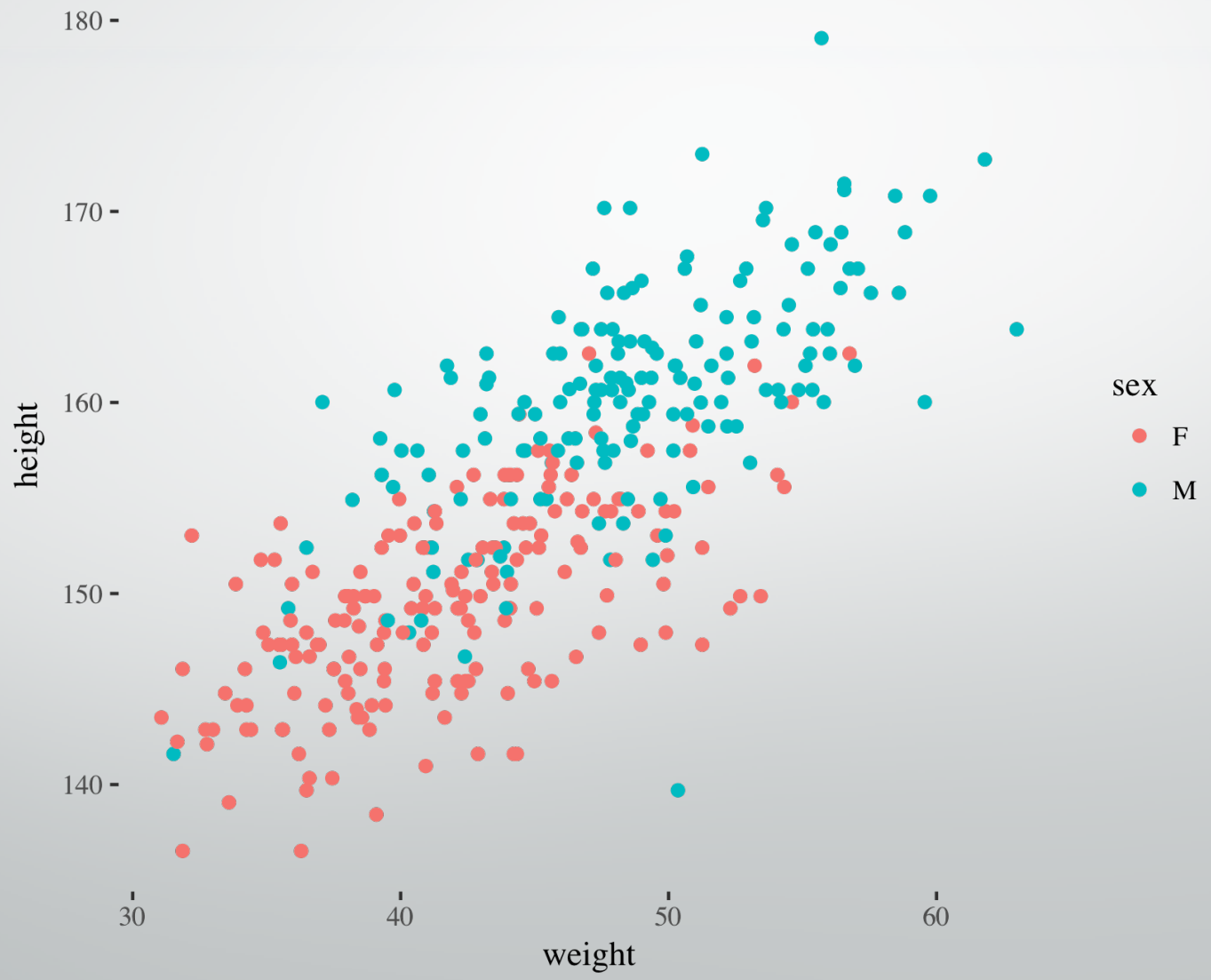


Problem → Model → Fake Data → Fit → Diagnostics → Graph fit → PPCs → **Compare**

Use these tools to compare predictive distributions and other quantities of interest among models.

Iterate!

Problem | Model | Fake Data | Fit | Diagnostics | Graph fit | PPCs | **Compare**



How do we use this to make a decision?

Our inferences present us with a posterior distribution which captures our beliefs about the world and their uncertainty.

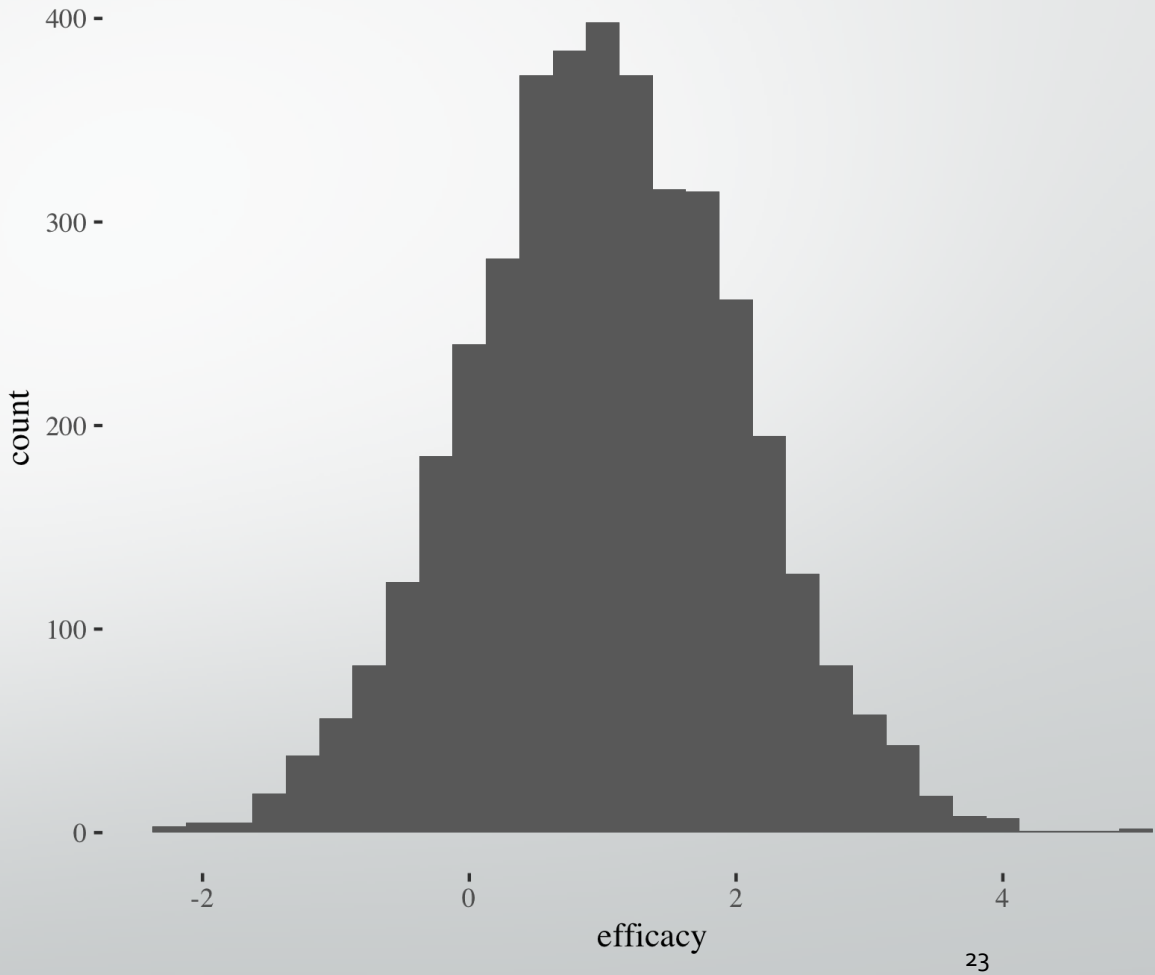
If we can encode our value system as a utility function, we can calculate the expected utility of each choice and find the one with the most utility.

Hypothetical drug company decision

We’ve done a preliminary study on a new drug and measured its efficacy.

Can release the drug if its effect is greater than 0.3, for which it will make $1e9 per unit of effect above that.

Costs $1e7 to develop the drug to that point.



Hypothetical drug company decision