Supplemental Resource: Brain and Cognitive Sciences
Statistics & Visualization for Data Analysis & Inference
January (IAP) 2009

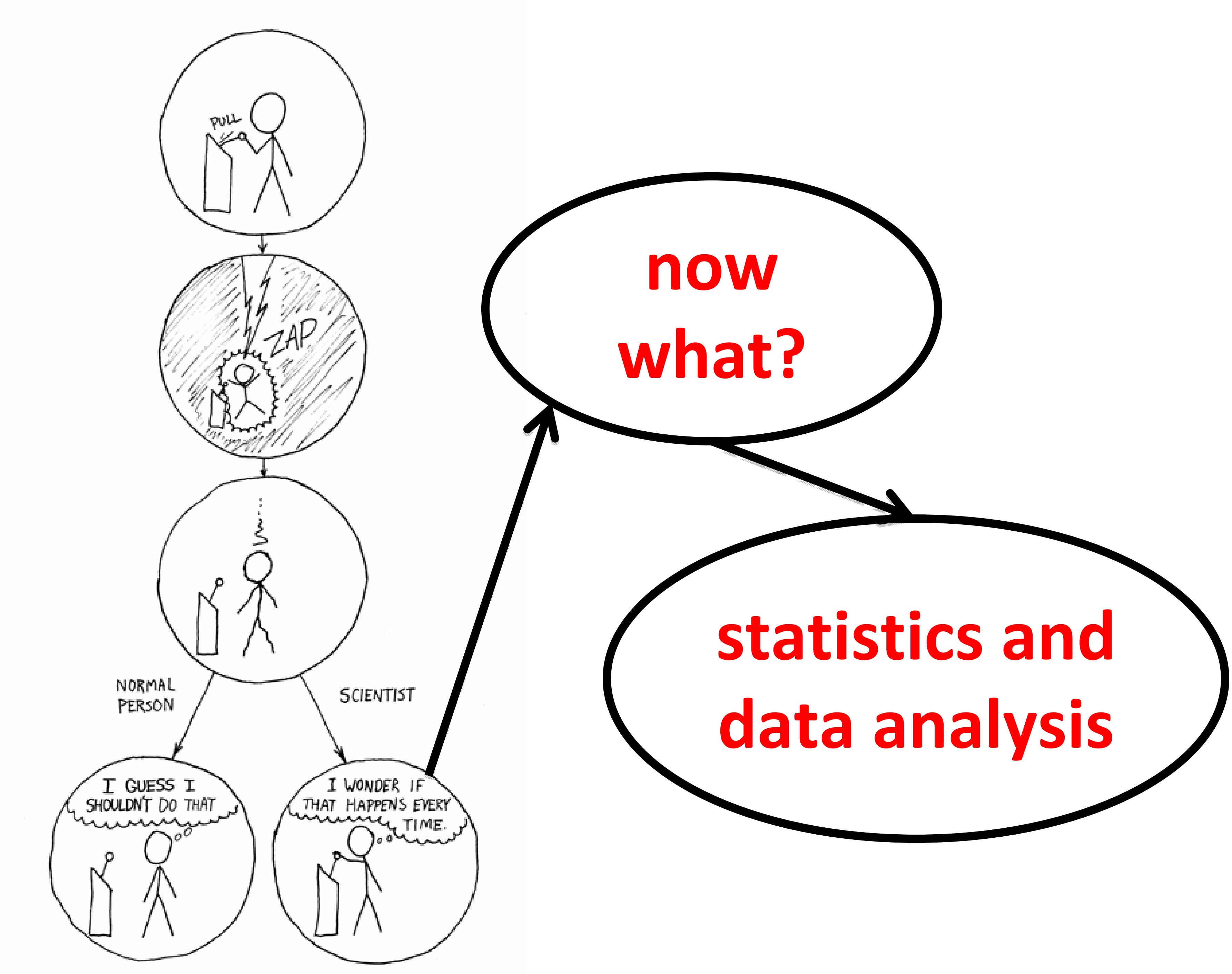
Statistics and Visualization for Data Analysis

Mike Frank & Ed Vul

IAP 2009

Today’s agenda

- Beliefs about the generative process
- Inverse probability and Bayes Theorem
- Numerically approximating inverse probability
- What might we believe about the generative process?
    - Gaussian
    - Log-Normal
    - Uniform
    - Beta
    - Binomial
    - Exponential
    - Von Mises
    - Poisson
    - Mixture

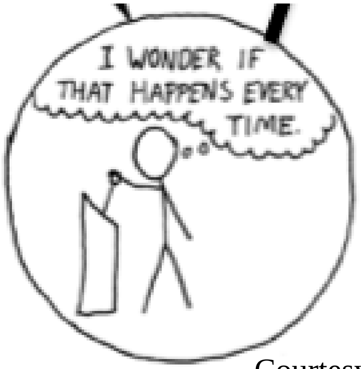

PULL
ZAP
NORMAL PERSON
SCIENTIST
I GUESS I SHOULDN'T DO THAT
I WONDER IF THAT HAPPENS EVERY TIME.

now what?

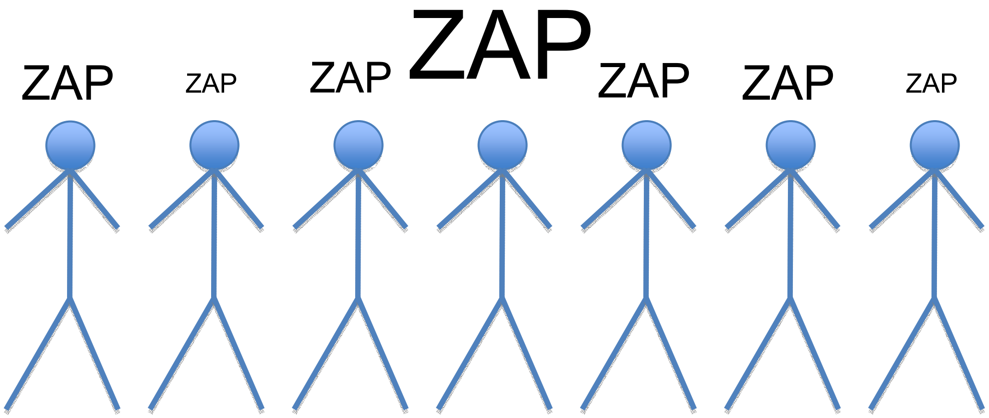
statistics and data analysis

Courtesy of xkcd.org

I wonder what are the underlying properties?

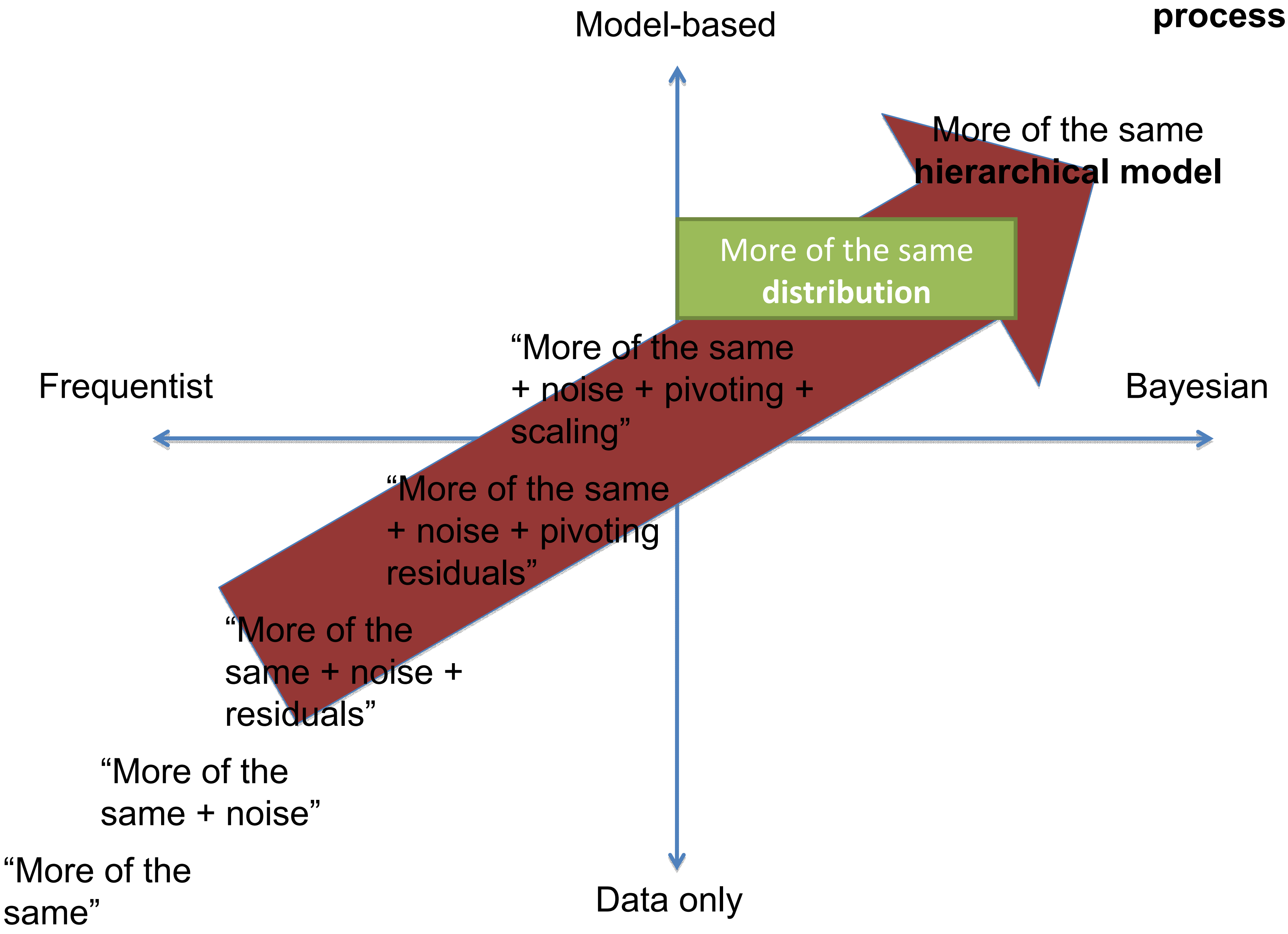


Courtesy of xkcd.org



now what?

Our class trajectory



Model-based

Data only

Frequentist

Bayesian

"More of the same"

"More of the same + noise"

"More of the same + noise + residuals"

"More of the same + noise + pivoting residuals"

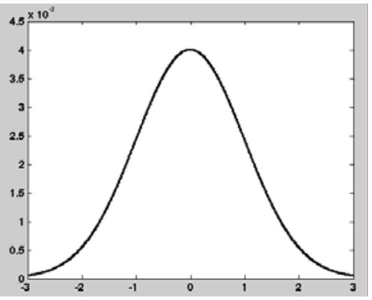
"More of the same + noise + pivoting + scaling"

More of the same **distribution**

More of the same **hierarchical model**

More of the same **generative process**

New assumption: a distribution



- The data I observed came from some process that produces different observations according to some **parametric distribution**
  - Or I just want to summarize data as such.
- Parametric distribution
  - Distribution: a function over possible outcomes, that assigns probability to each one.
  - Parametric: it has some parameters that limit the way this function might look.
- Now: We want to **figure out the parameters**

Figuring out the parameters

- Frequentist:
  - Define an “unbiased estimator”:
    - A measure on the data that estimates the parameter of interest, with no systematic bias (e.g., mean)
  - Given uncertainty about possible data-sets, we have uncertainty about the values from estimators.
  - Therefore we have “uncertainty” about the parameters
    - in so far as our estimators will come out slightly different on possible sets of data
    - Resampling methods + “estimators” as measures on data allow us to figure out parameters this way

Figuring out the parameters

- Bayesian
  - – What should I believe the parameter value is?
    - Ahh, that’s straight-forward.
  - – Use “inverse probability”

Inverse probability and Bayes Theorem

- Forward probability: the probability that a distribution with this parameter value would generate a particular data-set.
  P(D|H) (the "Likelihood")

- Inverse probability: the probability of this parameter, given that I observed a particular data-set
  P(H|D) (the "Posterior")

Inverse probability and Bayes Theorem

$$\underset{\text{Posterior}}{P(H|D)} = \underset{\text{Likelihood}}{P(D|H)}\,\underset{\text{Prior}}{P(H)}/\underset{\substack{\text{Probability of}\\ \text{all the alternatives}}}{P(D)}$$

- The probability that a parameter has a particular value (“H”ypothesis) reflects
    - Our prior belief (probability) about parameter values
    - The probability that a distribution with this parameter value produced our data
    - Normalized by this stuff computed for all alternative parameter values

Crippling Bayes, as is customary

$$P(H|D) = P(D|H)P(H)/P(D)$$

Posterior Likelihood Prior Probability of all the alternatives

- We want to plead ignorance about possible parameter values, so we will say our prior assigns each of them equal probability.
  - – Ignore that this is…
    - …not actually least informative
    - …not actually a proper prior
- This means we can do away with P(H), and our **posterior will be proportional to the likelihood**

The role of the prior

- As scientists, we have them, reflecting everything else we know.

- As statisticians, we “ignore” them to let the data speak.
    - And even if so, if we were sensible, we wouldn’t treat them as uniform (but ignore that)
    - But not in hierarchical statistical models

Inverting probability: the Likelihood

$$\underset{\text{Posterior}}{P(H|D)} = \underset{\text{Likelihood}}{P(D|H)} \, / \underset{\substack{\text{Likelihood under} \\ \text{all the alternatives}}}{P(D)}$$

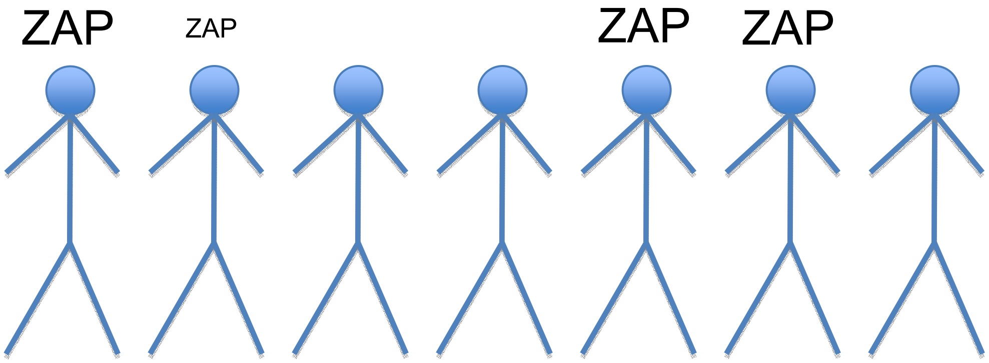
- So it seems we just need to figure out the “Likelihood” for every possible parameter value
  - That is: for each parameter value, figure out the probability that the data came from a distribution with that parameter value.

- How do we do that?

Computing likelihood

- There are various complicated ways of doing this using math.
- Lets avoid those, and do this approximately: numerically, capitalizing on the brute force of our computers.
  - Consider a bunch of possible parameter values.
  - Evaluate the likelihood under each one.
  - And call it a day.

What is the probability of a zap?



- What do we think the parametric distribution is?
- “Binomial”
  - – This function assigns a probability to a particular number of observed zaps given a particular number of total observations.

Binomial distribution

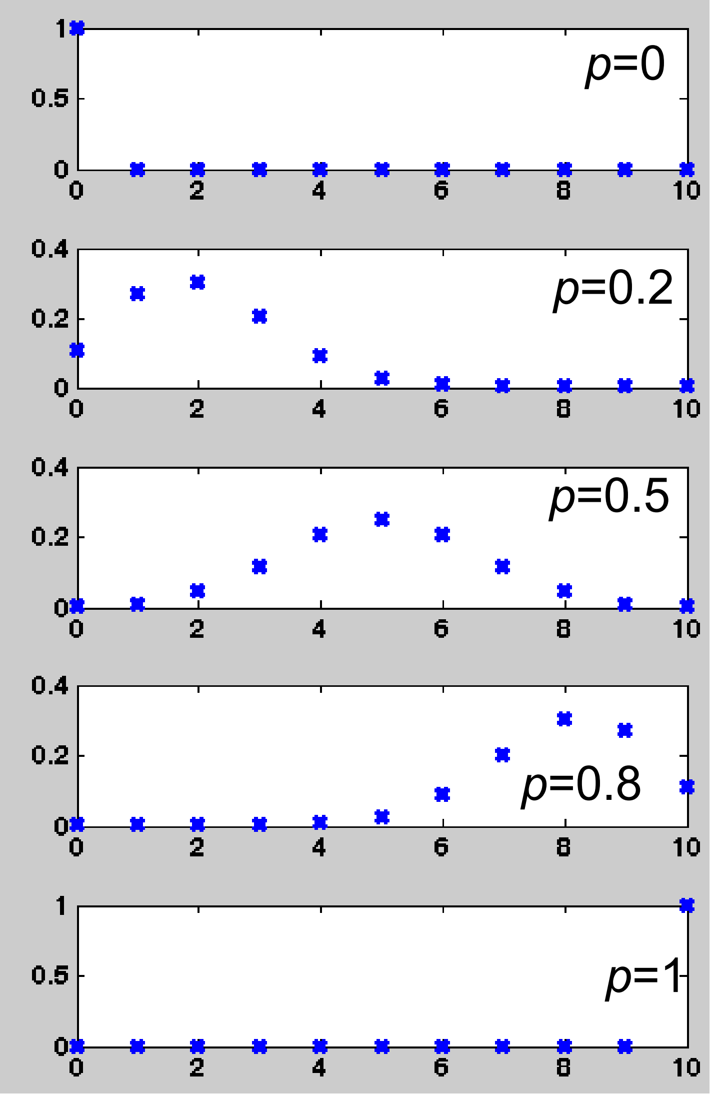
$$\binom{n}{k} p^k (1-p)^{n-k}$$

- “Ugh: Math!”
  - Sorry – I feel guilty describing a parametric distribution without writing down the function.
- “Ok, what’s it do?”
  - Assigns a probability to any possible number of observed zaps *k*
  - Given the number of observations *n*
  - Modulated by a parameter *p*

Binomial distribution

$$\binom{n}{k} p^k (1-p)^{n-k}$$

- "What does *p* do?"
  - – Changes the probability that we will observe one or another number *k* of zaps given *n* observations
- "What does *p* mean?"
  - – Probability that one observation will be a zap.



What is the probability of a zap?

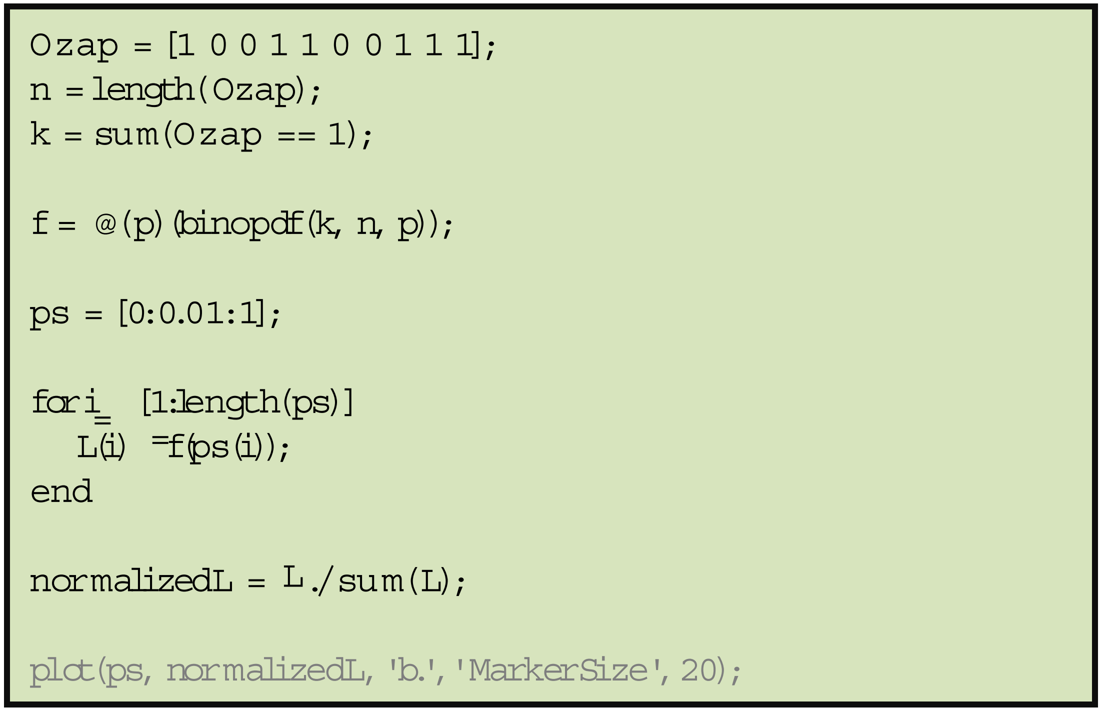
- Ok, so some “binomial” process generates this zap data. And I want to know what I should believe about this *p* parameter of the “binomial distribution”.
- …And I can figure this out somehow, by computing the “likelihood” of the data for every possible value of *p*
- …And I don’t really need to do it for *every* parameter, just a bunch.

Introducing: the Grid Search

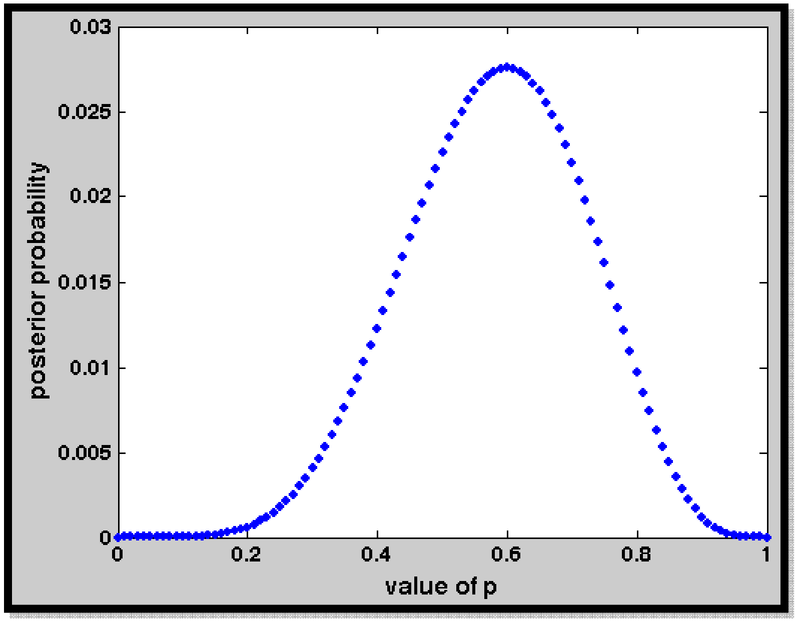
- Choose a reasonable range of plausible parameter values.
- Tessellate this range.
- Compute likelihood* for each point in the range.
  - Treat this as our approximation of the “likelihood function”
- Normalize these possible values.
- Treat that as our approximation of the “posterior”

What is the probability of a zap?

“What is the ‘posterior probability’ of particular value of *p* given my data?”

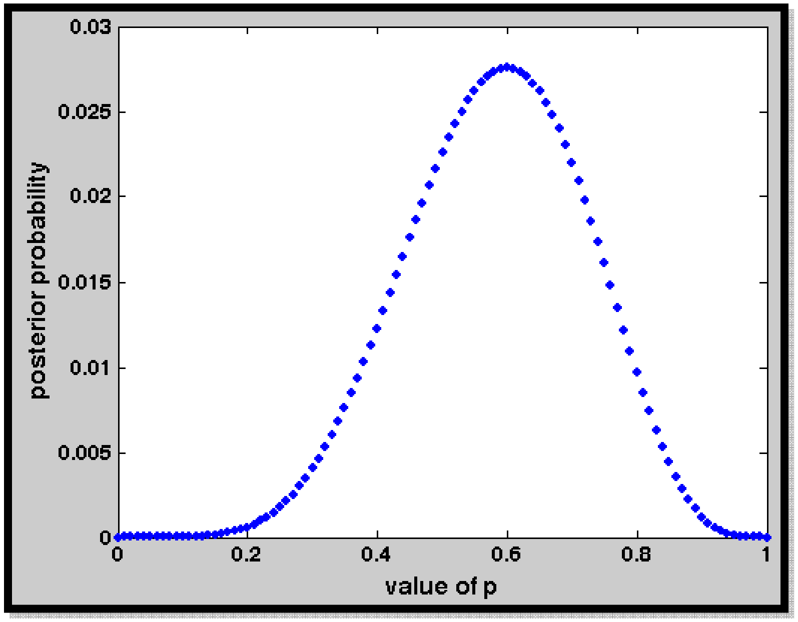


```
Ozap = [1 0 0 1 1 0 0 1 1 1];
n = length(Ozap);
k = sum(Ozap == 1);

f = @(p) (binopdf(k, n, p));

ps = [0:0.01:1];

for i = [1:length(ps)]
   L(i) = f(ps(i));
end

normalizedL = L./sum(L);

plot(ps, normalizedL, 'b.', 'MarkerSize', 20);
```

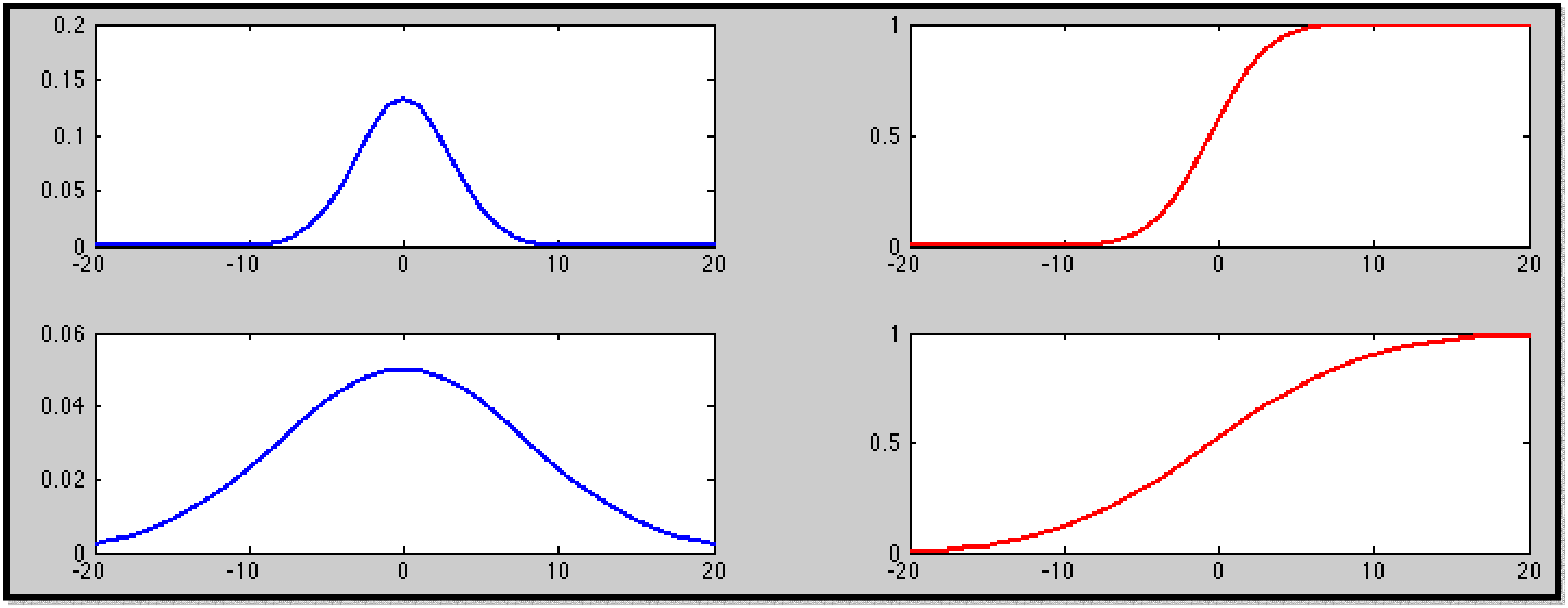


Again: Something more concise?

- Why not make a confidence interval for this?
- But how?

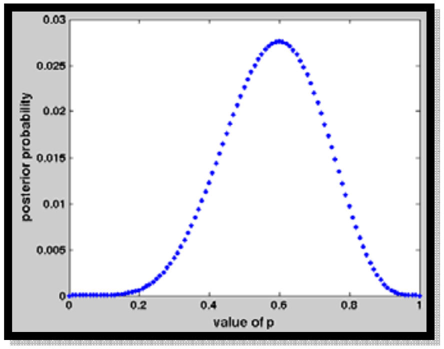


Cumulative density functions

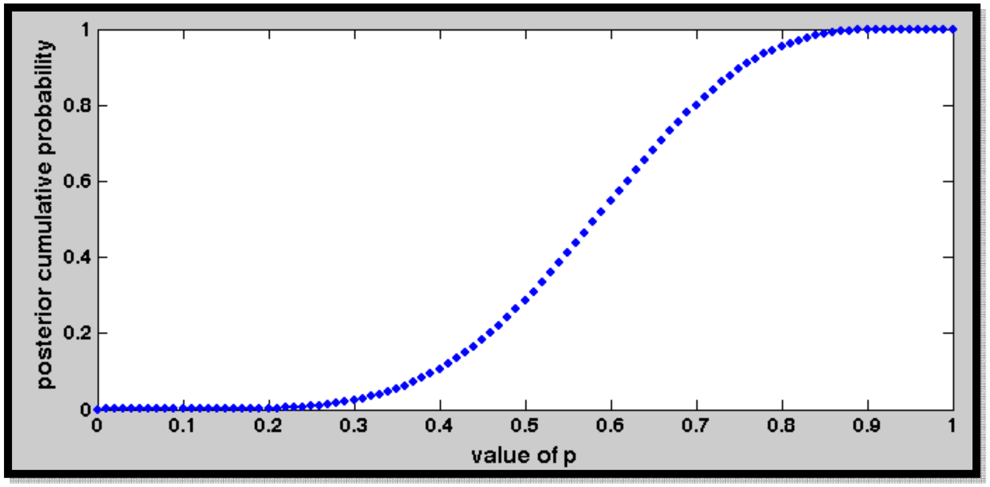


- Integral of f(x) from lower bound to x
- For each *x*, sum of all the probability that occurred at lower values
  - E.g., if a bulldozer were moving all the probability, how much probability would be in the bulldozer at this point

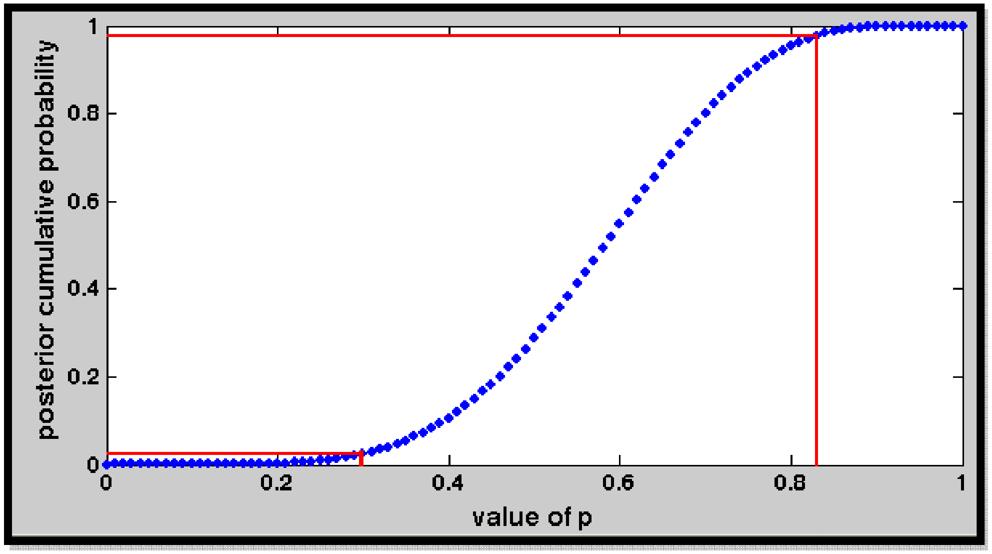
Confidence Intervals from Grids



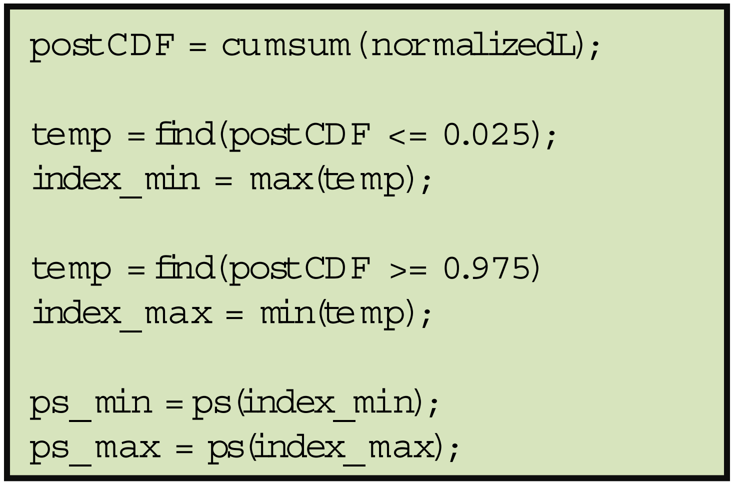
- Compute “cumulative probability” for each grid-point
  - Sum of probabilities from smallest grid point to here



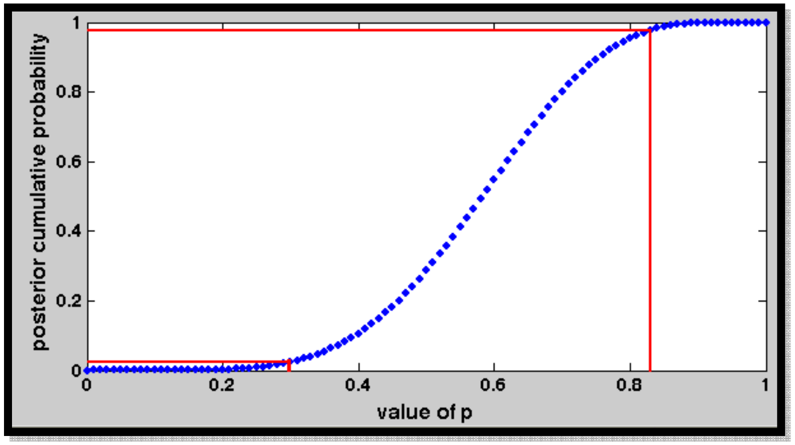
- Find grid points that are just inside the min and max percentiles of confidence interval

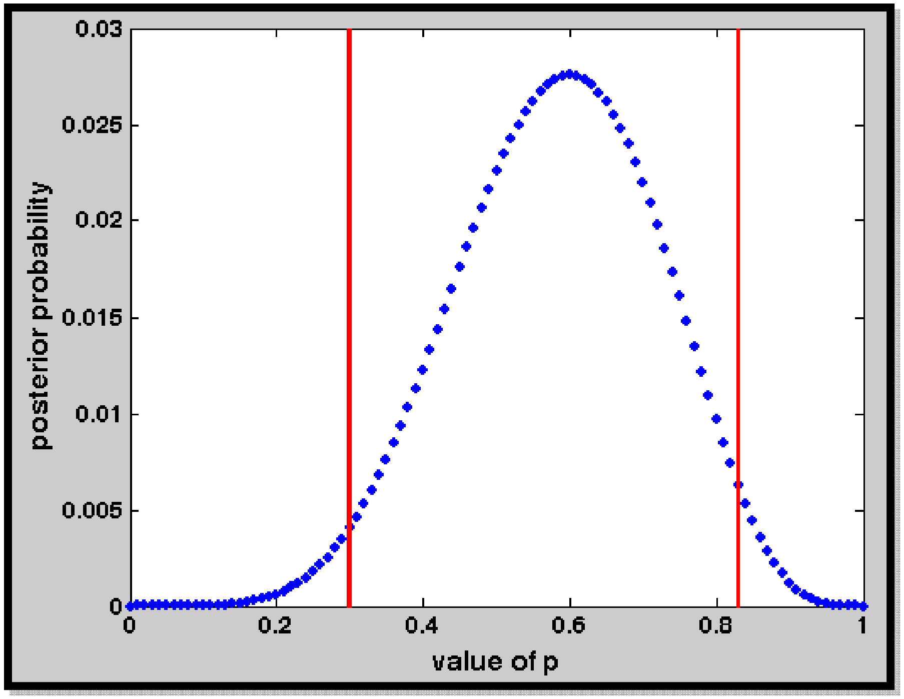


Confidence interval on probability of zap.



```
postCDF = cumsum(normalizedL);

temp = find(postCDF <= 0.025);
index_min = max(temp);

temp = find(postCDF >= 0.975)
index_max = min(temp);

ps_min = ps(index_min);
ps_max = ps(index_max);
```





The value of *p* is between 0.3 and 0.83
With 95% confidence.

Grid search limitations

- Can be slow (but so it goes)
- Choice of grid min, max, tessellation density
  (If it looks bad, try again.)
- Doesn't allow *exact* confidence intervals
  (If tessellation is fine enough, it doesn't matter)
- Doesn't allow to find "maximum likelihood" point
  (Try finer tessellation around max… you can always say max is between A and B with 100% confidence)
- If likelihood function is not smooth, has multiple modes, or is otherwise weird, easy to have a wrong approximation
  (Beware! But it won't matter for today's cases)

What distributions might I believe in?

- What are some possible distributions, and might I believe one or another describes the process generating my data?
- Considerations:
  - What is the “support”?
    - What observations are at all possible?
    - What do my data look like?
    - -Infinity to +Infinity (e.g., differences)
    - 0 to +Infinity (e.g., response times)
    - A to B
    - Integers

What could distributions might I believe in?

- What are some possible distributions, and might I believe one or another describes the process generating my data?
- Considerations:
  - What is the process like?
    - Perturbation of a value by many little factors, each equally likely to perturb up as down (with equal mag.)
    - Sum of a varying number of positive values
    - A combination of several (different) processes
    - Popping bubbles.
    - Etc.

What distributions might I believe in?

- What are some possible distributions, and might I believe one or another describes the process generating my data?
- Considerations:
  - Garbage in garbage out

The Gaussian (Normal) Distribution

$$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

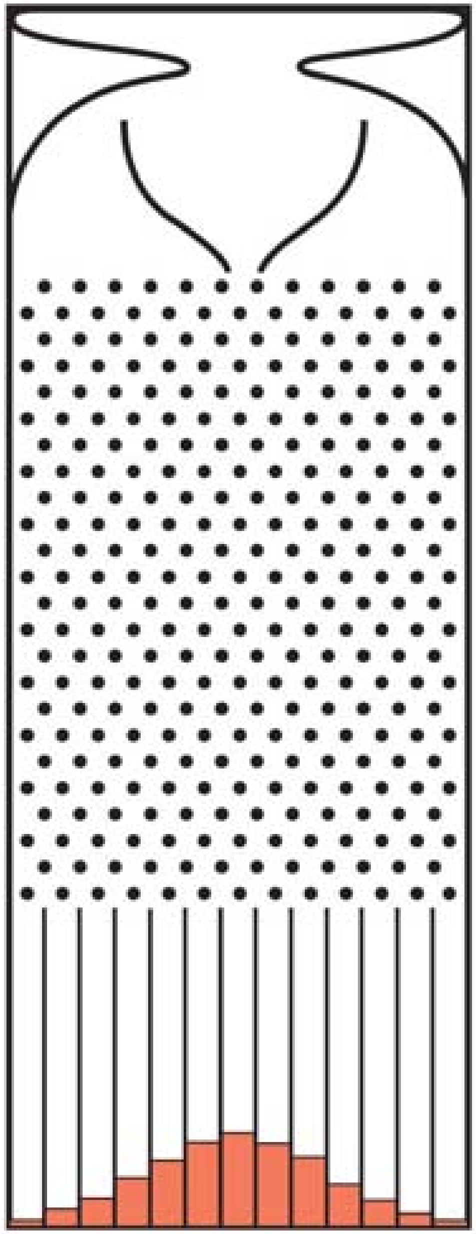
- What’s it do?
  - Assigns a probability to any possible observation: *x* between –Inf and +Inf
  - Given a particular ‘location’ parameter μ
  - And a particular ‘scale’ parameter σ

The Gaussian (Normal) Distribution

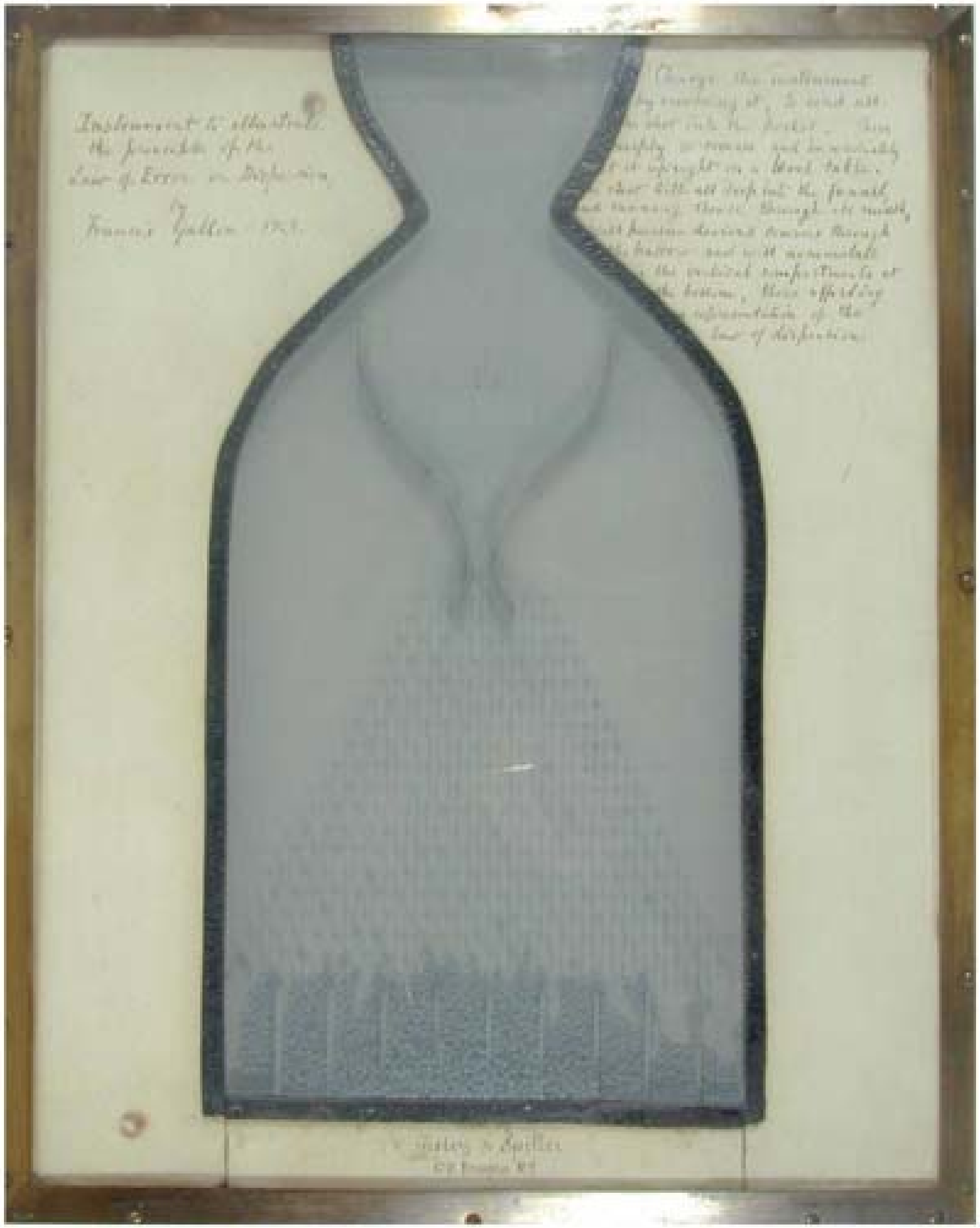
$$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- What’s it do?
  - Assigns a probability to any possible observation: *x* between –Inf and +Inf
  - Given a particular ‘location’ parameter μ
  - And a particular ‘scale’ parameter σ
- What sort of process?
  - Sum of many little factors equally likely to err positively or negatively (with eq. mag, finite var.)
  - The result of the law of large numbers

Galton's Quincunx



Courtesy of Macmillan. Used with permission.

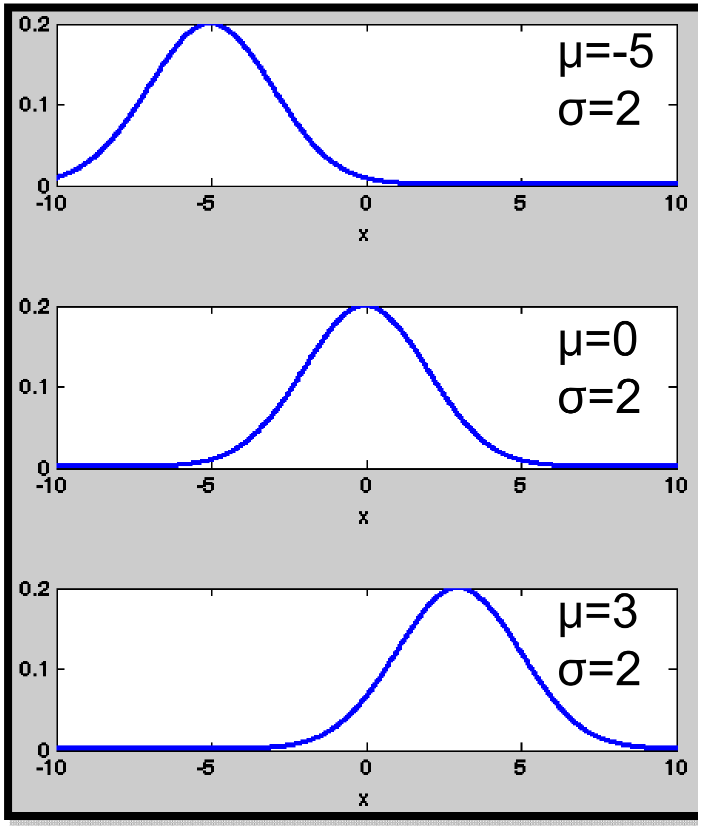


Courtesy of Galton Archives at University College London. Used with permission.

The Gaussian (Normal) Distribution

$$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

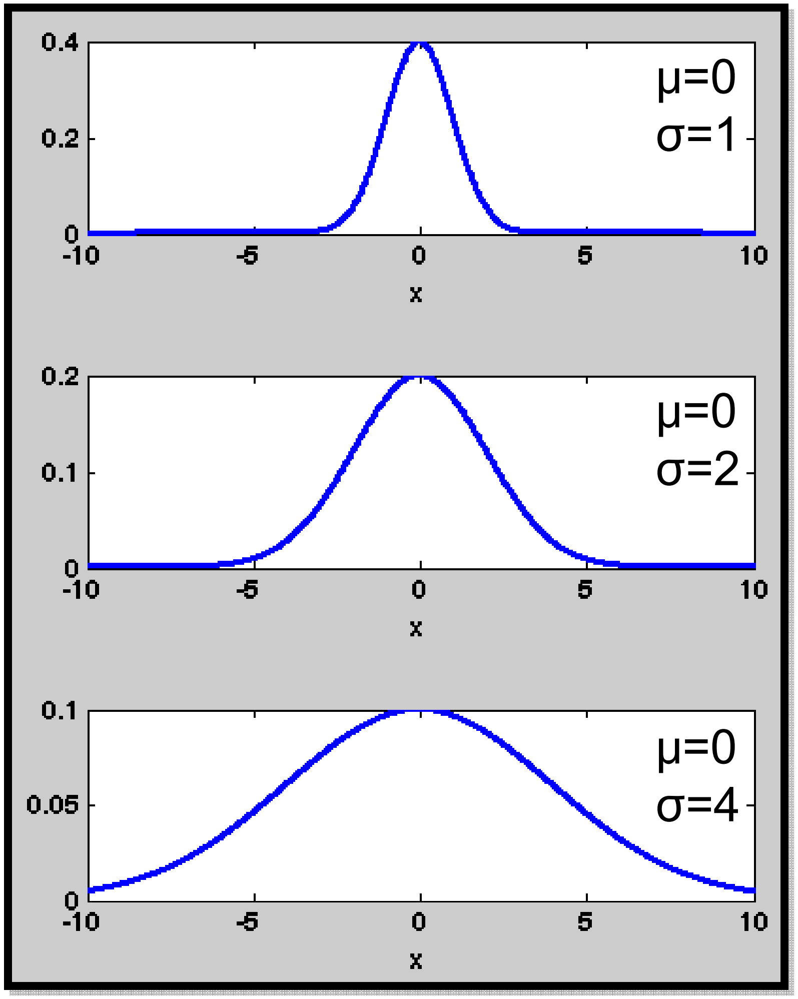
- What does ‘location’ (μ) do?
  - Determines which part of –Inf to +Inf is most likely



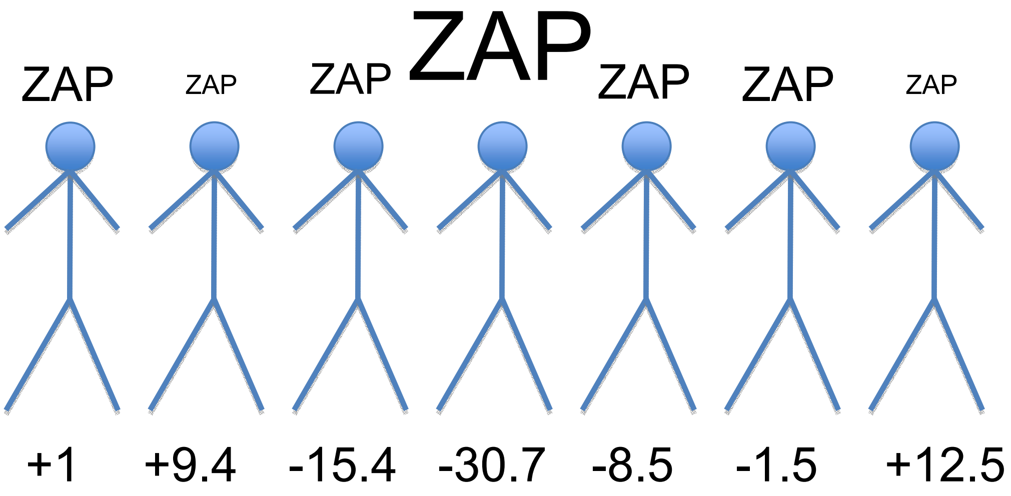
The Gaussian (Normal) Distribution

$$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- What does ‘scale’ (σ) do?
  - – Determines the “width” of the distribution.



Hedonic value of a zap.

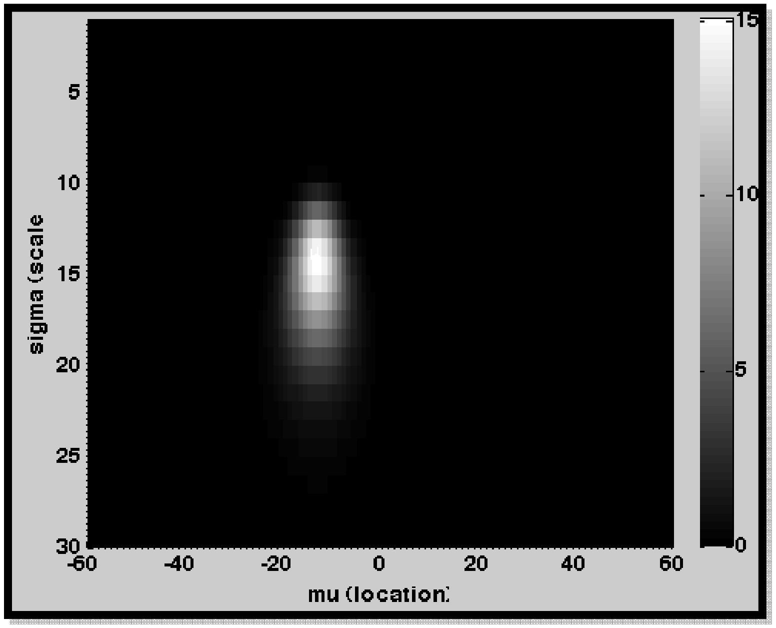


```
h_z = [ -12.4050
        0.9348
       -13.1701
       -15.3391
       -25.6529
       -9.8782
       -32.7065
       -3.9995
        8.7299
       -23.6849
       -1.9880
        3.5560
       -36.3122
       -34.1735
       -6.0039];
```

- Hedonic value may be +,-, real valued
- Arguably the sum of many little processes
- Let’s say its Gaussian

Hedonic value of zaps.

```
ms = [-60:1:60];
ss = [1:1:30];

for i = [1:length(ms)]
  for j = [1:length(ss)]
    L_hz = normpdf(h_z, ms(i), ss(j));
    ll_hz = log10(L_hz);
    LL(i,j) = sum(ll_hz);
  end
end

LL = LL + max(LL(:));
L = 10.^LL;
normL = L ./ sum(L(:));
```



Some trickery! Useful things built into this:

Probability of two independent events A and B [P(A&B)] = P(A)*P(B)

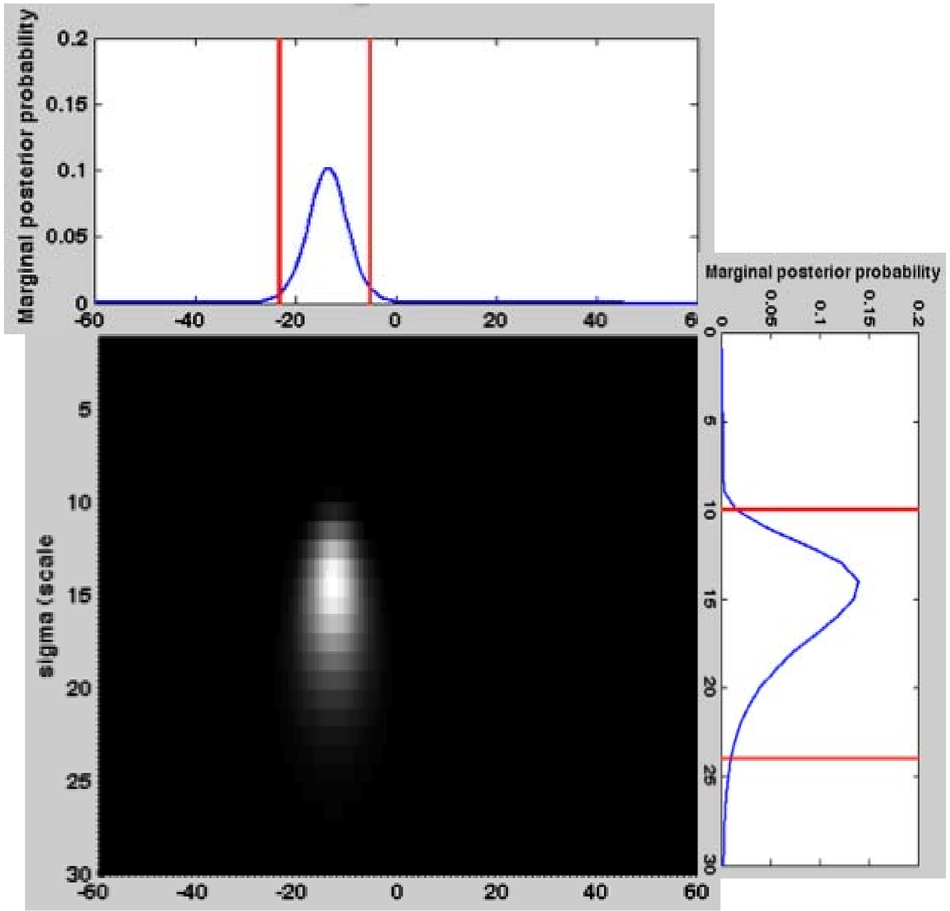
Multiplication is equivalent to the addition of logarithms

Using log likelihood prevents ‘under flow’ – numbers too small for machine precision

Taking out max log likelihood is scaling, makes no difference

Oy! A heat map? What am I to do with that?

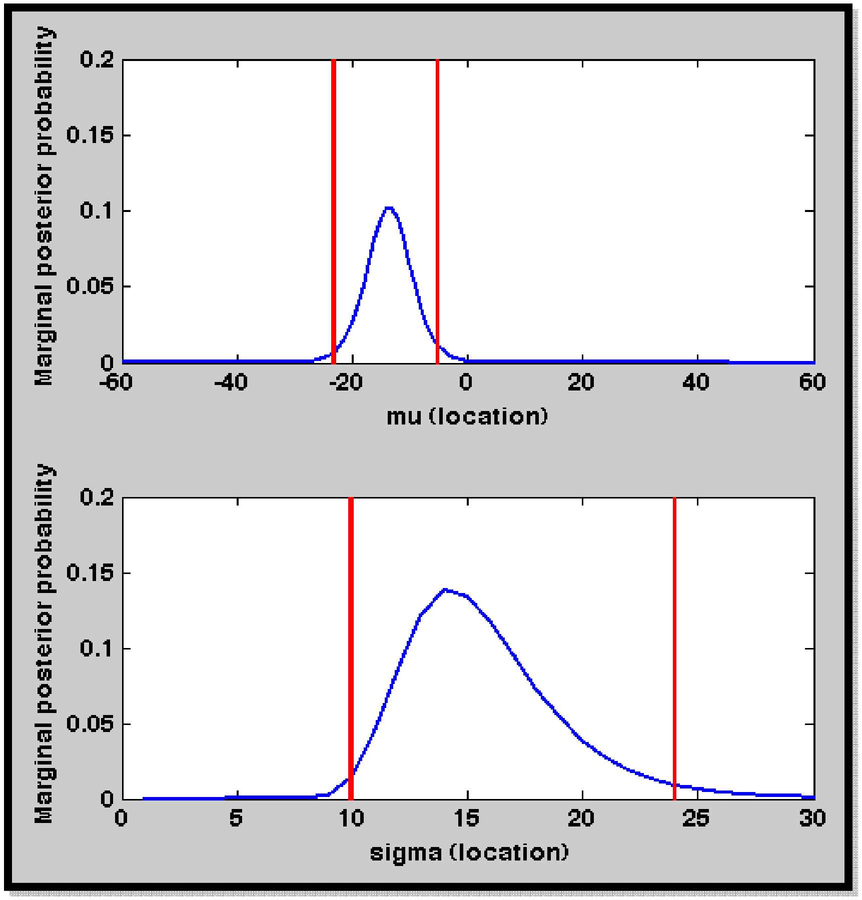
- Marginalize!
  - Sum probability over all but one dimension to compute “marginal probability” of that dimension.
- We lose dependencies, but usually that’s fine.



```
normL_m_marg = sum(normL, 2);
normL_s_marg = sum(normL, 1)';
```

Oy! A heat map? What am I to do with that?

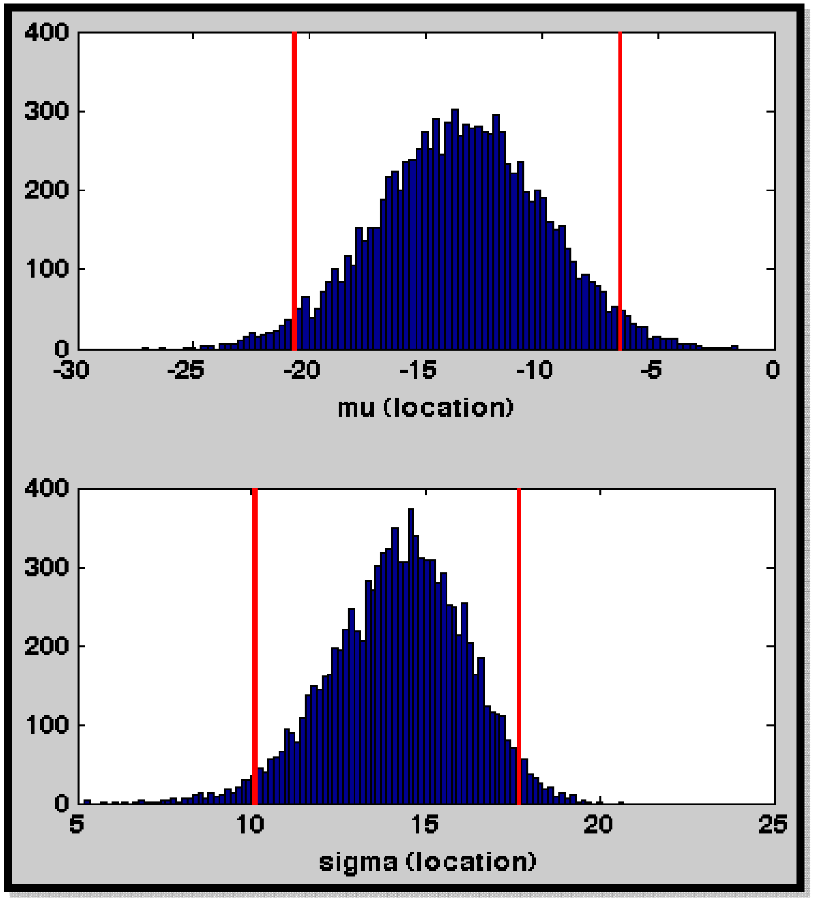
- Marginalize!
  - Sum probability over all but one dimension to compute “marginal probability” of that dimension.
- We lose dependencies, but usually that’s fine.



```
normL_m_marg = sum(normL, 2);
normL_s_marg = sum(normL, 1)';
```

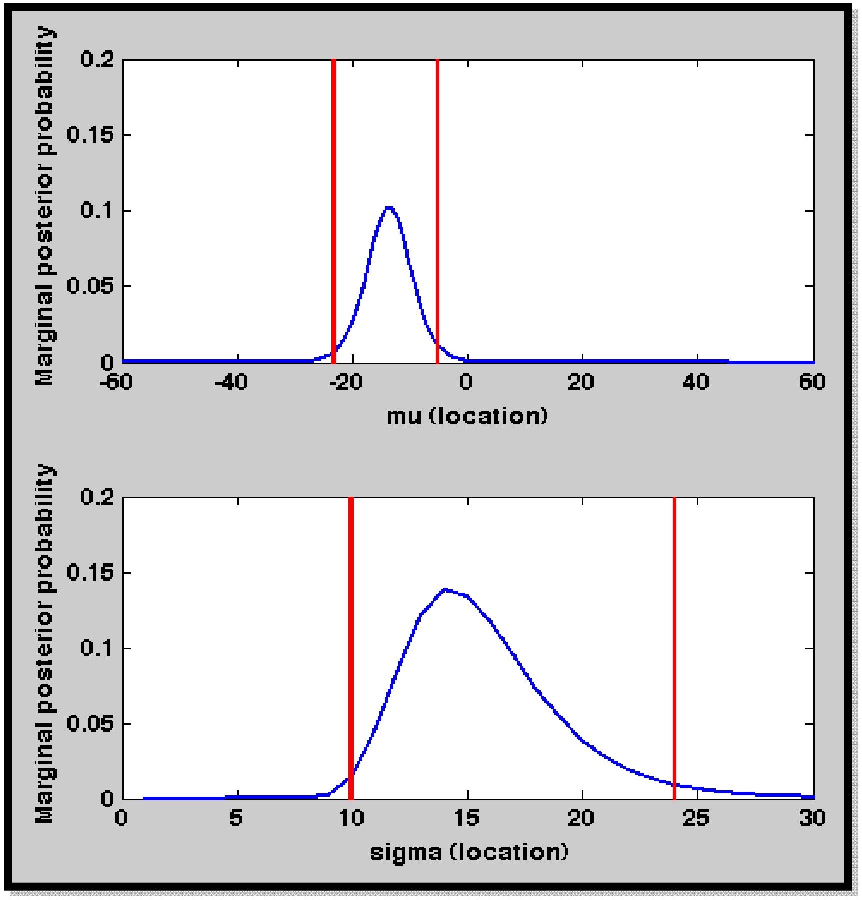
Comparing to bootstrapped estimators

-20 to -6.5



10 to 18

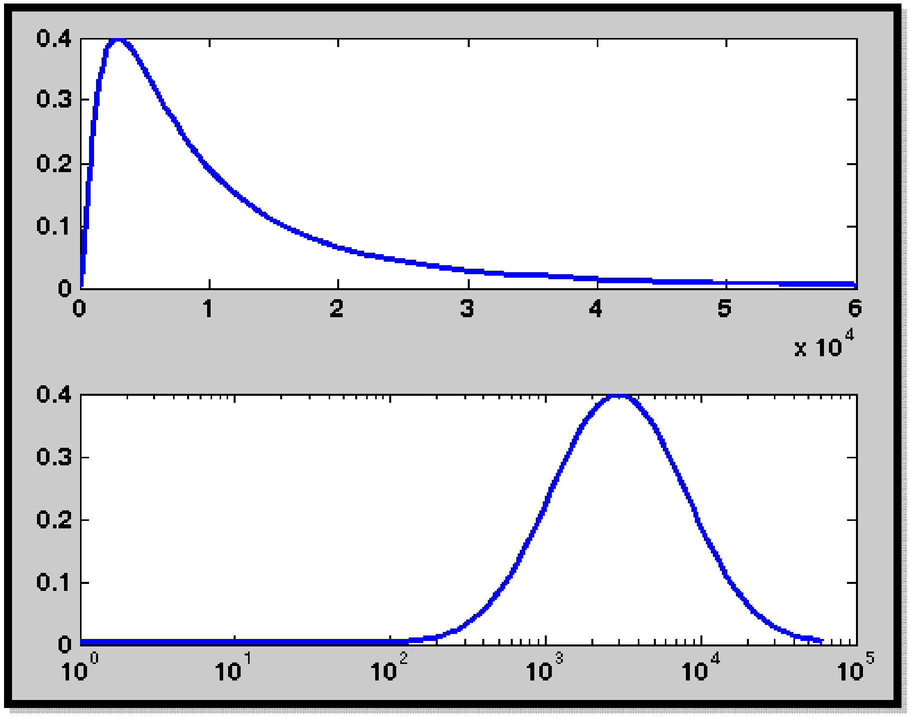
-23 to -5



10 to 24

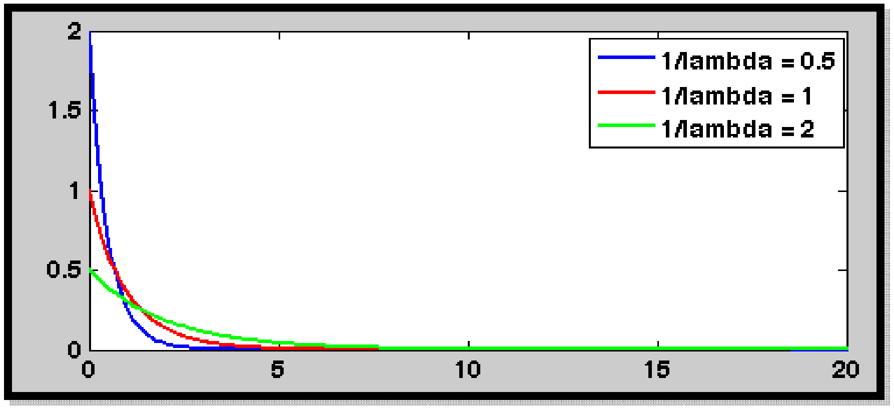
Log-Normal

- Just a Gaussian, applied to the logarithm of observations.
- Why?
  - Good for describing things between 0 and -Infinity



Exponential Distribution

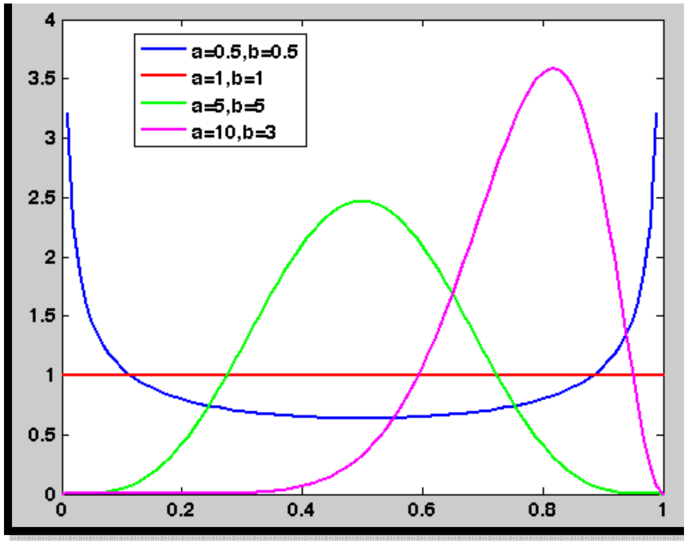
$$\lambda e^{-\lambda x}$$



- What's it do?
  - Assigns a probability to an x between 0 and +Infinity, something that is always decaying.
  - Given a particular count parameter α
  - And another count parameter β
- What's it good for?
  - Describing the probability of probabilities
  - E.g., over many sets of 5 observations each, the probability of getting zapped across these sets.

Beta

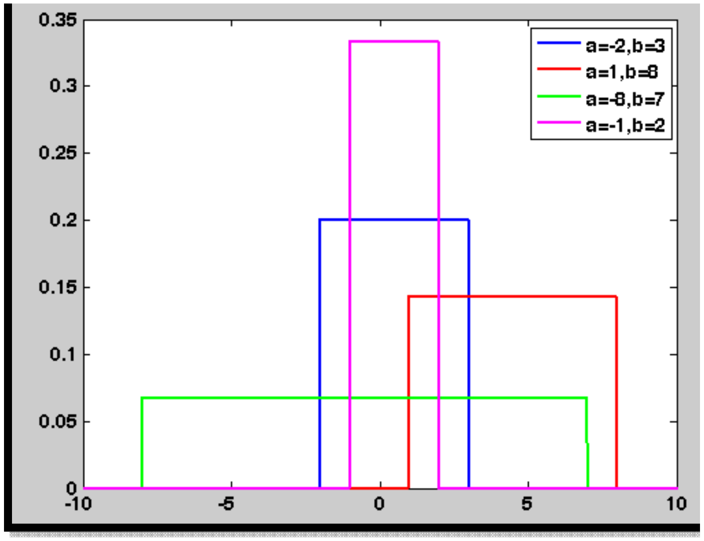
$$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}$$



- What’s it do?
  - Assigns a probability to something on an interval, typically 0 to 1, e.g., another probability
  - Given a particular count parameter α
  - And another count parameter β
- What’s it good for?
  - Describing the probability of probabilities
  - E.g., over many sets of 5 observations each, the probability of getting zapped across these sets.

Uniform

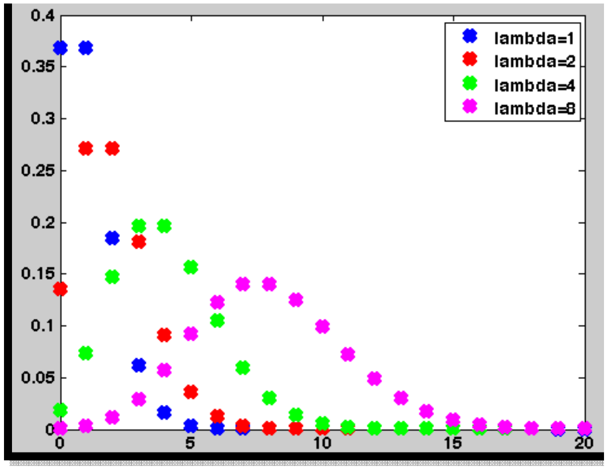
$$\begin{cases} \frac{1}{b-a} & \text{for } a \le x \le b \\ 0 & \text{for } x < a \text{ or } x > b \end{cases}$$



- What’s it do?
  - – Assigns equal probability density to all points within an interval between *a* and *b*
- – What’s it good for?
  - – Describing the “something weird might happen”
    - – E.g., “people might guess randomly”

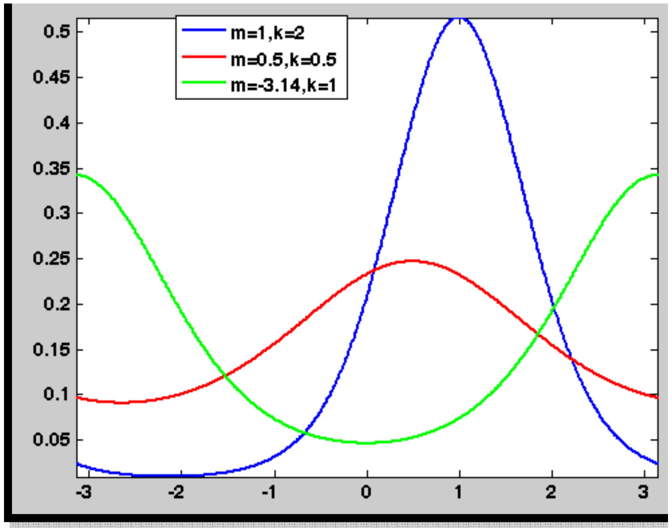
Poisson

$$\frac{e^{-\lambda}\lambda^{k}}{k!}$$



- What's it do?
  - – Probability of the number *k* of independent events occurring
  - – Given that λ events are expected on average
- – What's it good for?
  - – The number of fish caught in an hour.
  - – The number of words in a sentence.

Von Mises (circular Gaussian)

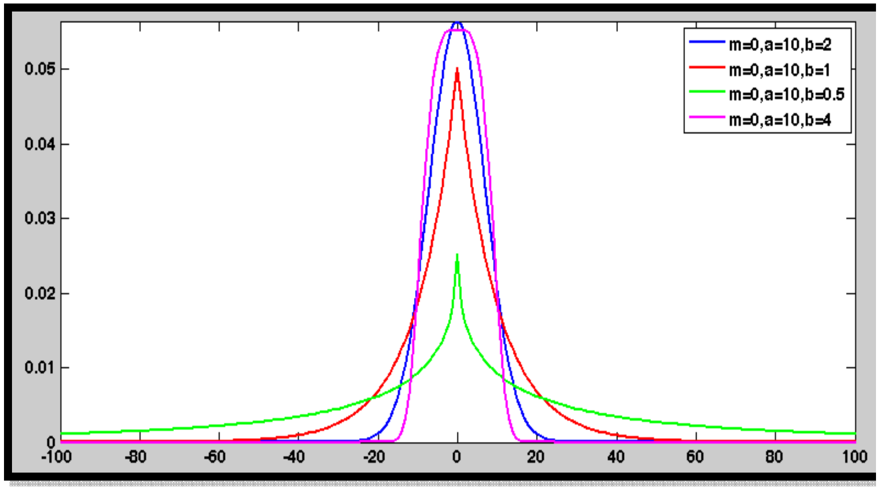


$$\frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)}$$

- What’s it do?
  - – Probability of an observation of cyclical data *x* (e.g., angle, phase, day of year)
  - – With ‘circular location’ μ
  - – Circular precision *κ*
- – What’s it good for?
  - – The phase of the beat…
  - – Errors of circular variables…

Generalized Gaussian Distribution

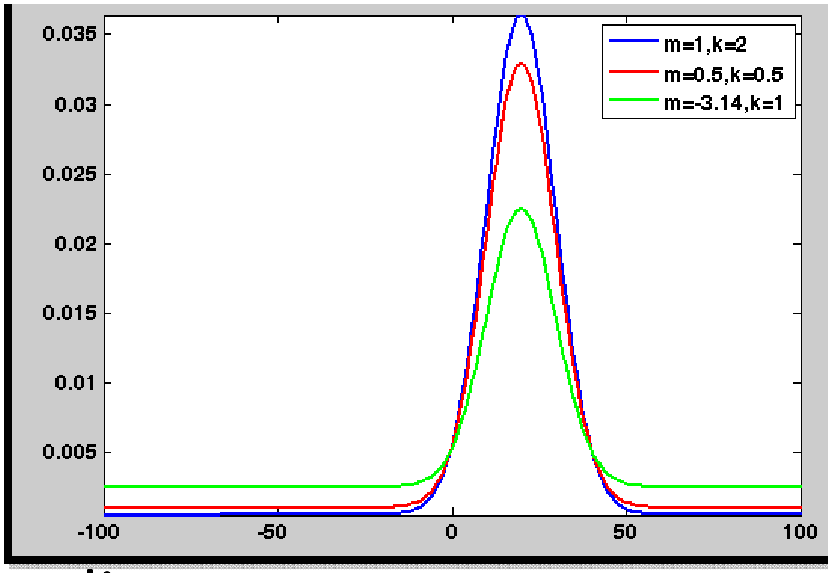
$$p(x)\,dx = \frac{1}{2a\Gamma(1+1/b)} \exp\left(-|x/a|^{b}\right) dx$$



- What’s it do?
  - Probability of an observation of *x* on –Inf to +Inf
  - With ‘location’ μ
  - scale *a*
  - Shape b
- What’s it good for?
  - Things that are not Gaussian (Errors! a.k.a. generalized error distribution)
  - Showing people that grid-search rules.

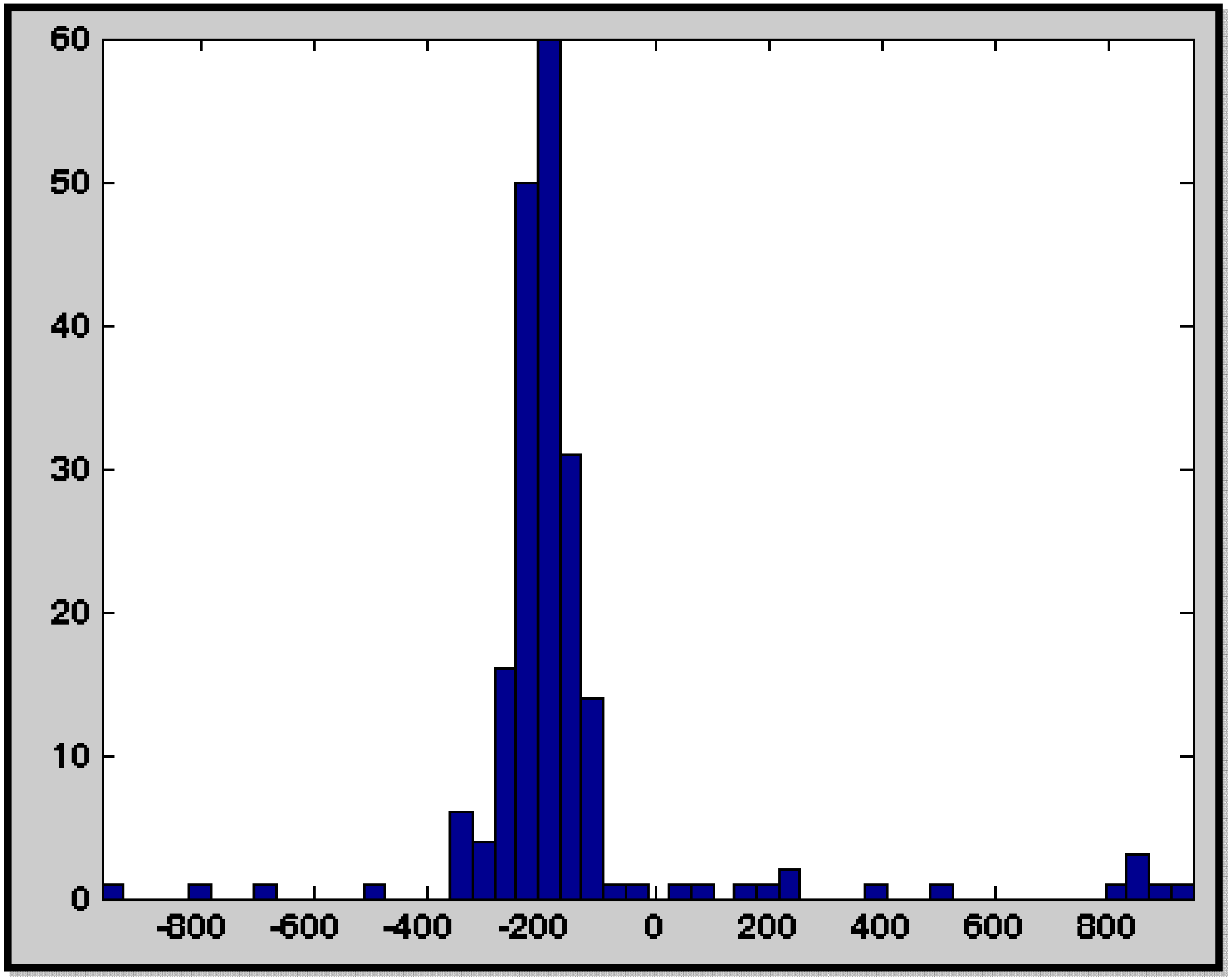
Mixture

mixP*onePDF + (1-mixP)*anotherPDF



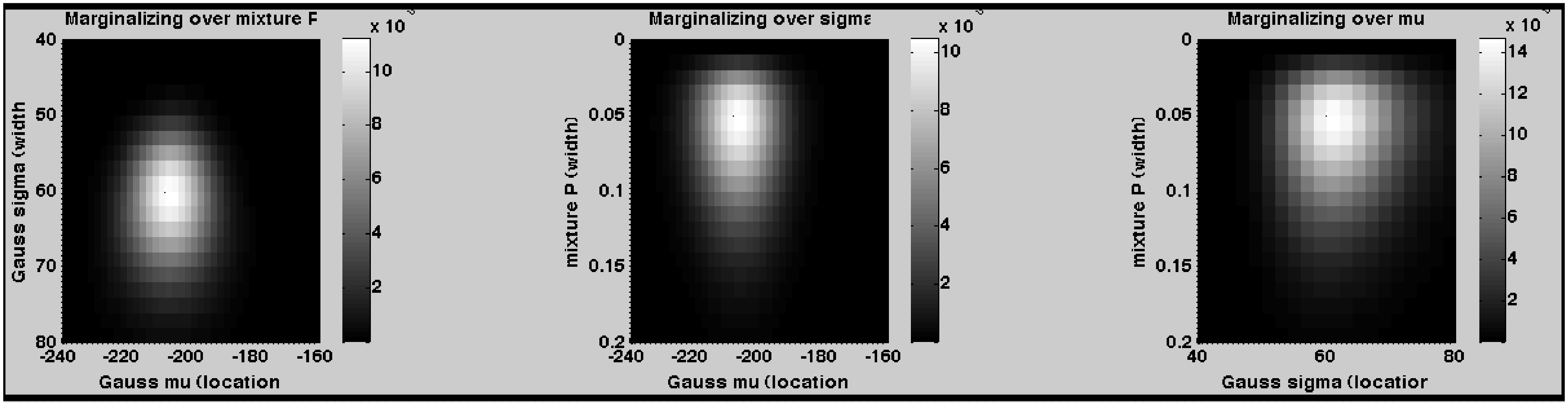
- What’s it do?
  - Assigns probability to x according to a combination of two other distributions. (here, gaussian and uniform)
  - *mixP* parameter determines proportion of each distribution involved
- What’s it good for?
  - Taking into account the possibility of outlandish errors
  - Robust estimation of non-noise data.

Estimating a mixture distribution



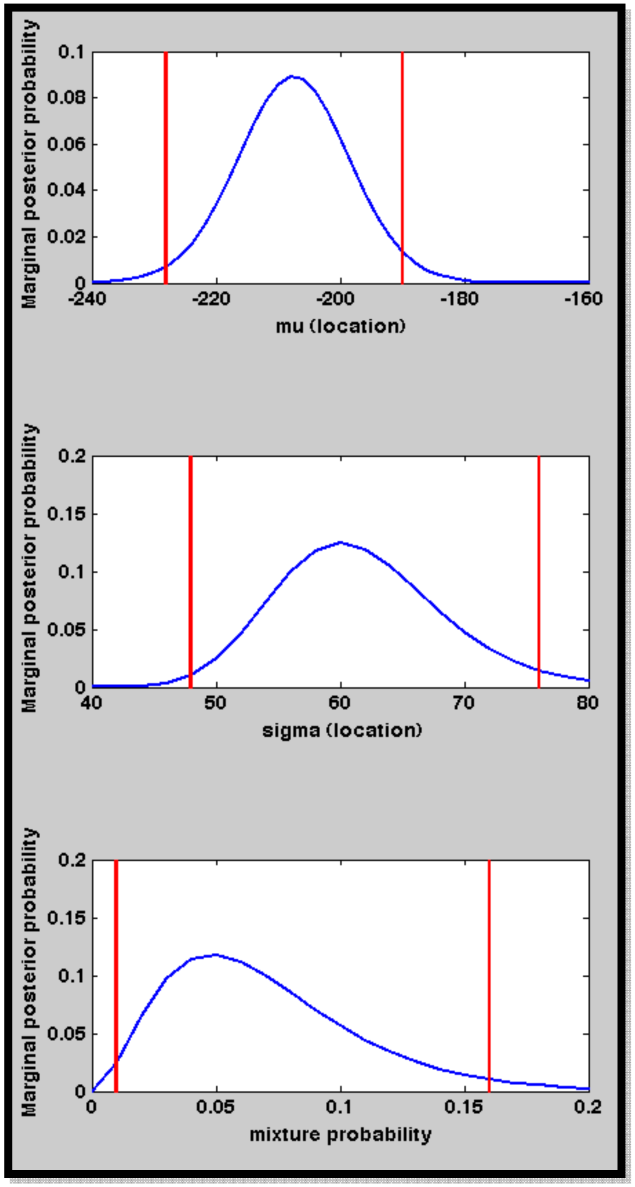
Estimating a mixture distribution

```
uni = [-1000 1000];
f = @(D, m, s, p, uni)(p.*1./(uni(2)-uni(1)) +(1-p).*normpdf(D, m, s));
ms = [-240:2:-160];
ss = [40:2:80];
ps = [0:0.01:0.2];

for i = [1:length(ms)]
   for j = [1:length(ss)]
      for k = [1:length(ps)]
         LL(i,j,k) =sum(log10(f(x, ms(i), ss(j), ps(k), uni)));
      end
   end
end
```



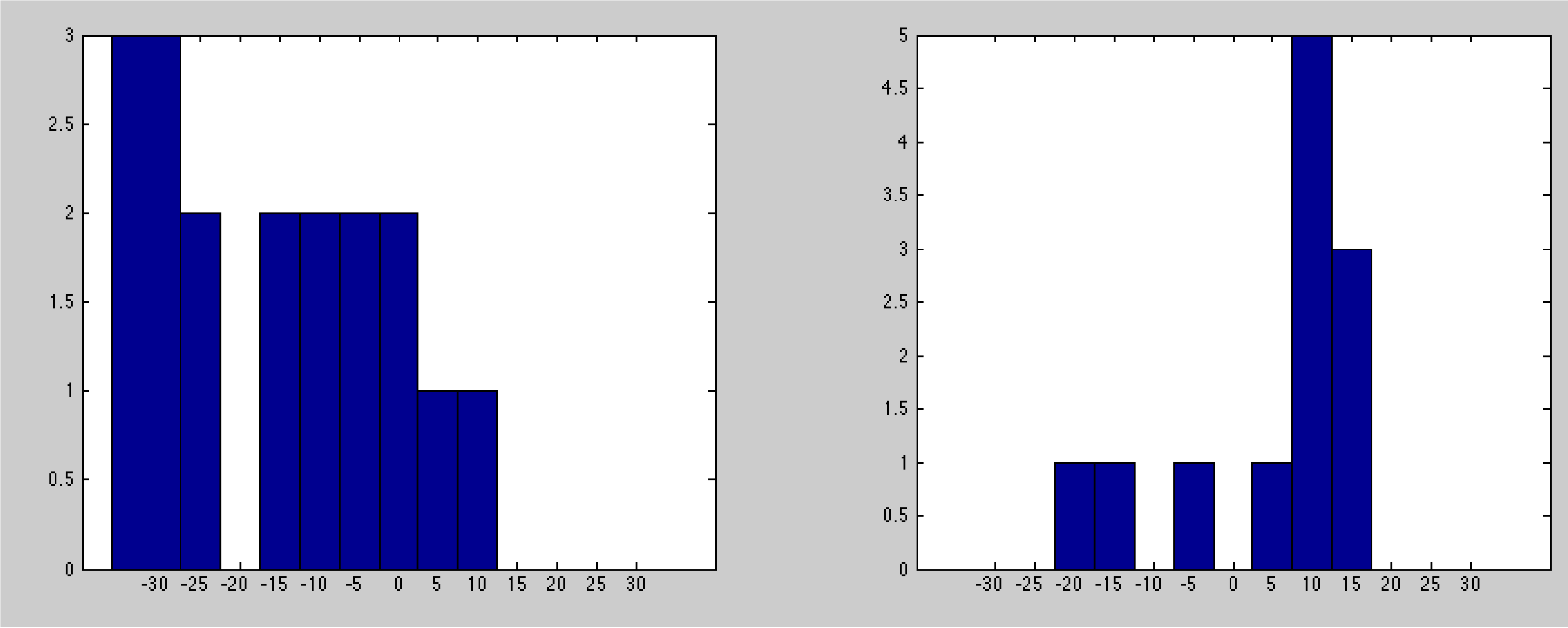
Marginalizing for each parameter

- Virtues of robustness
  - Without ‘robustness’ of mixture, our best estimate of standard deviation would have been “223”.
  - Estimate of mean would have been “-160”.
  - Both very wrong.

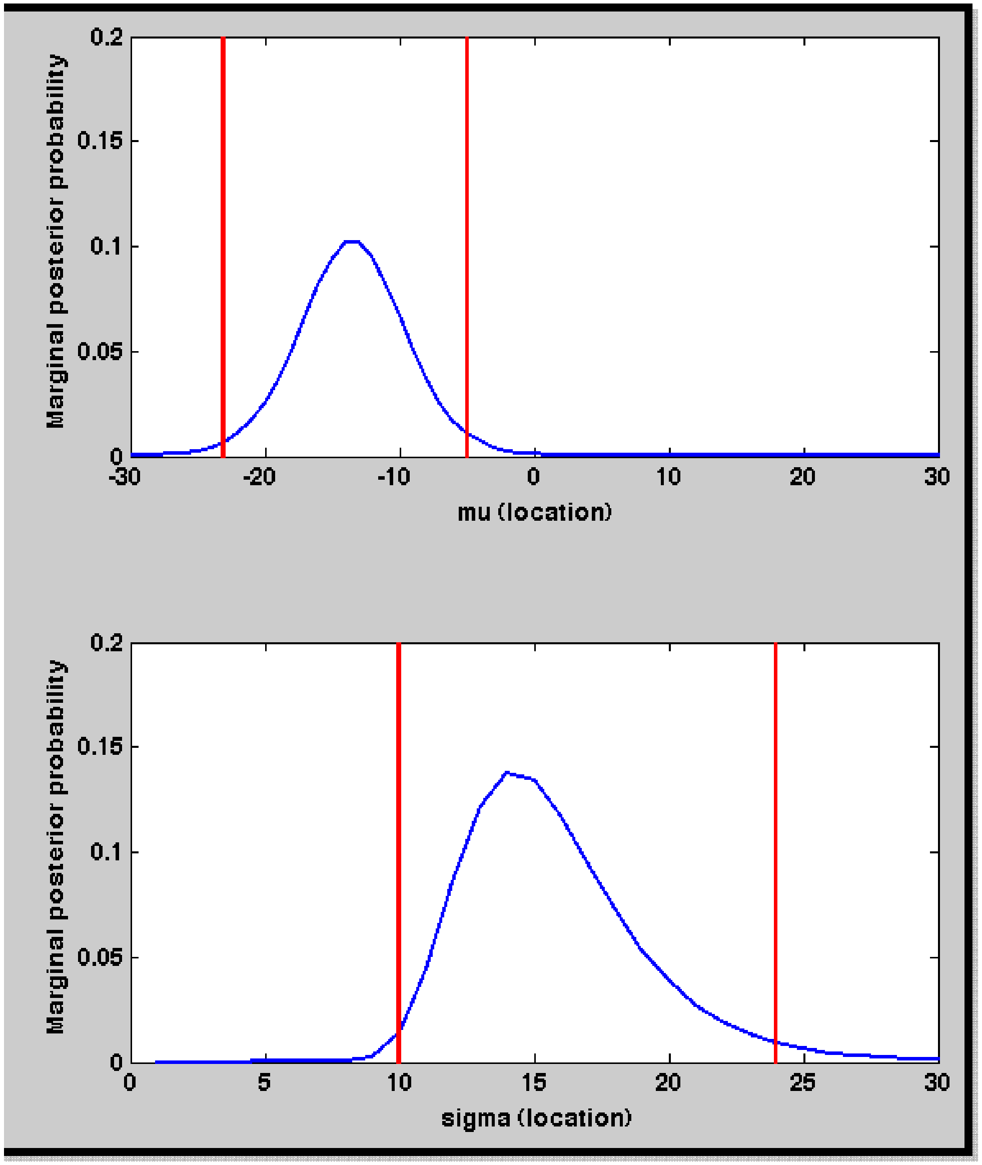


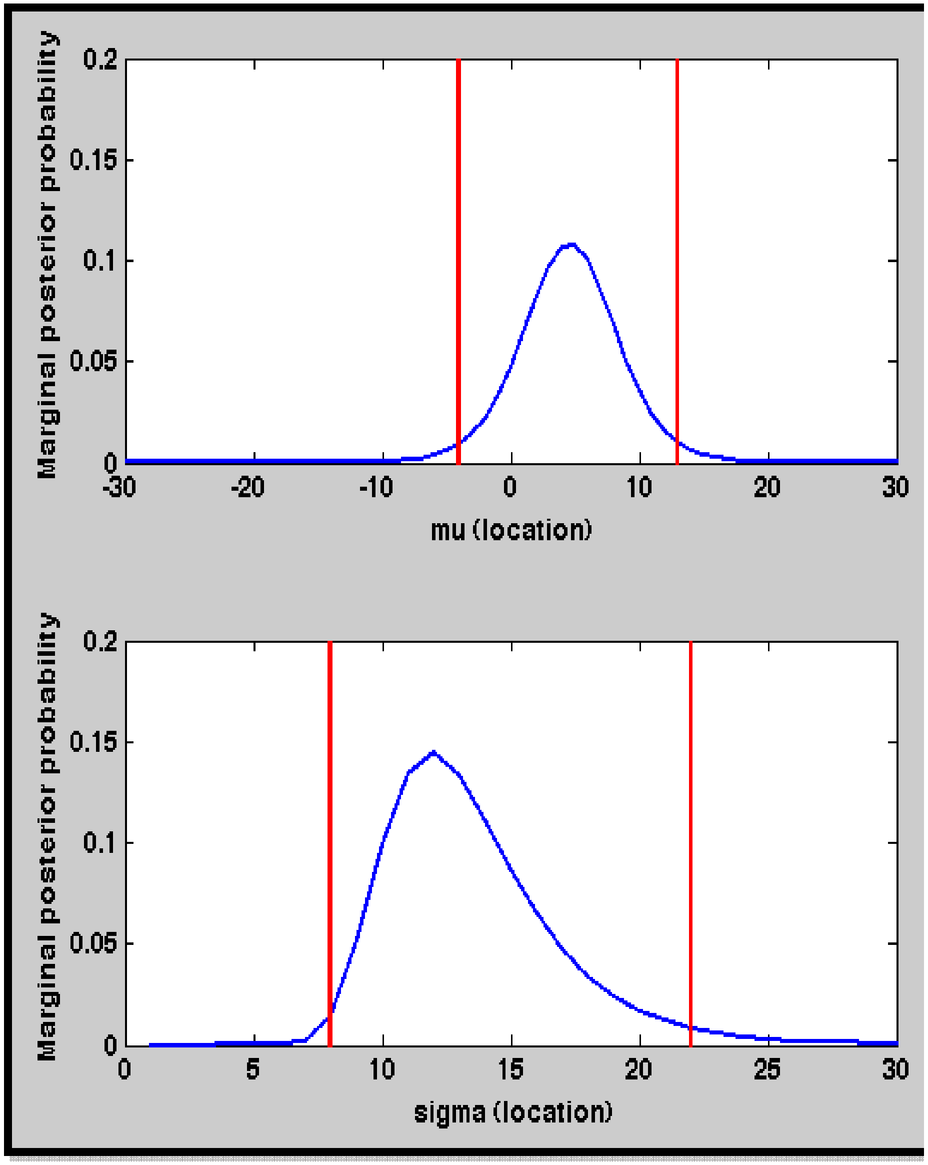
Posterior difference between means.

- How different are these two distributions?
- Assume they are Gaussian (underneath it all)
- Find posterior probability of the difference between means.



Posterior distributions of mus, sigmas





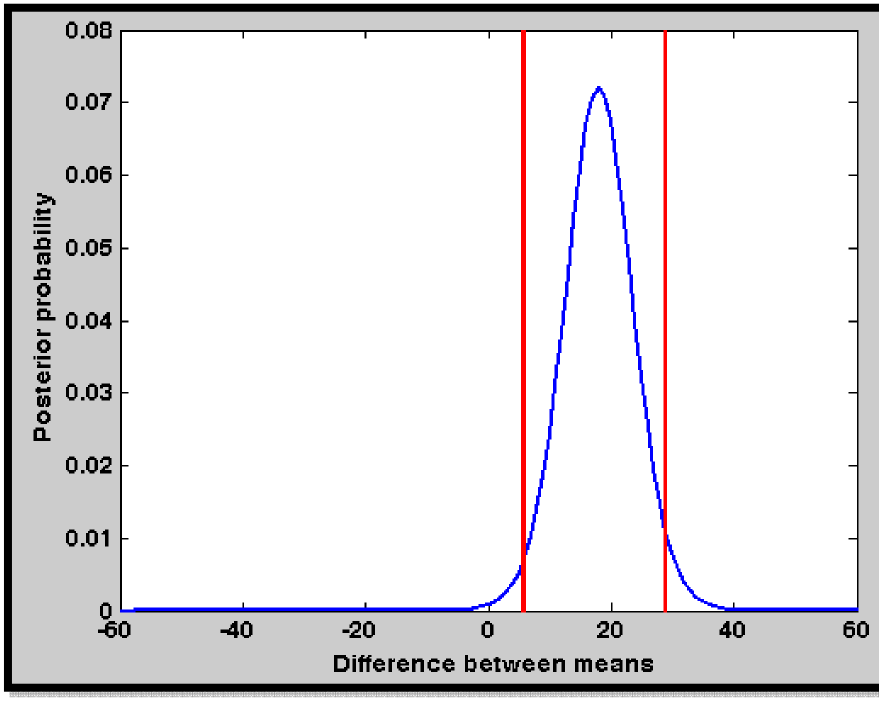
Combining posterior grids.

- For each combination of grid points
  - Compute the measure over both
  - Compute joint probability

- Combine all of these together.

Posterior difference

```
f = @(a,b)(a-b);

ms1 = ms;
ms2 = ms;
post_ms1 = sum(normL2, 2);
post_ms2 = sum(normL1, 2)
```



```
for i = [1:length(ms1)]
   for j = [1:length(ms2)]
      t = f(ms1(i),ms2(j));
      p = post_ms1(i).*post_ms2(j);
            old = find(f_ab == t);
      if(~isempty(old))
         p_f_ab(old) = p_f_ab(old) + p;
      else
         f_ab(end+1) = t;
         p_f_ab(end+1) = p;
      end
   end
end
```

To sum up

- It is useful to think of data as arising from ‘distributions’ that have ‘parameters’
- We want to ask questions about these parameters
- Inverse probability + Bayes theorem allows us to do this.
- We usually cripple Bayes to include only the likelihood.
- With that, we can use a grid search to estimate parameters of ay distribution

Why use a grid search?

- Because it is general, easy, and all you need to know is the likelihood.
- There are virtues of other numerical methods (Gibbs, MCMC, etc.)…
  - They allow you to work with large numbers of parameters and complicated models
- but they require doing quite a bit of math
  - Avoid it if we can
- Also, there are simple analytical solutions for some posterior distributions.
  - Use them! But they are not always available.
    (a grid always is)