Errors

- Basic idea popped out some meetings ago
- We want to do “statistics” for each prediction
- errors(features) or, even better, error-distribution(features)
- Assumption (for the moment at least):

  error1-distribution(feature1),
  error2-distribution(feature2),
  ...
- **Disclaimer: maybe what I am doing makes no sense :)** (in any case doing this I realized that there are some further issues)

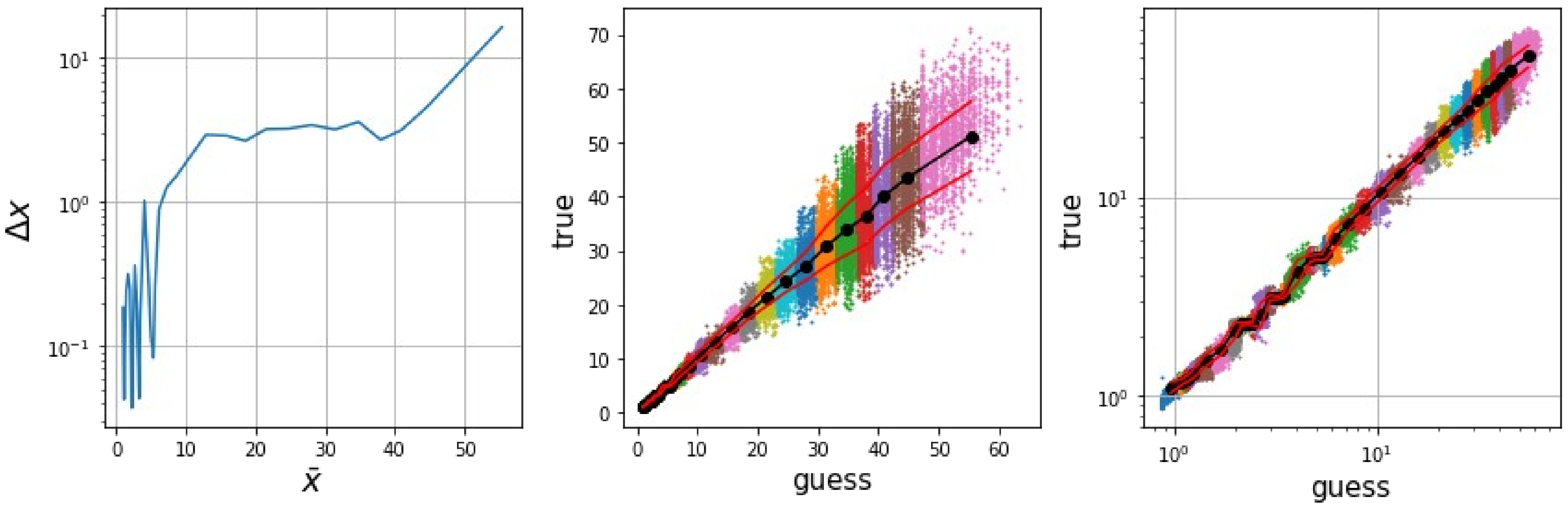
Finding error-distr(feature)

- We consider predicted/recovered (x-axis) vs injected (y-axis)
- In an ideal world, y=x. In practice for each x-value we have many y-values.
  More y-values → less reliable prediction, i.e. bigger error
- Let’s quantify: for each x-value, we want a probability distribution for the true value y.
- We need to create bins with a small dx, then we consider all the y-values in these bins and we hope that they follow a reasonable distribution (possibly not multimodal)

Choosing the bins

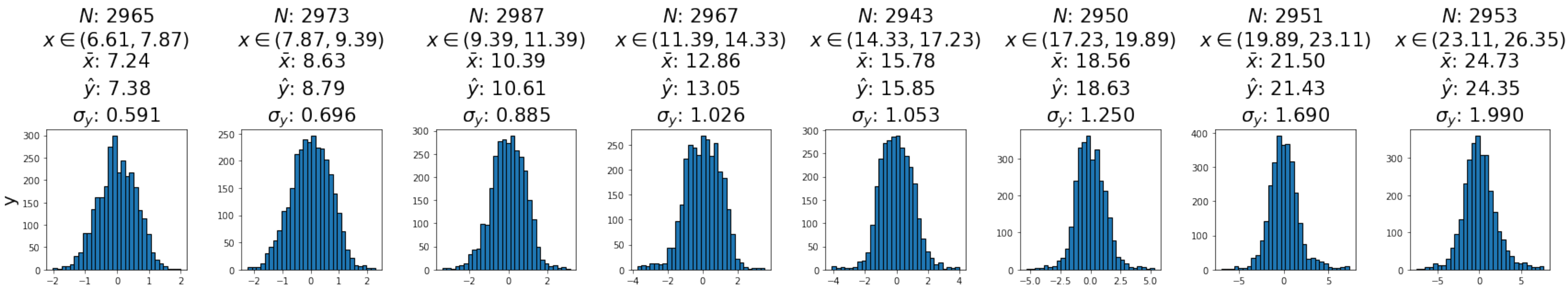
- Issues:
  1) If dx is too small, we do not have enough points: → statistics make no sense
  2) If dx too big, we are mixing x-values: → statistics make no sense
  3) Not uniformly distributed data → constant dx is not good
- For the moment I am fixing the number of points in each bin → issue (2) is quite relevant, let's ignore it just for a moment

Example: chirp mass



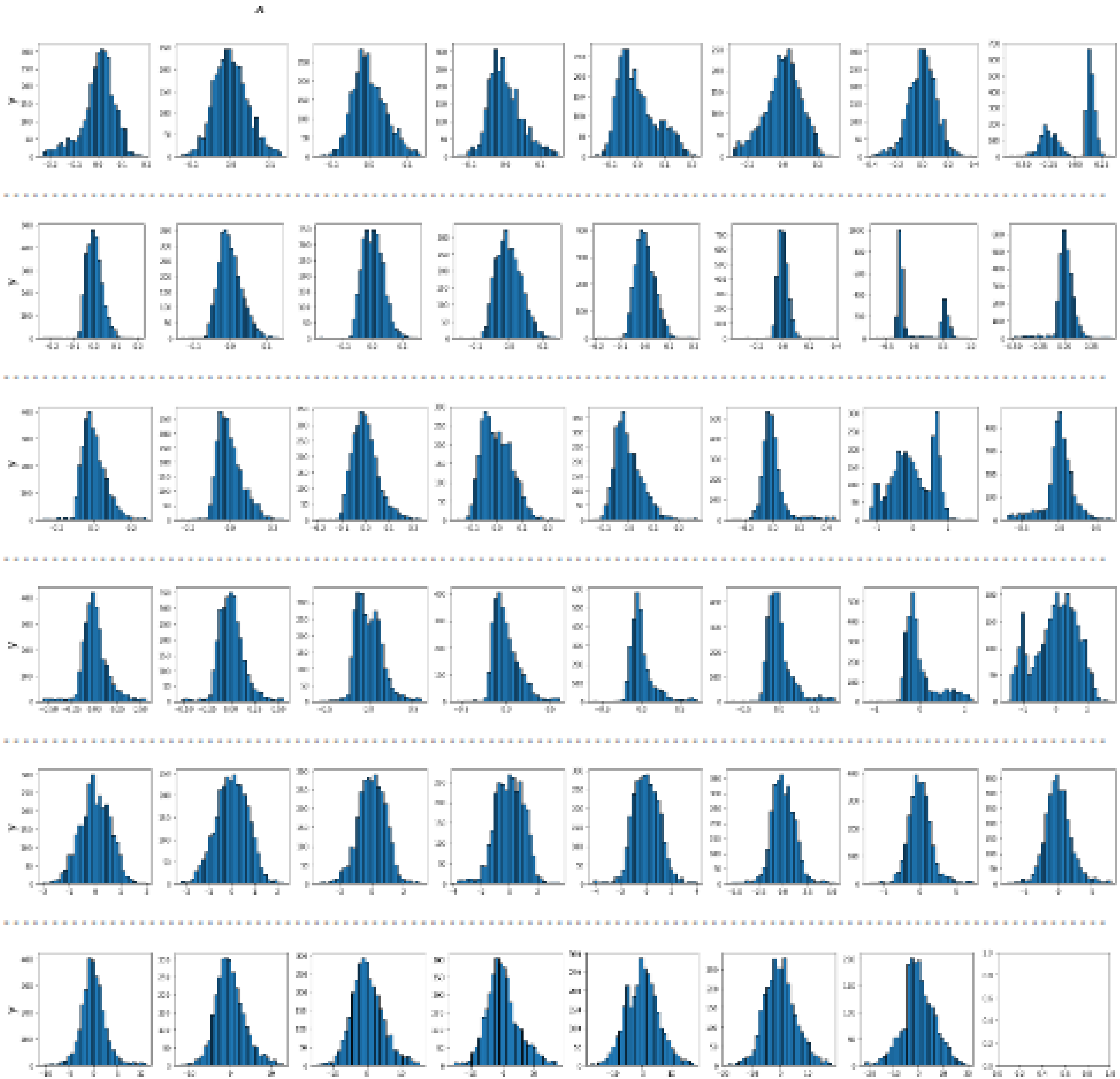
- guess is prediction
- 3k points for bins
- black dots: mean(y) for each bin
- red line: std(y) for each bin

Example: chirp mass



Some distributions:

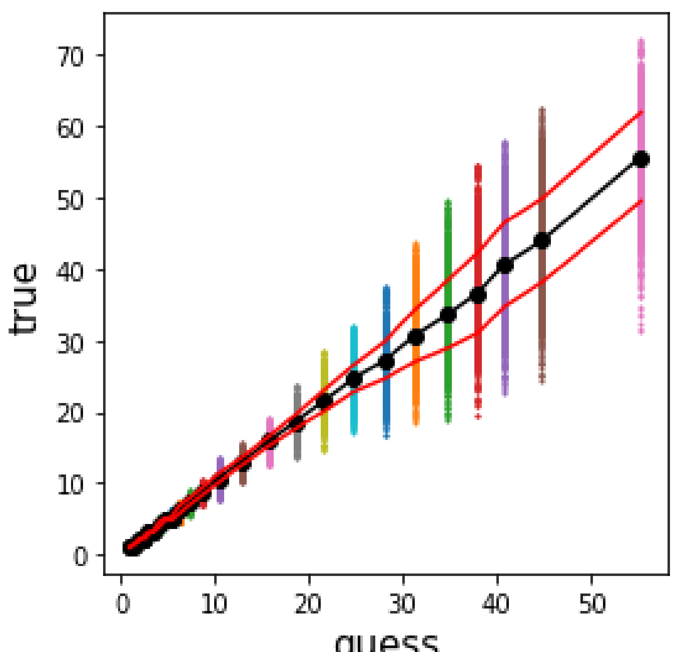
- N~3k, but slightly smaller → I am removing some outliers
- if x-middle is close to y-mean with a low y-std, we are happy
- Skewness != 0 is not a big issue (we will see why later)
- these distributions seem good, but it’s cherry picking

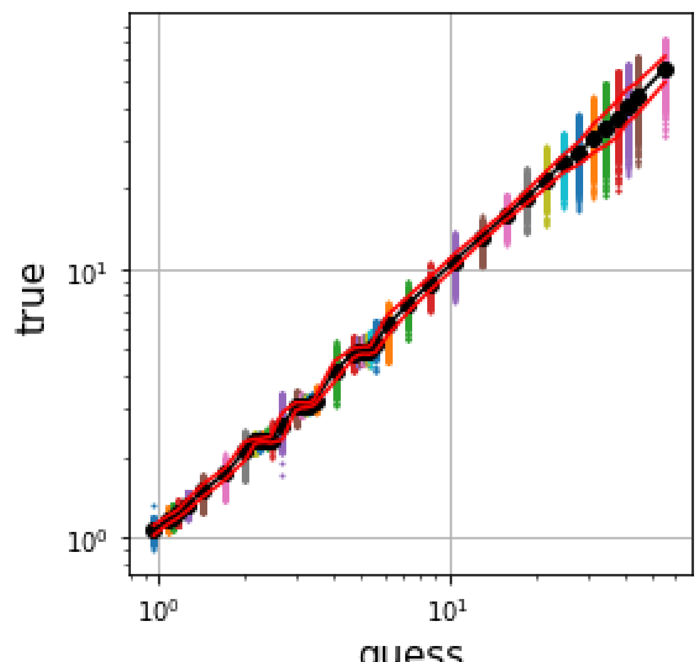


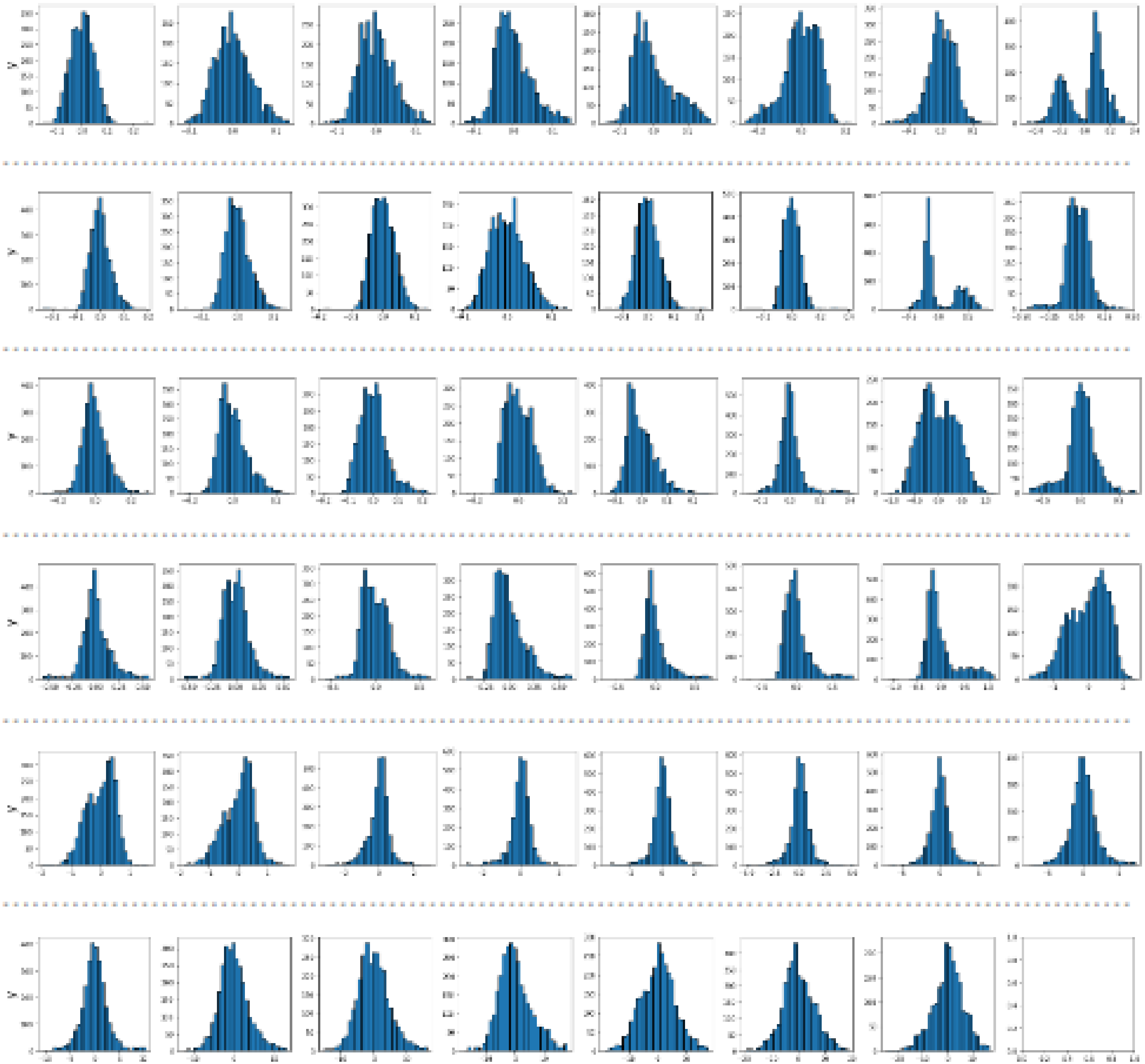
Low-quality image that shows all the distributions

Improve the bins-procedure

- Since dx can be quite big, the y-values do not refer to the same x (i.e., xmid is not very representative)
- → project the y values on the x=xmid line using the y=x line (in practice x’=xmid, y’=y+xmid-x)







Skew normal distribution

Fancy ref: https://doi.org/10.1093/biomet/83.4.715
Ref that I actually used : https://en.wikipedia.org/wiki/Skew_normal_distribution

- In a few cases the histo are bimodal. This is really bad.
- In most cases we have distr with non-zero skewness, so they are not Gaussians. This is not dramatic, let's consider a 1-parameter family of deformed gaussians:

Definition [edit]

Let $\phi(x)$ denote the standard normal probability density function

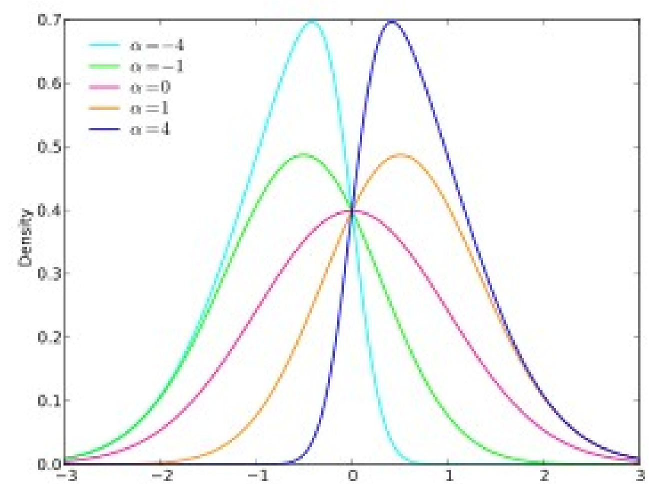
$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

with the cumulative distribution function given by

$$\Phi(x) = \int_{-\infty}^{x} \phi(t)\, dt = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right],$$

where "erf" is the error function. Then the probability density function (pdf) of the skew-normal distribution with parameter α is given by

$$\boxed{f(x) = 2\phi(x)\Phi(\alpha x).}$$



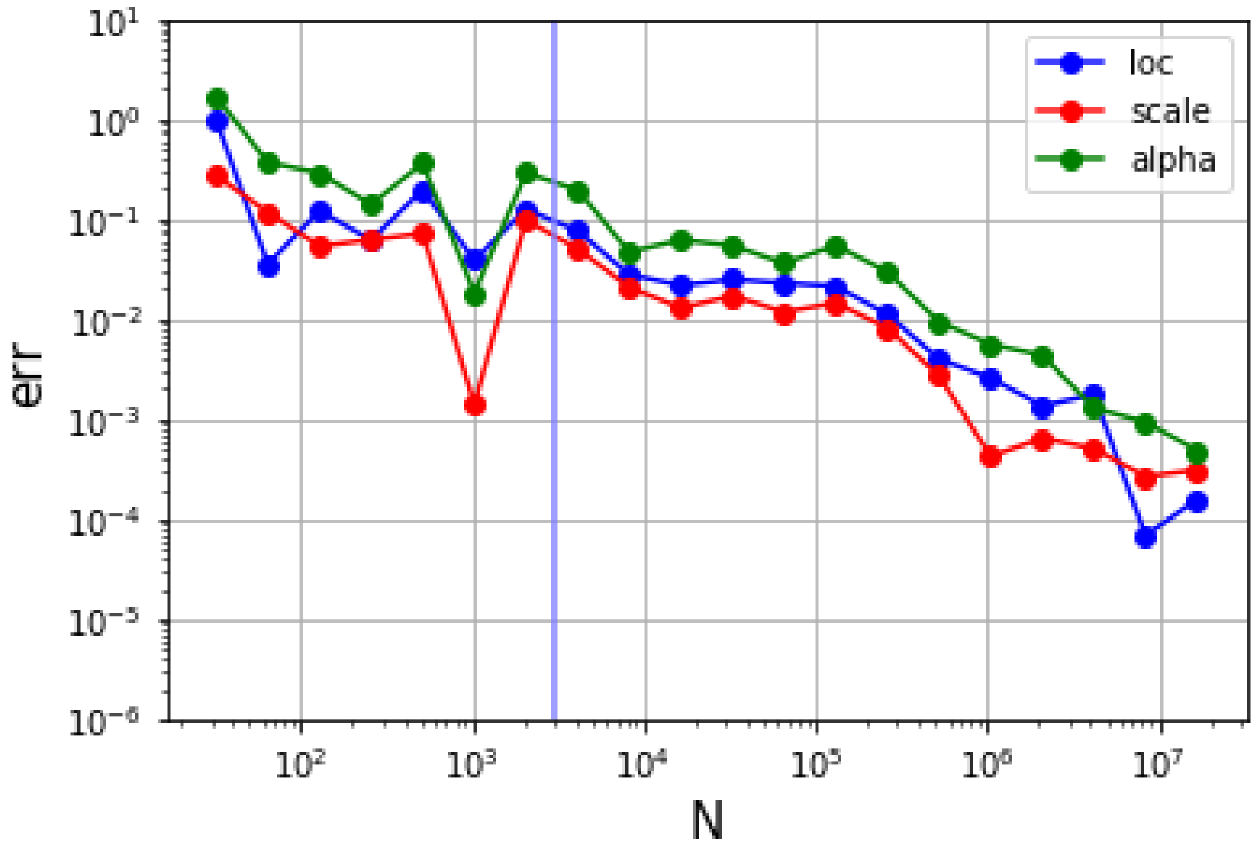
Skew normal distribution

- If alpha is zero we go back to Gaussian
- Given data, how do we find the corresponding distribution?
  Measure mean, variance, skewnees and compute loc, scale and shape
- Caveat: skew normal distribution can reproduce only skewness in (-1,1)

Parameters	ξ location (real) ω scale (positive, real) α shape (real)
Support	$x \in (-\infty; +\infty)$
PDF	$\frac{2}{\omega\sqrt{2\pi}} e^{-\frac{(x-\xi)^2}{2\omega^2}} \int_{-\infty}^{\alpha\left(\frac{x-\xi}{\omega}\right)} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}\ dt$
CDF	$\Phi\left(\frac{x-\xi}{\omega}\right) - 2T\left(\frac{x-\xi}{\omega}, \alpha\right)$ $T(h,a)$ is Owen's T function
Mean	$\xi + \omega\delta\sqrt{\frac{2}{\pi}}$ where $\delta = \frac{\alpha}{\sqrt{1+\alpha^2}}$
Mode	$\xi + \omega m_o(\alpha)$
Variance	$\omega^2\left(1 - \frac{2\delta^2}{\pi}\right)$
Skewness	$\gamma_1 = \frac{4-\pi}{2} \frac{\left(\delta\sqrt{2/\pi}\right)^3}{\left(1-2\delta^2/\pi\right)^{3/2}}$

Skew normal distribution

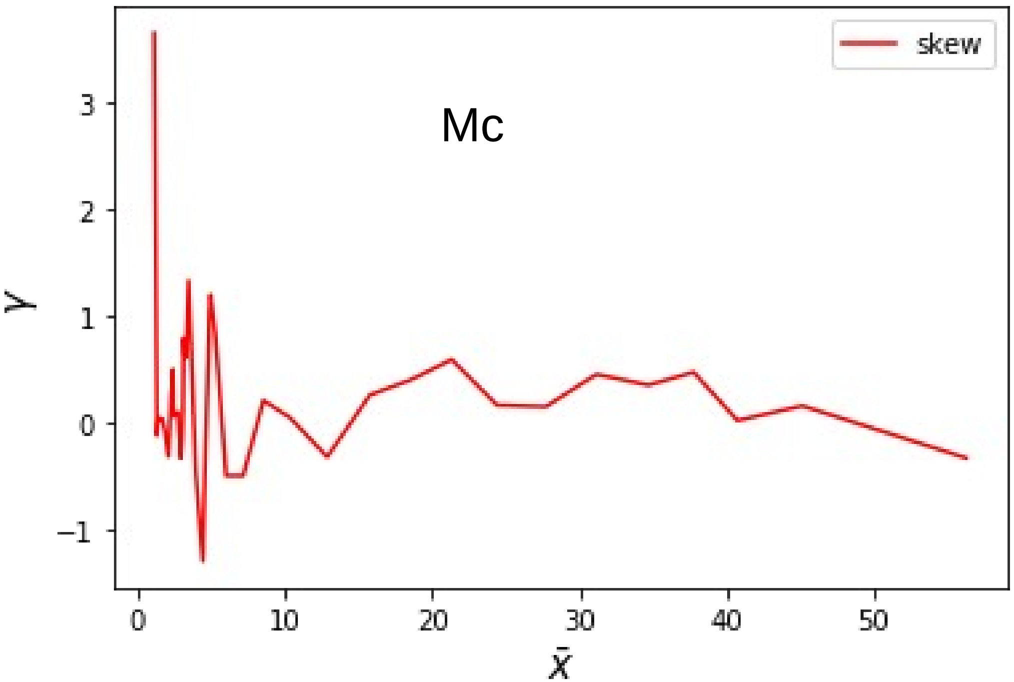
- How good is the loc/scale/shape recovery from mean/var/skewness for N~3k?

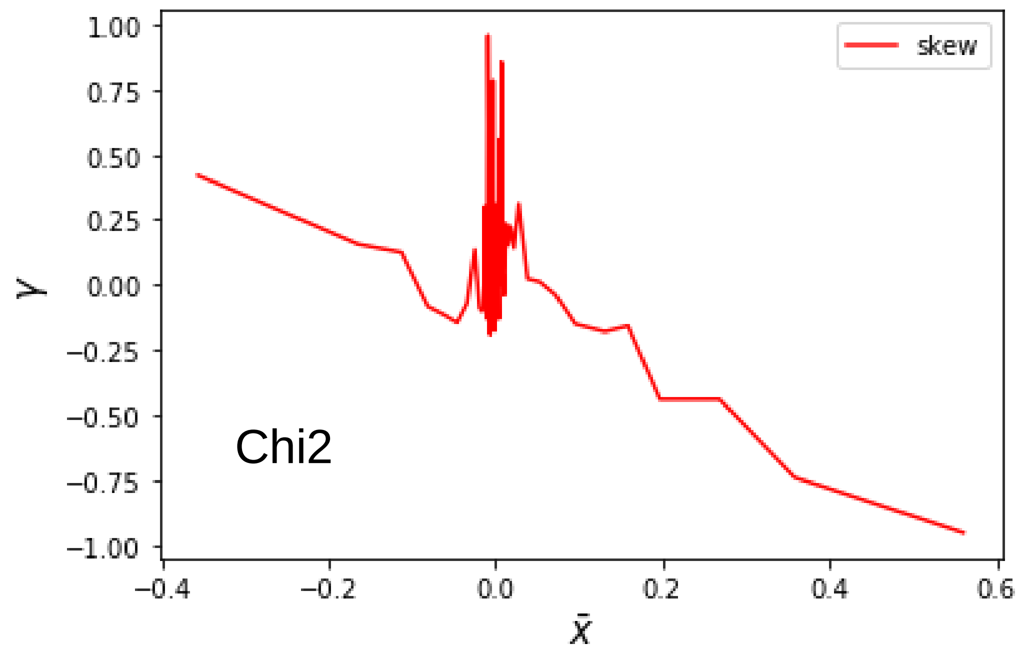


See IPAM2021_ML/algo/classy_NN/SkewNormalDistribution.ipynb

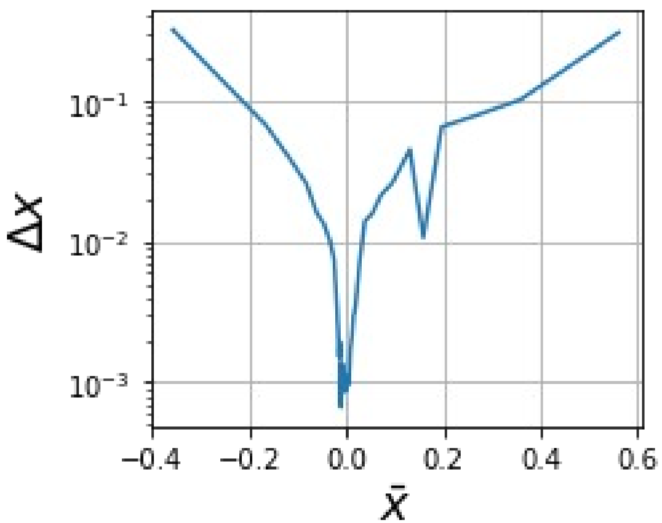
Skew normal distribution

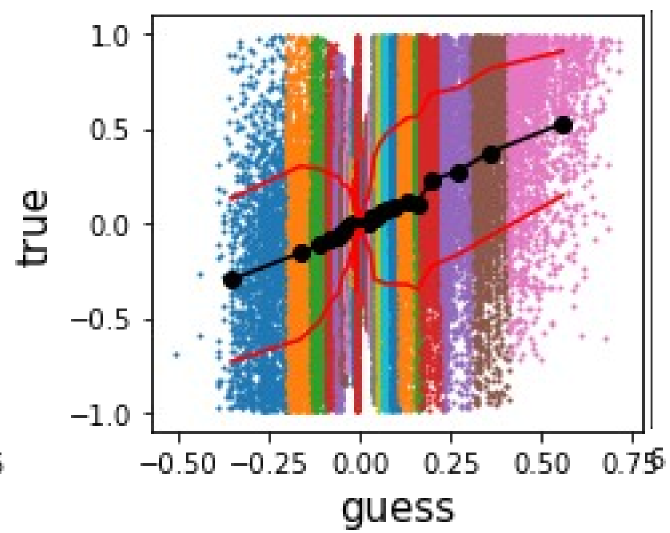
- Skewness in our case:

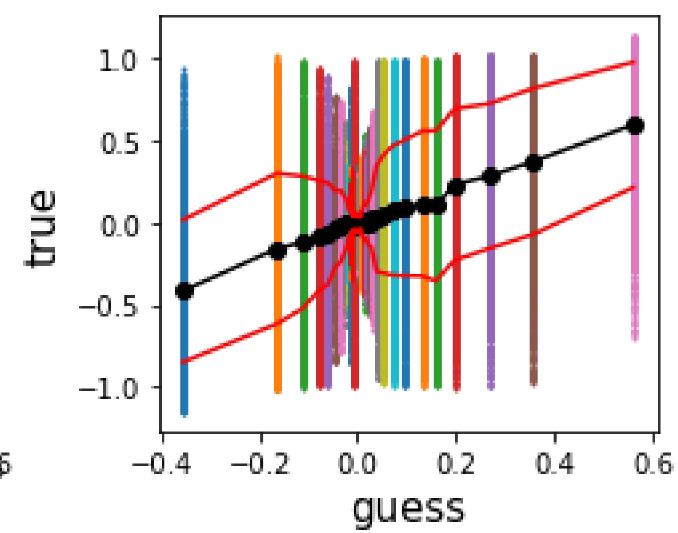




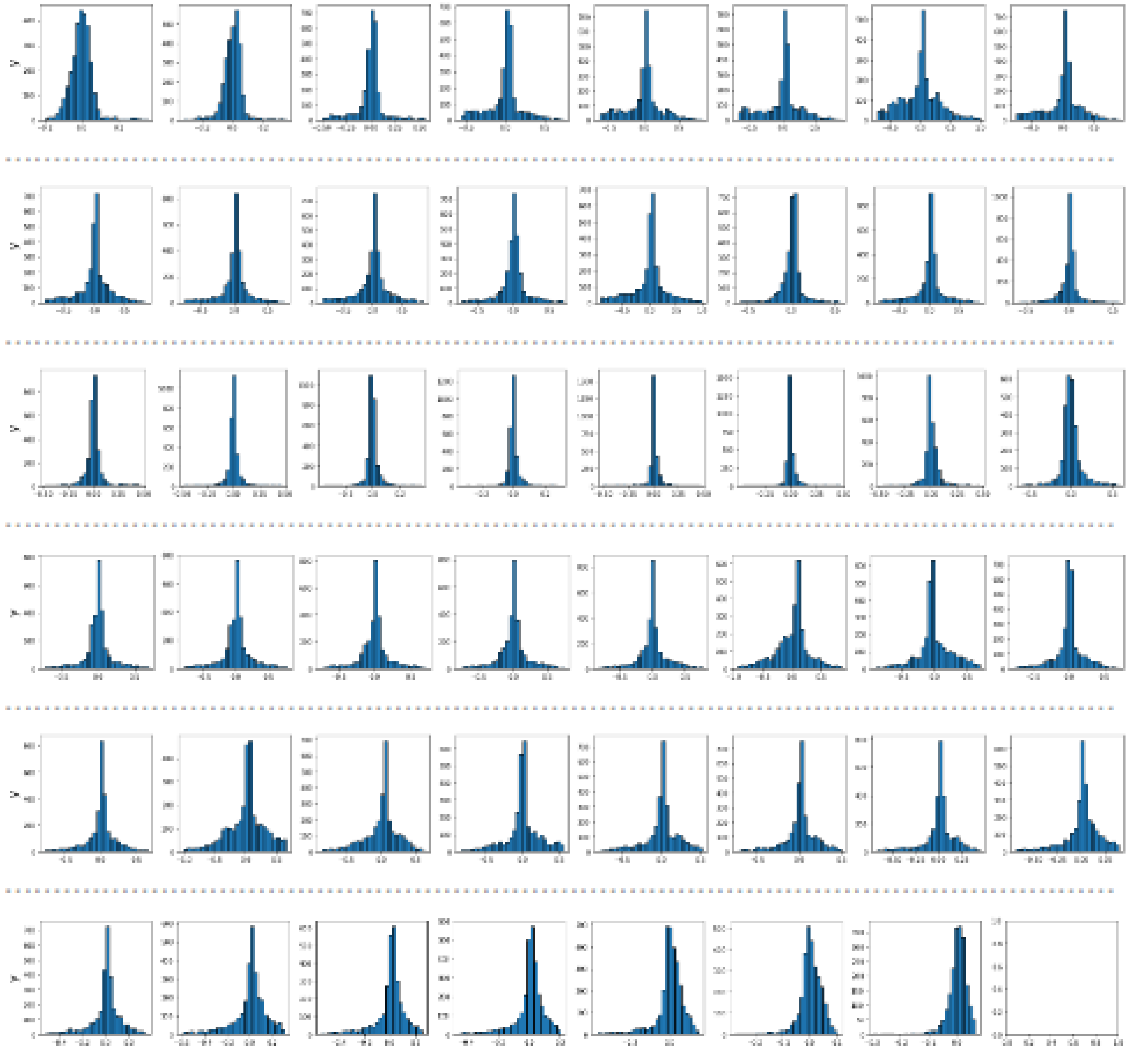
And btw, these are the plots for chi2







And btw, these are the plots for chi2



Summary (of the idea)

- Create bins
- Find mean, var, skewness
- If these three values change smoothly we can interpolate and for each feature we have these three values as function of the feature
- From mean, var, skew we can compute the distribution and compute errors/probability/stuff