Model Fitting from χ^2 to Maximum Likelihood

Keto Zhang

Regression

$$y(x) = \hat{y}(x; \alpha, \beta, \dots)$$

$$\hat{y}(x) = \alpha x + \beta$$

Least Squares

Fancy terms: loss and objective function

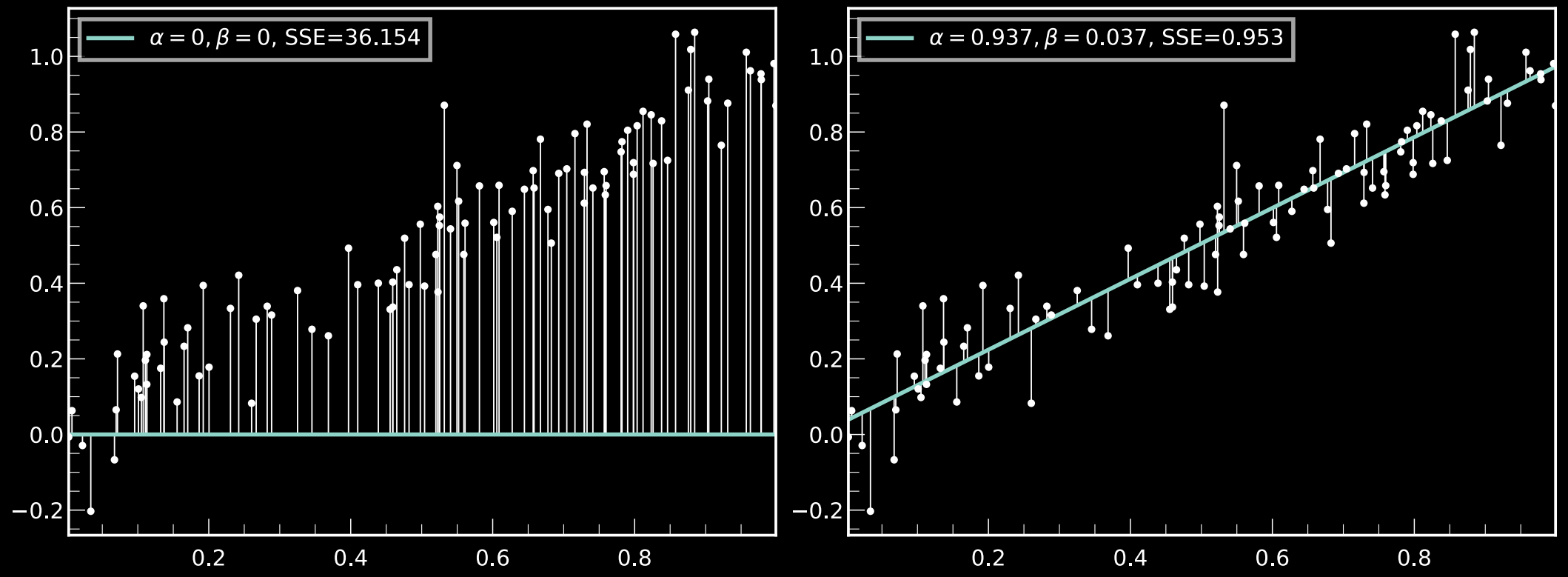
$$\text{SSE} = \sum_{i=1}^{N} \underbrace{(y_i - \hat{y})^2}_{\text{Loss Function}}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{Objective Function}}$$

- Minimize this letting θ in $\hat{y}$ be free

$$\mathrm{SSE} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

$$\hat{y}_i = \alpha x_i + \beta$$



```
def f(x, alpha, beta):
    return x * alpha + beta

def sse(params, x, y):
    alpha, beta = params
    ypred = f(x, alpha, beta)

    loss =(y - ypred)**2
    return np.sum(loss)

from scipy.optimize import minimize

# Let xdata and ydata be from some dataset
result = minimize(sse, x0=[0, 0], args=(xdata, ydata))
alpha_best, beta_best = result.x
```

$$\hat{y}_i = \alpha x_i + \beta$$

$$\text{SSE} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

$$\hat{\alpha}, \hat{\beta} = \underset{\alpha,\beta}{\text{argmin}}(\text{SSE})$$

Error Weighted Least Squares

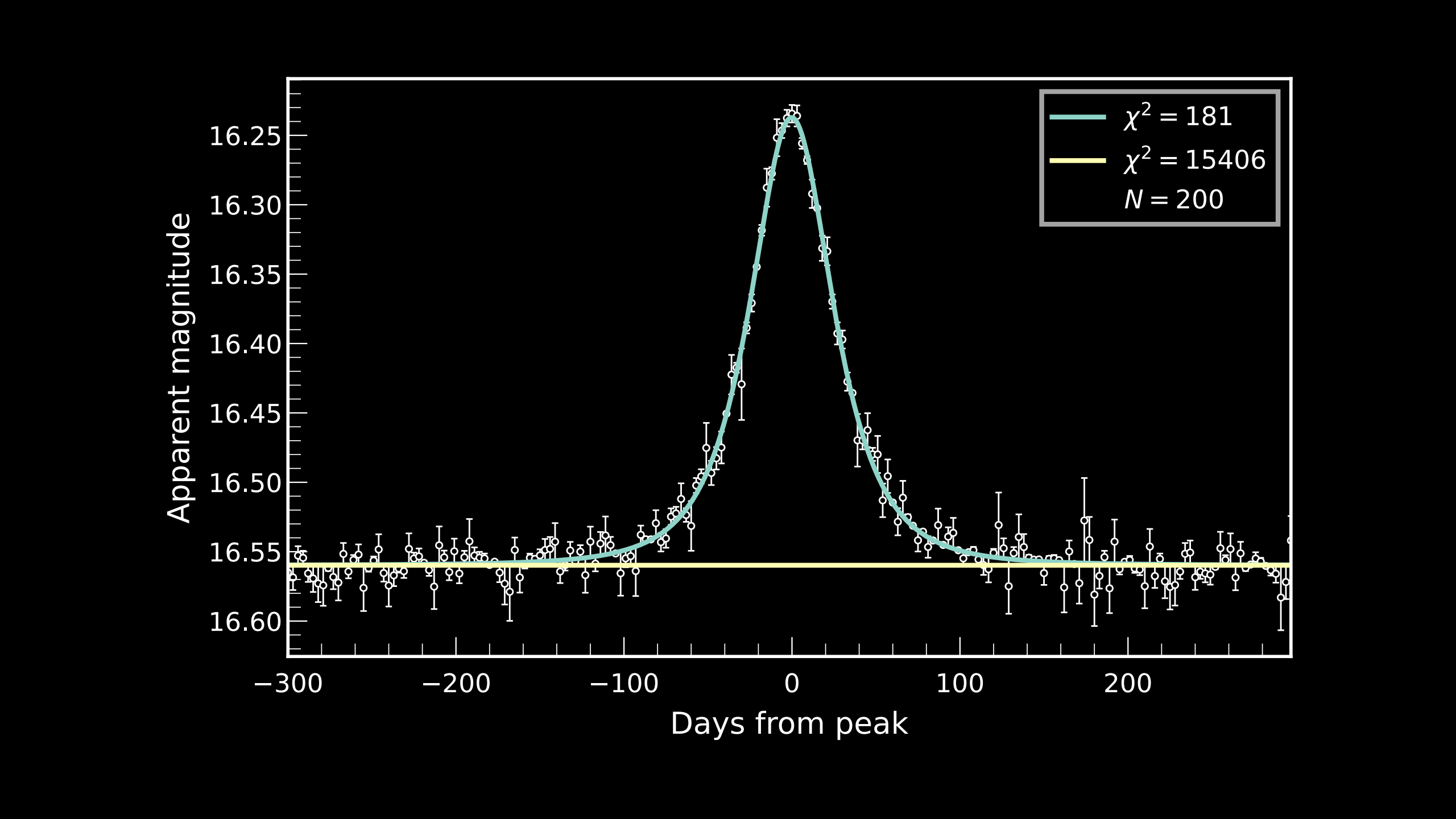
aka χ^2

It’s least squares but with noise

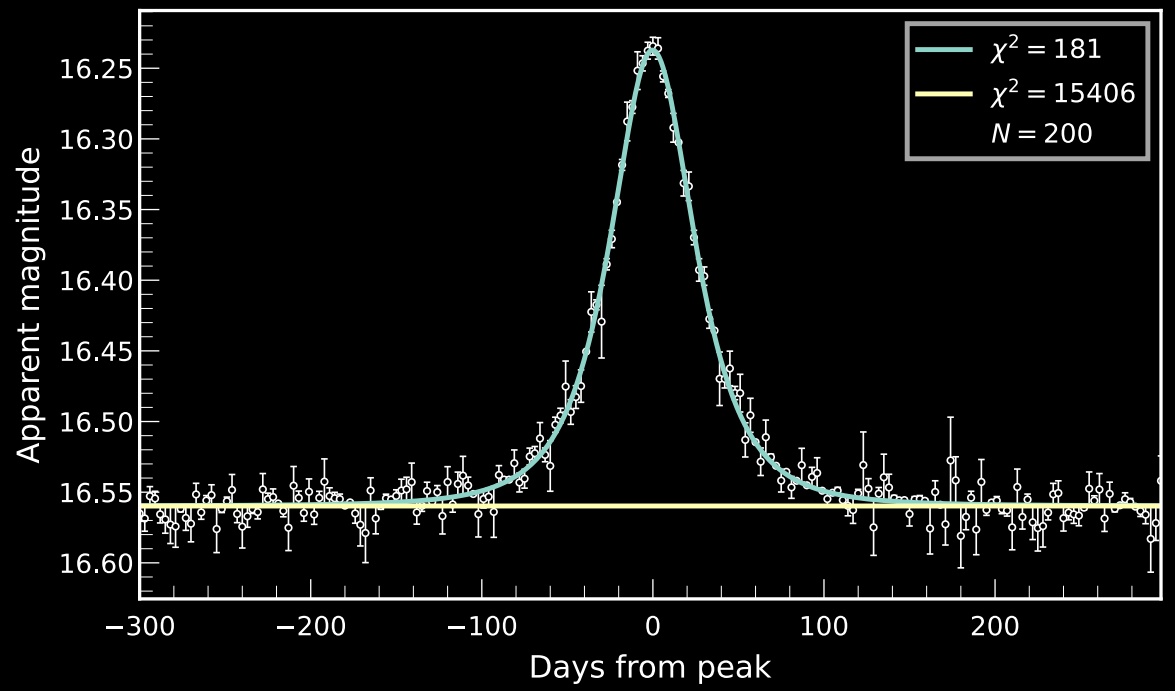
$$\chi^2 = \sum_{i=1}^{N} \frac{(y_i - \hat{y})^2}{\sigma_i^2}$$

Dispersion of noise

- Minimize this letting θ in $\hat{y}$ be free


$\chi^2 = 181$
$\chi^2 = 15406$
$N = 200$
Apparent magnitude
16.25
16.30
16.35
16.40
16.45
16.50
16.55
16.60
−300
−200
−100
0
100
200
Days from peak

Degree of Freedom	Probability of Exceeding the Critical Value								
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01
1	0.000	0.004	0.016	0.102	0.455	1.32	2.71	3.84	6.63
2	0.020	0.103	0.211	0.575	1.386	2.77	4.61	5.99	9.21
3	0.115	0.352	0.584	1.212	2.366	4.11	6.25	7.81	11.34
4	0.297	0.711	1.064	1.923	3.357	5.39	7.78	9.49	13.28
5	0.554	1.145	1.610	2.675	4.351	6.63	9.24	11.07	15.09
6	0.872	1.635	2.204	3.455	5.348	7.84	10.64	12.59	16.81
7	1.239	2.167	2.833	4.255	6.346	9.04	12.02	14.07	18.48
8	1.647	2.733	3.490	5.071	7.344	10.22	13.36	15.51	20.09
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67
10	2.558	3.940	4.865	6.737	9.342	12.55	15.99	18.31	23.21
11	3.053	4.575	5.578	7.584	10.341	13.70	17.28	19.68	24.72
12	3.571	5.226	6.304	8.438	11.340	14.85	18.55	21.03	26.22
13	4.107	5.892	7.042	9.299	12.340	15.98	19.81	22.36	27.69
14	4.660	6.571	7.790	10.165	13.339	17.12	21.06	23.68	29.14
15	5.229	7.261	8.547	11.037	14.339	18.25	22.31	25.00	30.58
16	5.812	7.962	9.312	11.912	15.338	19.37	23.54	26.30	32.00
17	6.408	8.672	10.085	12.792	16.338	20.49	24.77	27.59	33.41
18	7.015	9.390	10.865	13.675	17.338	21.60	25.99	28.87	34.80
19	7.633	10.117	11.651	14.562	18.338	22.72	27.20	30.14	36.19
20	8.260	10.851	12.443	15.452	19.337	23.83	28.41	31.41	37.57
22	9.542	12.338	14.041	17.240	21.337	26.04	30.81	33.92	40.29
24	10.856	13.848	15.659	19.037	23.337	28.24	33.20	36.42	42.98
26	12.198	15.379	17.292	20.843	25.336	30.43	35.56	38.89	45.64
28	13.565	16.928	18.939	22.657	27.336	32.62	37.92	41.34	48.28
30	14.953	18.493	20.599	24.478	29.336	34.80	40.26	43.77	50.89
40	22.164	26.509	29.051	33.660	39.335	45.62	51.80	55.76	63.69
50	27.707	34.764	37.689	42.942	49.335	56.33	63.17	67.50	76.15
60	37.485	43.188	46.459	52.294	59.335	66.98	74.40	79.08	88.38




```
>>> import scipy.stats as st

>>> 1 - st.chi2.cdf(15406, df=200)
0.0

>>> 1 - st.chi2.cdf(181, df=200)
0.8285004581033072
```

Maximum Likelihood

The dual problem to χ^2

Probabilistic Regression

$$y(x_i) = \hat{y}(x_i) + \epsilon_i$$

stats + sys

Probabilistic Regression

$$\epsilon_i \sim \text{Normal}\left(0, \sigma_i^2\right)$$

aka Gaussian

Probabilistic Regression

$$y(x_i) = \hat{y}(x_i) + \epsilon_i$$

$$\epsilon_i \sim \text{Normal}(0, \sigma_i^2)$$

$$y \sim \text{Normal}(\hat{y}_i, \sigma_i^2)$$

- Because ϵ is a random variable, y is a random variable
- Because ϵ is normally distributed, y is normally distributed but with mean $\hat{y}$

Likelihood Function

A goodness of fit metric calculated with the assumption that your data is has some probability distribution. For fixed parameters $\boldsymbol{\theta}$, the likelihood function gives the probability* that your data is observed out of all possible outcomes.

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^{N} P[y_i \mid \hat{y}\,(\boldsymbol{\theta}; x_i)]$$

- **<u>Maximize</u>** this letting θ in $\hat{y}$ be free

*Not really

Gaussian Likelihood Looks Familiar

$$P[\, y_i \mid \hat{y}\,(\boldsymbol{\theta}; x_i)\,] = A_i \cdot \exp\left[-\frac{(y_i - \hat{y}_i)^2}{\sigma_i^2}\right]$$

$$\log \mathcal{L} = \sum_{i=1}^{N}\left[\log A_i - \frac{(y_i - \hat{y}_i)^2}{\sigma_i^2}\right]$$

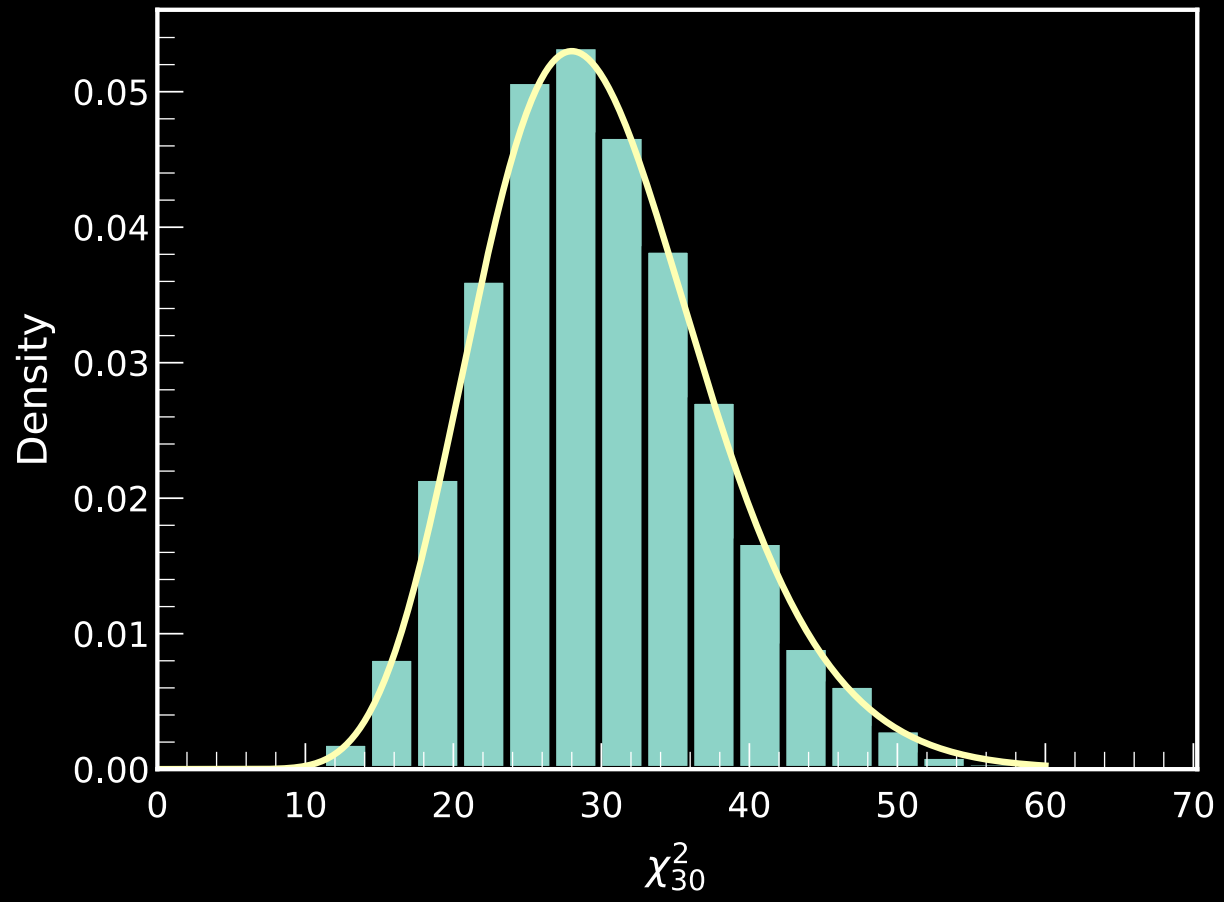
$$-\log \mathcal{L} \propto \chi^2$$

Extra: χ^2 does imply Gaussian noise

The sum of squares of independent Gaussian random variables is distributed as the χ^2 with k degrees of freedom.

$$\chi^2 = \sum_{i=1}^{N} \frac{\epsilon_i^2}{\sigma_i^2}$$

$$\chi^2 \sim \chi^2(N)$$



Extra: Bayesian Maximum a Posteriori

$$P(\theta \mid y_1, y_2, \ldots, y_N) \propto \prod_{i=1}^{N} P(y_i \mid \theta) P(\theta)$$

$$\theta_{\mathrm{MAP}} \equiv \mathrm{Mode}(\theta)$$

- In Bayesian, θ is a random variable (as well as y)
- If $P(\theta) = 1$ (uniform), then $\theta_{\mathrm{MLE}} = \theta_{\mathrm{MAP}}$

Conclusion

- If your data has appreciable error, use χ^2 to weigh the fitting on more reliable data points.
- Literature are always doing probabilistic regression without stating it
- In most cases, experimental data comes with the assumption of measurements are random due to Gaussian noise
- Gaussian maximum likelihood and χ^2 regression produce the same results
- Probability and statistics is not intuitive, don’t ignore formalism and fundamentals.

Resources

- *An Introduction to Statistical Learning,* James G., Witten D., Hastie T., Tibshirani R., https://www.statlearning.com/
- Prob 140, http://prob140.org/
- Data 100, https://ds100.org/