a practical introduction to

Gaussian Processes

for astronomy

Dan Foreman-Mackey

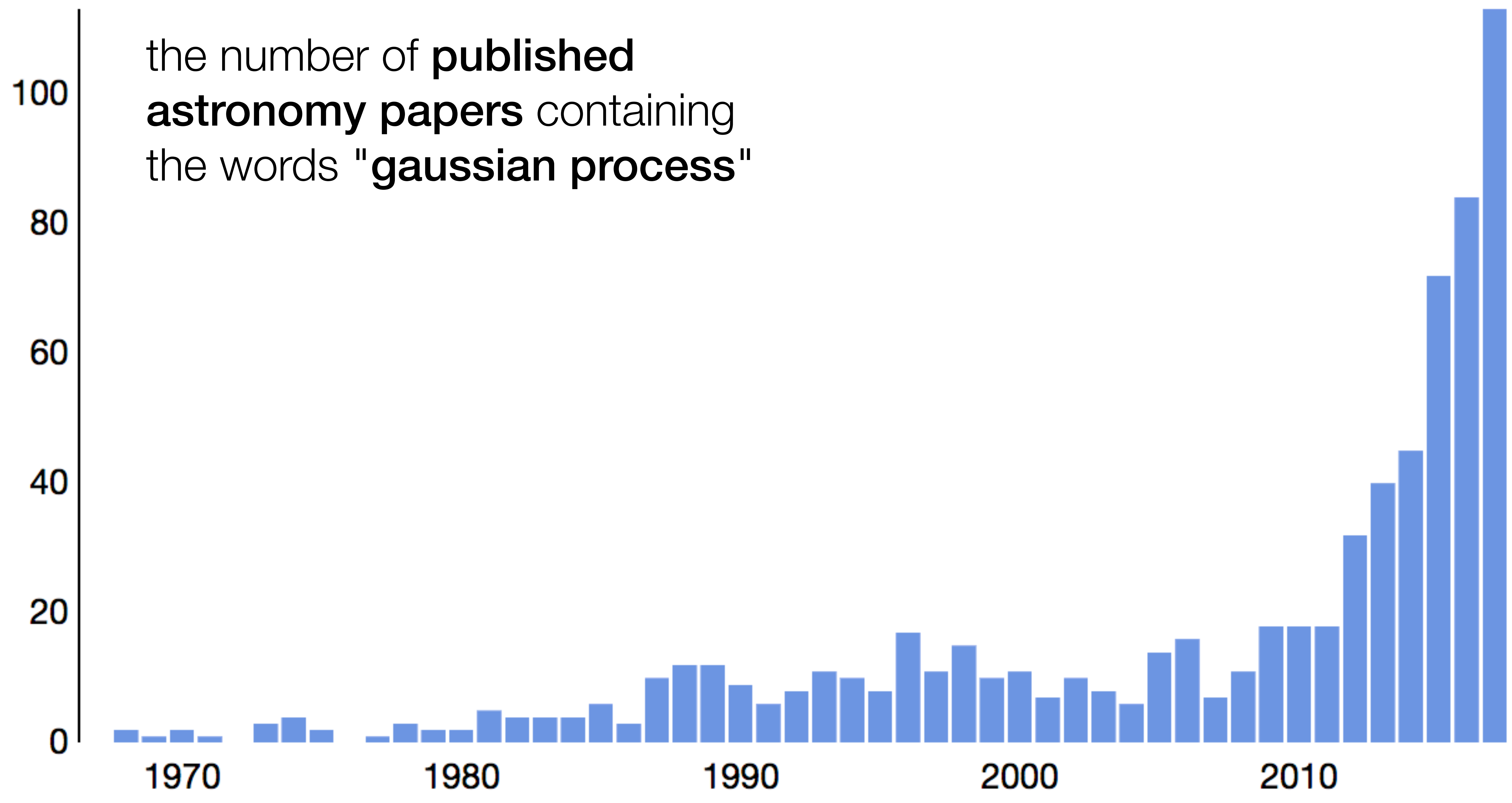
Flatiron Institute // dfm.io // github.com/dfm // @exoplaneteer

Resources

- a gaussianprocess.org/gpml
- b george.readthedocs.io
- c dfm.io/gp.js
- d github.com/dfm/gp
- e foreman.mackey@gmail.com

1

Gaussian **P**rocesses

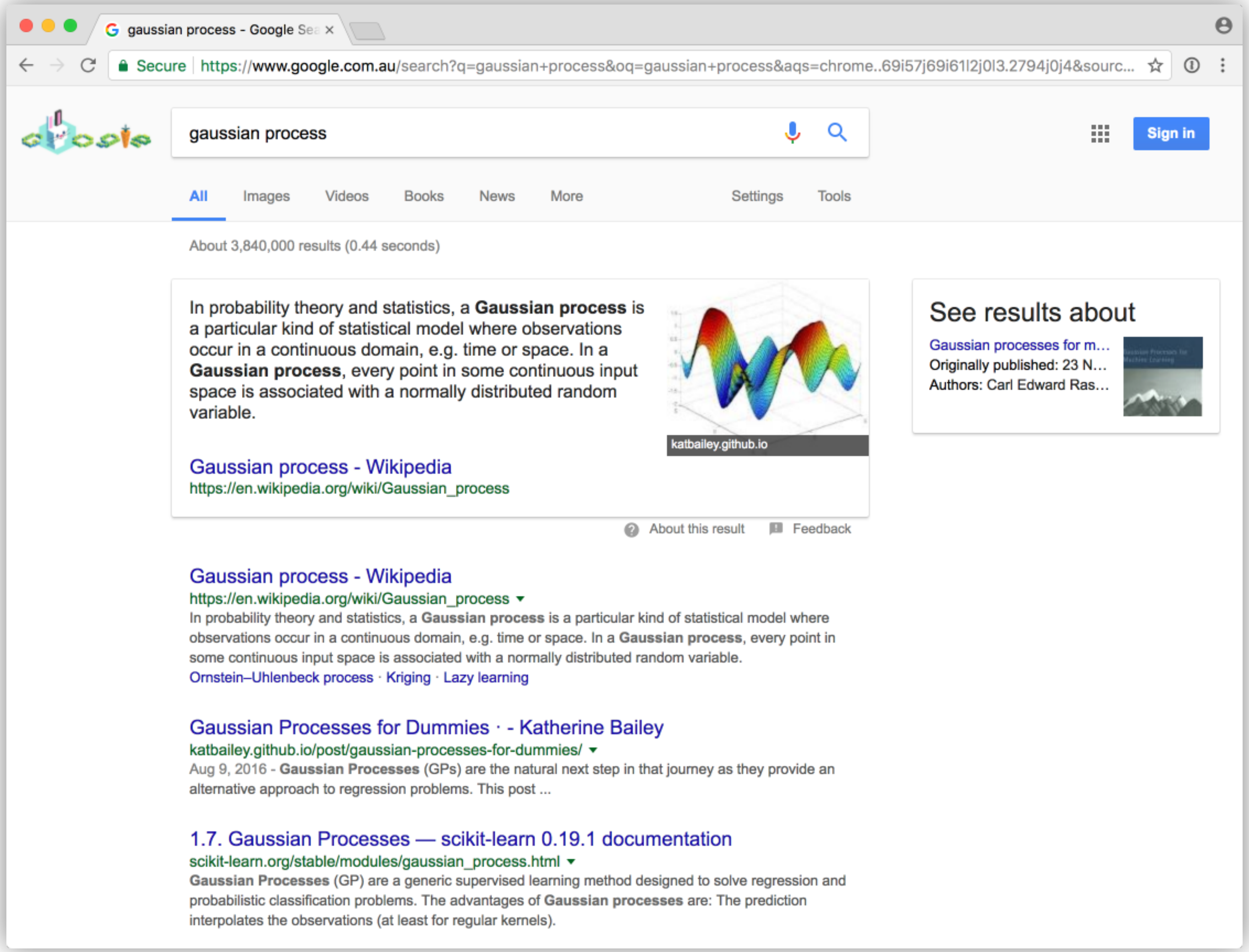


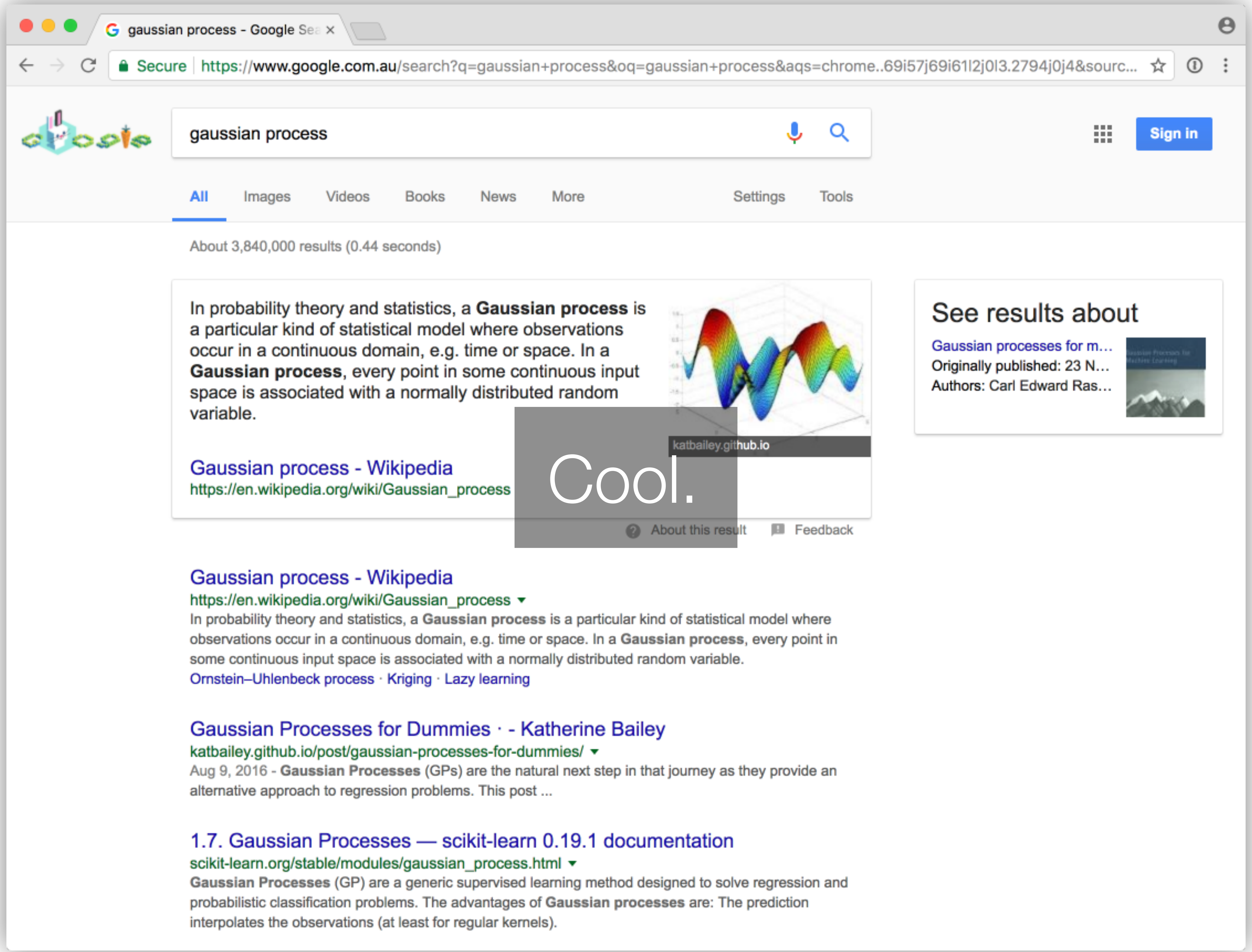
Screenshot from **ui.adsabs.harvard.edu**

Don't you think **you** should be using **Gaussian Processes**?

Photo by **Adi Goldstein** on Unsplash

Photo by **Adi Goldstein** on Unsplash


gaussian process - Google Sea
Secure https://www.google.com.au/search?q=gaussian+process&oq=gaussian+process&aqs=chrome..69i57j69i61l2j0l3.2794j0j4&sourc...
gaussian process
Sign in
All
Images
Videos
Books
News
More
Settings
Tools
About 3,840,000 results (0.44 seconds)
In probability theory and statistics, a Gaussian process is a particular kind of statistical model where observations occur in a continuous domain, e.g. time or space. In a Gaussian process, every point in some continuous input space is associated with a normally distributed random variable.
katbailey.github.io
Gaussian process - Wikipedia
https://en.wikipedia.org/wiki/Gaussian_process
About this result
Feedback
Gaussian process - Wikipedia
https://en.wikipedia.org/wiki/Gaussian_process
In probability theory and statistics, a Gaussian process is a particular kind of statistical model where observations occur in a continuous domain, e.g. time or space. In a Gaussian process, every point in some continuous input space is associated with a normally distributed random variable.
Ornstein–Uhlenbeck process · Kriging · Lazy learning
Gaussian Processes for Dummies · - Katherine Bailey
katbailey.github.io/post/gaussian-processes-for-dummies/
Aug 9, 2016 - Gaussian Processes (GPs) are the natural next step in that journey as they provide an alternative approach to regression problems. This post ...
1.7. Gaussian Processes — scikit-learn 0.19.1 documentation
scikit-learn.org/stable/modules/gaussian_process.html
Gaussian Processes (GP) are a generic supervised learning method designed to solve regression and probabilistic classification problems. The advantages of Gaussian processes are: The prediction interpolates the observations (at least for regular kernels).
See results about
Gaussian processes for m...
Originally published: 23 N...
Authors: Carl Edward Ras...


gaussian process - Google Sea
Secure
https://www.google.com.au/search?q=gaussian+process&oq=gaussian+process&aqs=chrome..69i57j69i61l2j0l3.2794j0j4&sourc...
gaussian process
Sign in
All
Images
Videos
Books
News
More
Settings
Tools
About 3,840,000 results (0.44 seconds)
In probability theory and statistics, a Gaussian process is a particular kind of statistical model where observations occur in a continuous domain, e.g. time or space. In a Gaussian process, every point in some continuous input space is associated with a normally distributed random variable.
katbailey.github.io
Gaussian process - Wikipedia
https://en.wikipedia.org/wiki/Gaussian_process
Cool.
About this result
Feedback
See results about
Gaussian processes for m...
Originally published: 23 N...
Authors: Carl Edward Ras...
Gaussian process - Wikipedia
https://en.wikipedia.org/wiki/Gaussian_process
In probability theory and statistics, a Gaussian process is a particular kind of statistical model where observations occur in a continuous domain, e.g. time or space. In a Gaussian process, every point in some continuous input space is associated with a normally distributed random variable.
Ornstein–Uhlenbeck process · Kriging · Lazy learning
Gaussian Processes for Dummies · - Katherine Bailey
katbailey.github.io/post/gaussian-processes-for-dummies/
Aug 9, 2016 - Gaussian Processes (GPs) are the natural next step in that journey as they provide an alternative approach to regression problems. This post ...
1.7. Gaussian Processes — scikit-learn 0.19.1 documentation
scikit-learn.org/stable/modules/gaussian_process.html
Gaussian Processes (GP) are a generic supervised learning method designed to solve regression and probabilistic classification problems. The advantages of Gaussian processes are: The prediction interpolates the observations (at least for regular kernels).

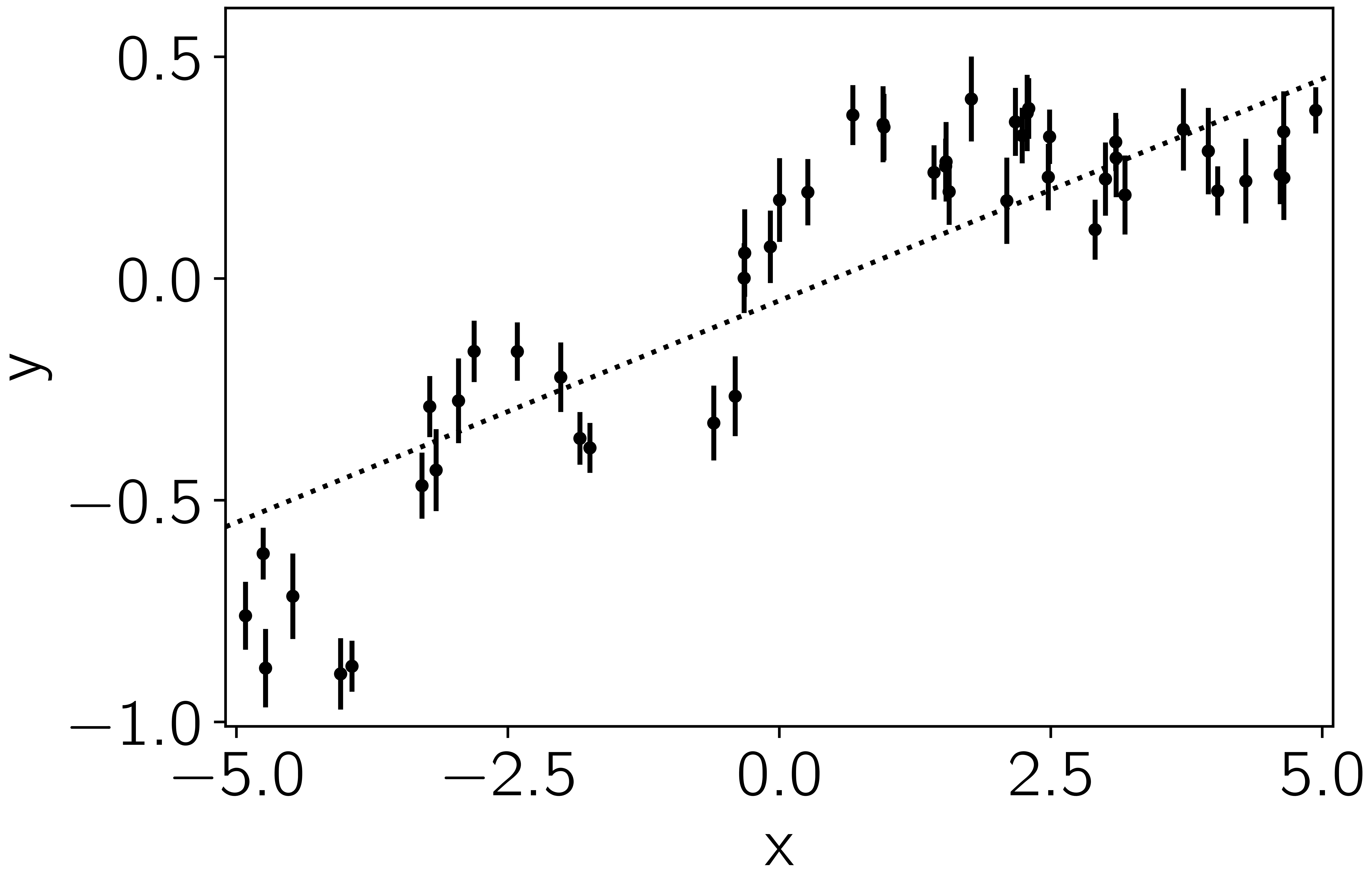
Today

1. **Why?**
2. **What?**
3. **How?**

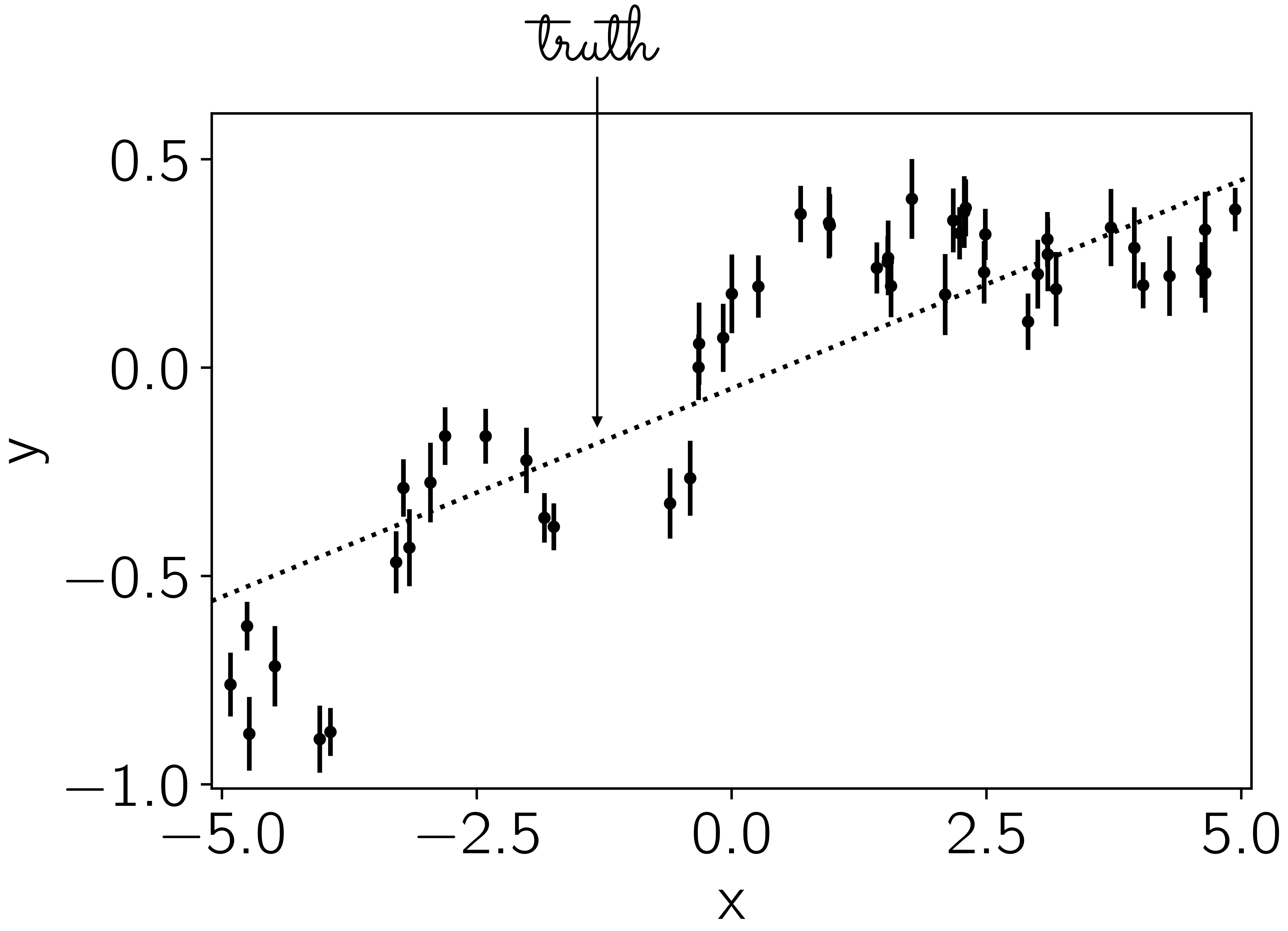
2

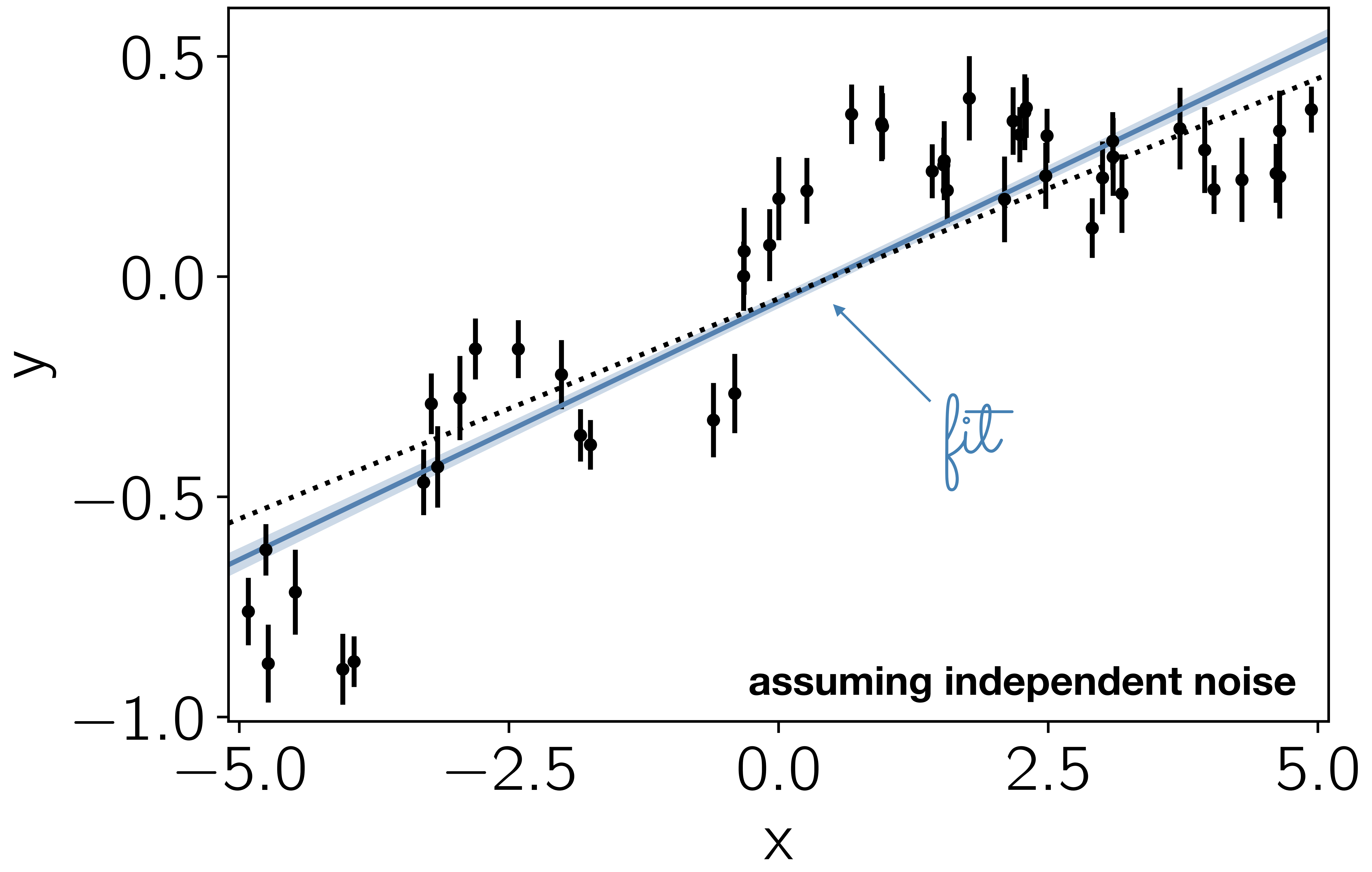
The **importance** of **correlated noise**

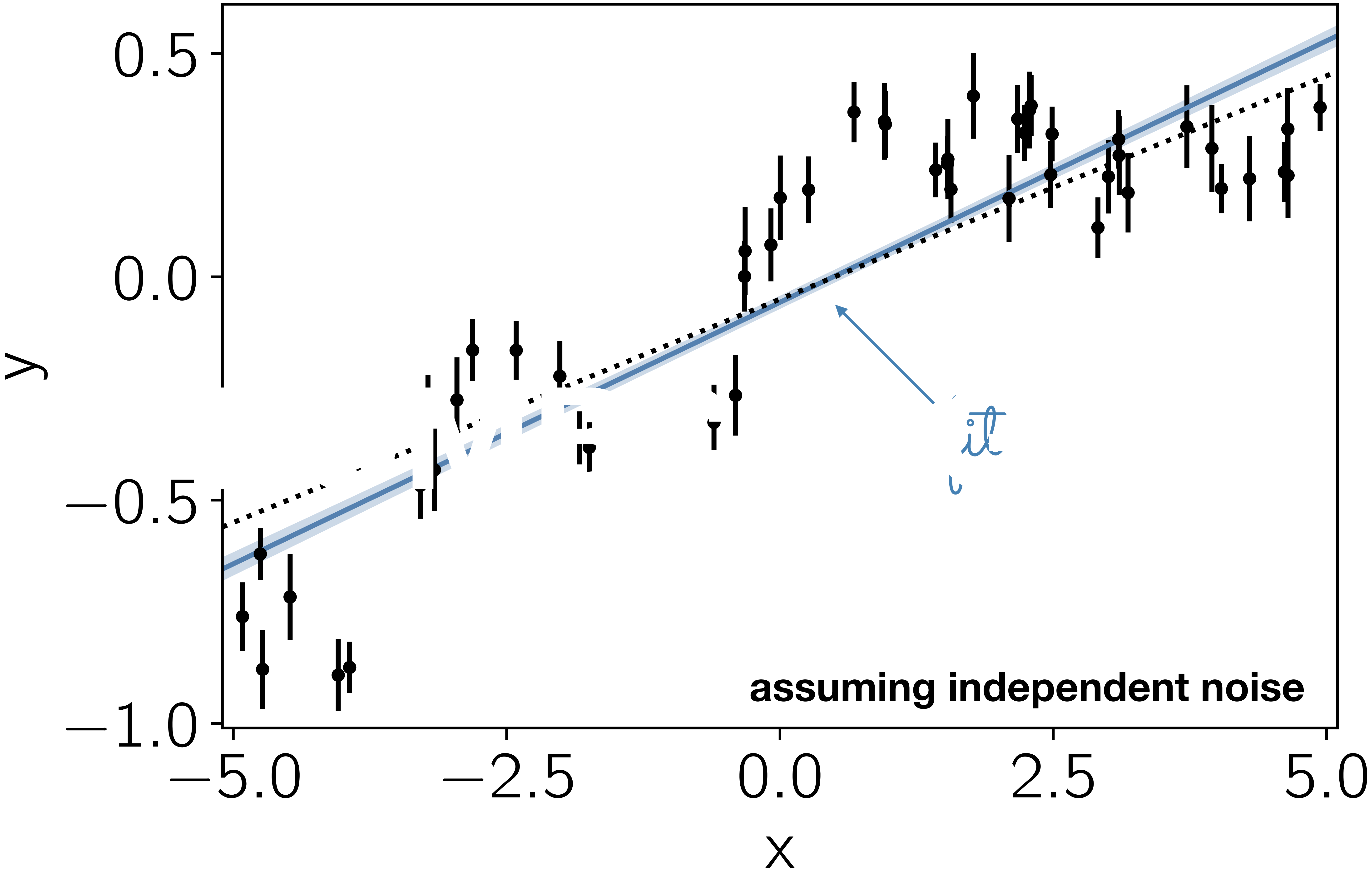
a motivating example



0.5
0.0
−0.5
−1.0
y
−5.0
−2.5
0.0
2.5
5.0
x

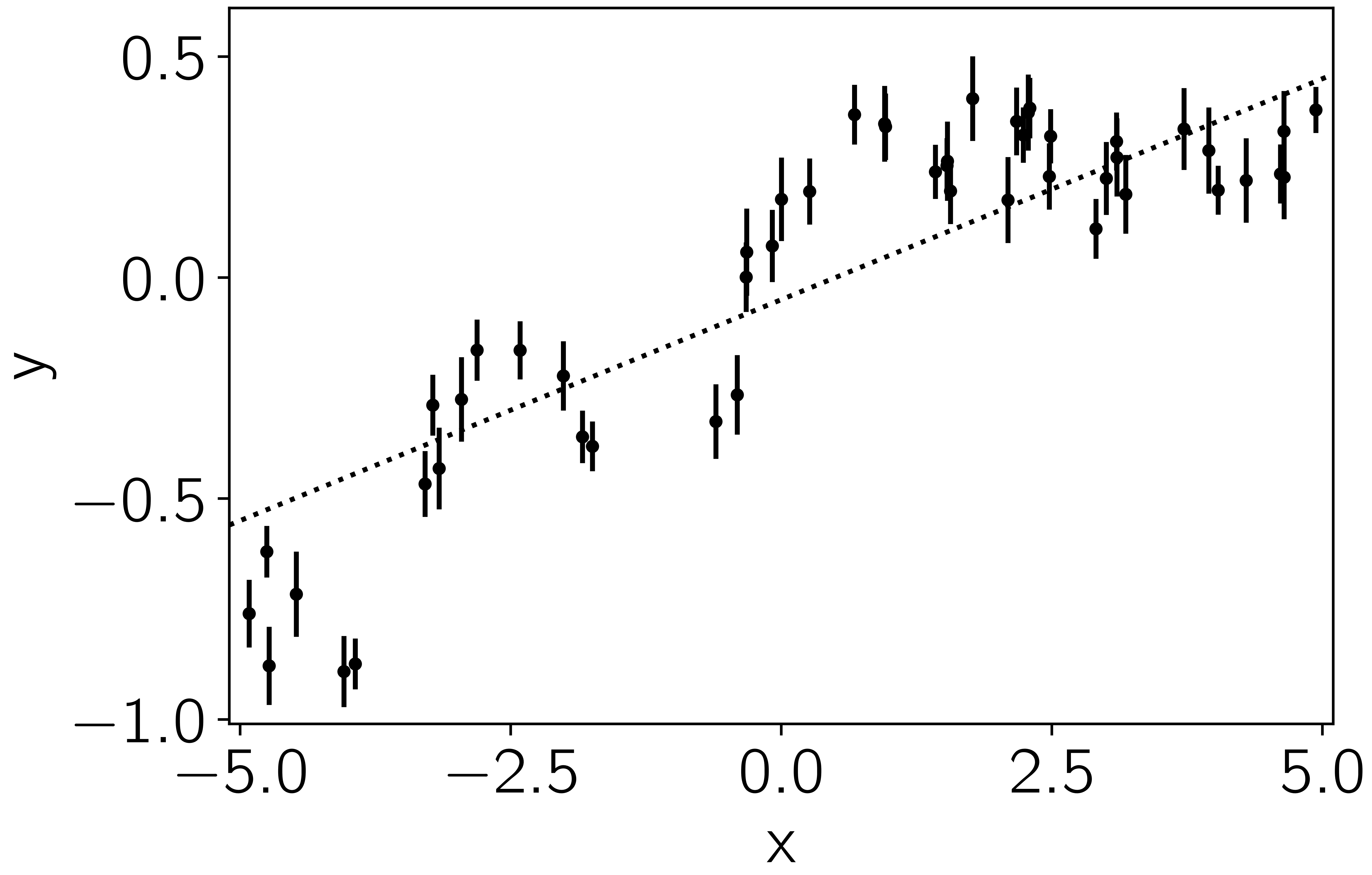

truth
0.5
0.0
−0.5
−1.0
y
−5.0
−2.5
0.0
2.5
5.0
x

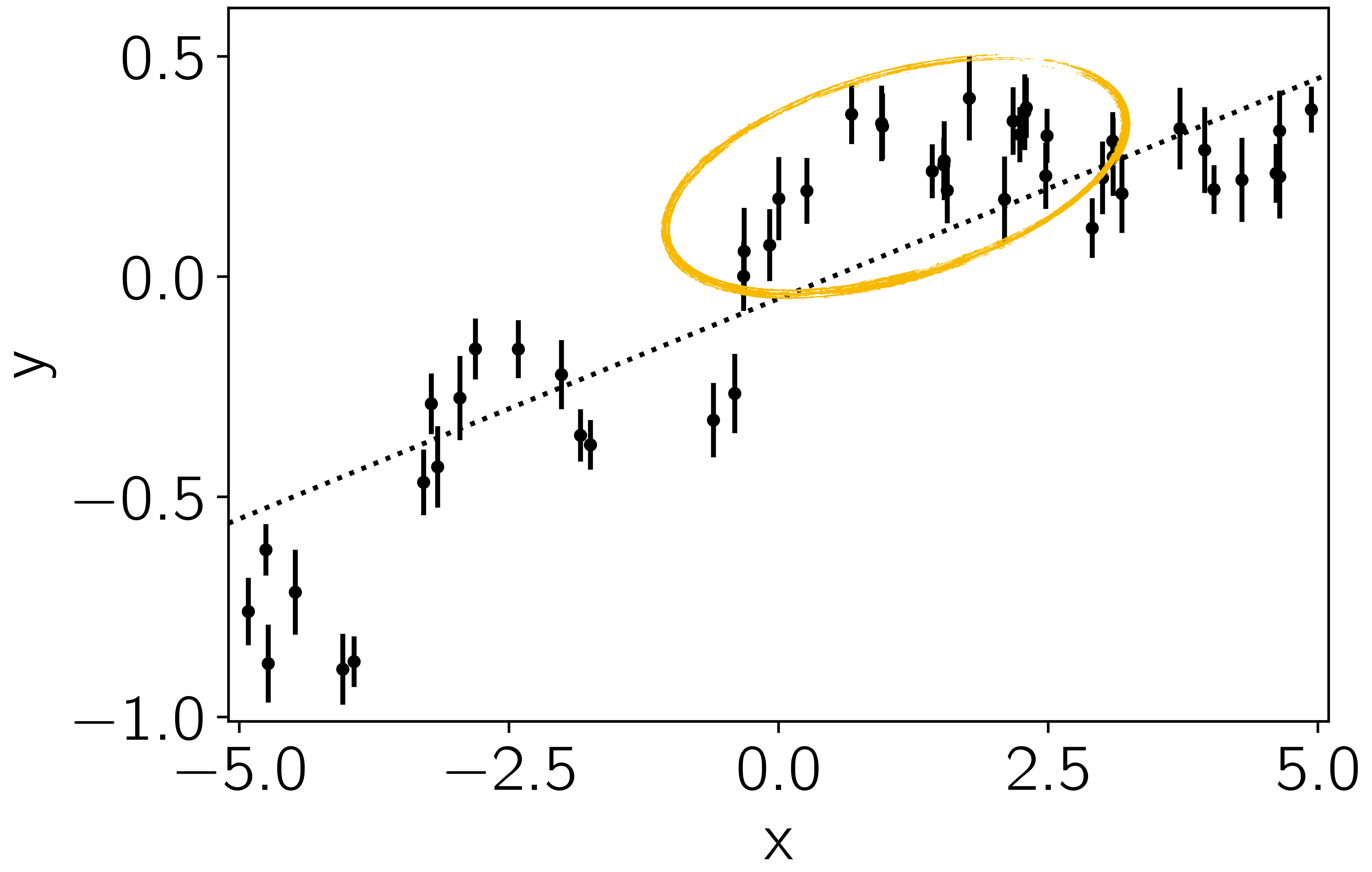

0.5
0.0
−0.5
−1.0
−5.0
−2.5
0.0
2.5
5.0
y
x
fit
assuming independent noise



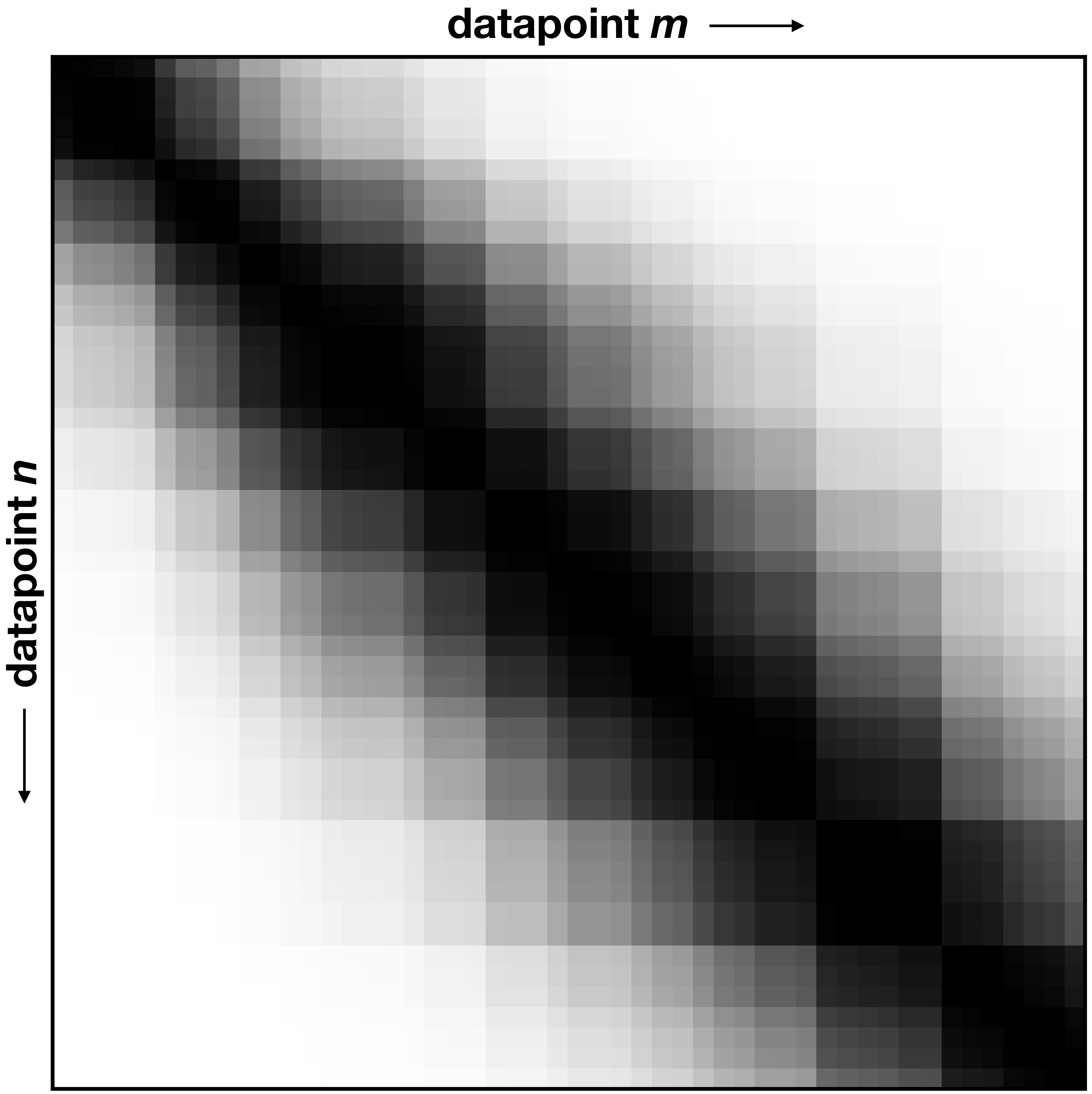
0.5
0.0
−0.5
−1.0
y
−5.0
−2.5
0.0
2.5
5.0
x
assuming independent noise

What **happened**?


0.5
0.0
−0.5
−1.0
y
−5.0
−2.5
0.0
2.5
5.0
x


0.5
0.0
−0.5
−1.0
y
−5.0
−2.5
0.0
2.5
5.0
x

The true **covariance** matrix



$$\log p(\{y_n\}\,|\,\theta) = -\frac{1}{2}\,\sum_{n=1}^{N}\left[\frac{[y_n - m_n]^2}{{\sigma_n}^2} + \log(2\,\pi\,{\sigma_n}^2)\right]$$

↓

$$\log p(\{y_n\}\,|\,\theta) = -\frac{1}{2}\,\boldsymbol{r}^{\mathrm{T}}\,C^{-1}\,\boldsymbol{r} - \frac{1}{2}\log\det C - \frac{N}{2}\log(2\,\pi)$$

if...

$$\boldsymbol{r} = \begin{pmatrix} y_1 - m_1 \\ \vdots \\ y_N - m_N \end{pmatrix}$$ and $$C = \begin{pmatrix} {\sigma_1}^2 & & 0 \\ & \ddots & \\ 0 & & {\sigma_N}^2 \end{pmatrix}$$

$$\log p(\{y_n\}\,|\,\theta) = -\frac{1}{2}\sum_{n=1}^{N}\left[\frac{[y_n - m_n]^2}{{\sigma_n}^2} + \log(2\,\pi\,{\sigma_n}^2)\right]$$

$$\log p(\{y_n\}\,|\,\theta) = -\frac{1}{2}\,\boldsymbol{r}^{\mathrm{T}}\,C^{-1}\,\boldsymbol{r} - \frac{1}{2}\log\det C - \frac{N}{2}\log(2\,\pi)$$

if...

$$\boldsymbol{r} = \begin{pmatrix} y_1 - m_1 \\ \vdots \\ y_N - m_N \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} {\sigma_1}^2 & & 0 \\ & \ddots & \\ 0 & & {\sigma_N}^2 \end{pmatrix}$$

$$\log p(\{y_n\}\,|\,\theta) = -\frac{1}{2}\sum_{n=1}^{N}\left[\frac{[y_n - m_n]^2}{{\sigma_n}^2} + \log(2\,\pi\,{\sigma_n}^2)\right]$$

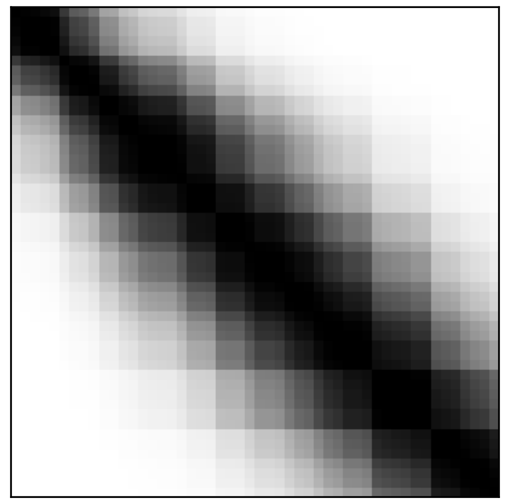
$$\log p(\{y_n\}\,|\,\theta) = -\frac{1}{2}\,\boldsymbol{r}^{\mathrm{T}}\,C^{-1}\,\boldsymbol{r} - \frac{1}{2}\log\det C - \frac{N}{2}\log(2\,\pi)$$

if...

$$\boldsymbol{r} = \begin{pmatrix} y_1 - m_1 \\ \vdots \\ y_N - m_N \end{pmatrix}$$

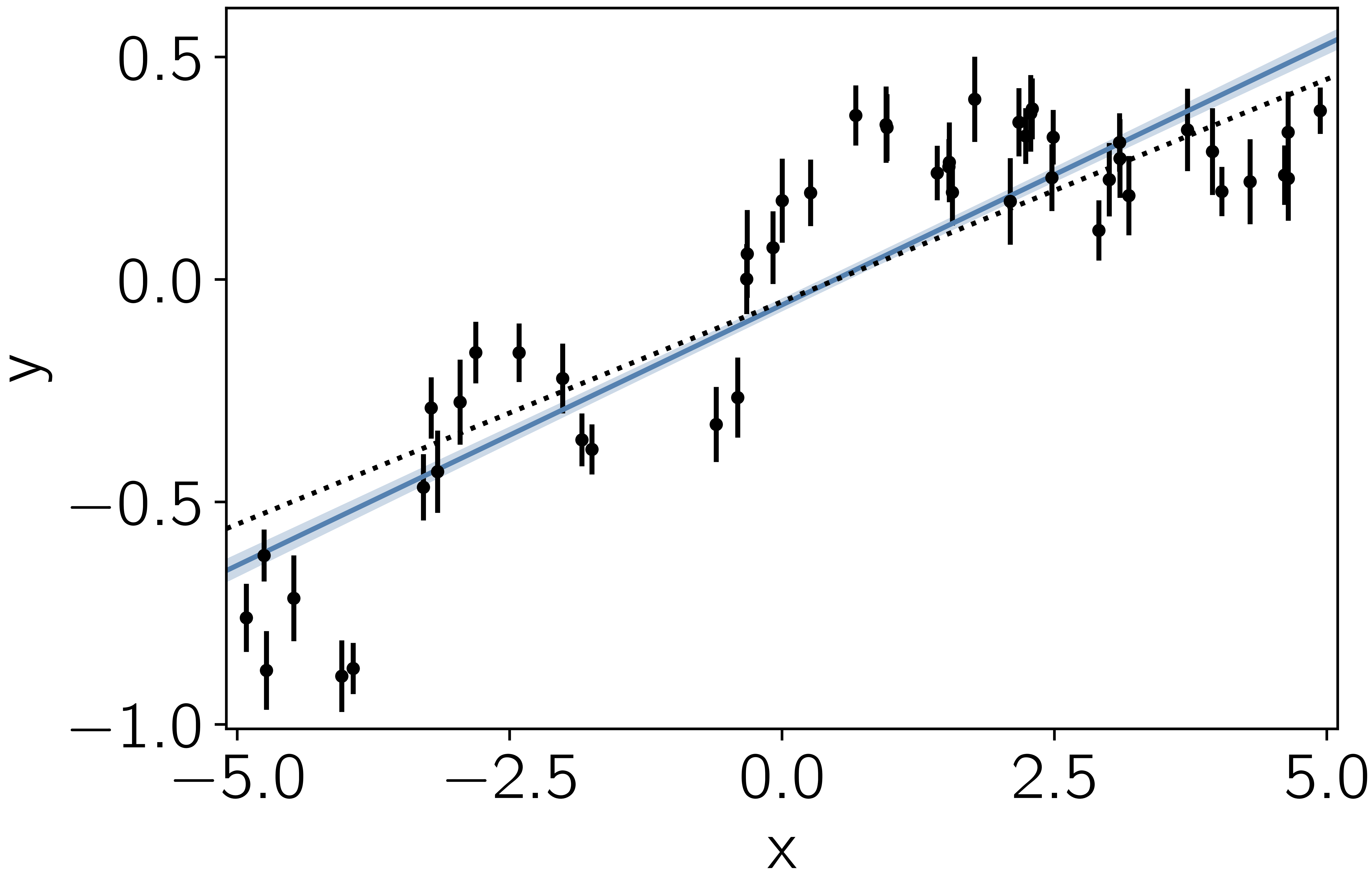
and

$C = \Bigg($

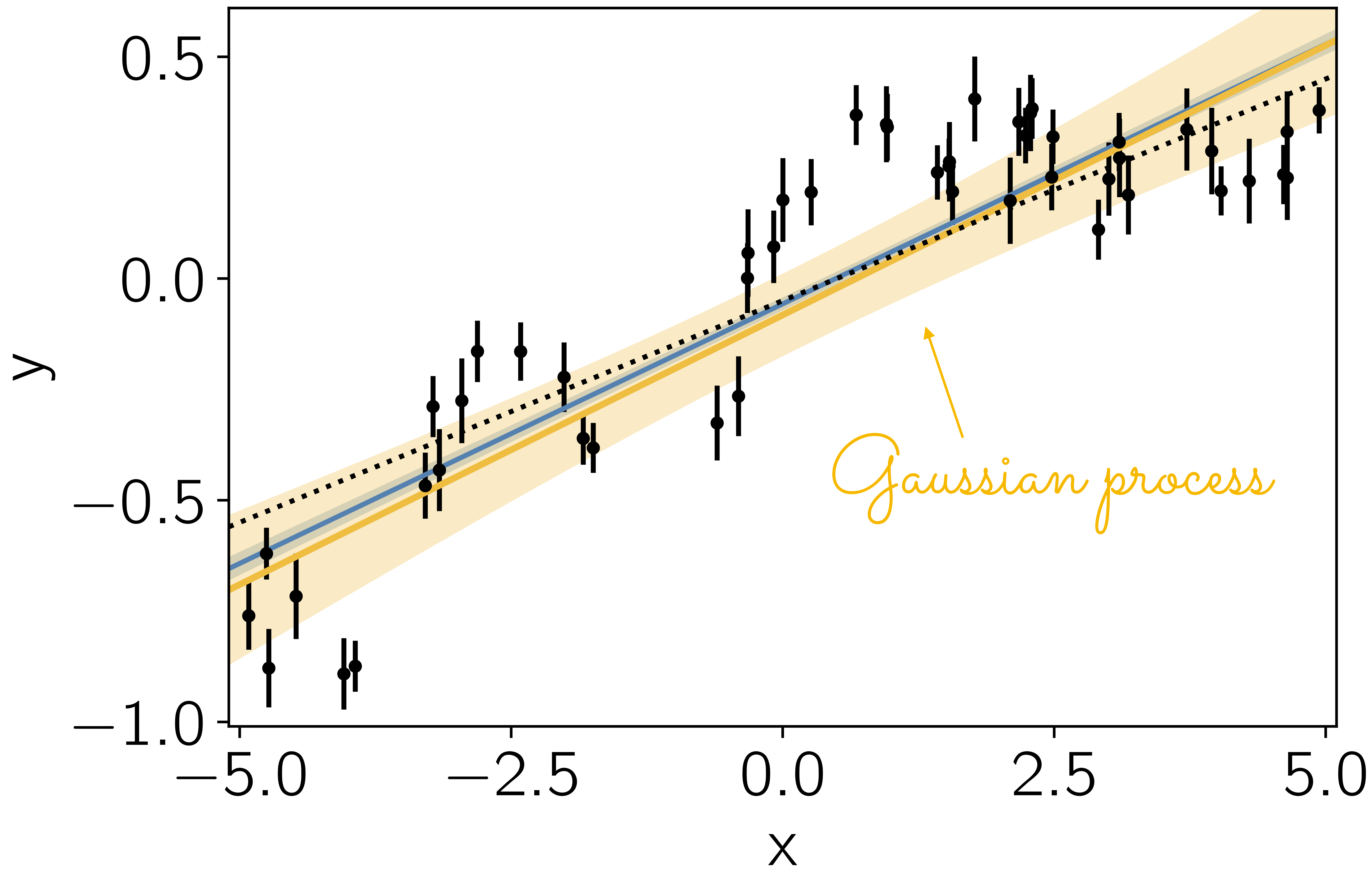


$^{*}\Bigg)$

** Note: this part has **everything** to do with Gaussian processes*

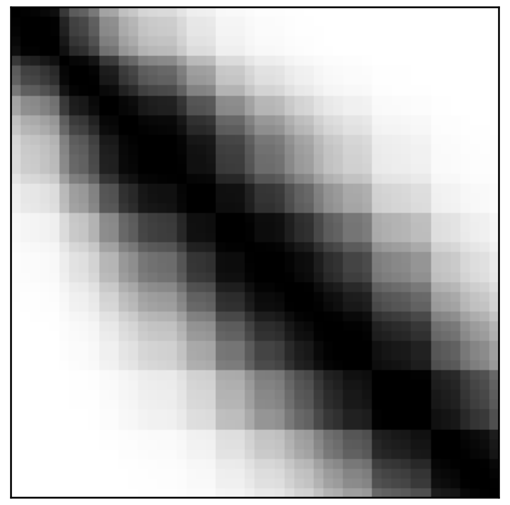


0.5
0.0
−0.5
−1.0
y
−5.0
−2.5
0.0
2.5
5.0
x



0.5
0.0
−0.5
−1.0
y
−5.0
−2.5
0.0
2.5
5.0
x
Gaussian process

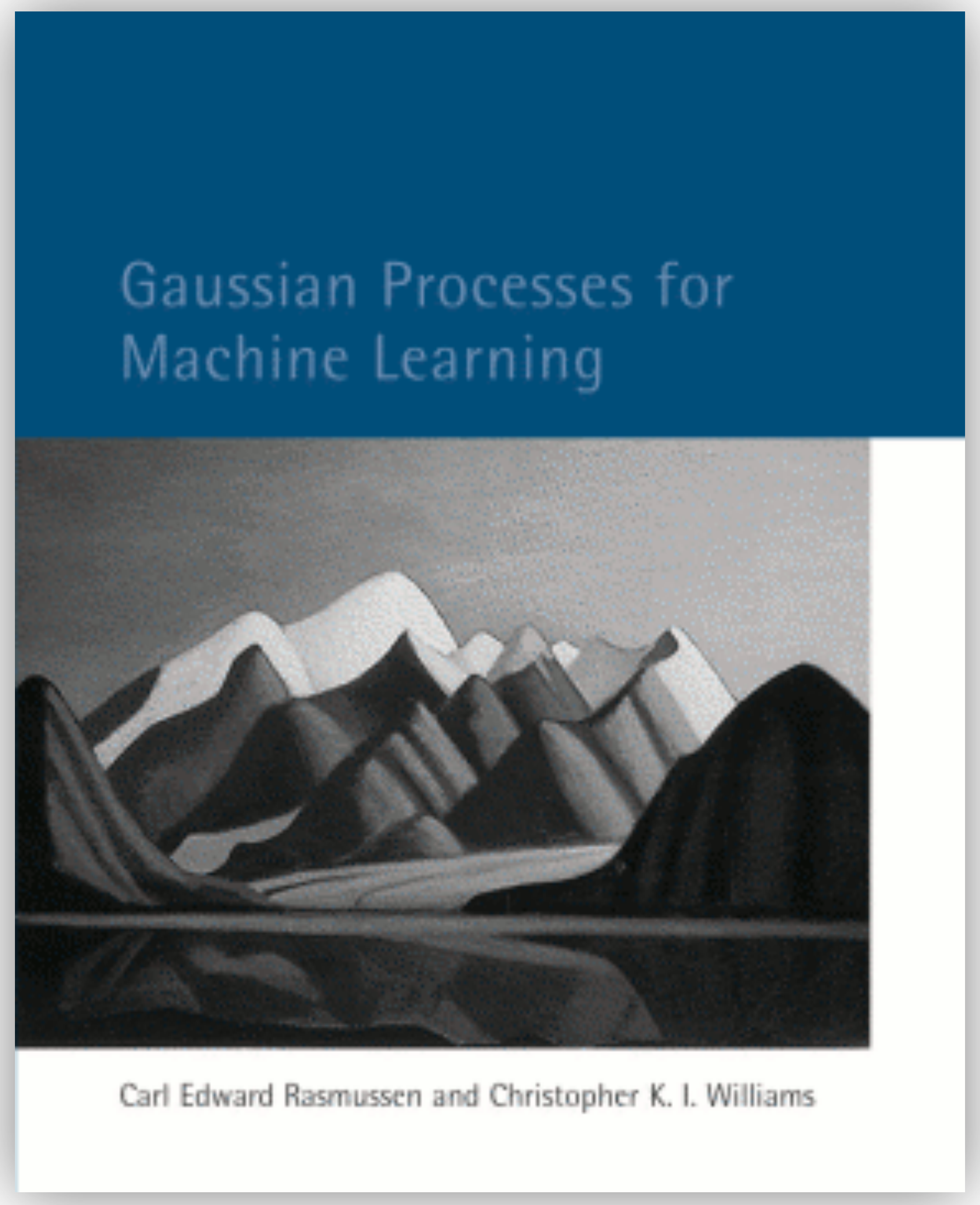
Excellent.

But: we don't know  *

** we need to <u>fit</u> for it.*

3

The **math** of **Gaussian processes**



Rasmussen & Williams
gaussianprocess.org/gpml

parameters of
mean model

covariance
matrix

$$\log p(\{y_n\} \,|\, \theta,\, \alpha) = -\frac{1}{2}\,\boldsymbol{r}_\theta{}^{\mathrm{T}}\, K_\alpha{}^{-1}\,\boldsymbol{r}_\theta - \frac{1}{2}\log\det K_\alpha - \frac{N}{2}\log(2\,\pi)$$

parameters of
covariance model

residual
vector

parameters of mean model

covariance matrix

$$\log p(\{y_n\}\,|\,\theta,\,\alpha) = -\frac{1}{2}\,\boldsymbol{r}_\theta{}^{\mathrm{T}}\,K_\alpha{}^{-1}\,\boldsymbol{r}_\theta - \frac{1}{2}\log\det K_\alpha - \frac{N}{2}\log(2\,\pi)$$

parameters of covariance model

residual vector

This is the equation for **an *N*-dimensional Gaussian***

* *hint: this is where the name comes from...*

generative model

covariance matrix

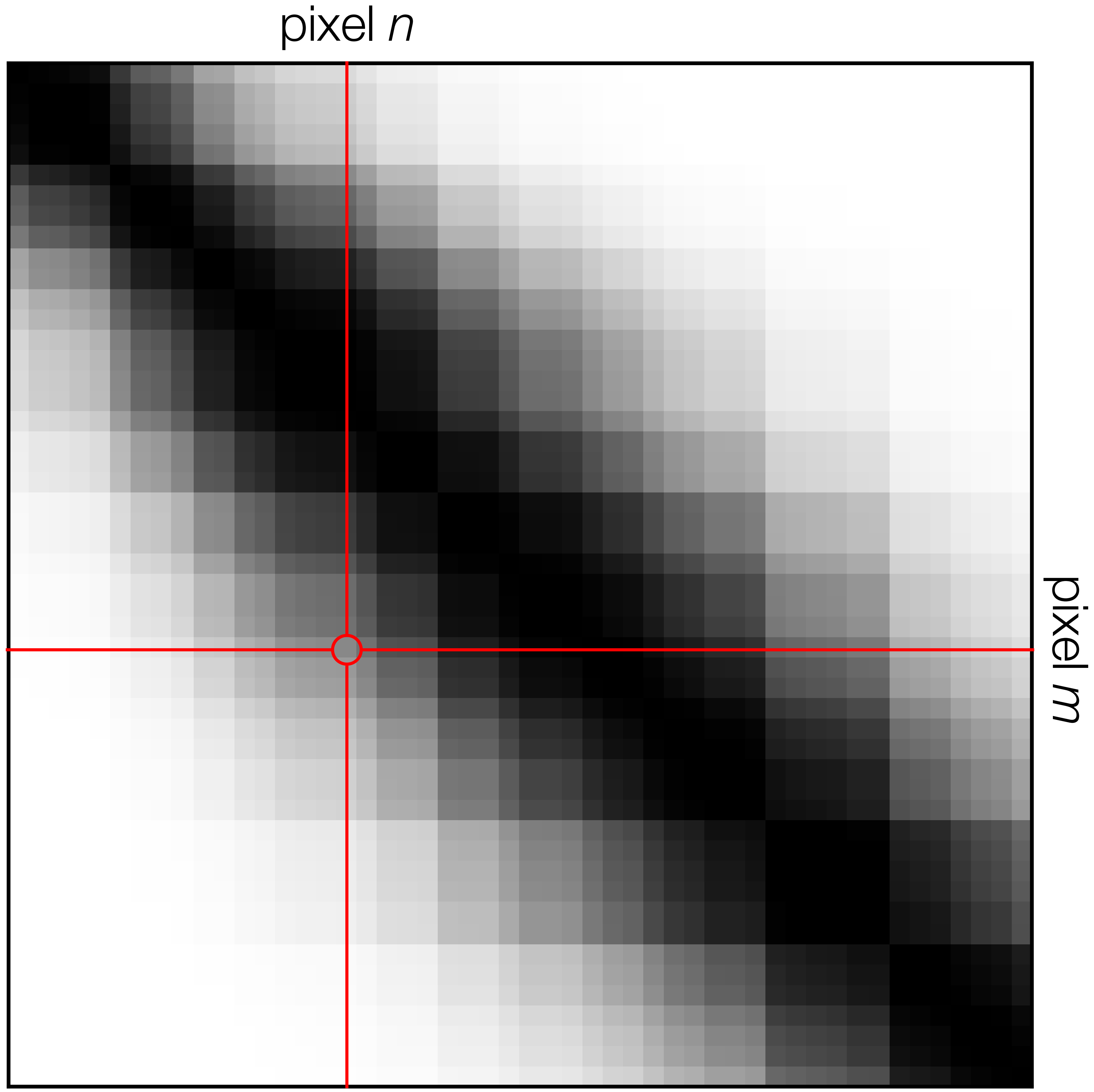
$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{m}_\theta,\, K_\alpha)$$

mean model

This is the equation for **an *N*-dimensional Gaussian***

* *hint: this is where the name comes from...*

$$[K_\alpha]_{nm} =$$



$$[K_\alpha]_{nm} =$$

$$[K_\alpha]_{nm} = {\sigma_n}^2 \, \delta_{nm}$$

$$[K_\alpha]_{nm} = {\sigma_n}^2\,\delta_{nm} + k_\alpha(\boldsymbol{x}_n,\, \boldsymbol{x}_m)$$

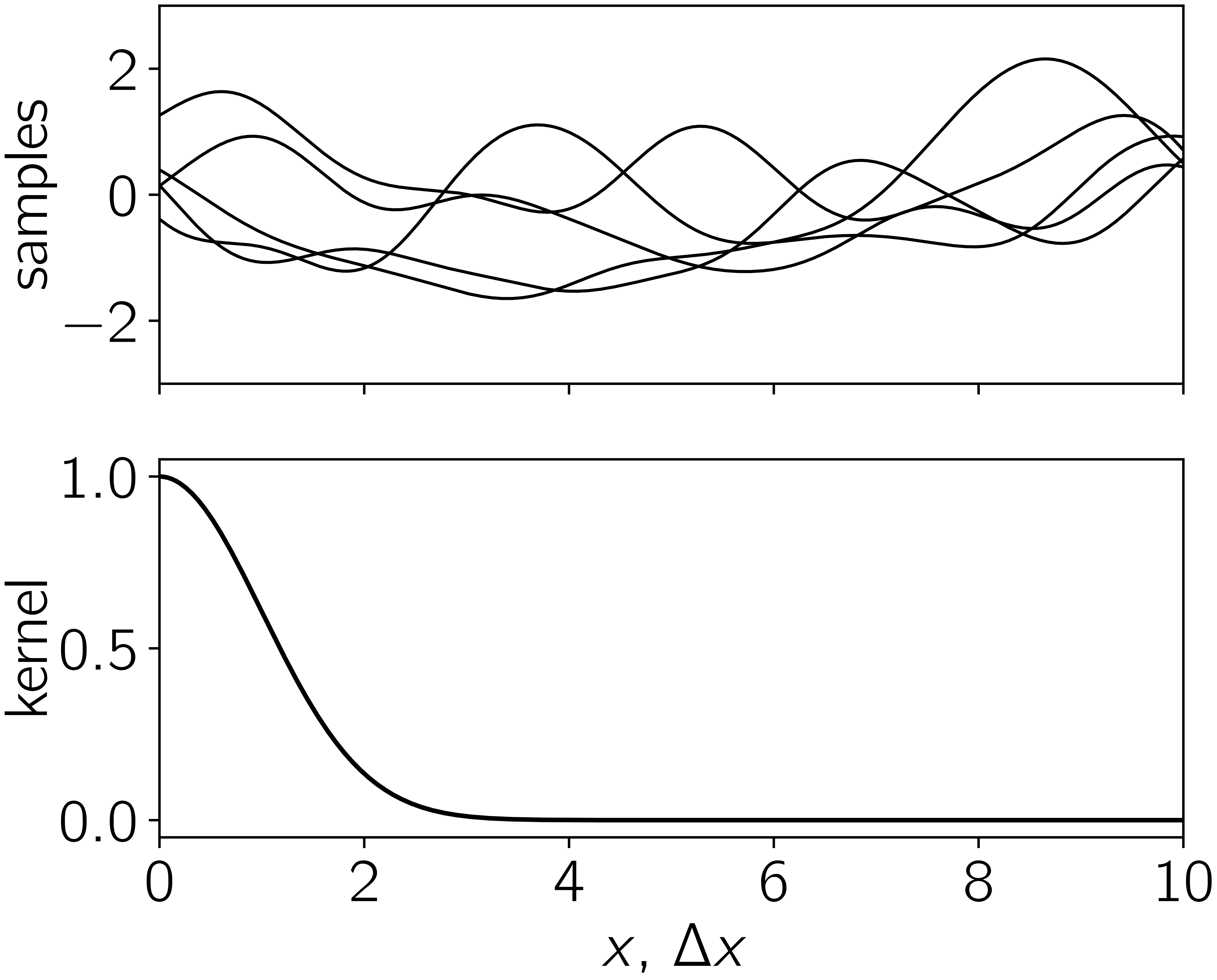
$$[K_\alpha]_{nm} = {\sigma_n}^2\,\delta_{nm} + k_\alpha(\boldsymbol{x}_n,\,\boldsymbol{x}_m)$$

the "kernel" function

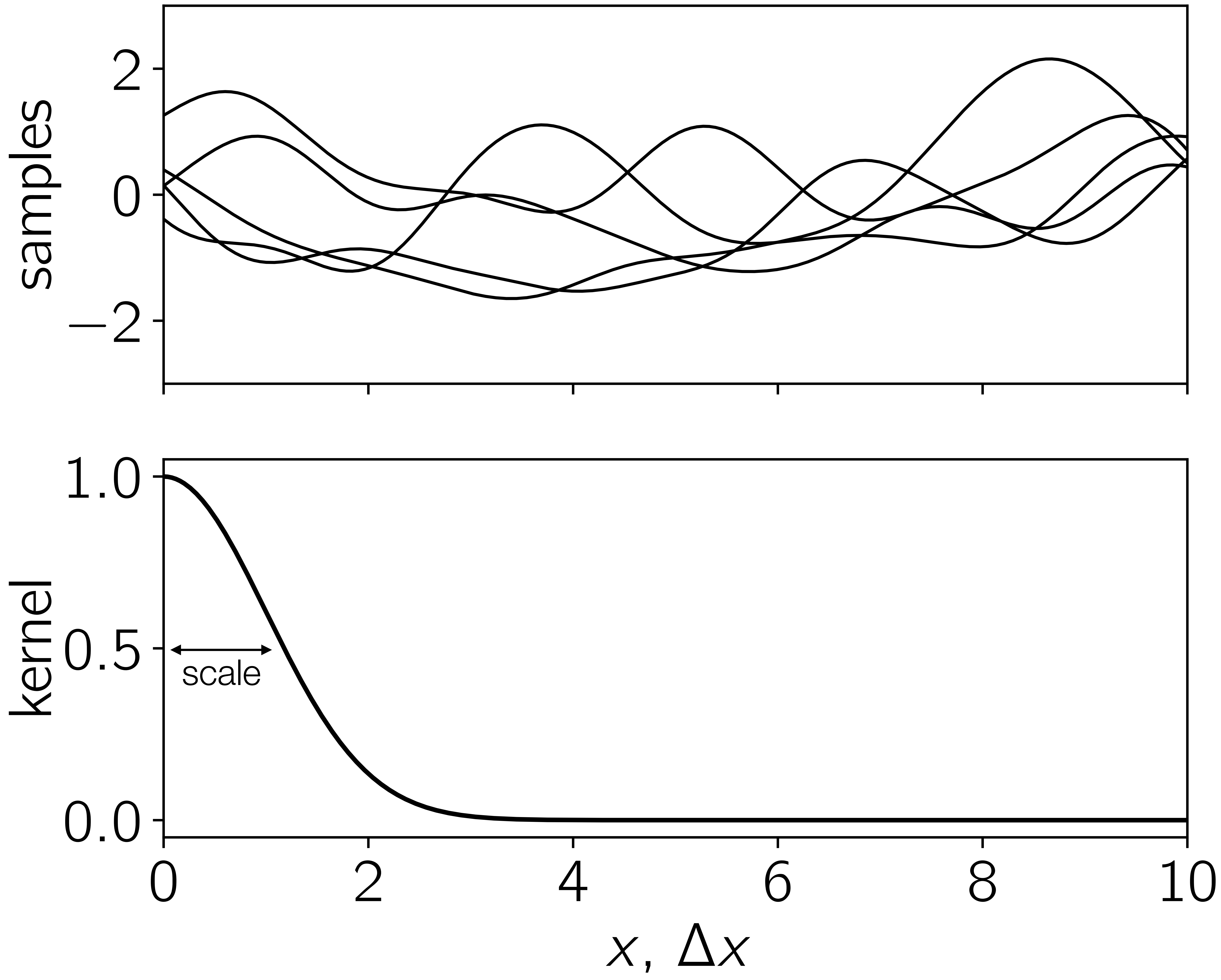
for example*:

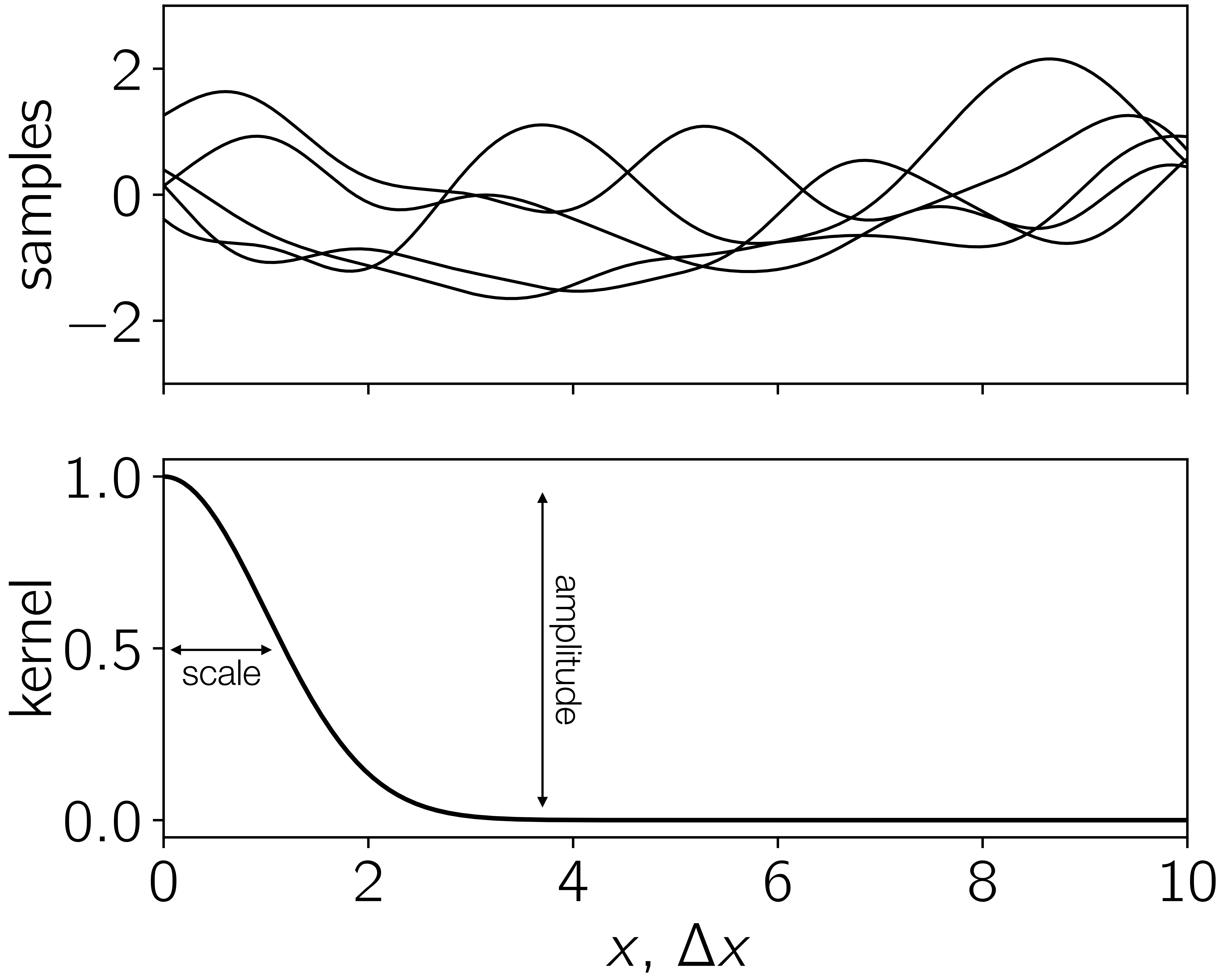
$$k_\alpha(\boldsymbol{x}_n,\,\boldsymbol{x}_m) = a^2\ \exp\left(-\frac{(x_n - x_m)^2}{2\,\tau^2}\right)$$

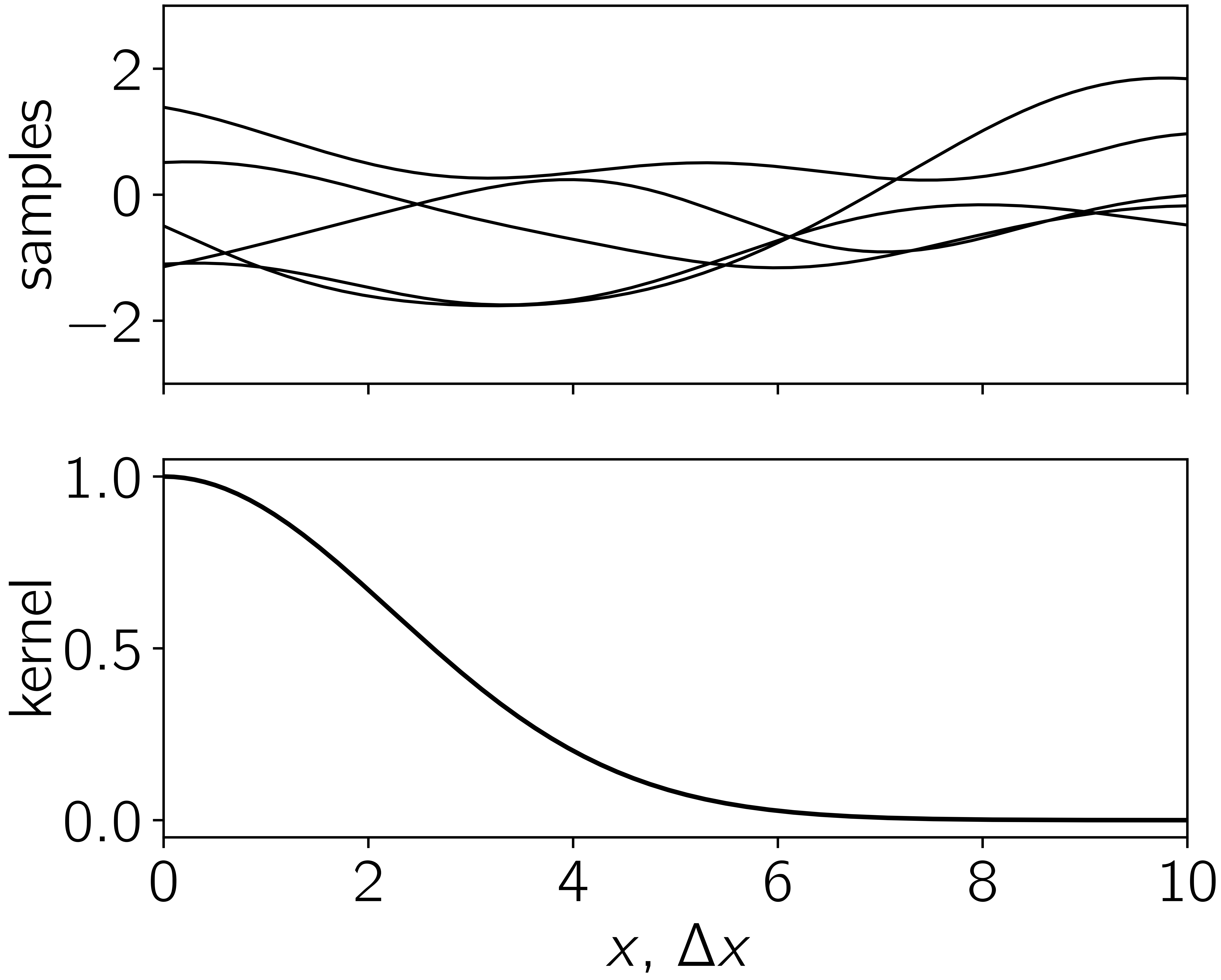
* *Note: this part has nothing to do with the name* ***Gaussian*** *process*

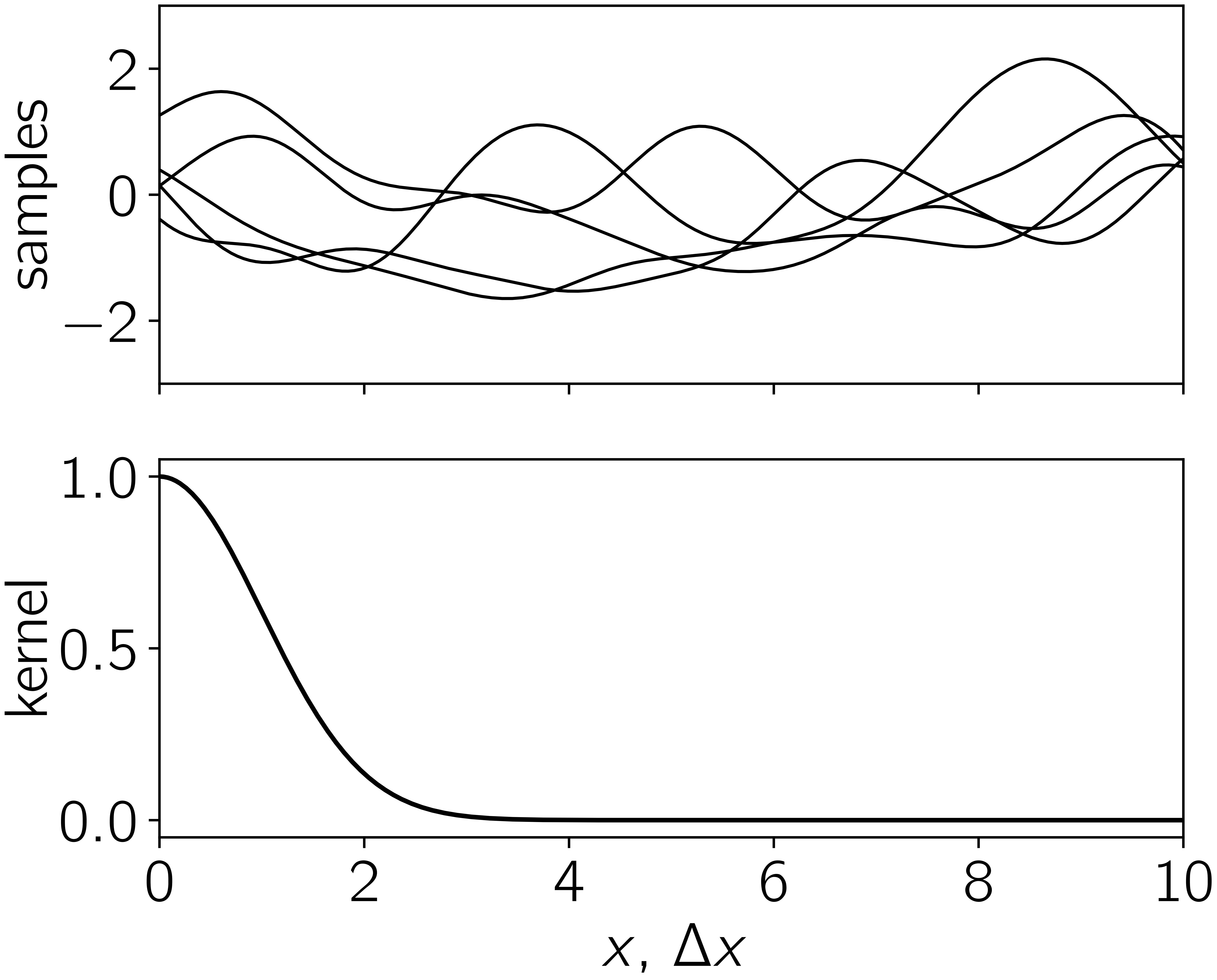


samples
2
0
−2
kernel
1.0
0.5
0.0
0
2
4
6
8
10
x, Δx

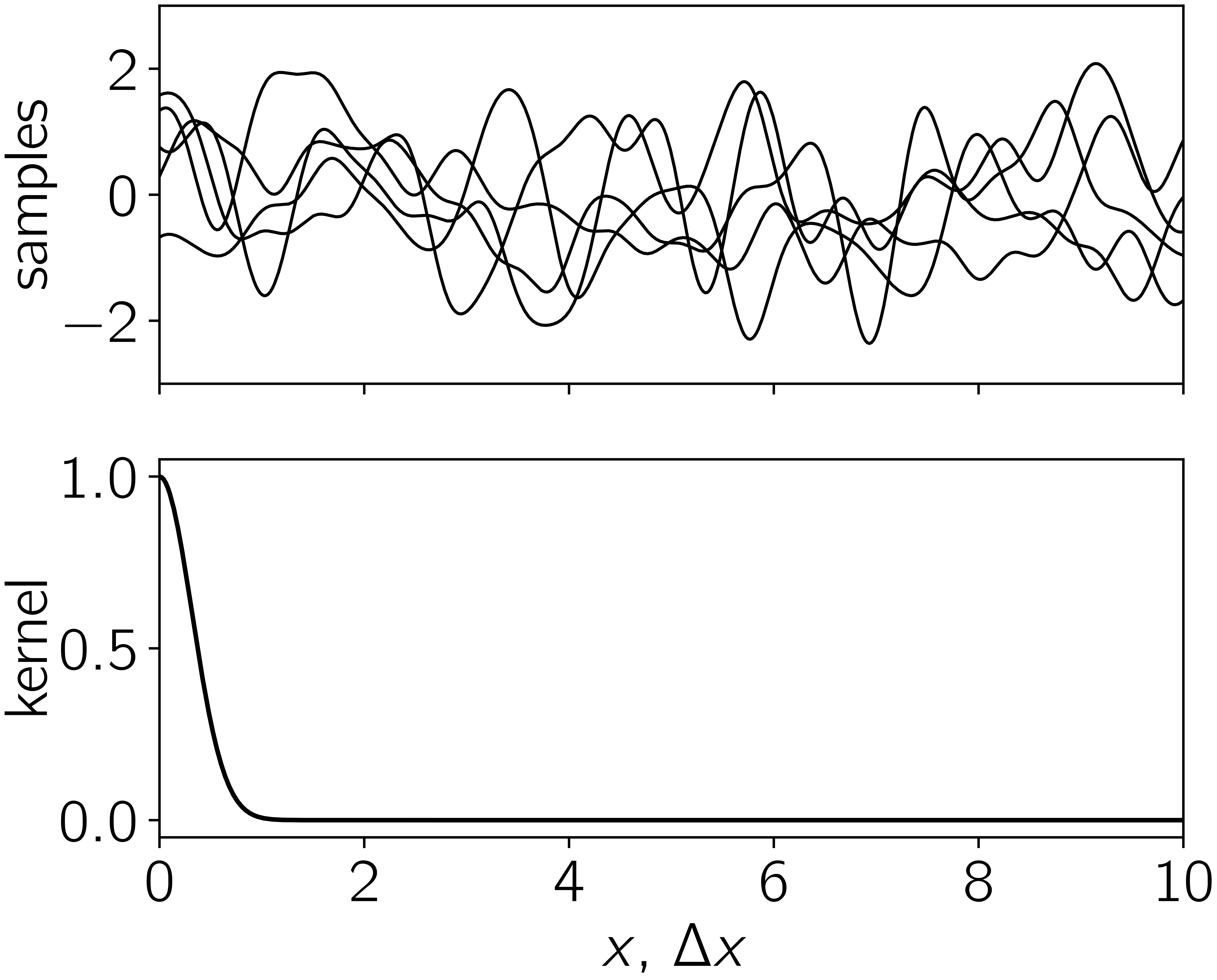

samples
2
0
−2
kernel
1.0
0.5
0.0
scale
0
2
4
6
8
10
x, Δx

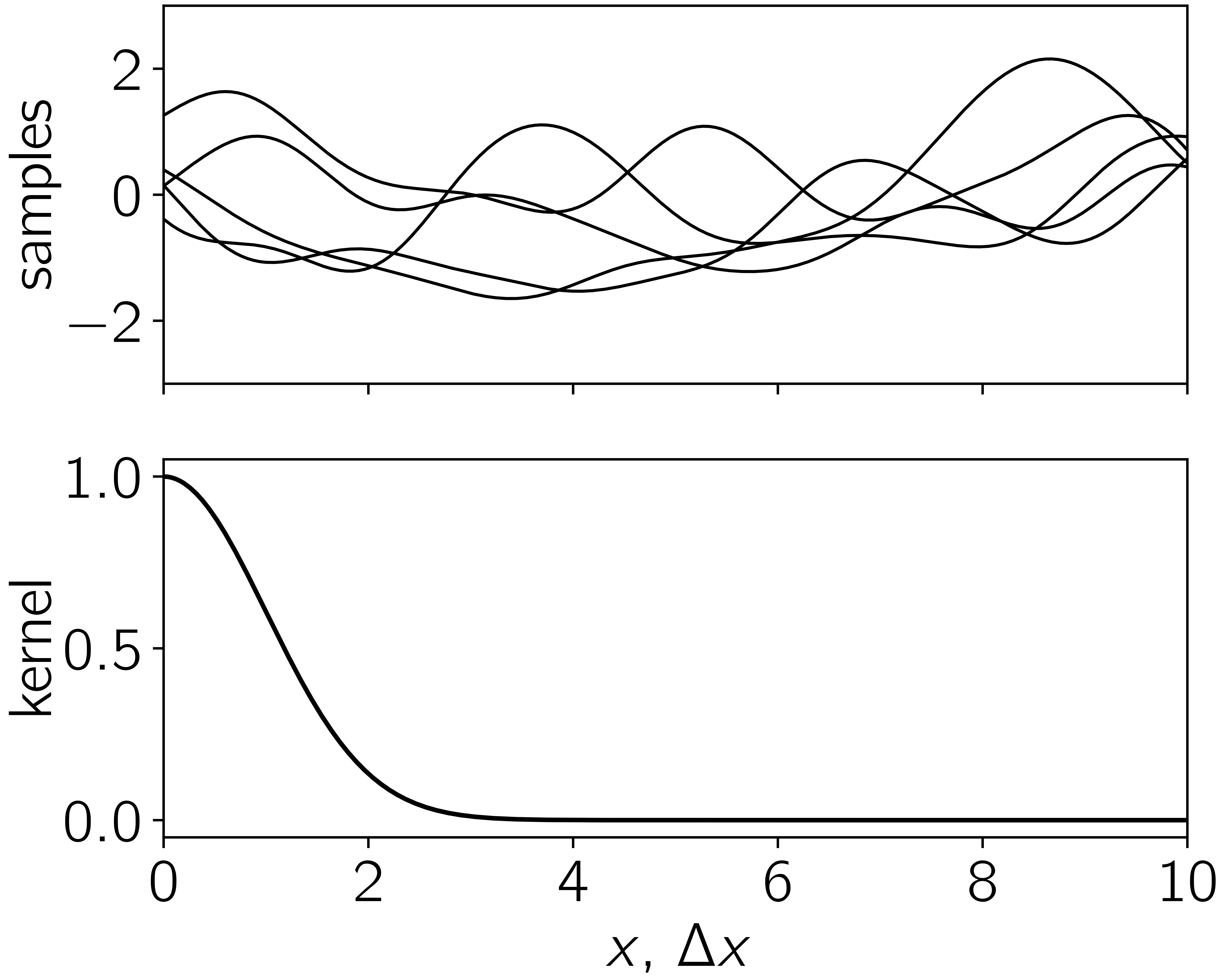

samples
2
0
−2
kernel
1.0
0.5
0.0
scale
amplitude
0
2
4
6
8
10
x, Δx


samples
2
0
−2
kernel
1.0
0.5
0.0
0
2
4
6
8
10
x, Δx

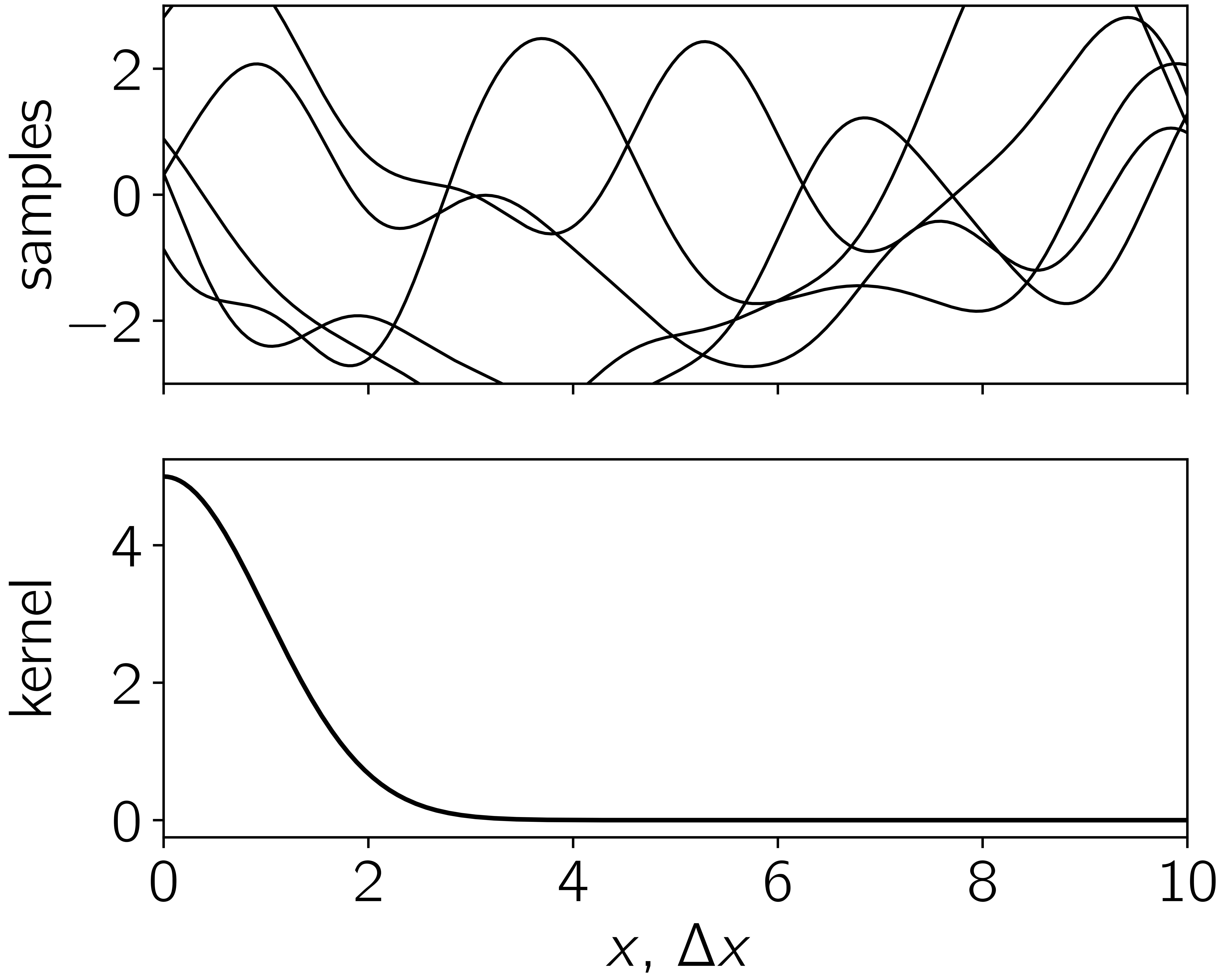


2
0
−2
samples
1.0
0.5
0.0
kernel
0
2
4
6
8
10
x, Δx


samples
2
0
−2
kernel
1.0
0.5
0.0
0
2
4
6
8
10
x, Δx



2
0
−2
samples
1.0
0.5
0.0
kernel
0
2
4
6
8
10
x, Δx



samples
2
0
−2
kernel
4
2
0
0
2
4
6
8
10
x, Δx

$$\log p(\{y_n\} \,|\, \theta,\, \alpha) = -\frac{1}{2}\,\boldsymbol{r}_\theta{}^{\mathrm{T}}\,K_\alpha{}^{-1}\,\boldsymbol{r}_\theta - \frac{1}{2}\log\det K_\alpha - \frac{N}{2}\log(2\,\pi)$$

a **drop-in replacement** for χ^2

A ~~fully~~ functional GP implementation in Python

```
import numpy as np

def gp_log_like(params, x, y, yerr):
    K = params[0]**2 * np.exp(-0.5*(x[:, None]-x[None, :])**2/params[1]**2)
    K[np.diag_indices_from(K)] += yerr**2
    ll = np.dot(y, np.linalg.solve(K, y))
    ll += np.linalg.slogdet(K)[1]
    return -0.5*ll
```

+ **scipy.optimize** or **emcee**

Using GPs in Python

1. george
2. scikit-learn
3. GPy
4. PyMC3
5. etc.

Using GPs in Python

1. george
2. scikit-learn
3. GPy
4. PyMC3
5. etc.

4

Why? The **problems** with Gaussian processes

$$\log p(\{y_n\}\,|\,\theta,\,\alpha) = -\frac{1}{2}\,\boldsymbol{r}_\theta{}^{\mathrm{T}}\,K_\alpha{}^{-1}\,\boldsymbol{r}_\theta - \frac{1}{2}\,\log\det K_\alpha - \frac{N}{2}\,\log(2\,\pi)$$

a

Choosing the kernel

$$\log p(\{y_n\} \,|\, \theta,\, \alpha) = -\frac{1}{2}\, \boldsymbol{r}_\theta{}^{\mathrm{T}}\, {K_\alpha}^{-1}\, \boldsymbol{r}_\theta - \frac{1}{2} \log \det K_\alpha - \frac{N}{2} \log(2\,\pi)$$

a

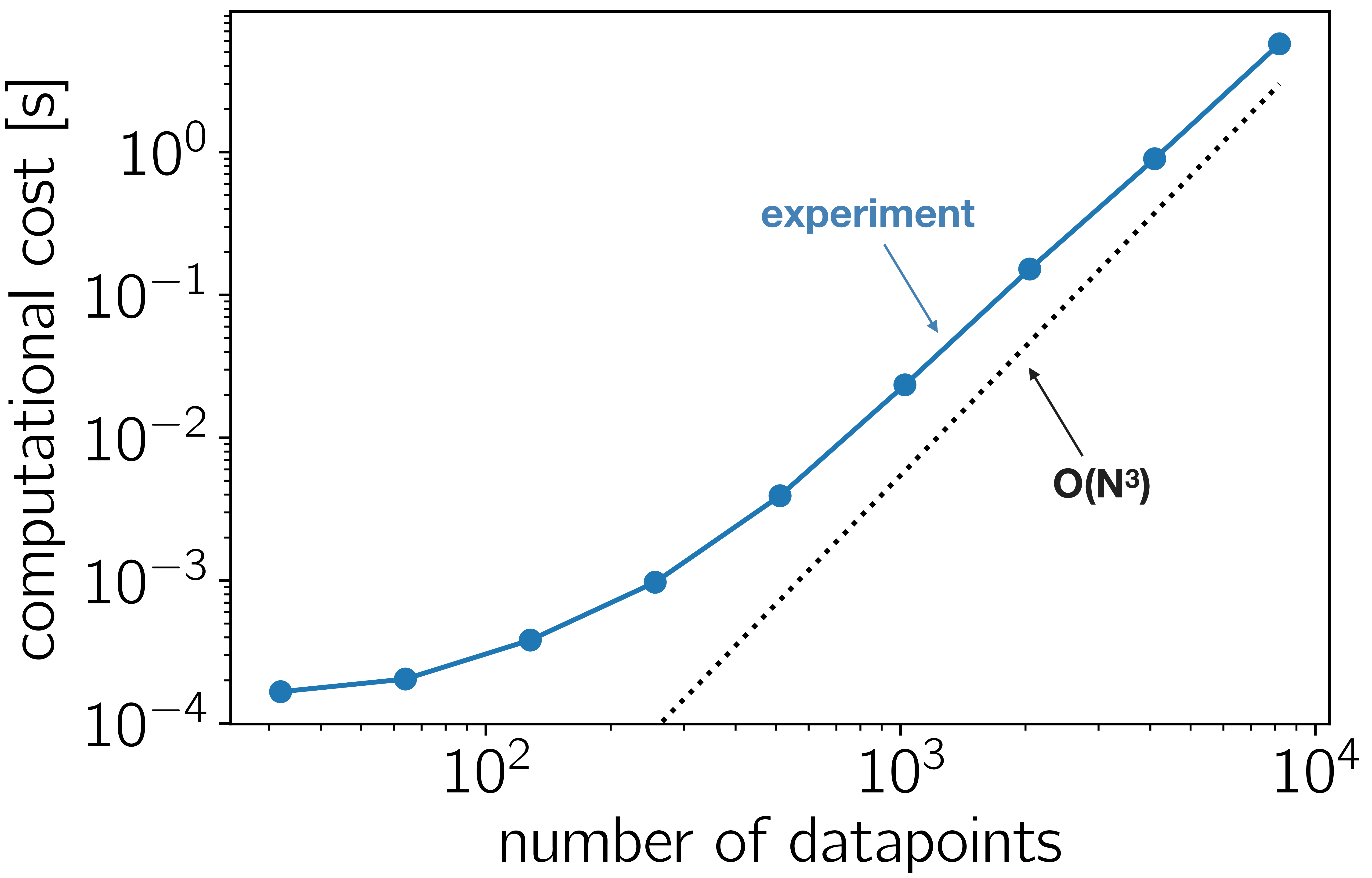
Choosing the kernel

$$\log p(\{y_n\} \,|\, \theta,\, \alpha) = -\frac{1}{2}\, \boldsymbol{r}_\theta{}^{\mathrm{T}}\, {K_\alpha}^{-1}\, \boldsymbol{r}_\theta - \frac{1}{2}\log\det K_\alpha - \frac{N}{2}\log(2\,\pi)$$

b

Scaling to large datasets

b Scaling to large datasets



computational cost [s]
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{2}
10^{3}
10^{4}
number of datapoints
experiment
$O(N^3)$

a Approximation

b Structure

Approximation methods

i **Subsample**

ii **Sparsity**

iii **Low-rank approximations**
(see HODLR solver in george)

iv **etc.**

Structured models

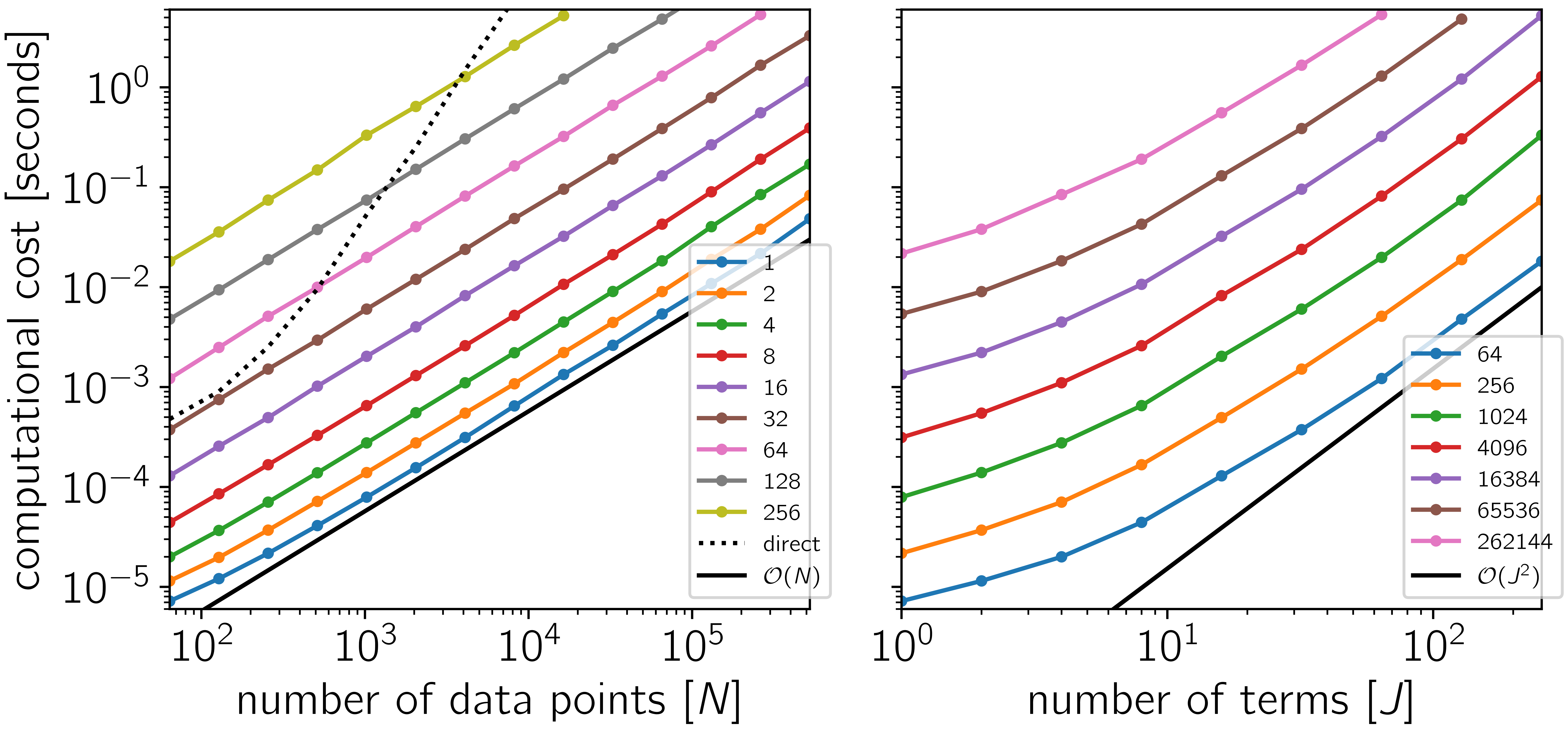
i **Kronecker products**

ii **Evenly sampled data**

iii **Semi-separable kernels**
(see celerite)

iv **etc.**

celerite.readthedocs.io



5

Recap

if there is **correlated noise** (instrumental or astrophysical)
in your data*, try a **Gaussian process**:

$$\log p(\{y_n\}\,|\,\theta,\,\alpha) = -\frac{1}{2}\,\boldsymbol{r}_\theta{}^{\mathrm{T}}\,K_\alpha{}^{-1}\,\boldsymbol{r}_\theta - \frac{1}{2}\log\det K_\alpha - \frac{N}{2}\log(2\,\pi)$$

to do this in **Python**, try:

```
import george
```

george.readthedocs.io

** Hint: there is!*

Resources

- a gaussianprocess.org/gpml
- b george.readthedocs.io
- c dfm.io/gp.js
- d github.com/dfm/gp
- e foreman.mackey@gmail.com

a practical introduction to

Gaussian Processes

for astronomy

Dan Foreman-Mackey

Flatiron Institute // dfm.io // github.com/dfm // @exoplaneteer