UNIVERSITY OF BIRMINGHAM | SCHOOL OF PHYSICS AND ASTRONOMY

Time series analysis

Dealing with correlated signal or noise

GUY R. DAVIES

@CitationWarrior



Session 3 - Correlation in the data	
Section 1 ~30 Mins	Motivation - Correlation in the data
Section 2 ~40 Mins	An introduction to a Gaussian Process
Section 2.1 ~40 Mins	Applying Gaussian Processes
Section 3 ~30 Mins	An introduction to Celerite

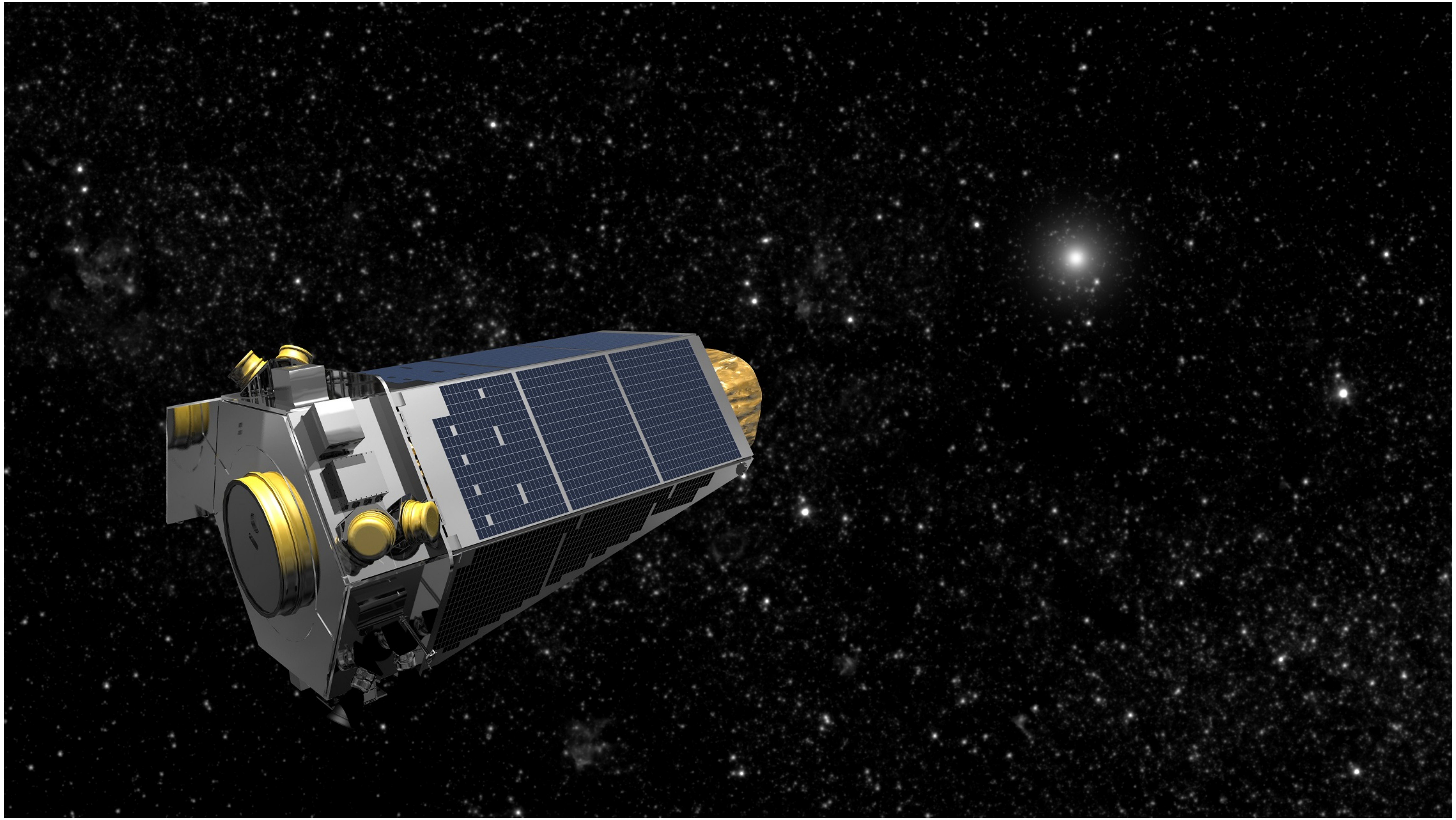
Key Take Aways From This Week/Session:

- Noise can be correlated as a result of physical processes
- Don't neglect correlation
- A GP is an extension of a multivariate normal distribution
- A GP has very interesting properties that we can use to model correlated noise
- There are a bunch of GP software implementations but one that ties in nicely with a physical interpretation is Celerite

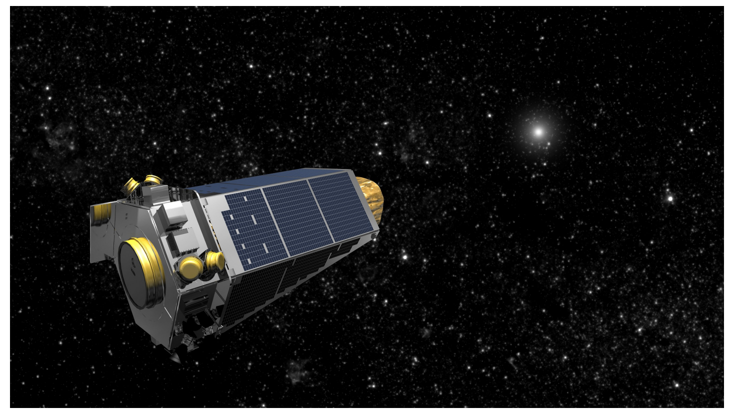
Session 3 - Correlation in the data

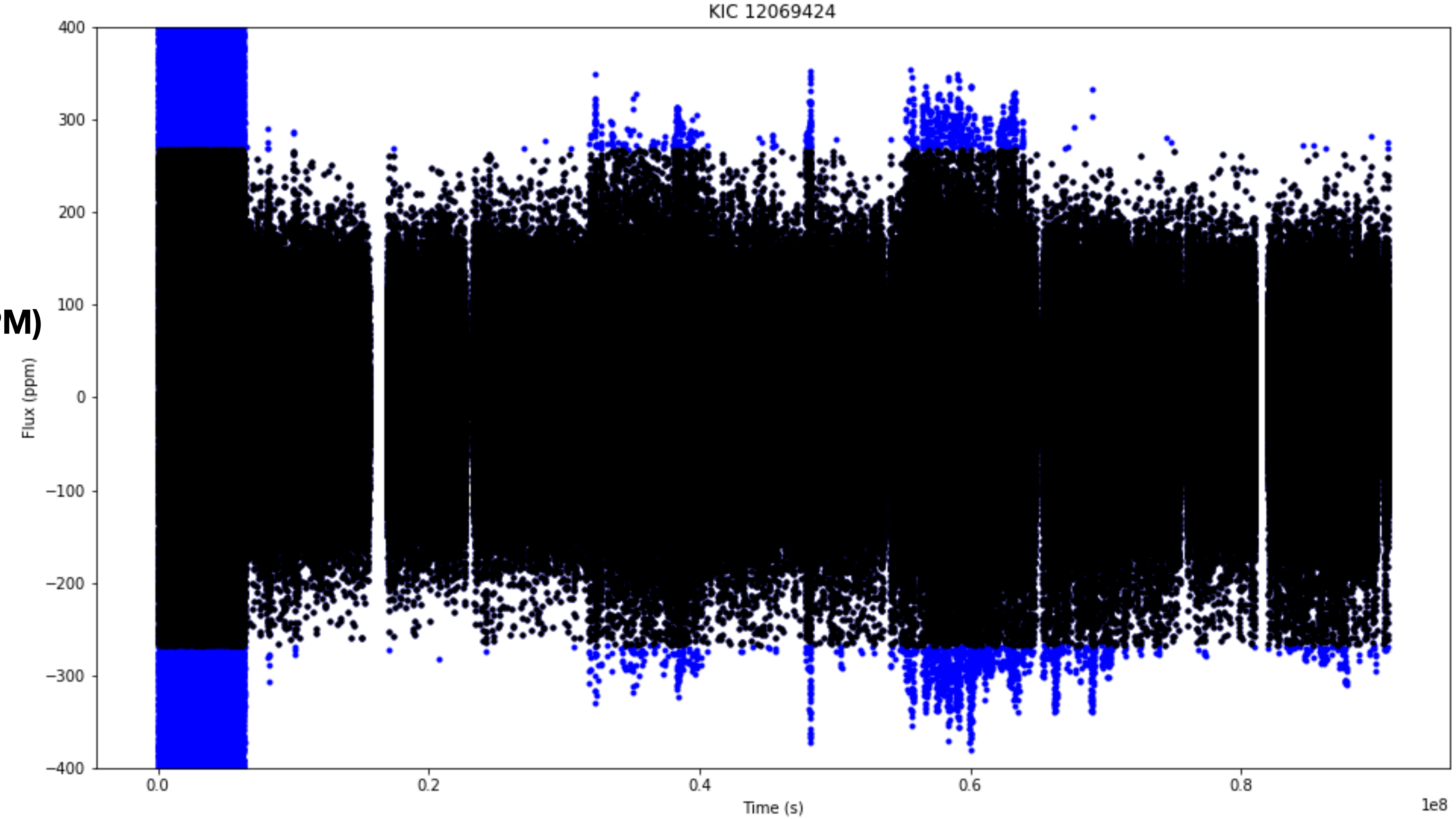
Section 1 ~30 Mins	Motivation - Correlation in the data
Section 2 ~40 Mins	An introduction to a Gaussian Process
Section 2.1 ~40 Mins	Applying Gaussian Processes
Section 3 ~30 Mins	An introduction to Celerite

Motivation - Time series analysis



Motivation - Time series analysis

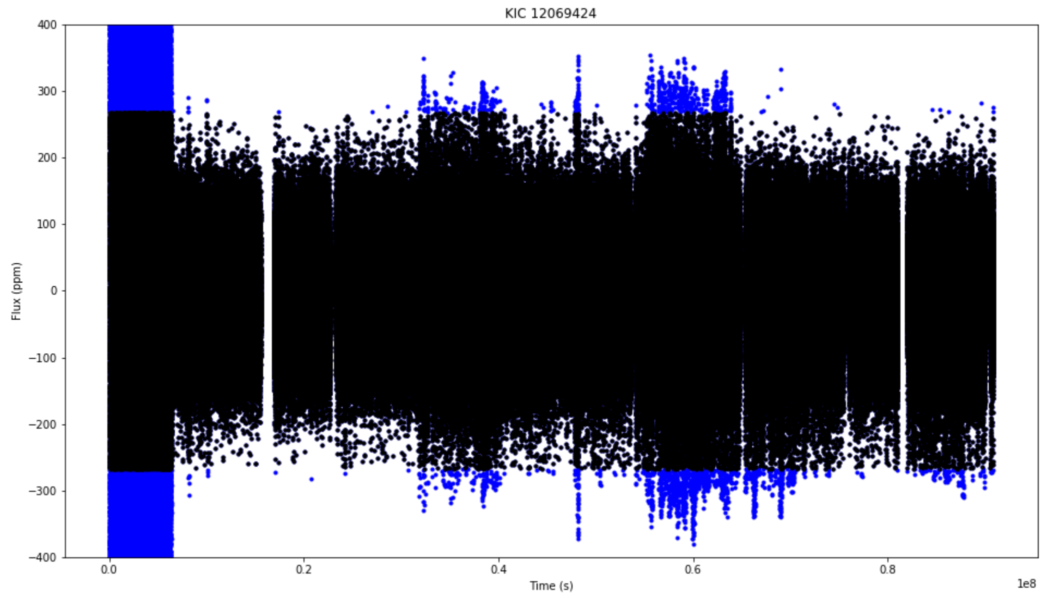




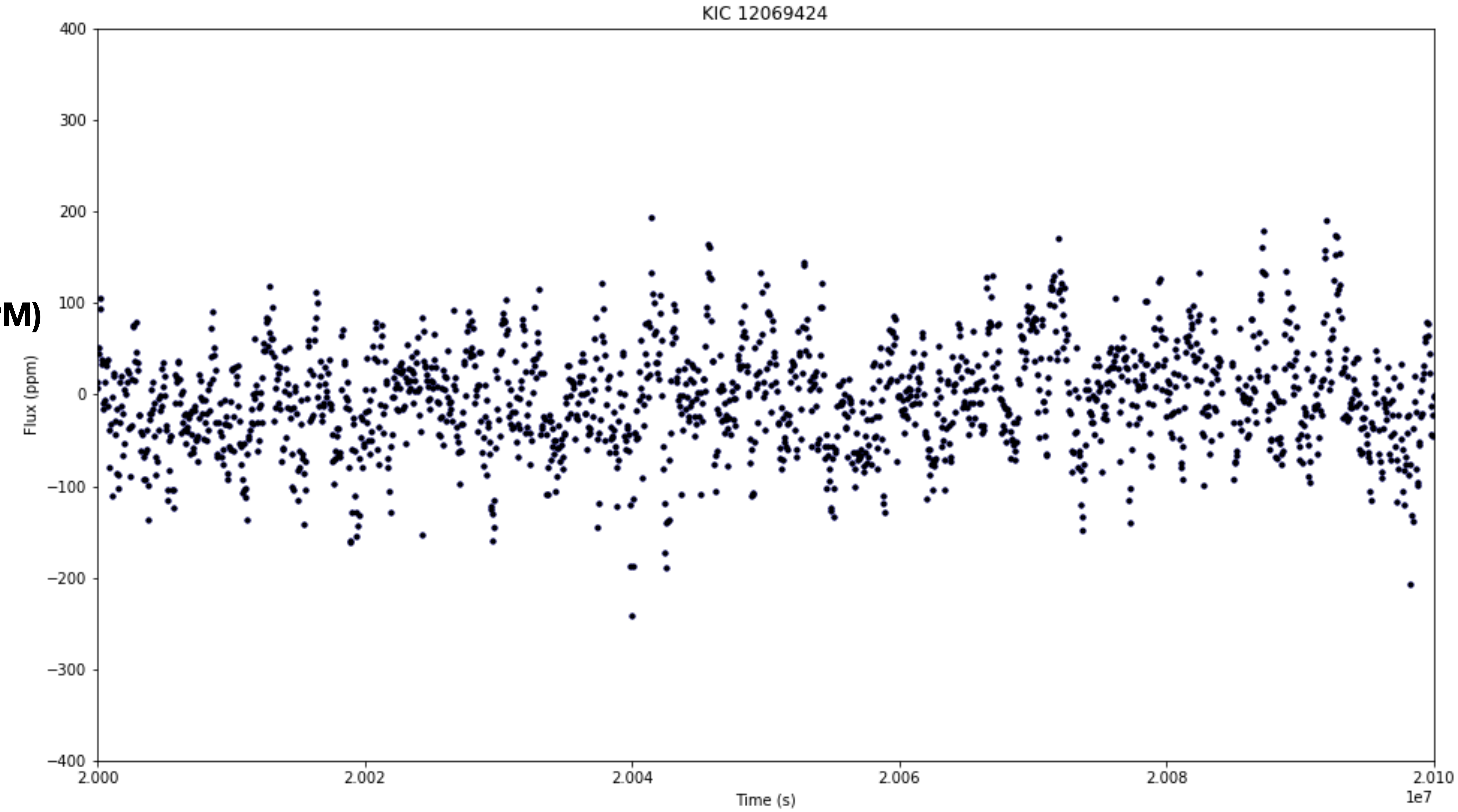
FLUX (PPM)

TIME (s)

MOTIVATION - TIME SERIES ANALYSIS

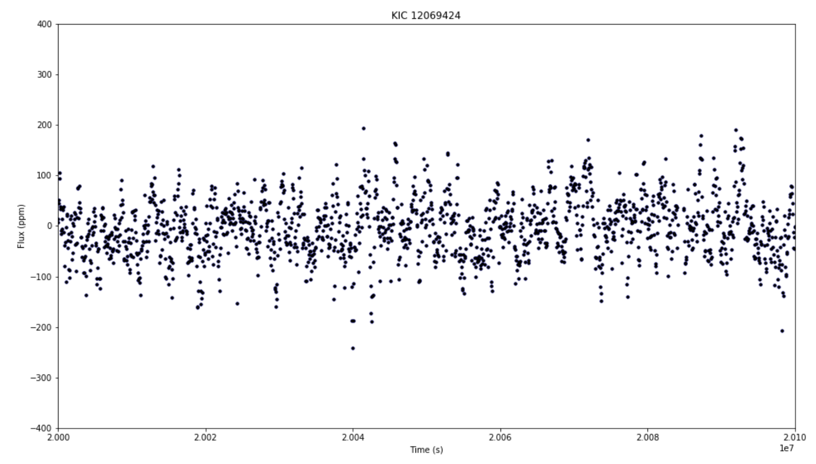


FLUX (PPM)

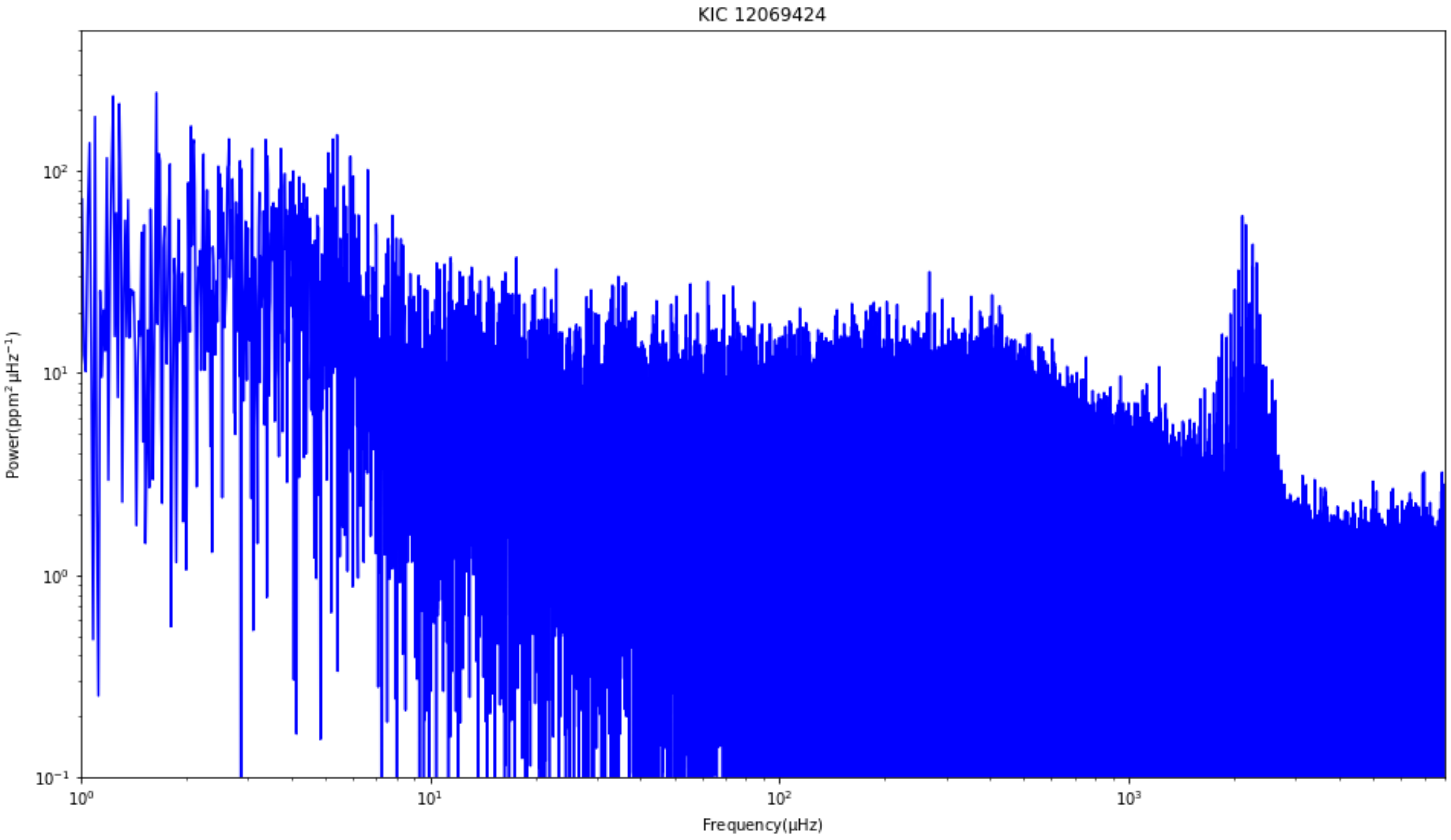


TIME (s)

Motivation - Time series analysis

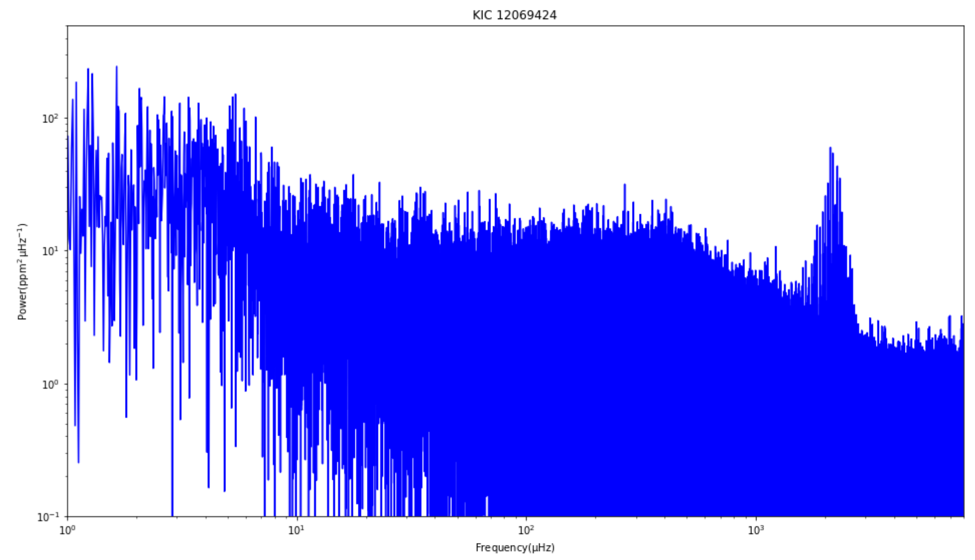


Power

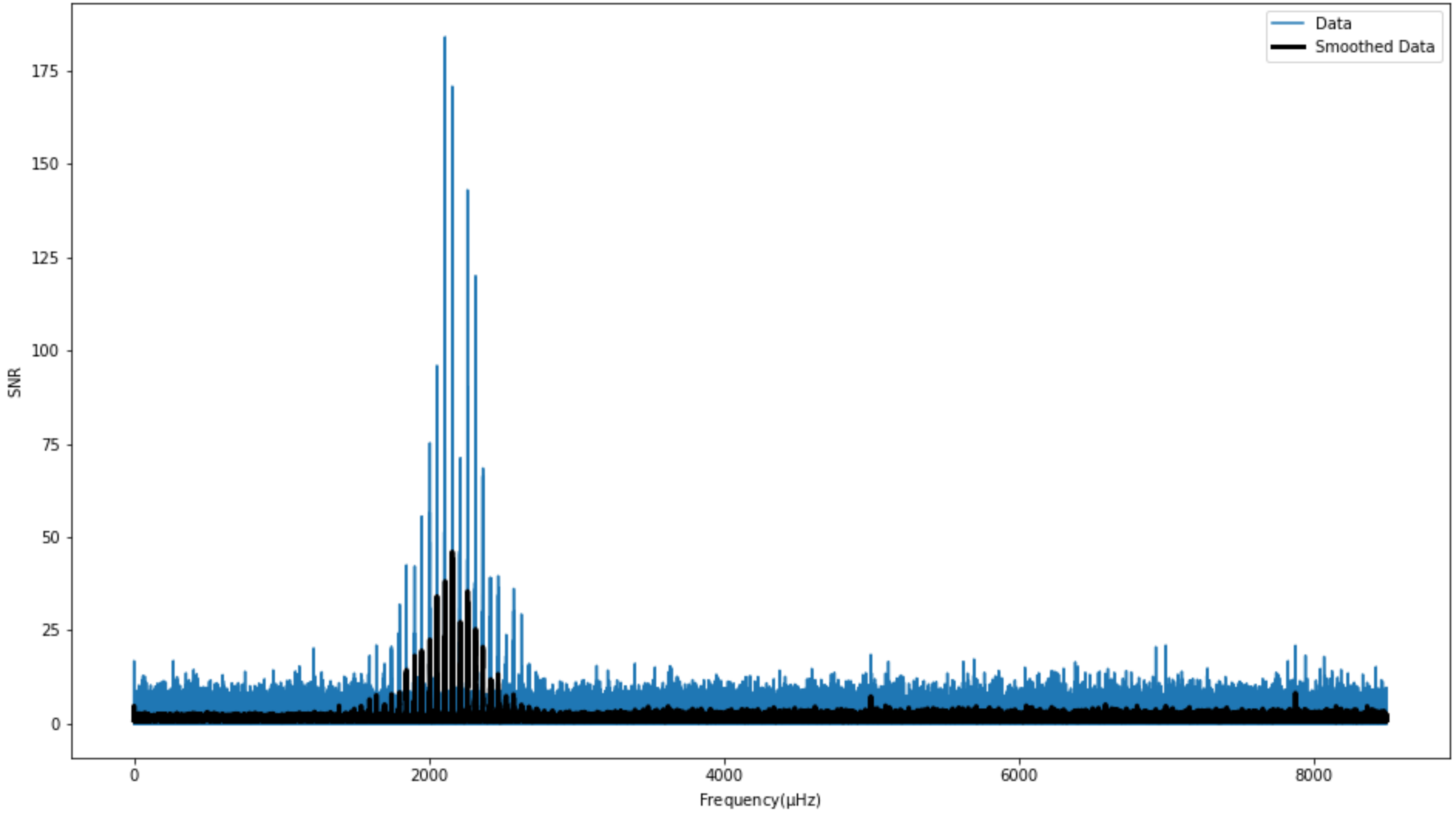


Frequency

Motivation - Time series analysis

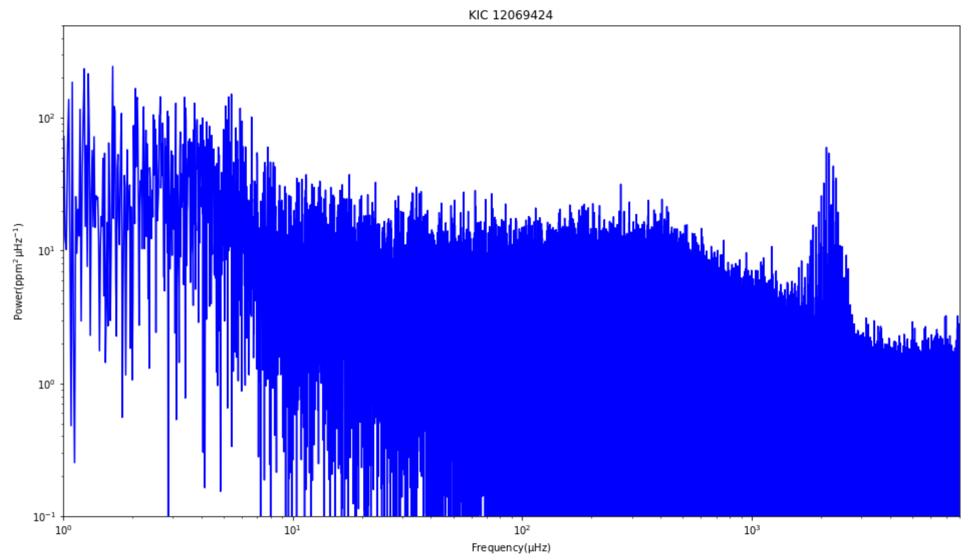


SNR

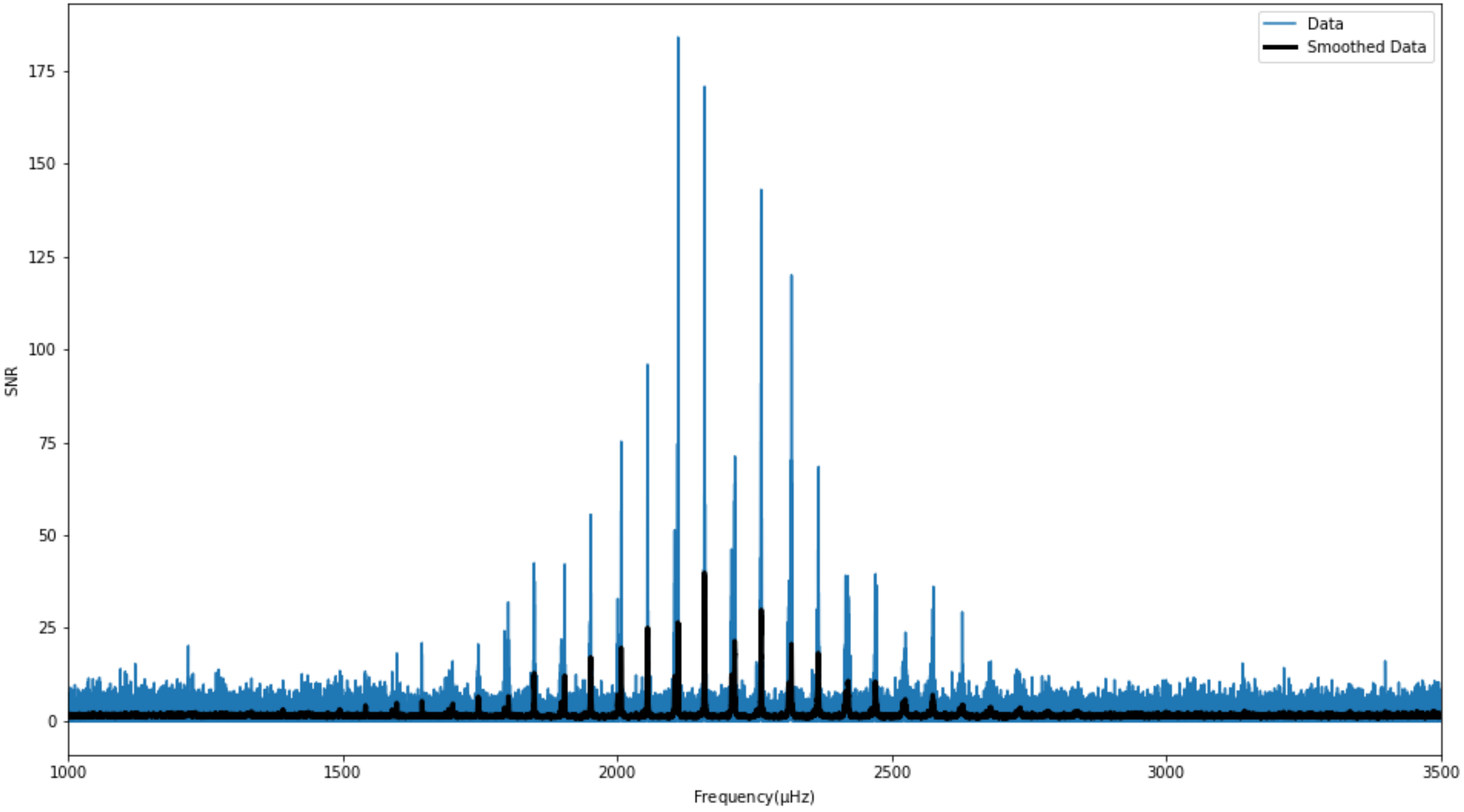


Frequency

MOTIVATION - TIME SERIES ANALYSIS

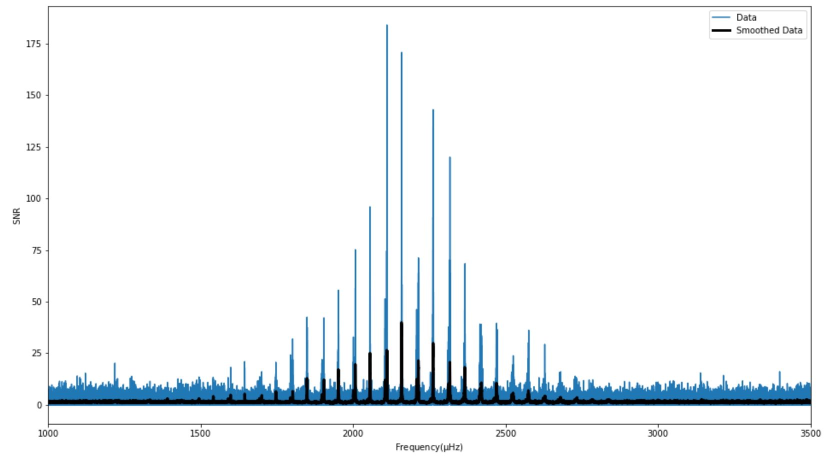


SNR

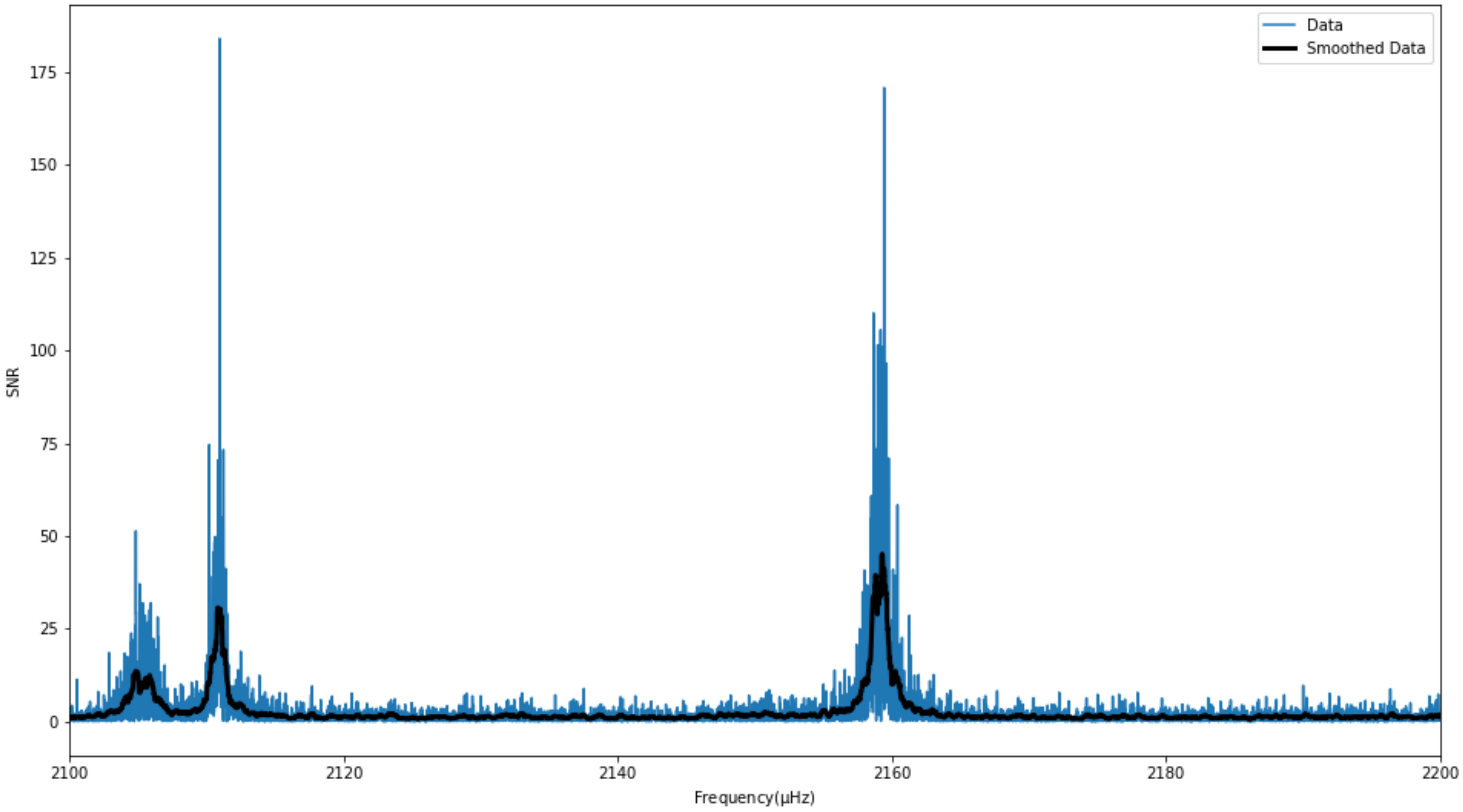


Frequency

MOTIVATION - TIME SERIES ANALYSIS

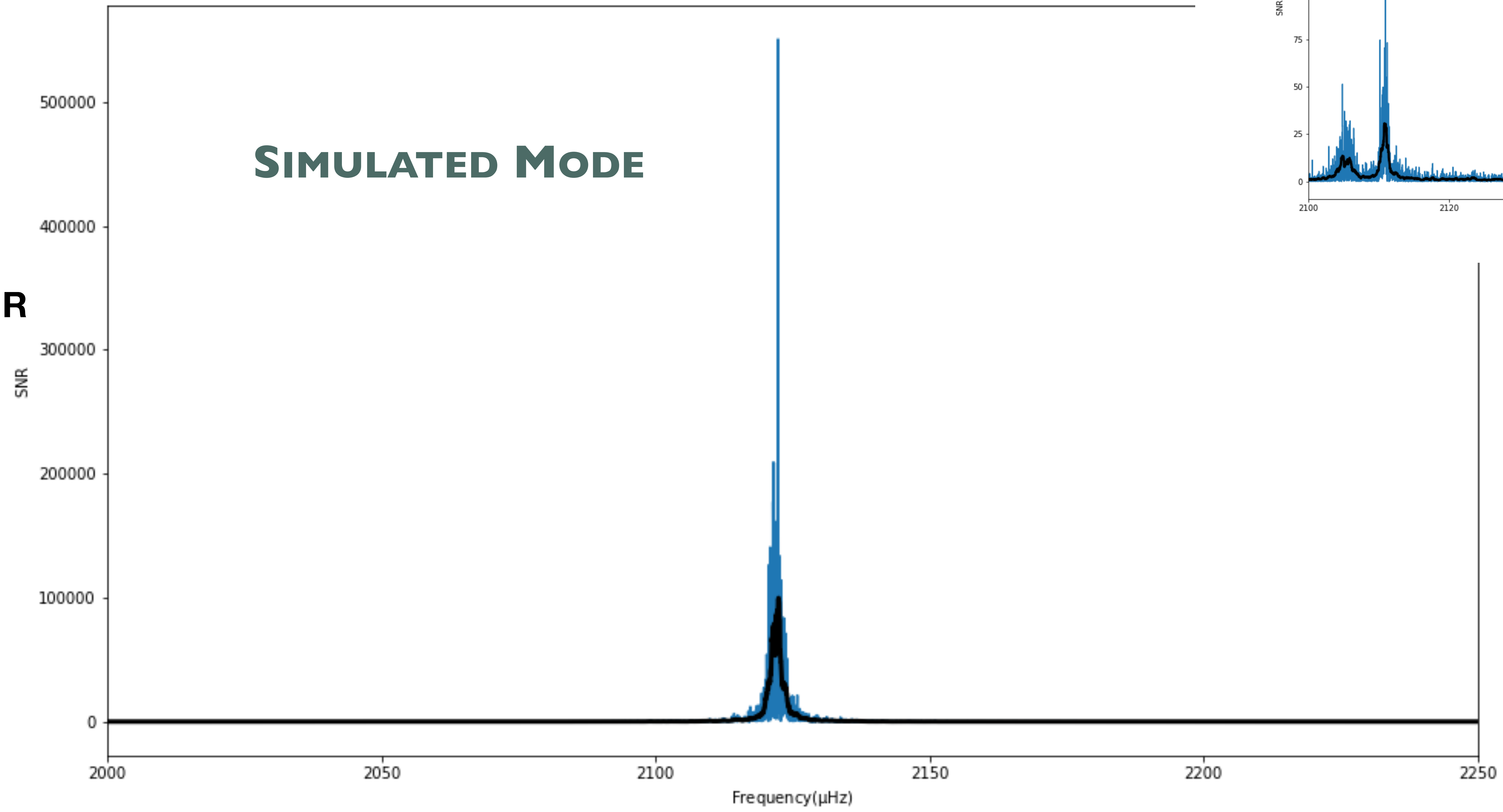


SNR



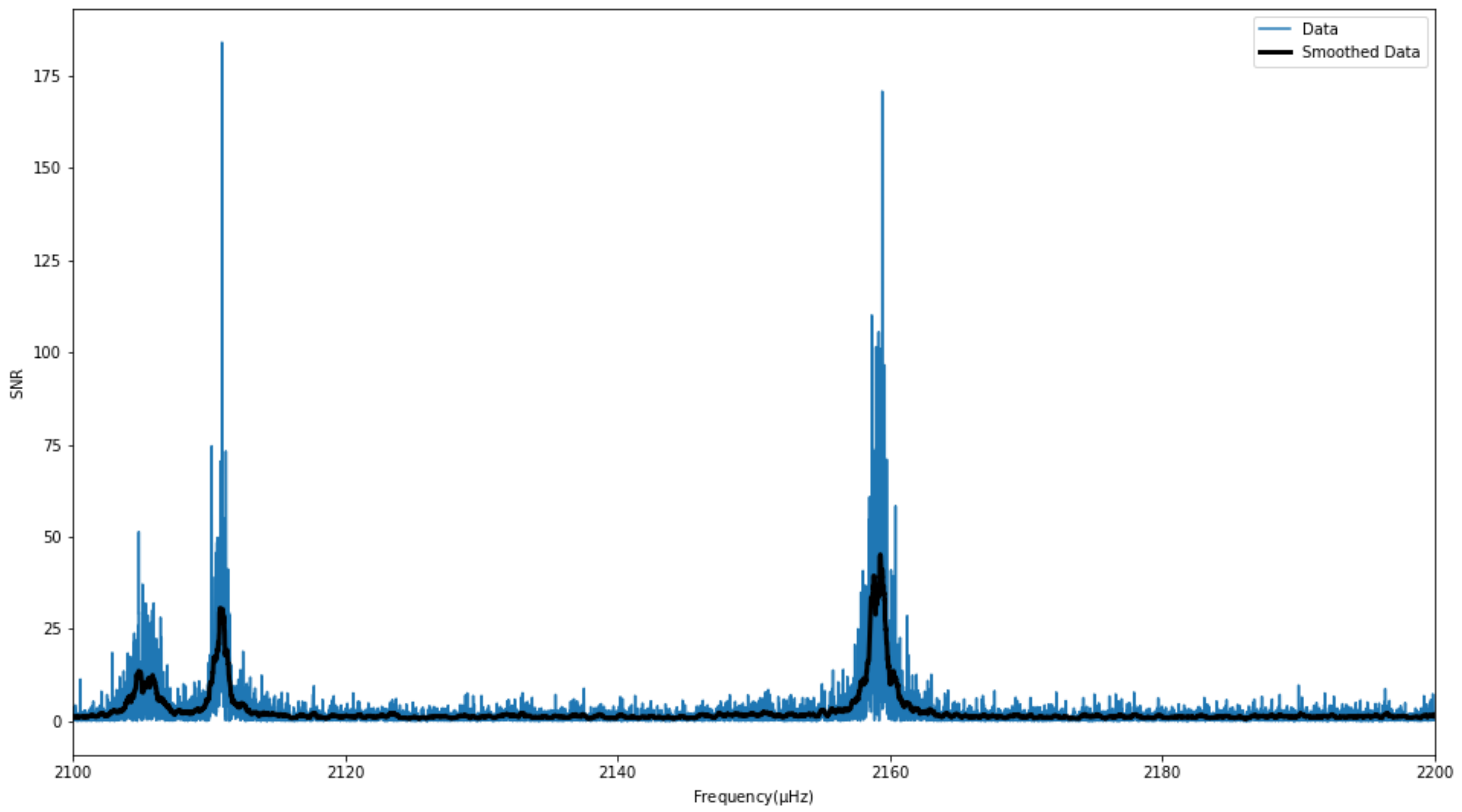
Frequency

Motivation - Time series analysis

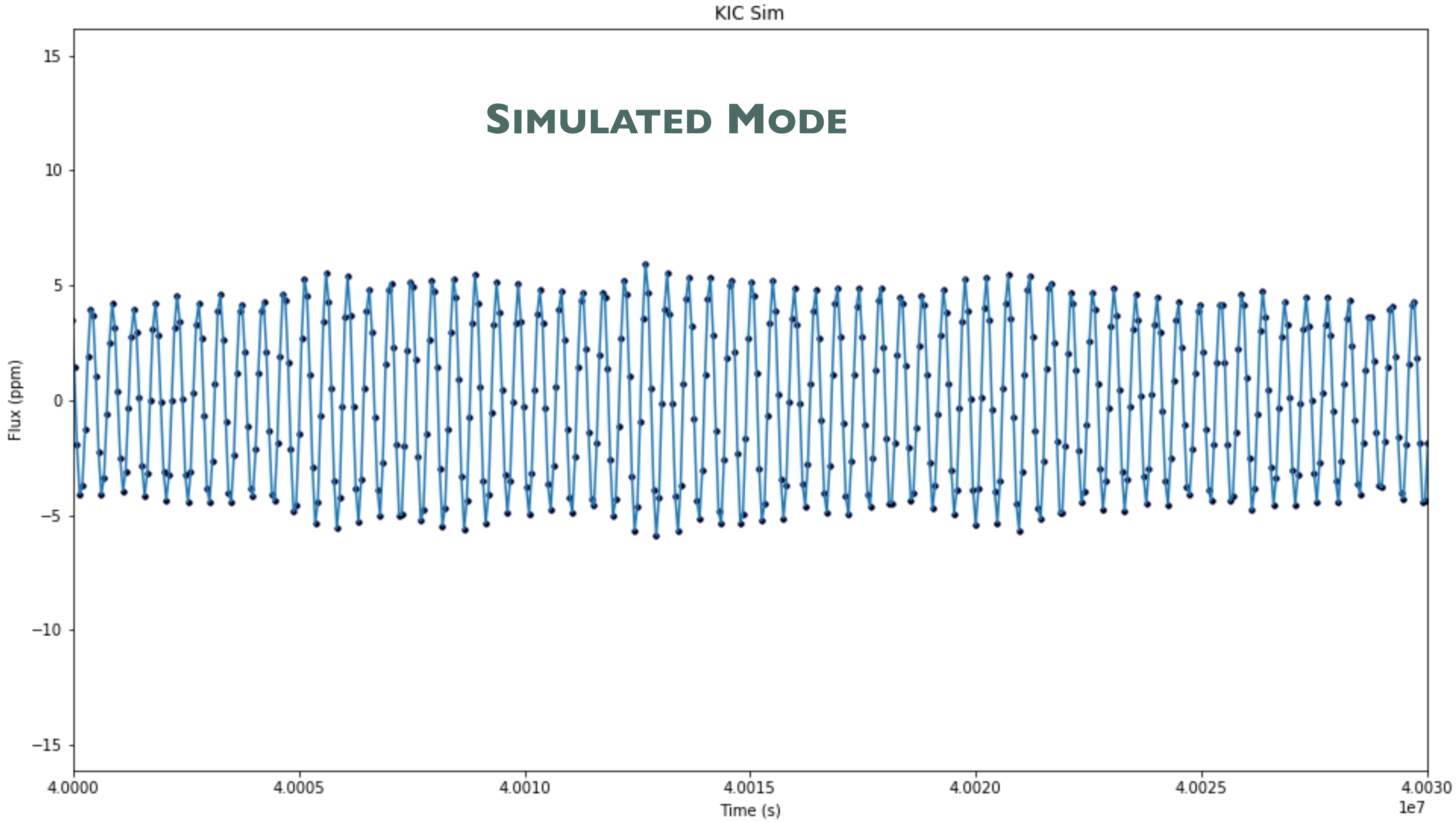


SNR

Frequency

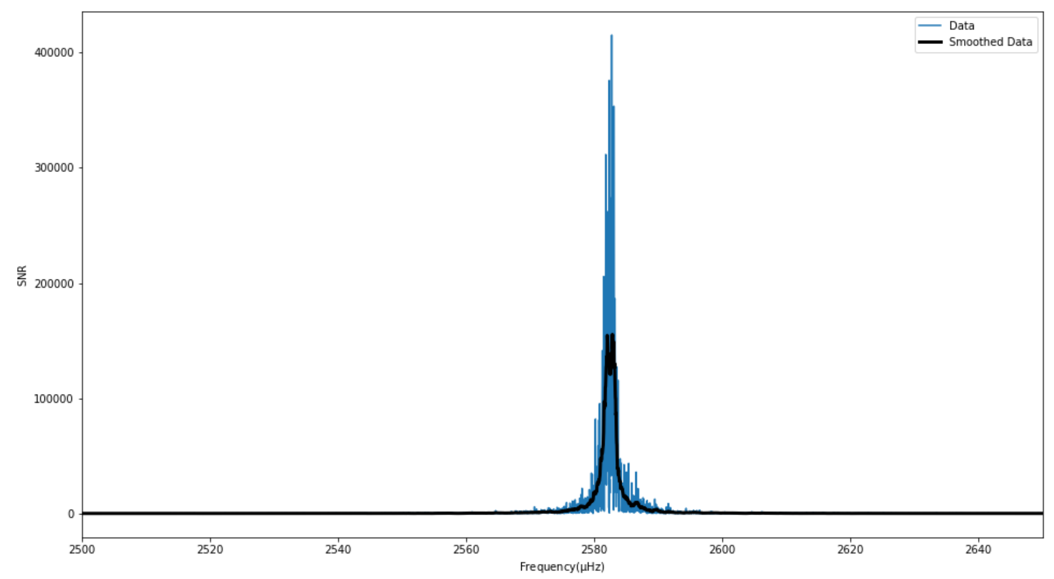


Motivation - Time series analysis

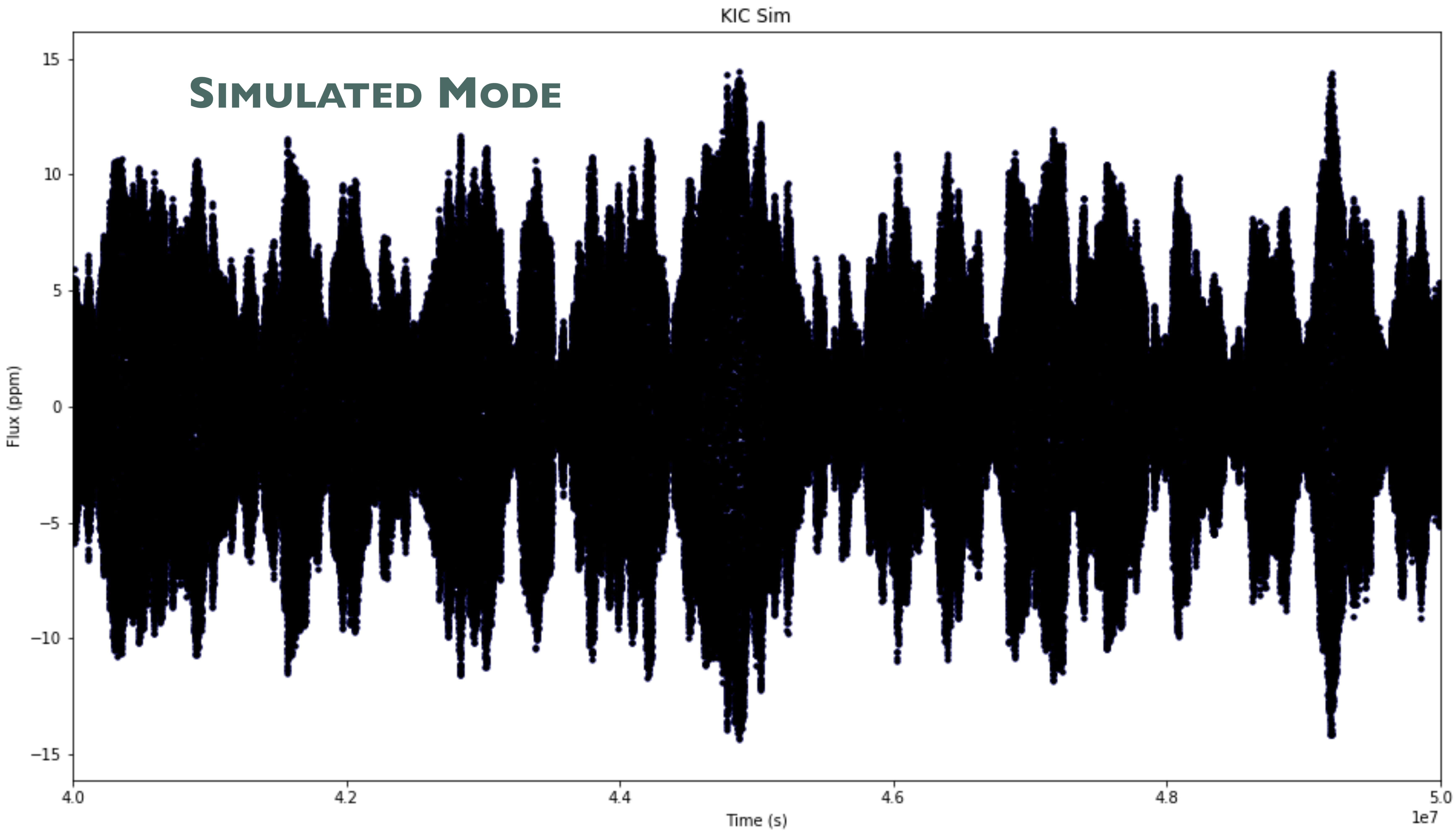


Flux

Time

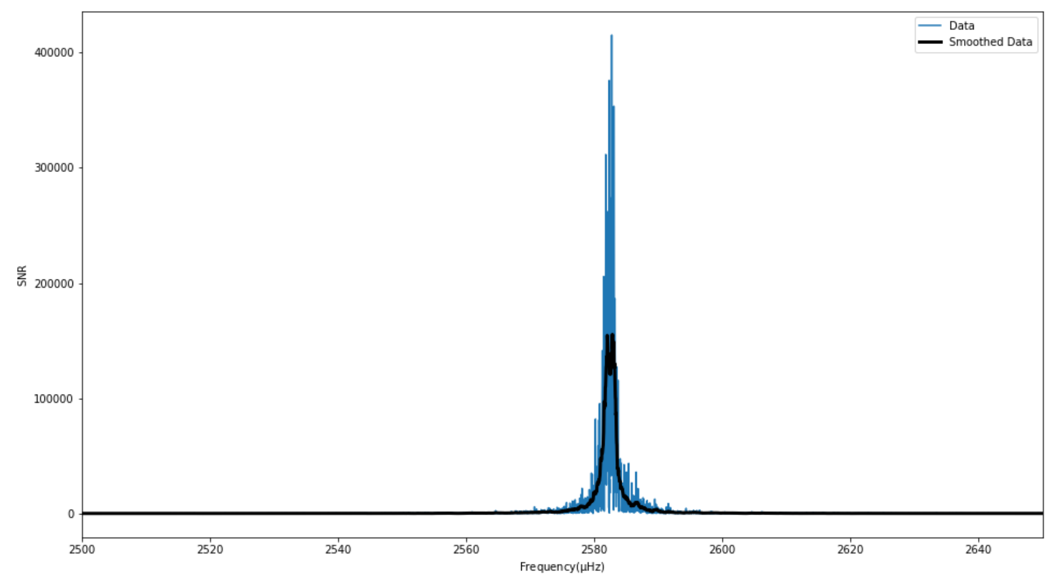


Motivation - Time series analysis

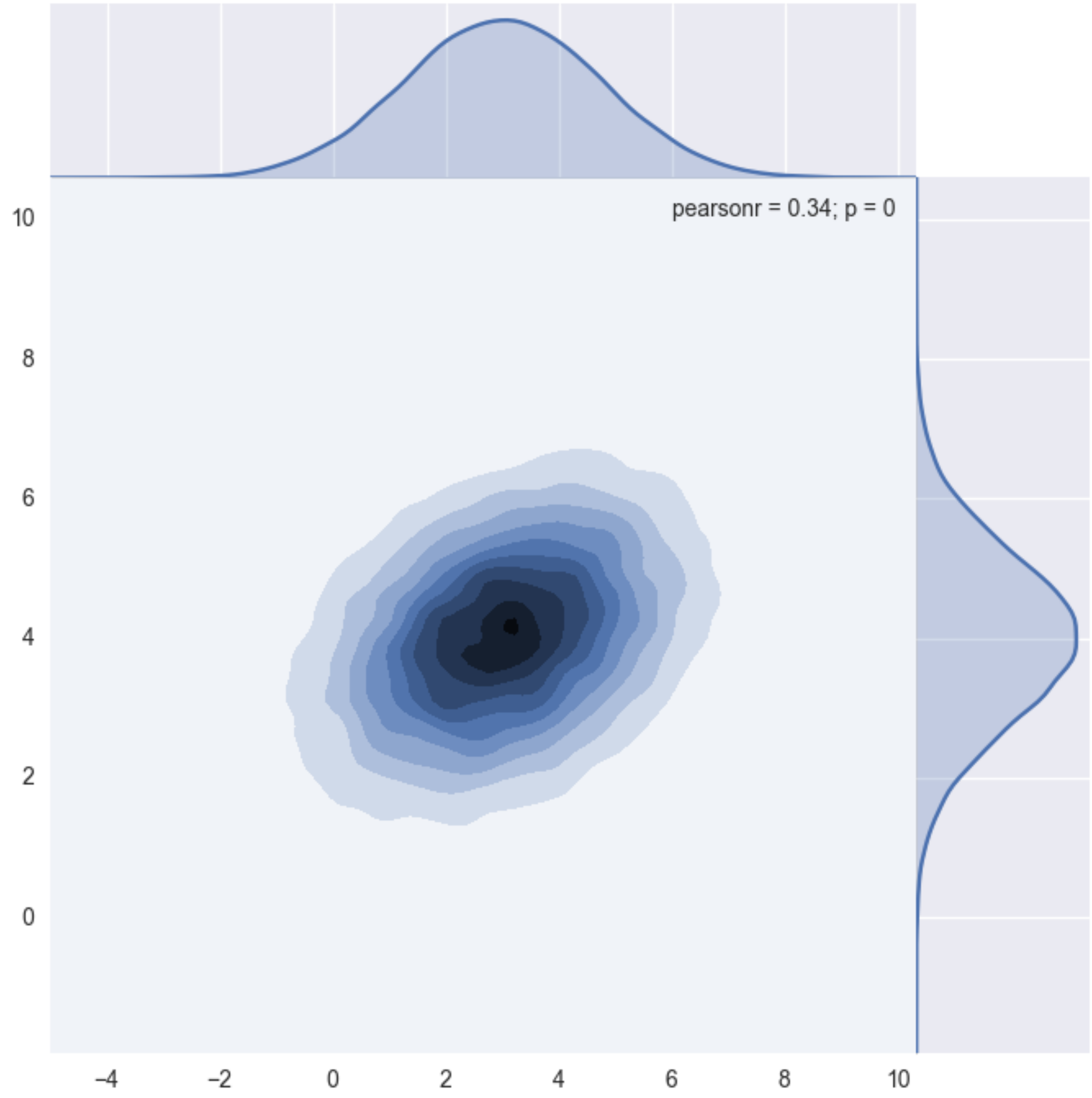


Flux

Time



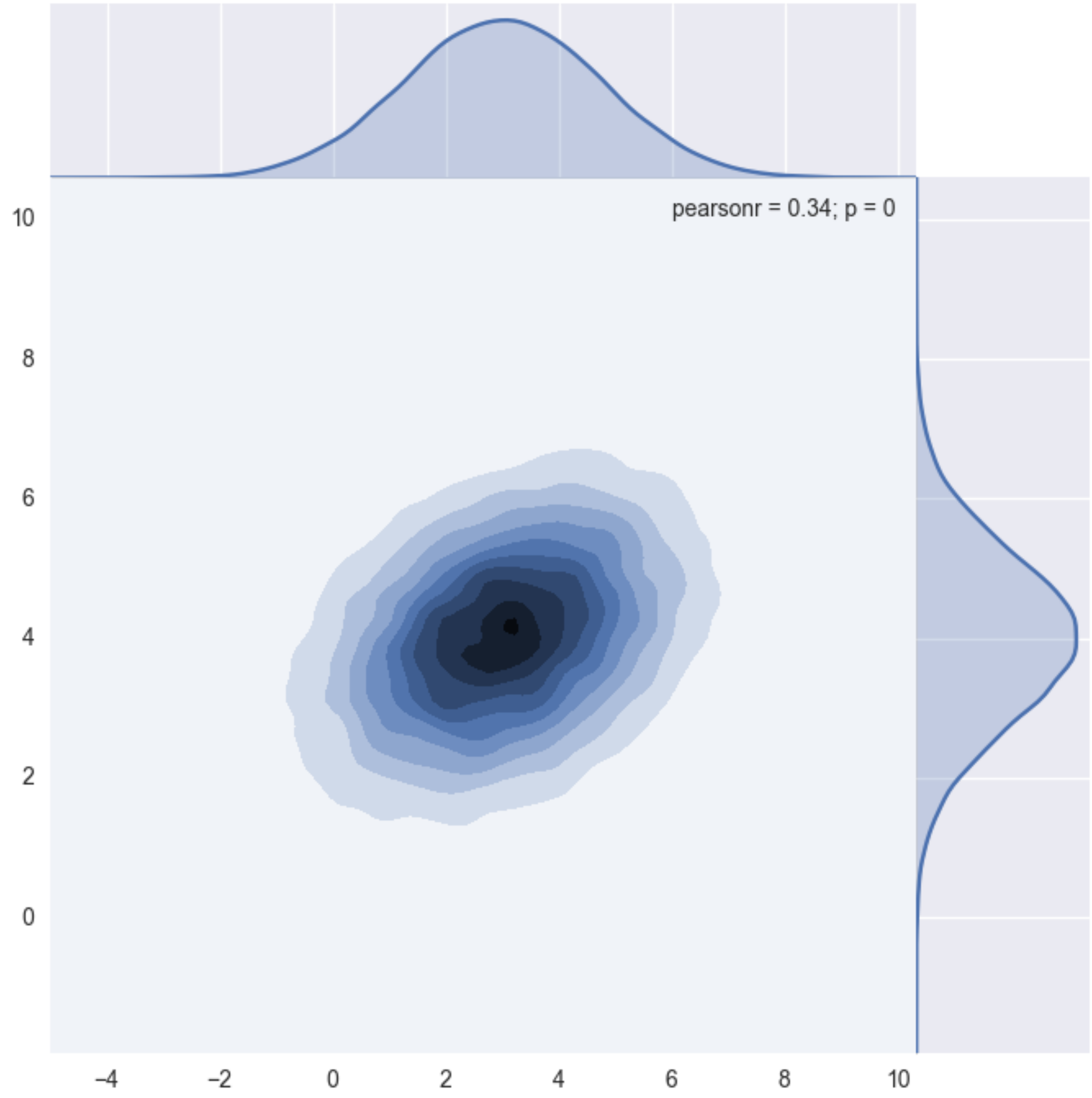
MOTIVATION - CORRELATED DATA BASICS



$$\begin{pmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) \end{pmatrix} = \begin{pmatrix} 3.0 & 0.7 \\ 0.7 & 1.5 \end{pmatrix}$$

$$\mathcal{N}\left[\begin{pmatrix} 3 \\ 4 \end{pmatrix}, \begin{pmatrix} 3.0 & 0.7 \\ 0.7 & 1.5 \end{pmatrix}\right]$$

Motivation - Correlated Data Basics

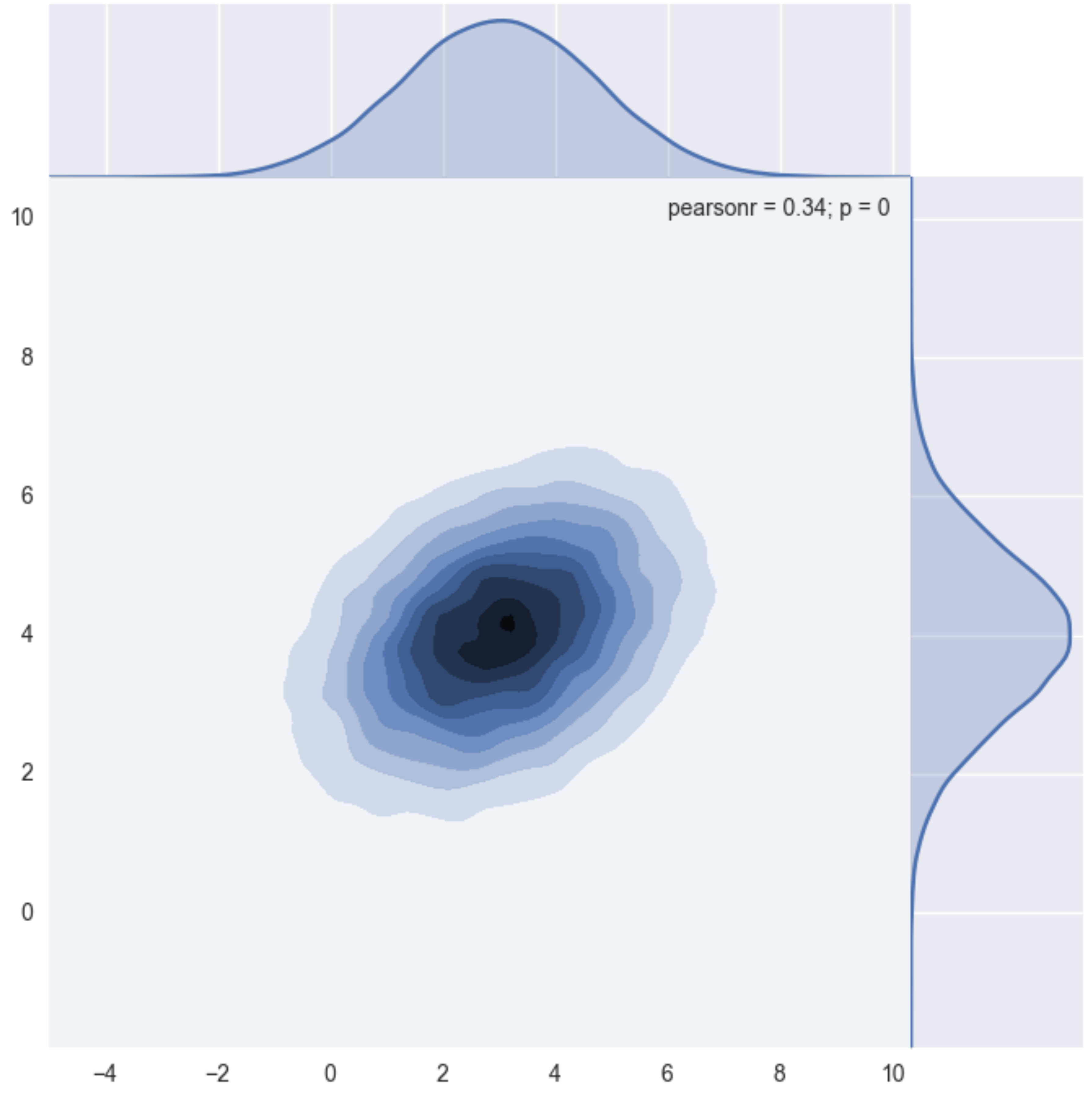


$$\mathrm{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - E(X))(y_i - E(Y)).$$

$$\begin{pmatrix} \mathrm{cov}(x,x) & \mathrm{cov}(x,y) \\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) \end{pmatrix} = \begin{pmatrix} 3.0 & 0.7 \\ 0.7 & 1.5 \end{pmatrix}$$

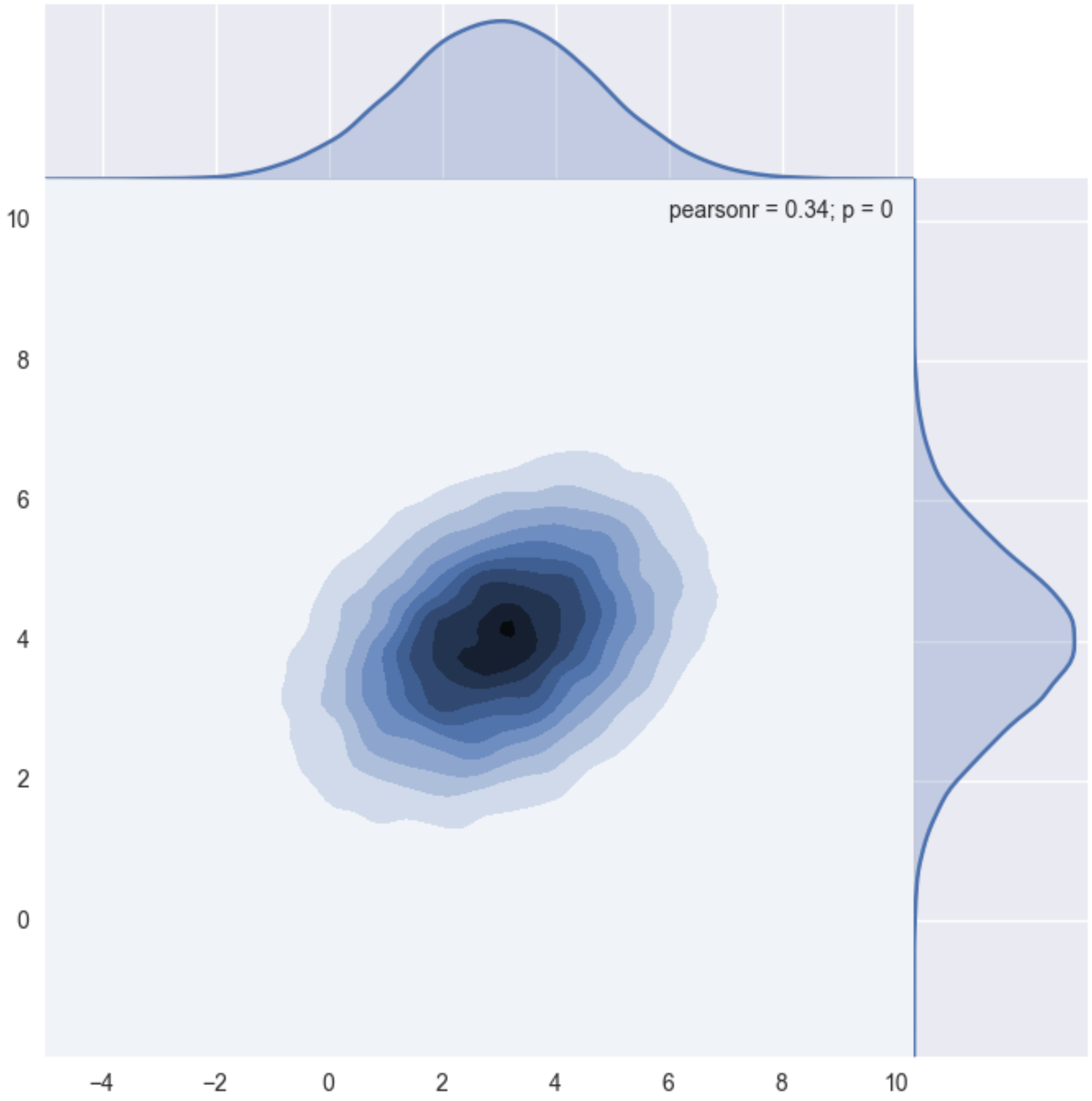
$$\mathcal{N}\left[\begin{pmatrix} 3 \\ 4 \end{pmatrix}, \begin{pmatrix} 3.0 & 0.7 \\ 0.7 & 1.5 \end{pmatrix}\right]$$

Motivation - Correlated Data Basics



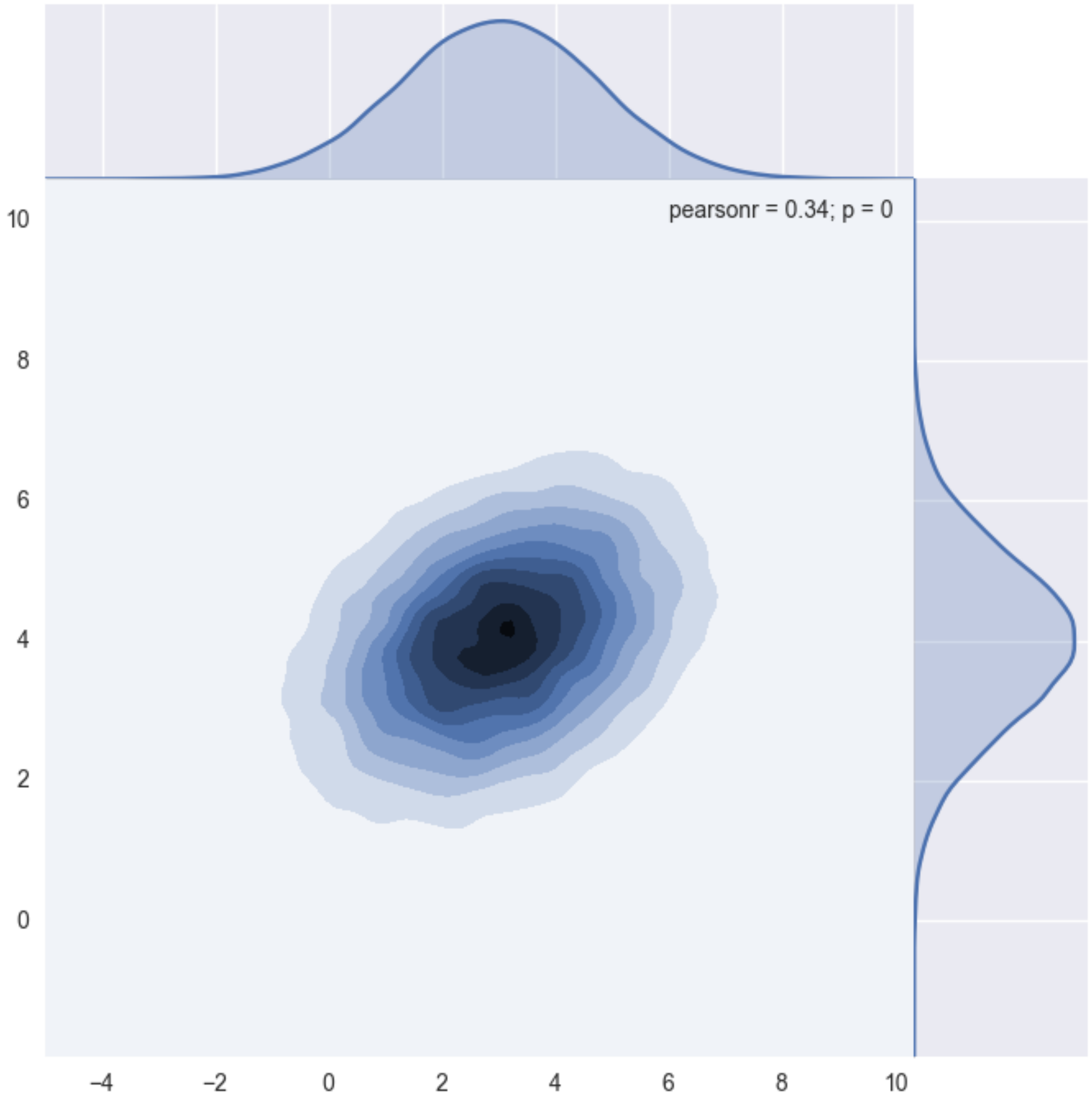
$$\mu(x+y) = \mu(x) + \mu(y)$$

$$\sigma^2(x+y) = \sigma^2(x) + \sigma^2(y) + 2\,\mathrm{cov}(x,y)$$



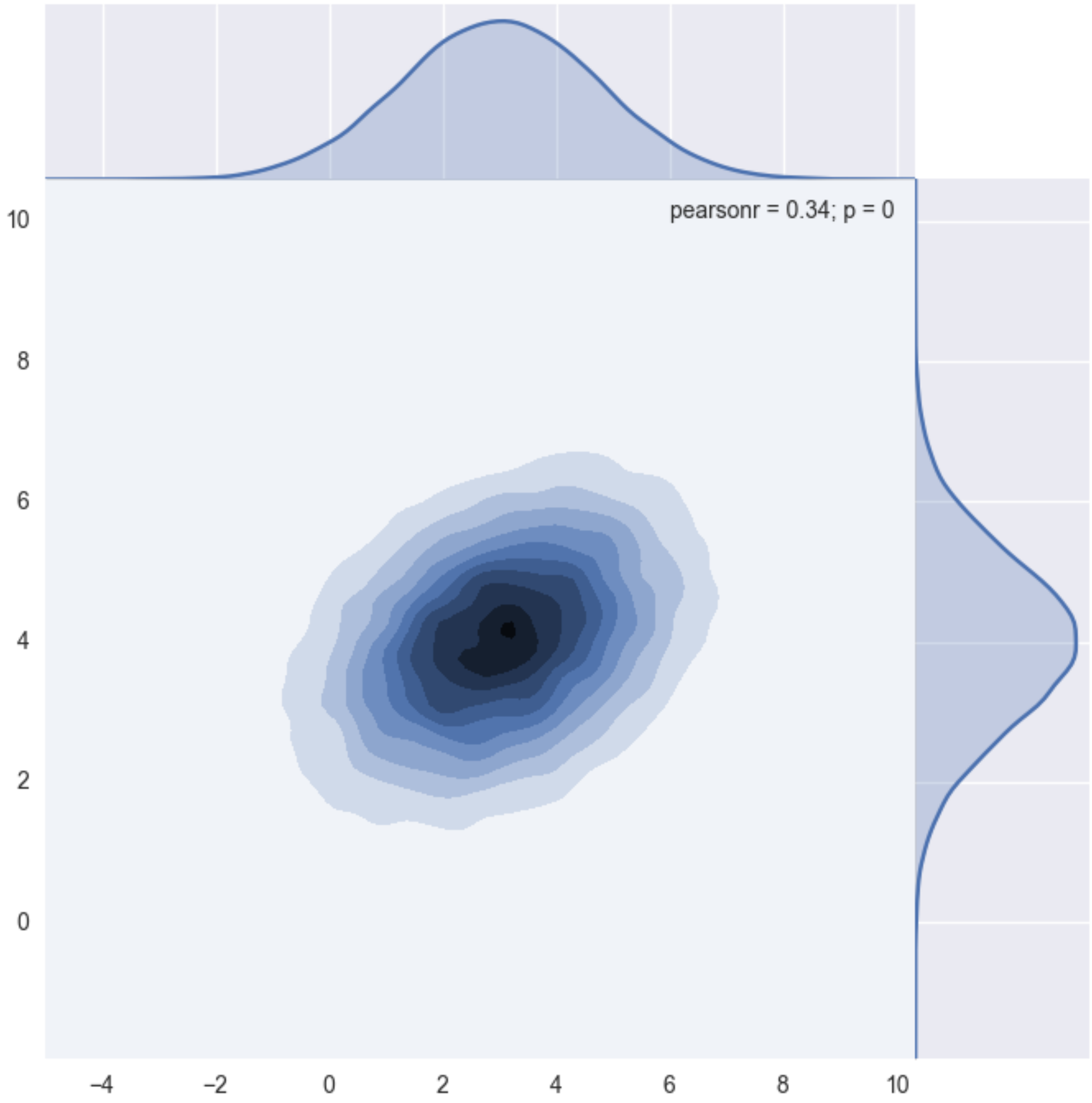
$$p(\mathbf{y}) = \det(2\pi\Sigma)^{-\frac{1}{2}} \ \exp\left(-\tfrac{1}{2}(\mathbf{y}-\mu)\Sigma^{-1}(\mathbf{y}-\mu)^T\right)$$

$$\mathcal{N}\left[\begin{pmatrix}3\\4\end{pmatrix}, \begin{pmatrix}3.0 & 0.7\\0.7 & 1.5\end{pmatrix}\right]$$



$$p(\mathbf{y}) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mu)\Sigma^{-1}(\mathbf{y}-\mu)^T\right)$$

$$\mathcal{N}\left[\begin{pmatrix}3\\4\end{pmatrix}, \begin{pmatrix}3.0 & 0.7\\0.7 & 1.5\end{pmatrix}\right]$$



$$p(\mathbf{y}) = \det(2\pi\Sigma)^{-\frac{1}{2}}\ \exp\left(-\tfrac{1}{2}(\mathbf{y}-\mu)\Sigma^{-1}(\mathbf{y}-\mu)^T\right)$$

$$\rightarrow \quad p(y) = (2\pi\sigma^2)^{-\frac{1}{2}}\ \exp\left(-\frac{1}{2}\frac{(y-\mu)^2}{\sigma^2}\right)$$

Key Take Aways:

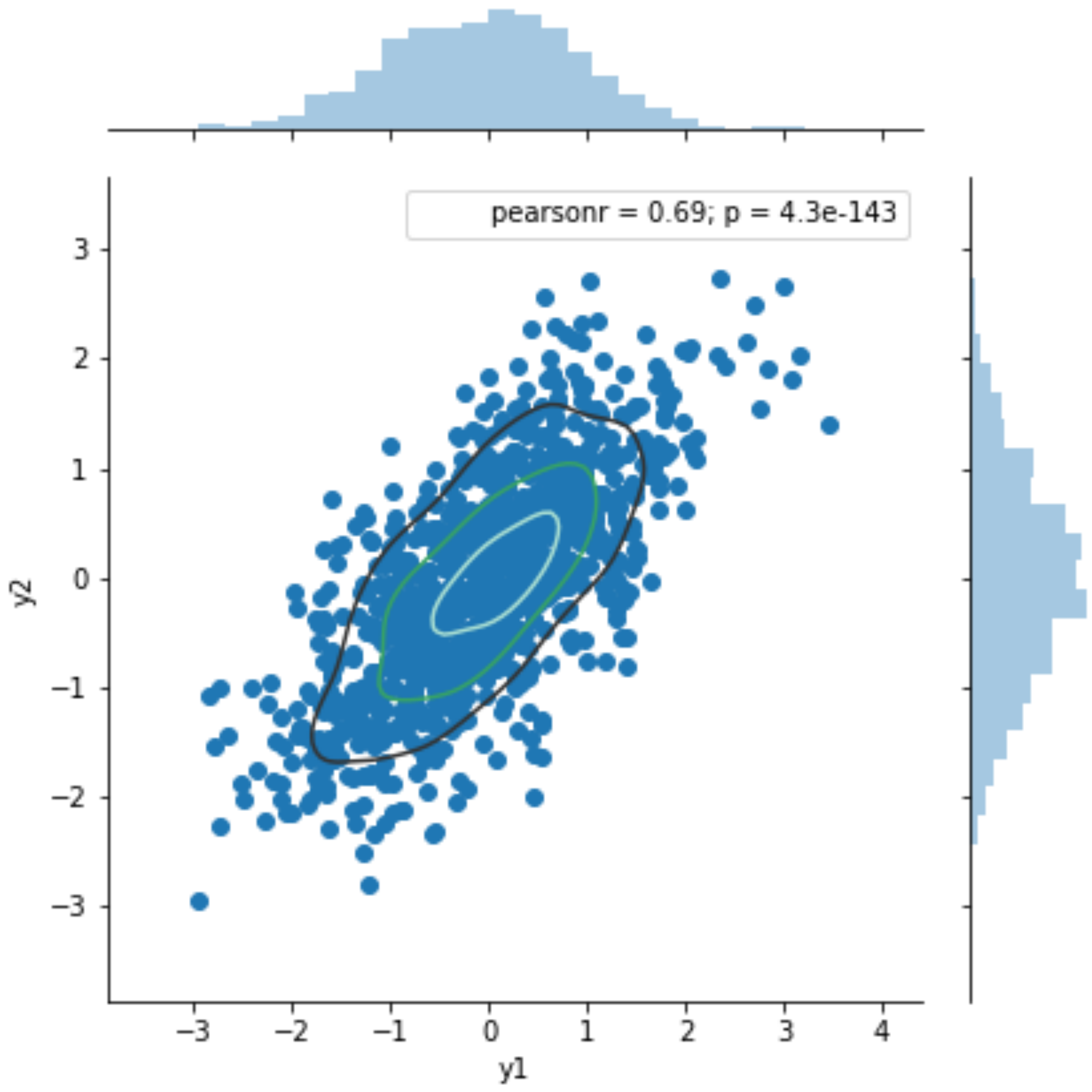
- Noise can be correlated but we can deal with this
- Physically, the current state depends on the previous state
- For some this is noise
- For others this is signal

Session 3 - Correlation in the data

Section 1 ~30 Mins	Motivation - Correlation in the data
Section 2 ~40 Mins	An introduction to a Gaussian Process
Section 2.1 ~40 Mins	Applying Gaussian Processes
Section 3 ~30 Mins	An introduction to Celerite

GP's - Sources of Information

- http://www.gaussianprocess.org/gpml/ Rasmussen & Williams 2006
- You Tube - J Cunningham : MLSS 2012
- You Tube - Richard Turner : ML Tutorial Gaussian Processes
- http://katbailey.github.io/post/gaussian-processes-for-dummies/

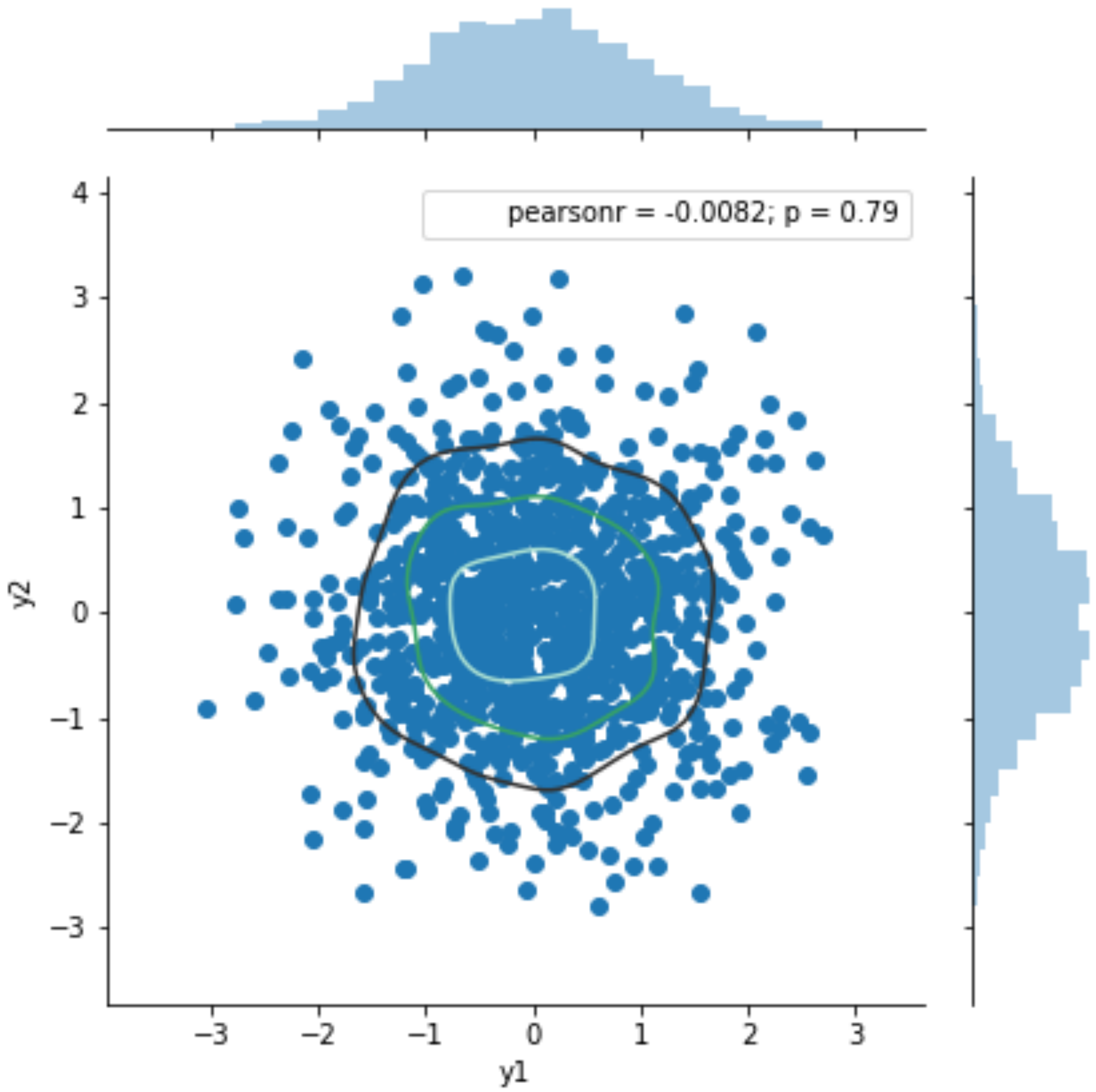


Multinomial

$$p(\mathbf{y}|\boldsymbol{\Sigma}) \propto \exp\left(-\frac{1}{2}\mathbf{y}^T\boldsymbol{\Sigma}^{-1}\mathbf{y}\right)$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1.0 & 0.7 \\ 0.7 & 1.0 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

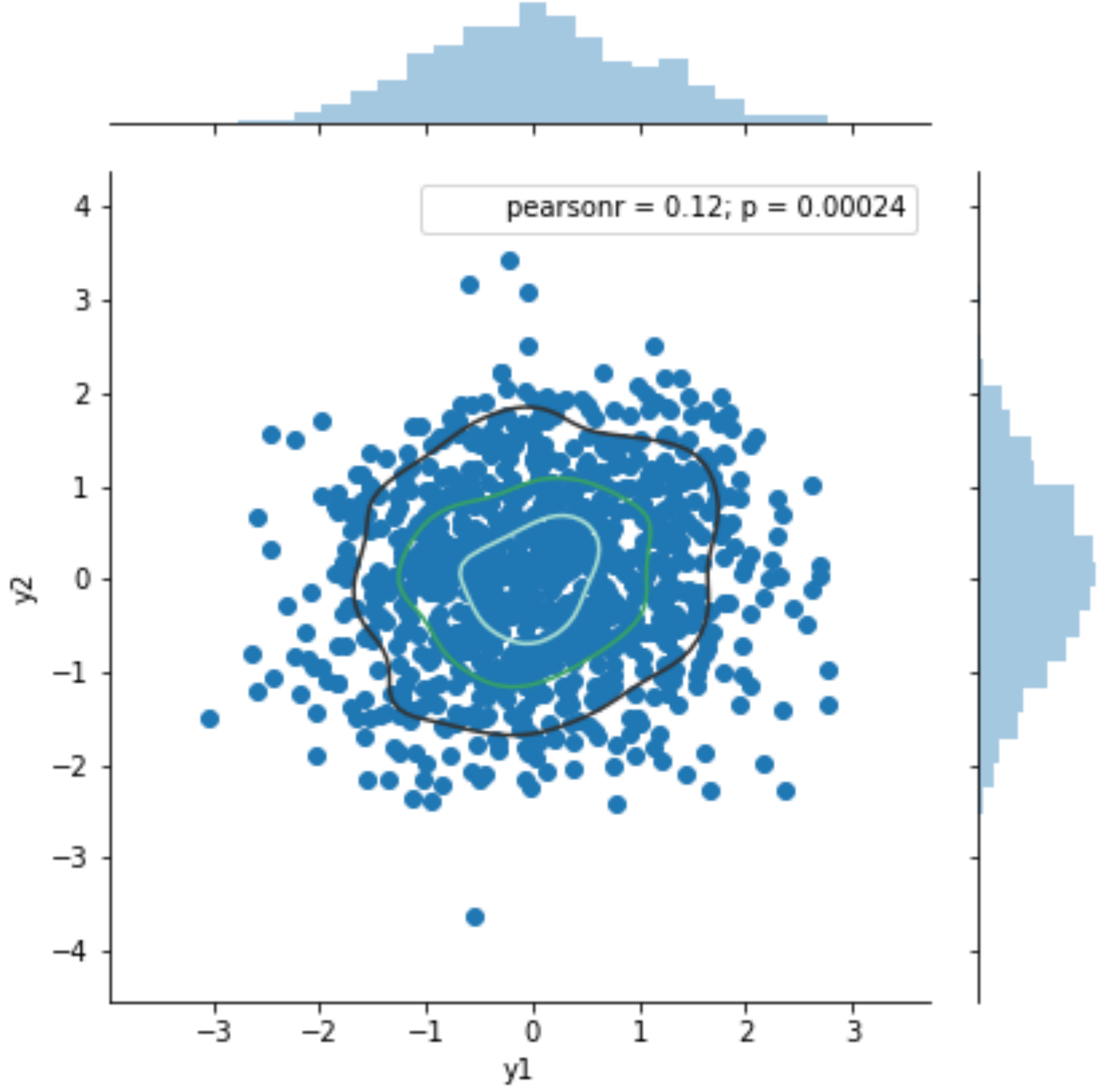


Multinomial

$$p(\mathbf{y}|\boldsymbol{\Sigma}) \propto \exp\left(-\frac{1}{2}\mathbf{y}^T\boldsymbol{\Sigma}^{-1}\mathbf{y}\right)$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1.0 & \mathbf{0.0} \\ \mathbf{0.0} & 1.0 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

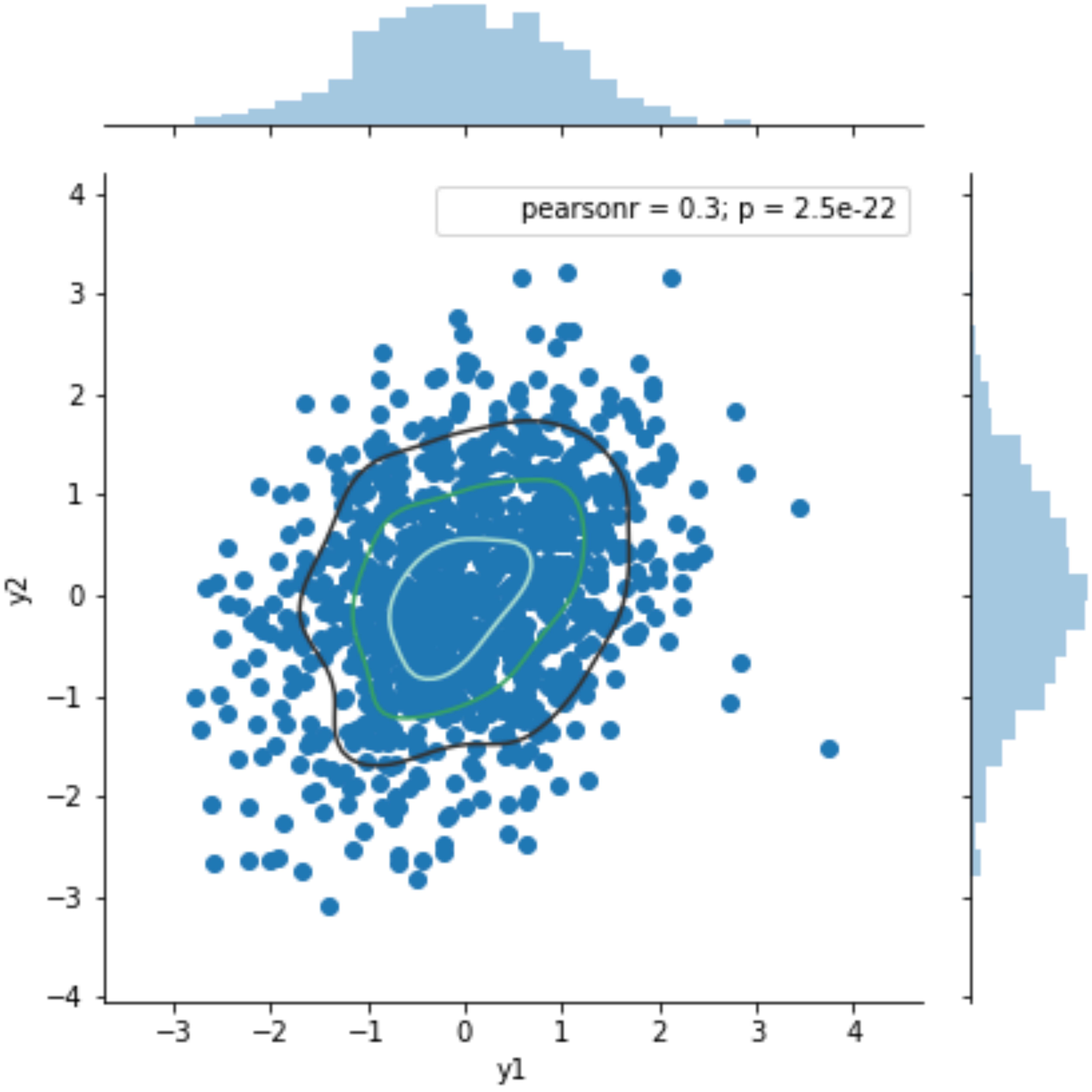


Multinomial

$$p(\mathbf{y}|\boldsymbol{\Sigma}) \propto \exp\left(-\tfrac{1}{2}\mathbf{y}^T\boldsymbol{\Sigma}^{-1}\mathbf{y}\right)$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1.0 & \mathbf{X.X} \\ \mathbf{X.X} & 1.0 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

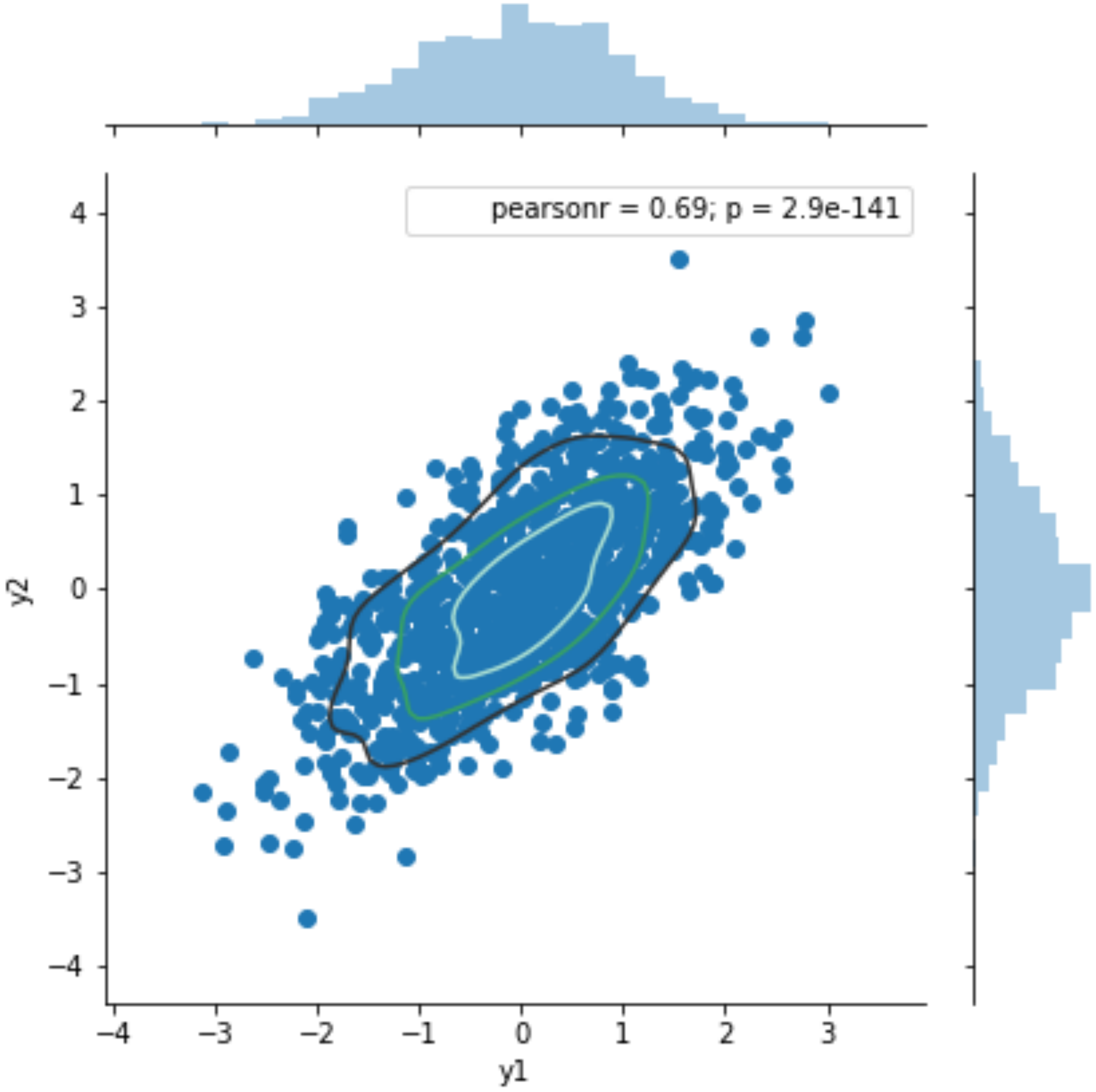


Multinomial

$$p(\mathbf{y}|\mathbf{\Sigma}) \propto \exp\left(-\frac{1}{2}\mathbf{y}^T\mathbf{\Sigma}^{-1}\mathbf{y}\right)$$

$$\mathbf{\Sigma} = \begin{bmatrix} 1.0 & \mathbf{X.X} \\ \mathbf{X.X} & 1.0 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

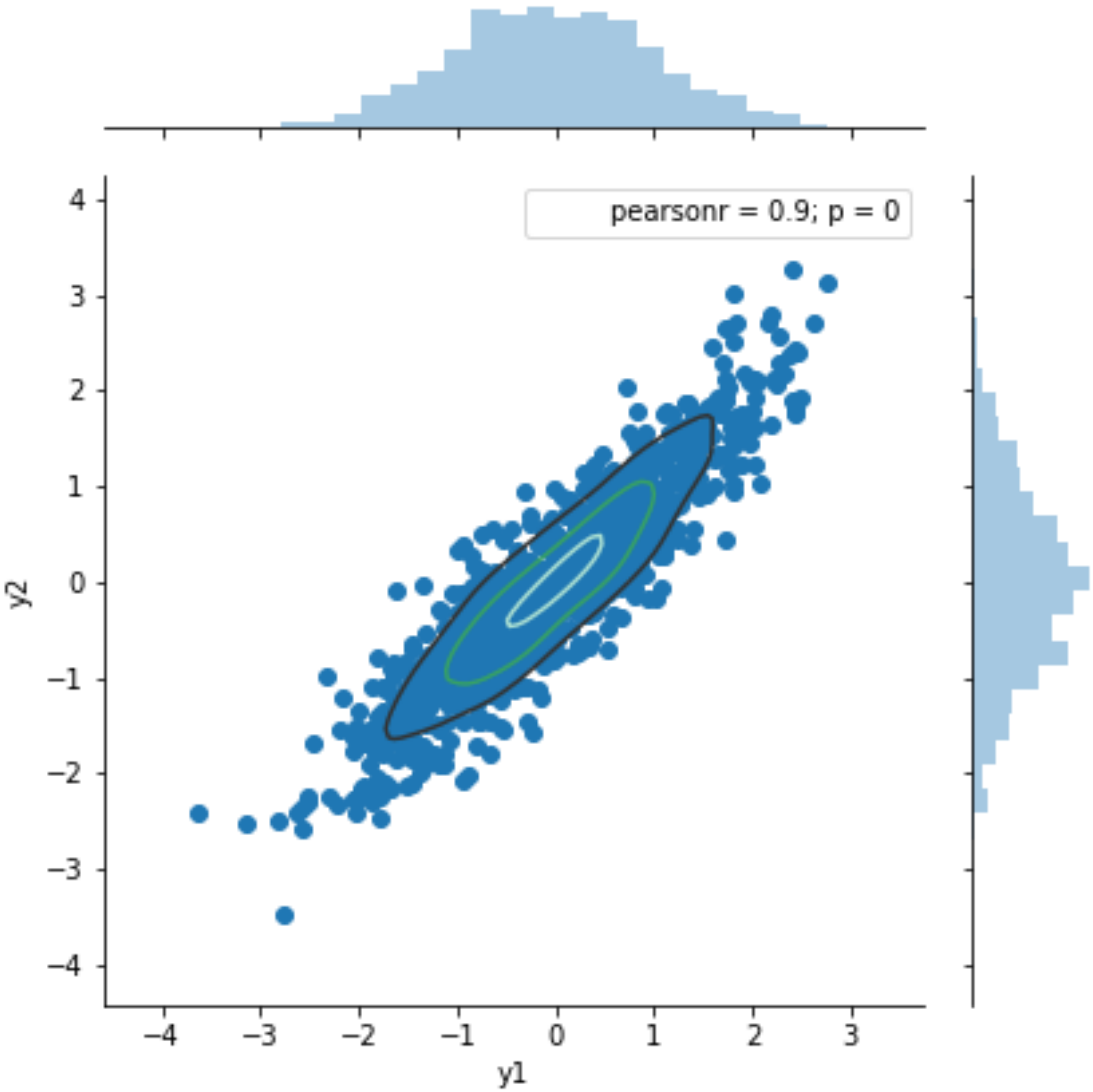


Multinomial

$$p(\mathbf{y}|\mathbf{\Sigma}) \propto \exp\left(-\frac{1}{2}\mathbf{y}^T\mathbf{\Sigma}^{-1}\mathbf{y}\right)$$

$$\mathbf{\Sigma} = \begin{bmatrix} 1.0 & \mathbf{X.X} \\ \mathbf{X.X} & 1.0 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

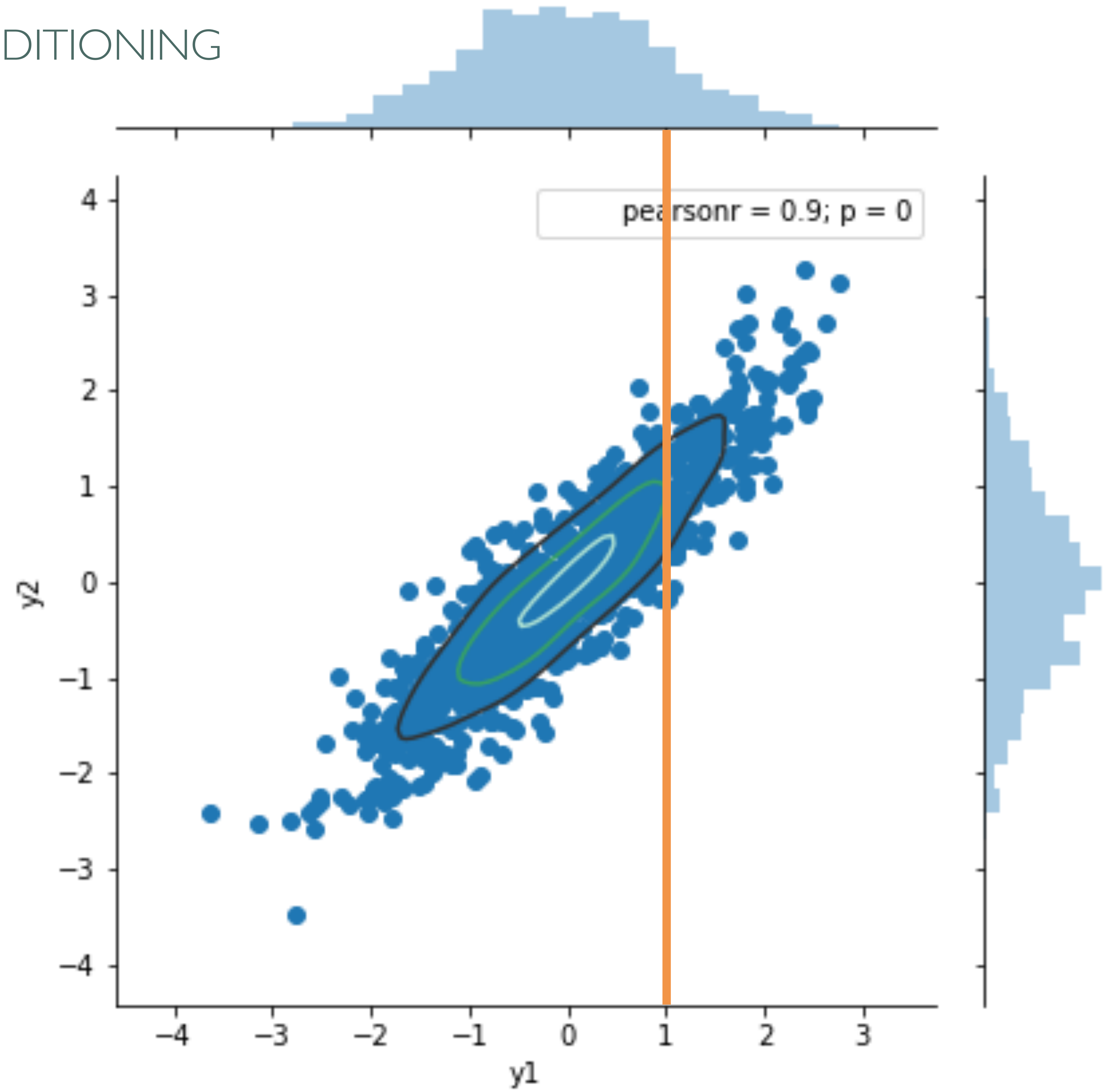


Multinomial

$$p(\mathbf{y}|\mathbf{\Sigma}) \propto \exp\left(-\frac{1}{2}\mathbf{y}^T\mathbf{\Sigma}^{-1}\mathbf{y}\right)$$

$$\mathbf{\Sigma} = \begin{bmatrix} 1.0 & \textbf{X.X} \\ \textbf{X.X} & 1.0 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$



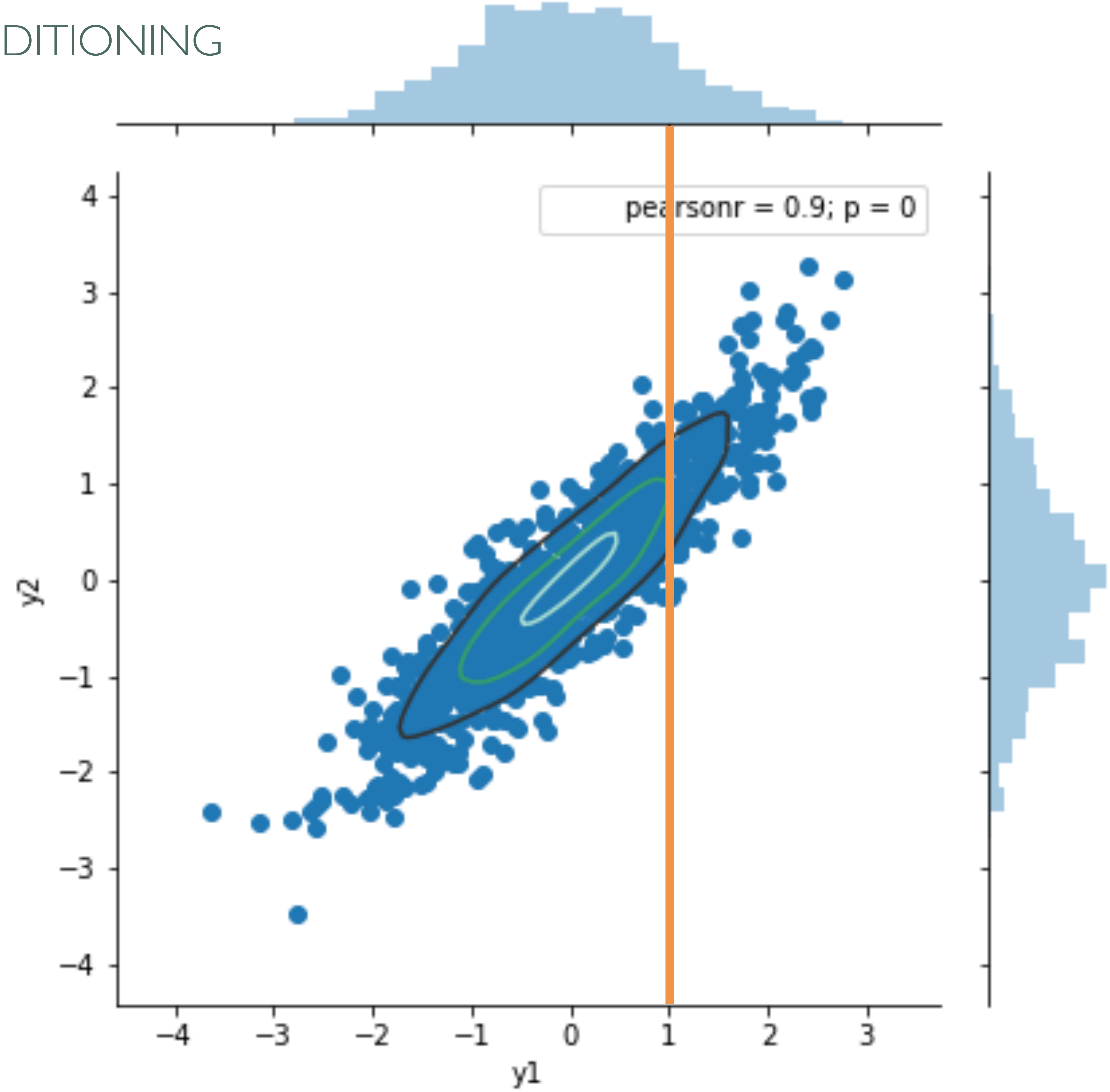
Conditioning

$$p(y_2 | y_1, \mathbf{\Sigma}) \propto \exp\left(-\frac{1}{2}(y_2 - \mu_\star)\,\Sigma_\star^{-1}\,(y_2 - \mu_\star)\right)$$

$$\mathbf{\Sigma} = \begin{bmatrix} 1.0 & 0.9 \\ 0.9 & 1.0 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Conditioning

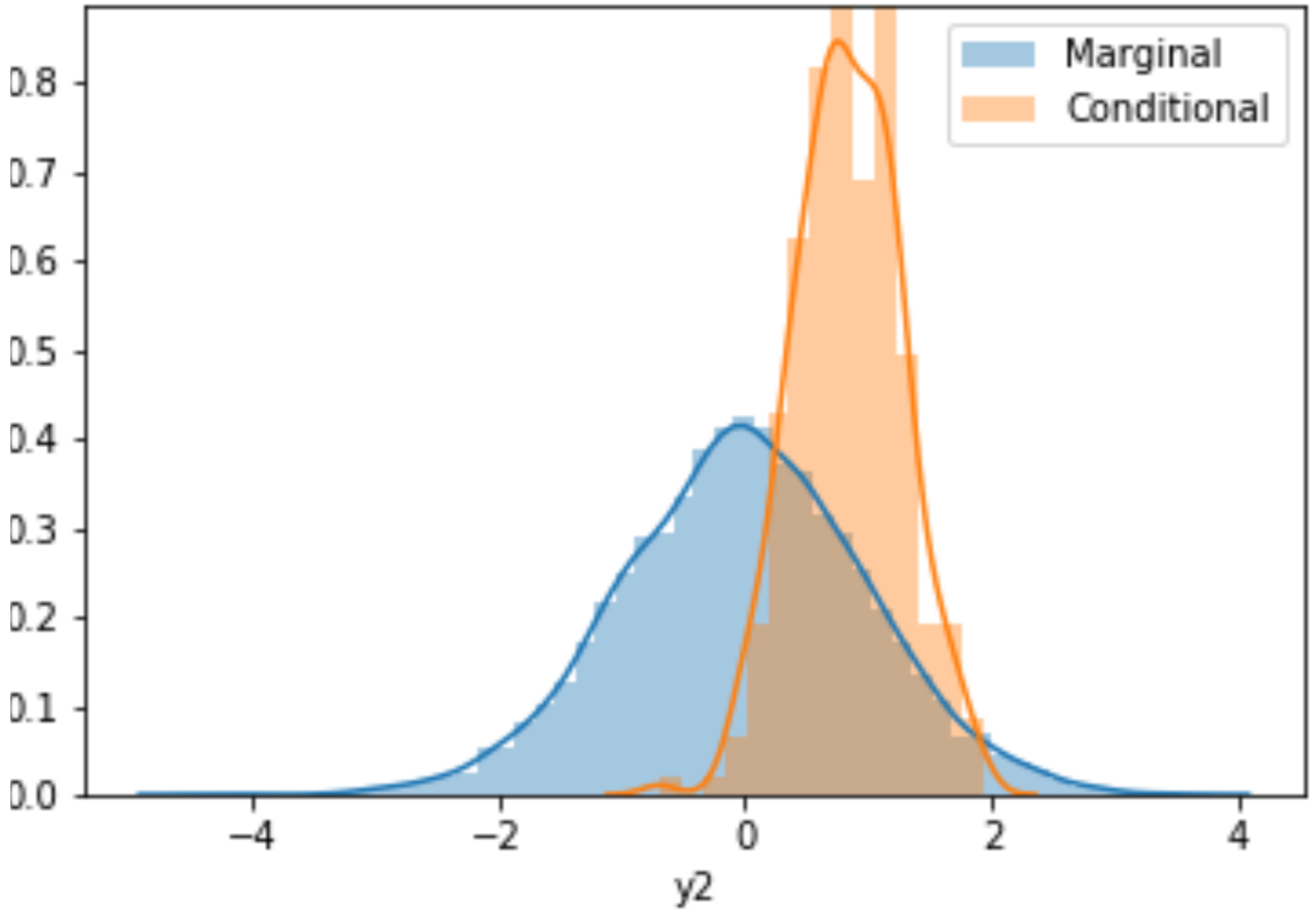


Conditioning

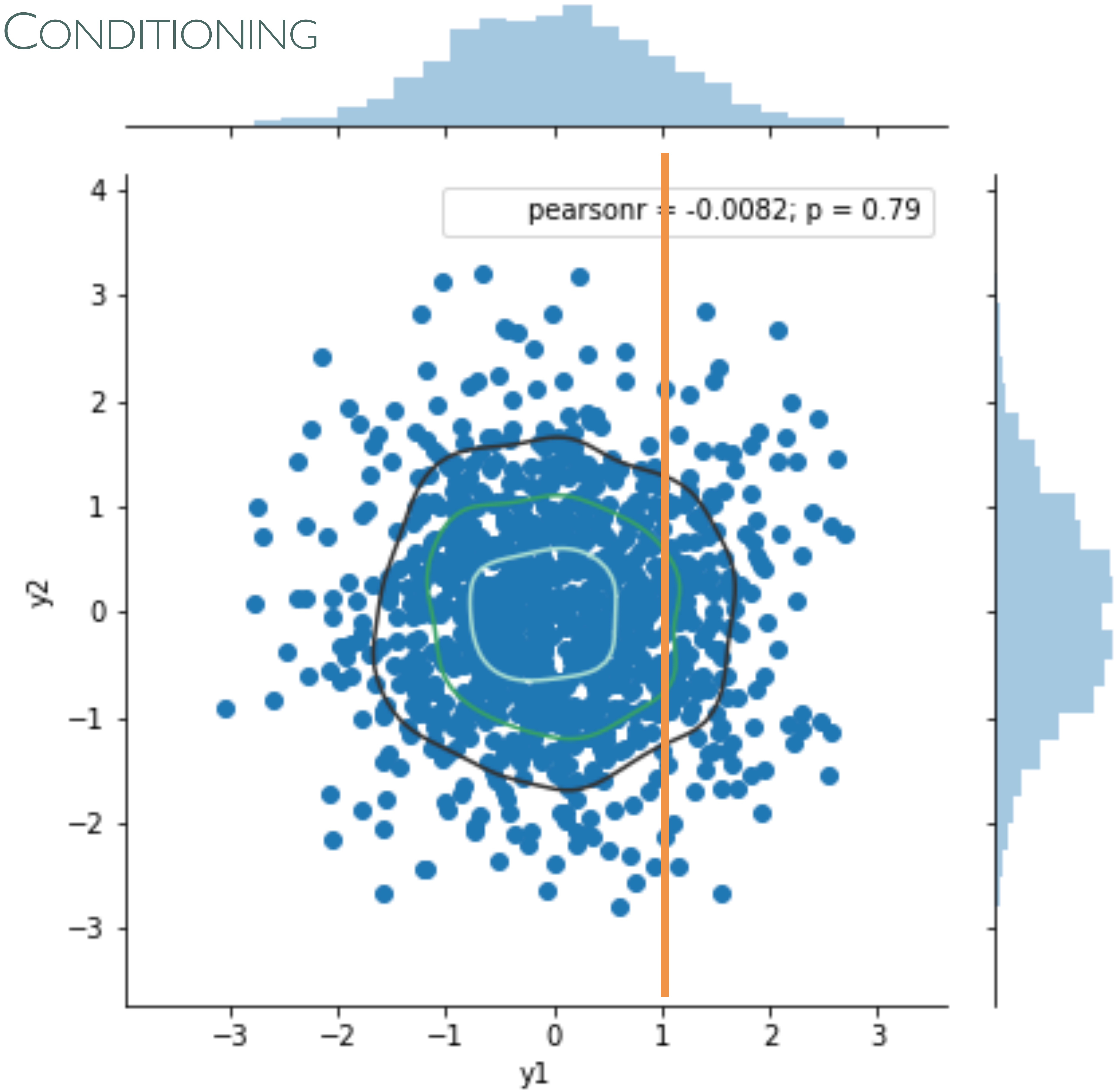
$$p(y_2 | y_1, \mathbf{\Sigma}) \propto \exp\left(-\tfrac{1}{2}(y_2 - \mu_\star)\, \Sigma_\star^{-1}\, (y_2 - \mu_\star)\right)$$

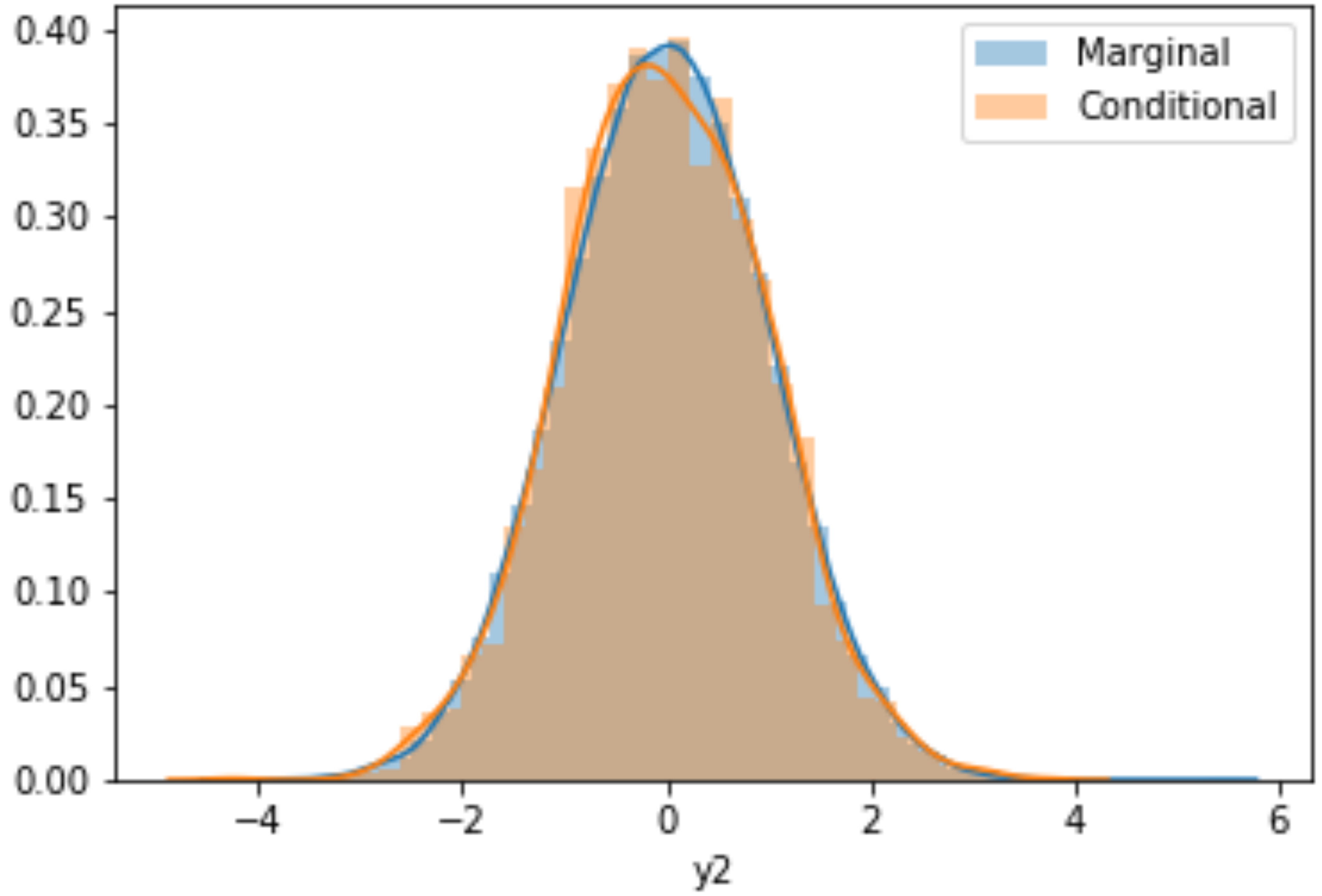
$$\mathbf{\Sigma} = \begin{bmatrix} 1.0 & 0.9 \\ 0.9 & 1.0 \end{bmatrix}$$

$$\mu_\star = W\, y_1 = 1.0\, ,\, \Sigma_\star = 0.2$$

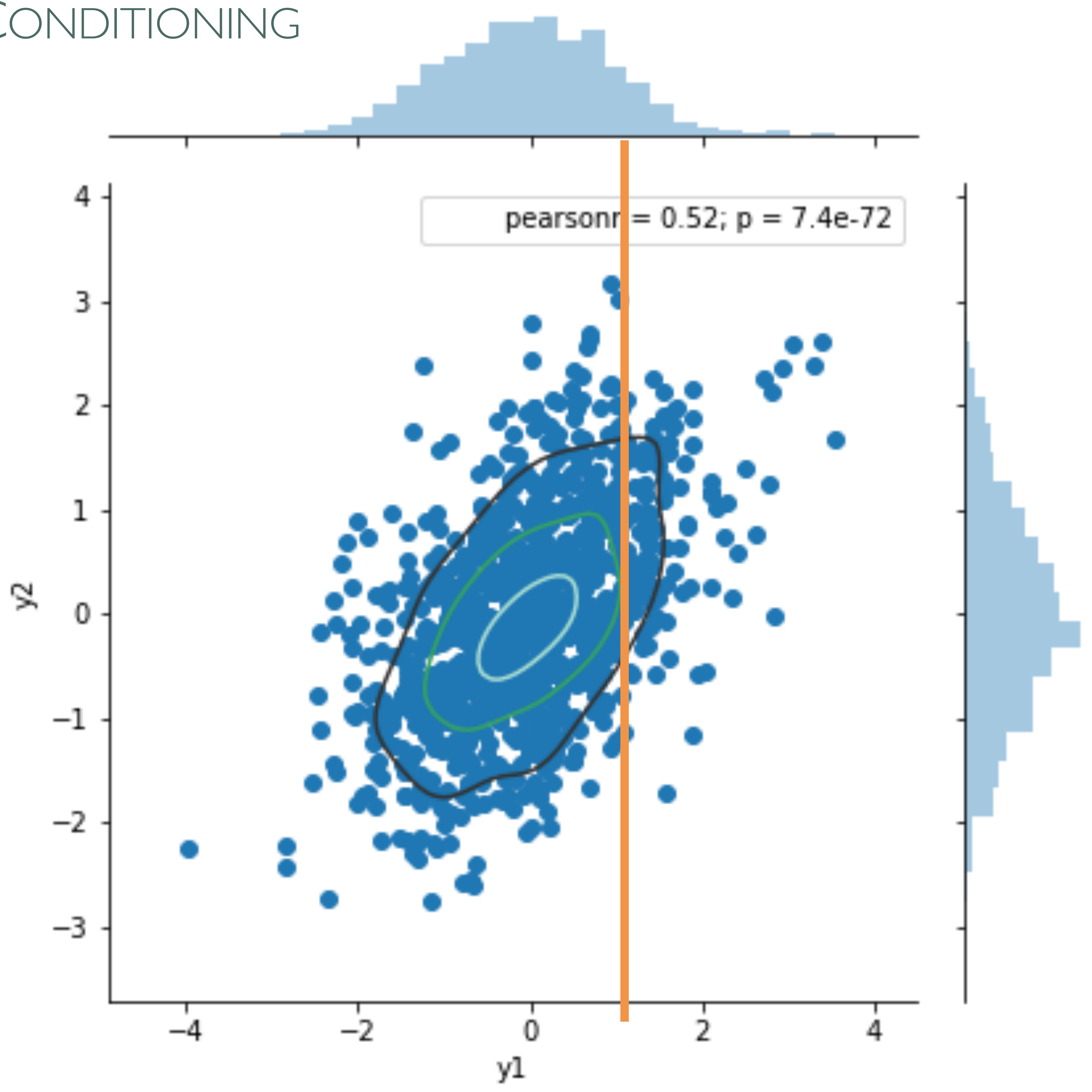


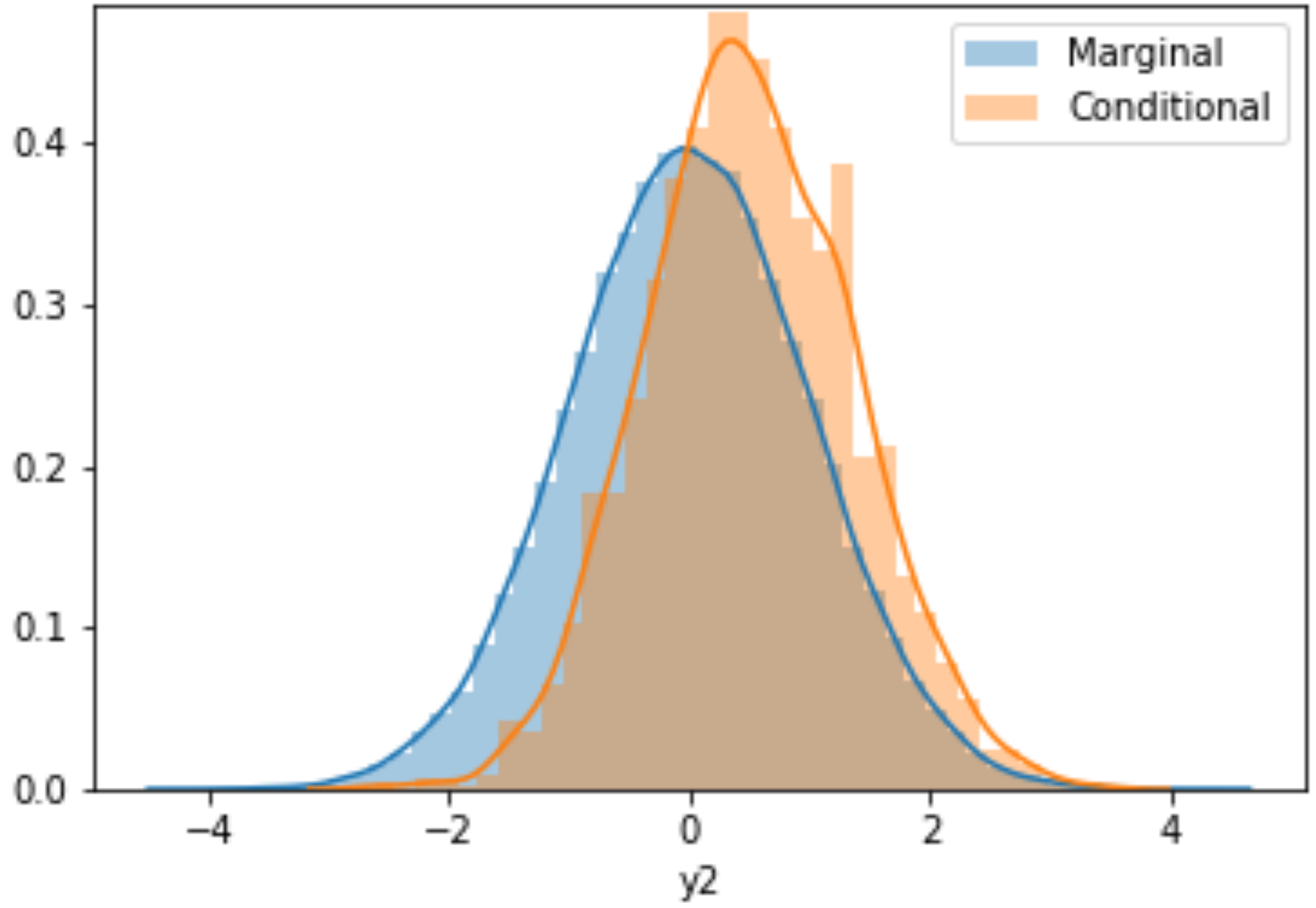
CONDITIONING



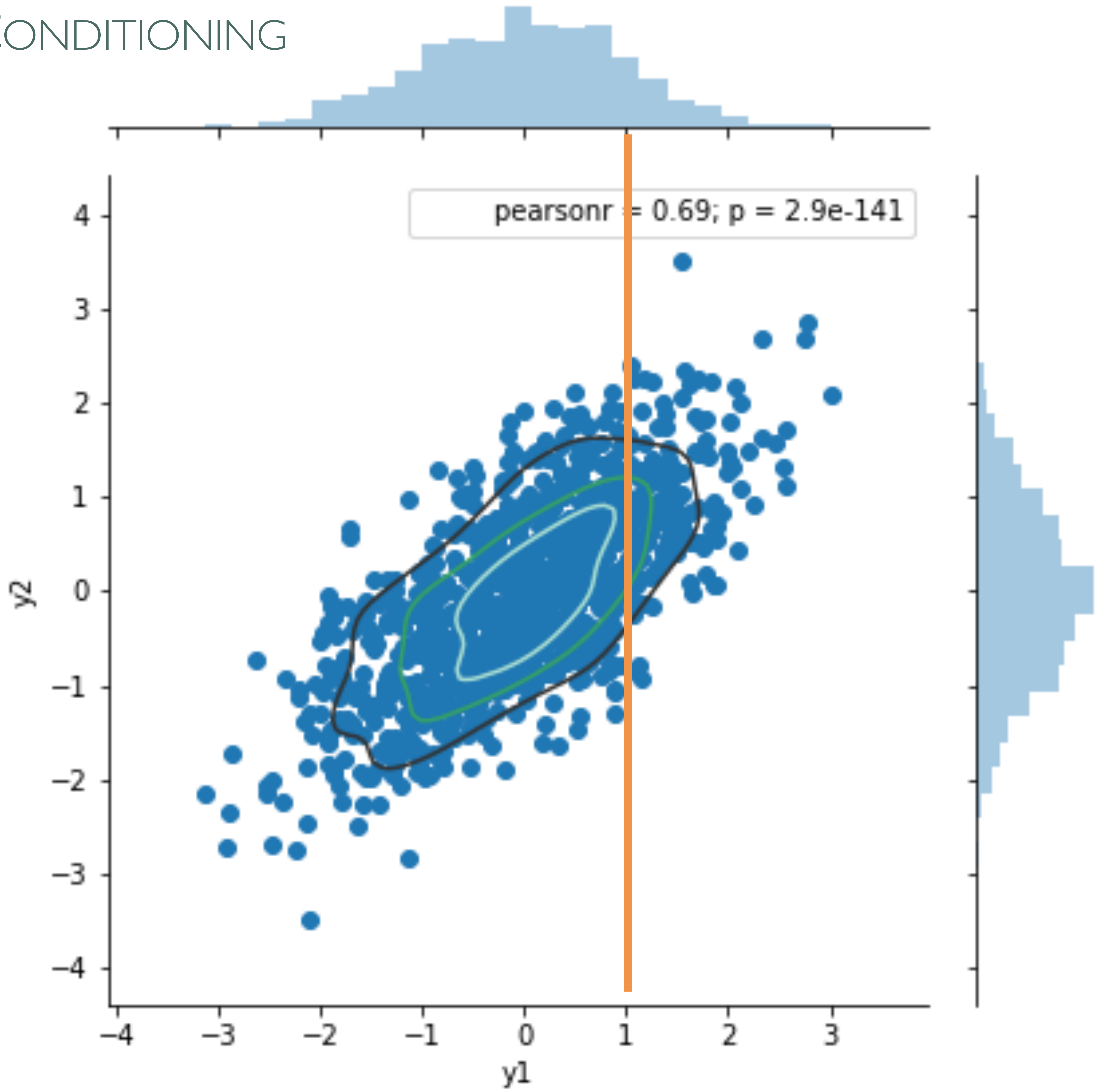


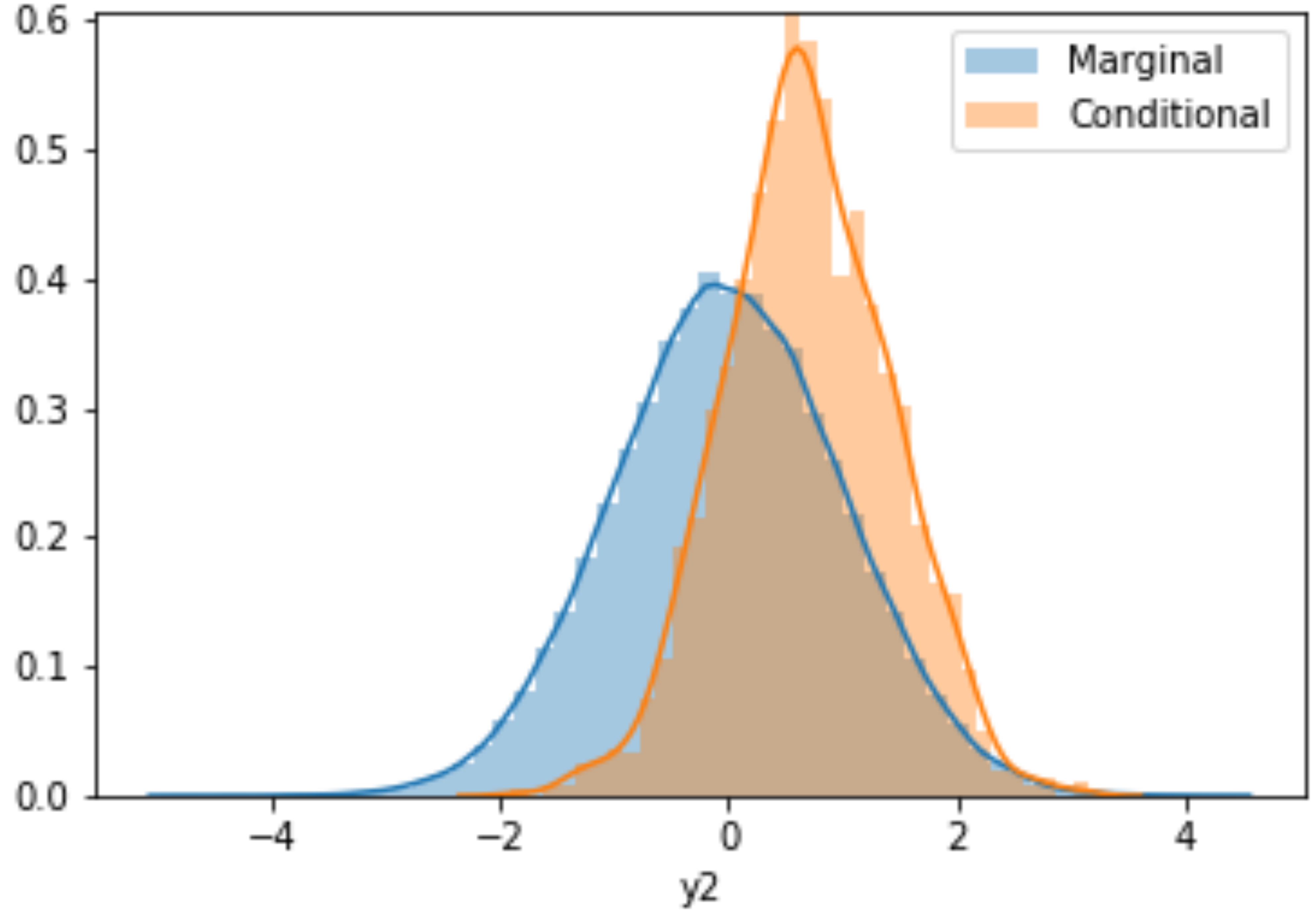
Conditioning



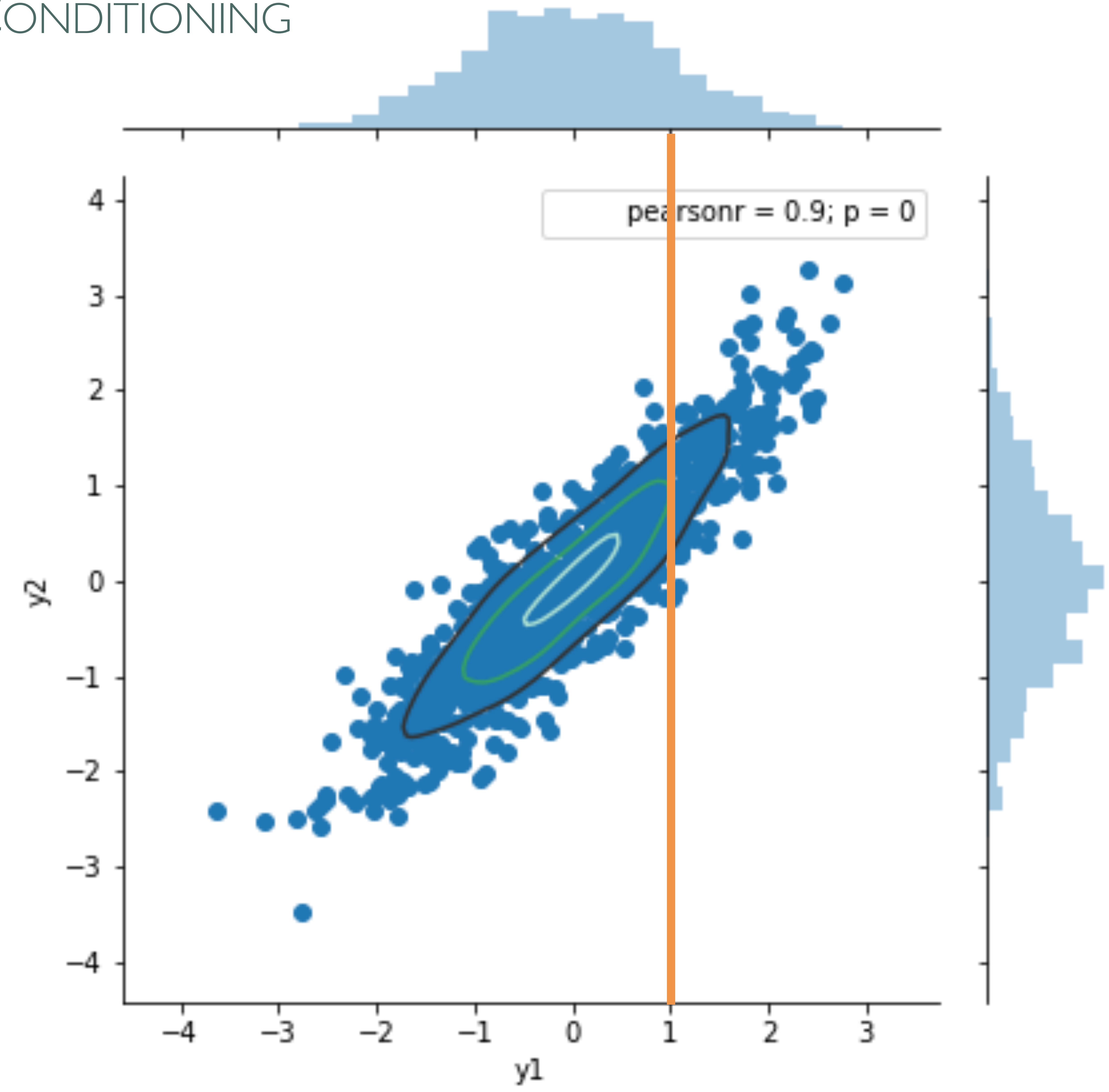


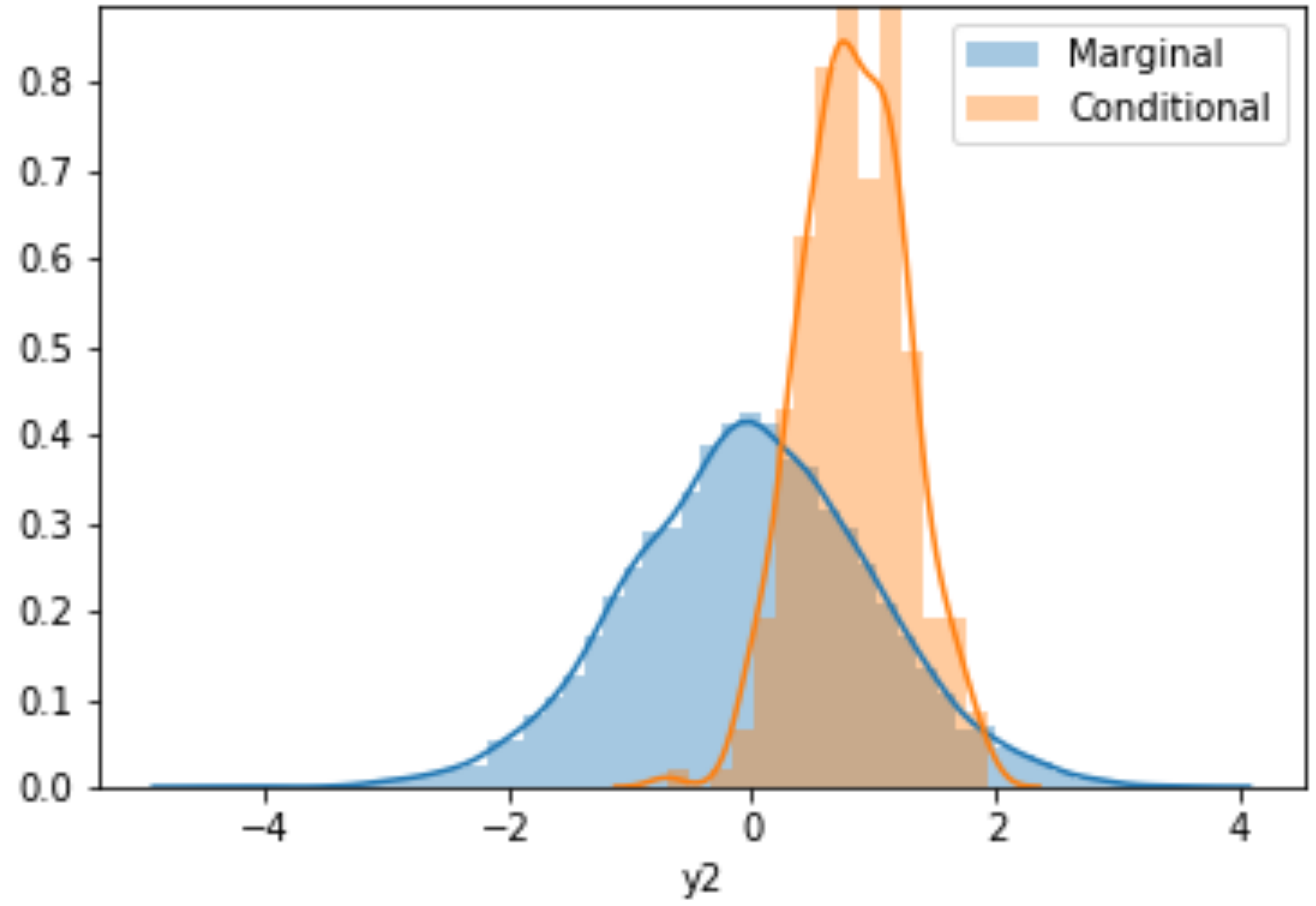
Conditioning



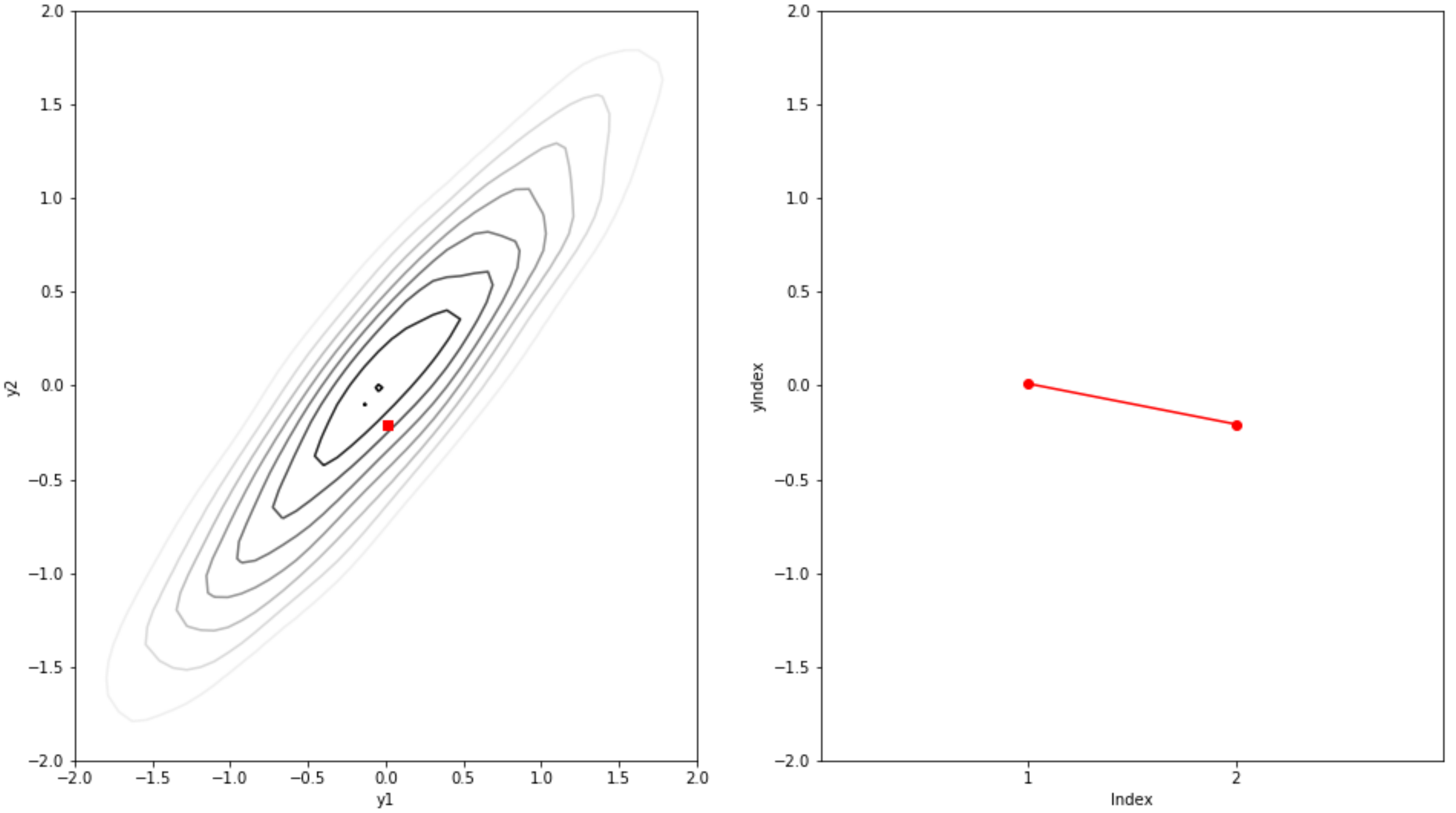


CONDITIONING

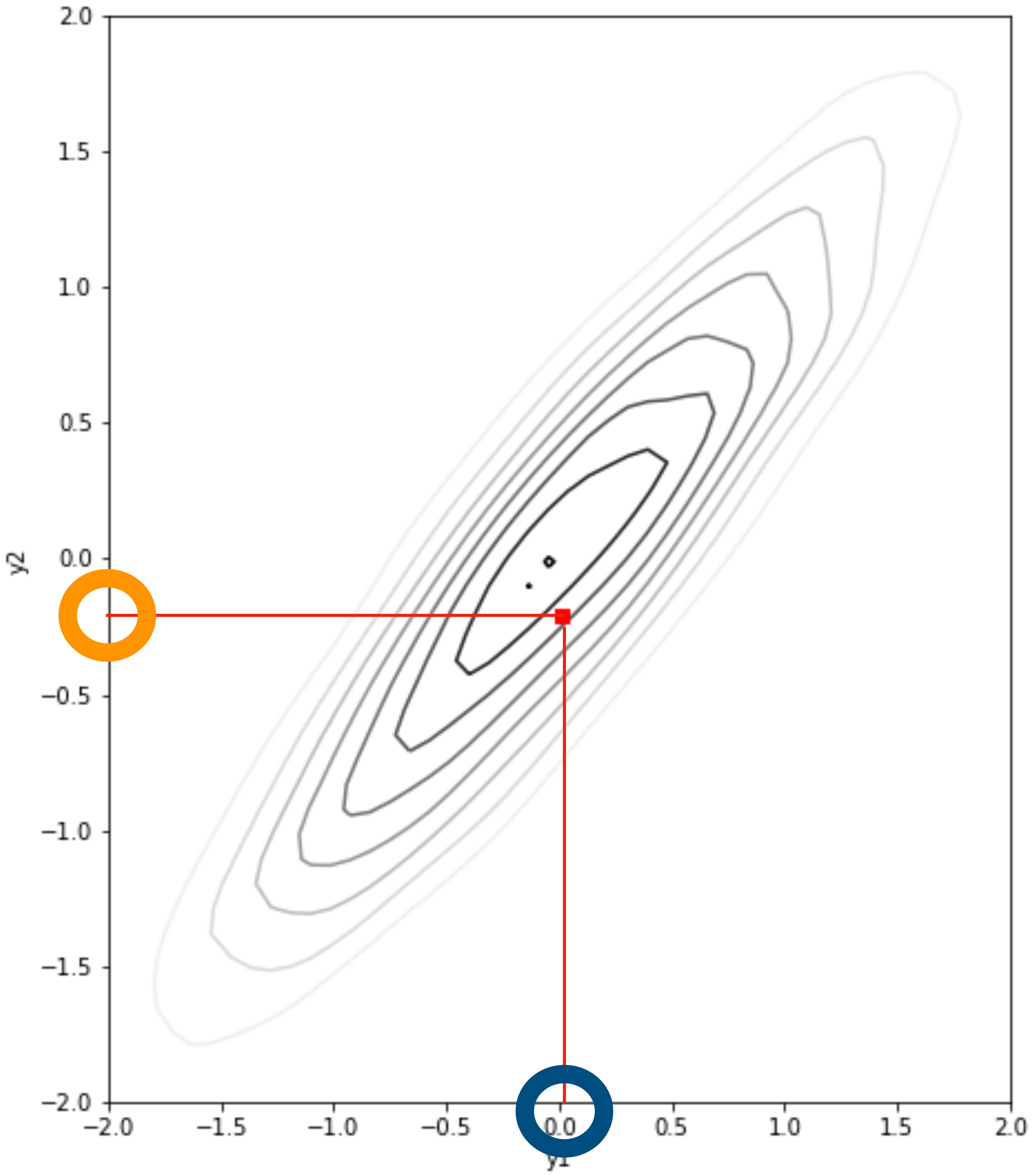


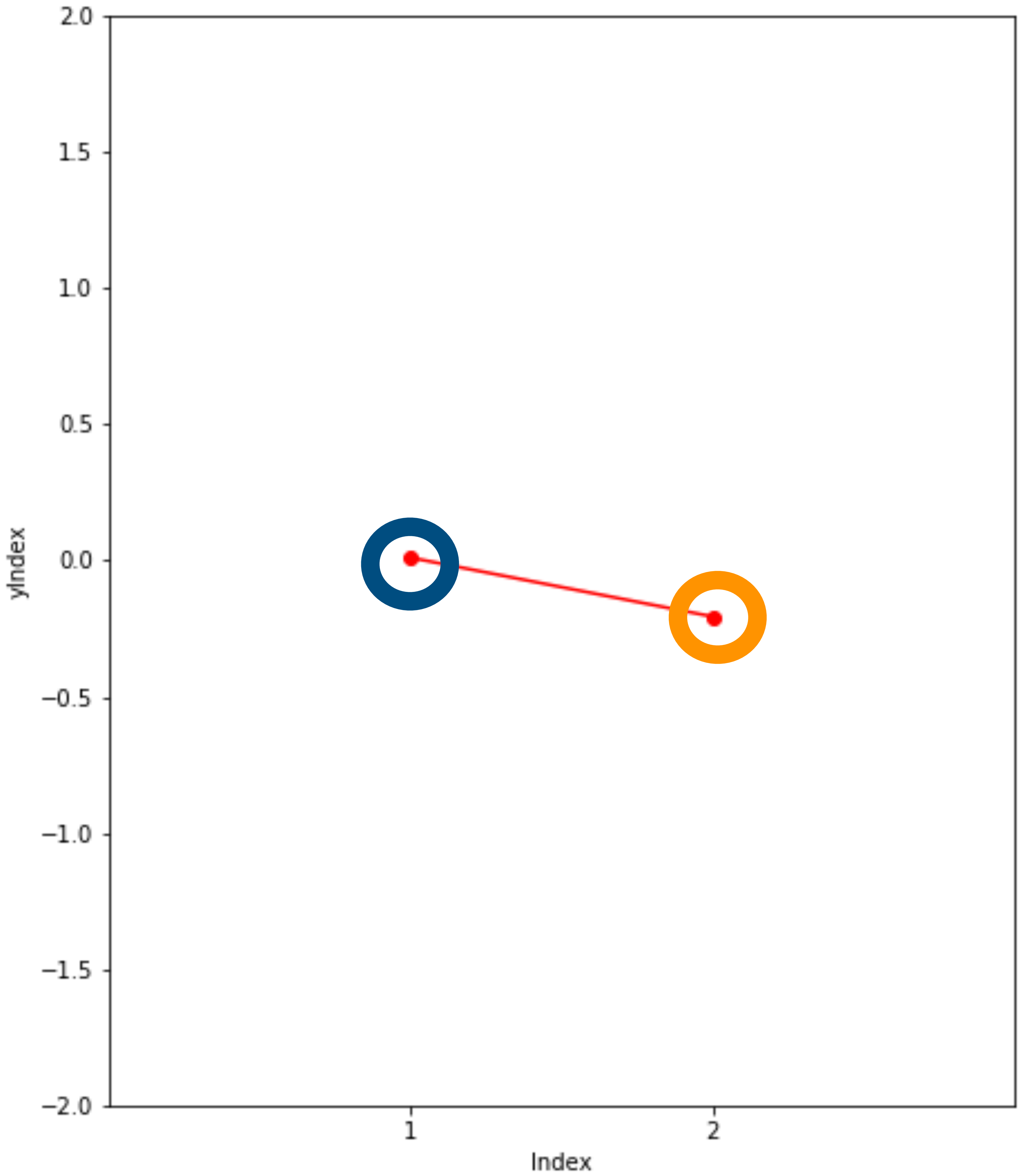


VISUALISATION

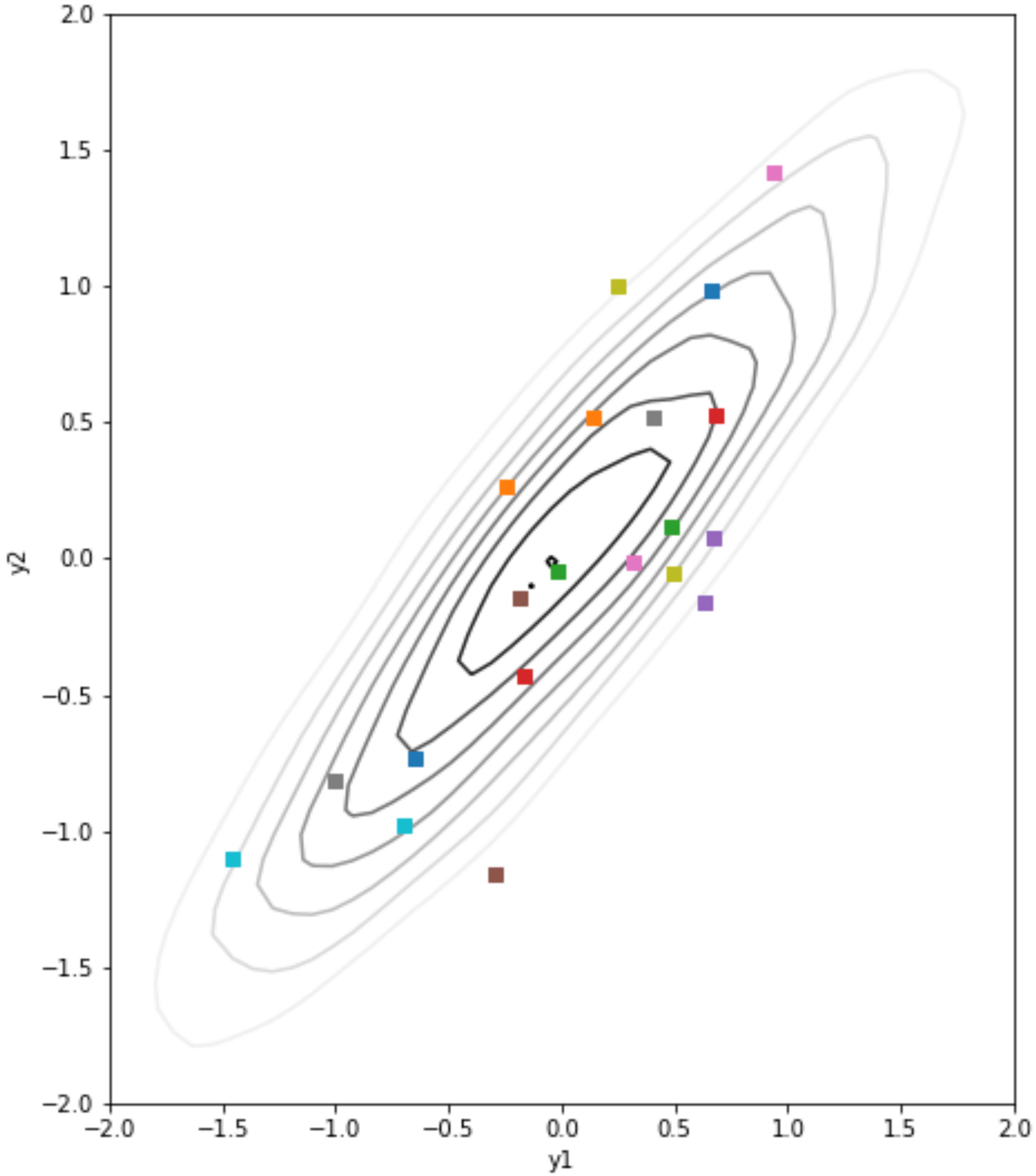


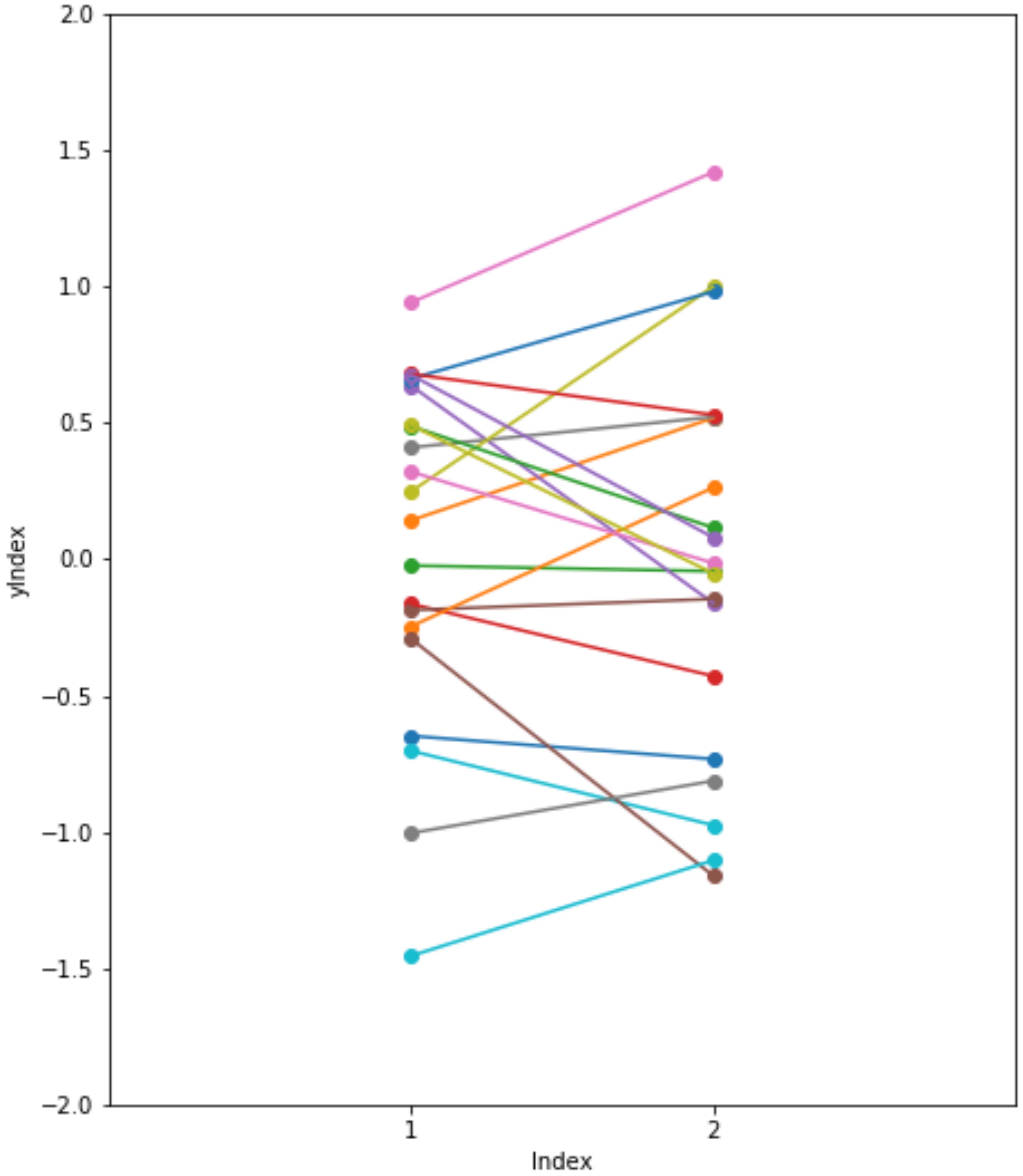
VISUALISATION



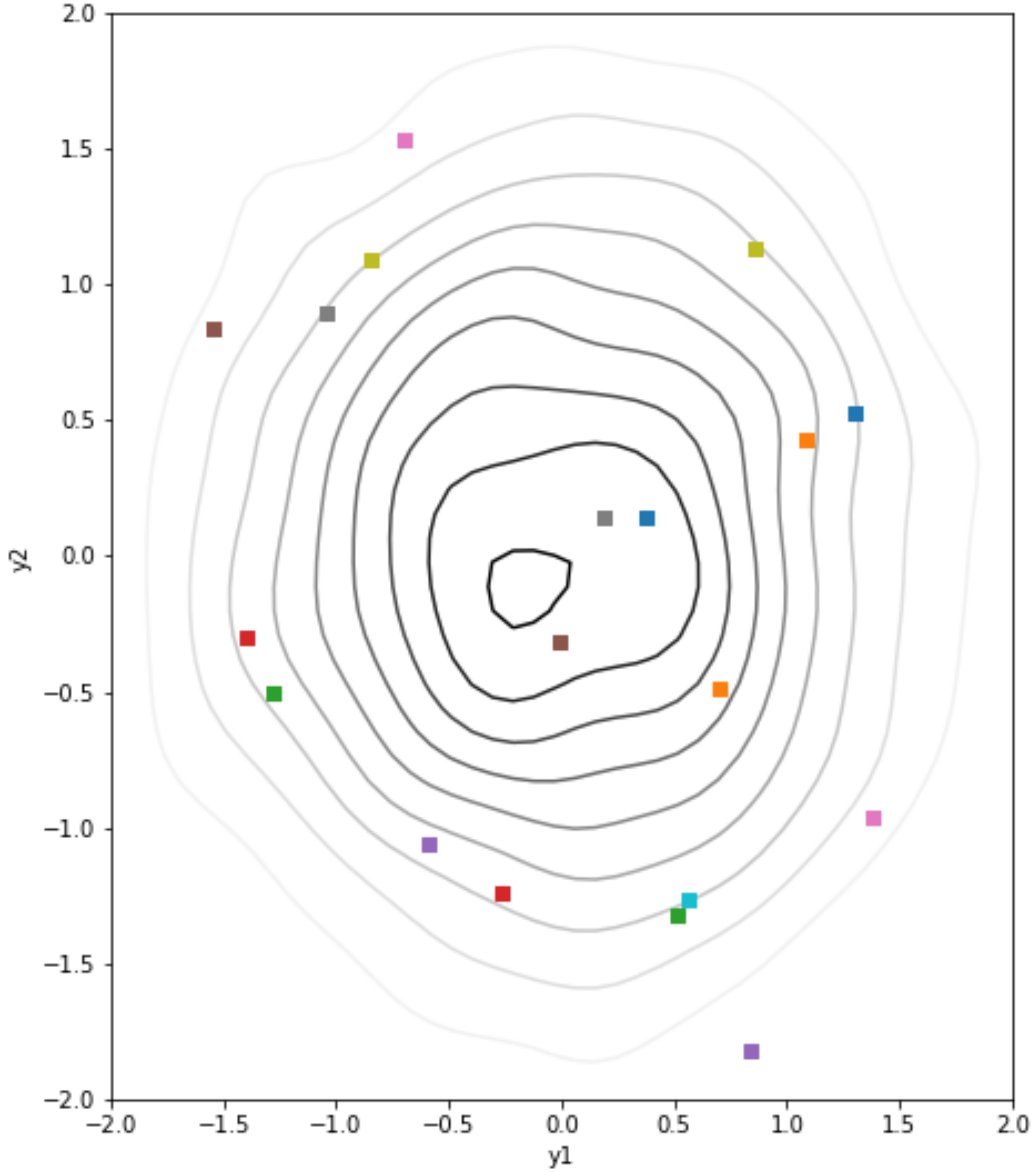


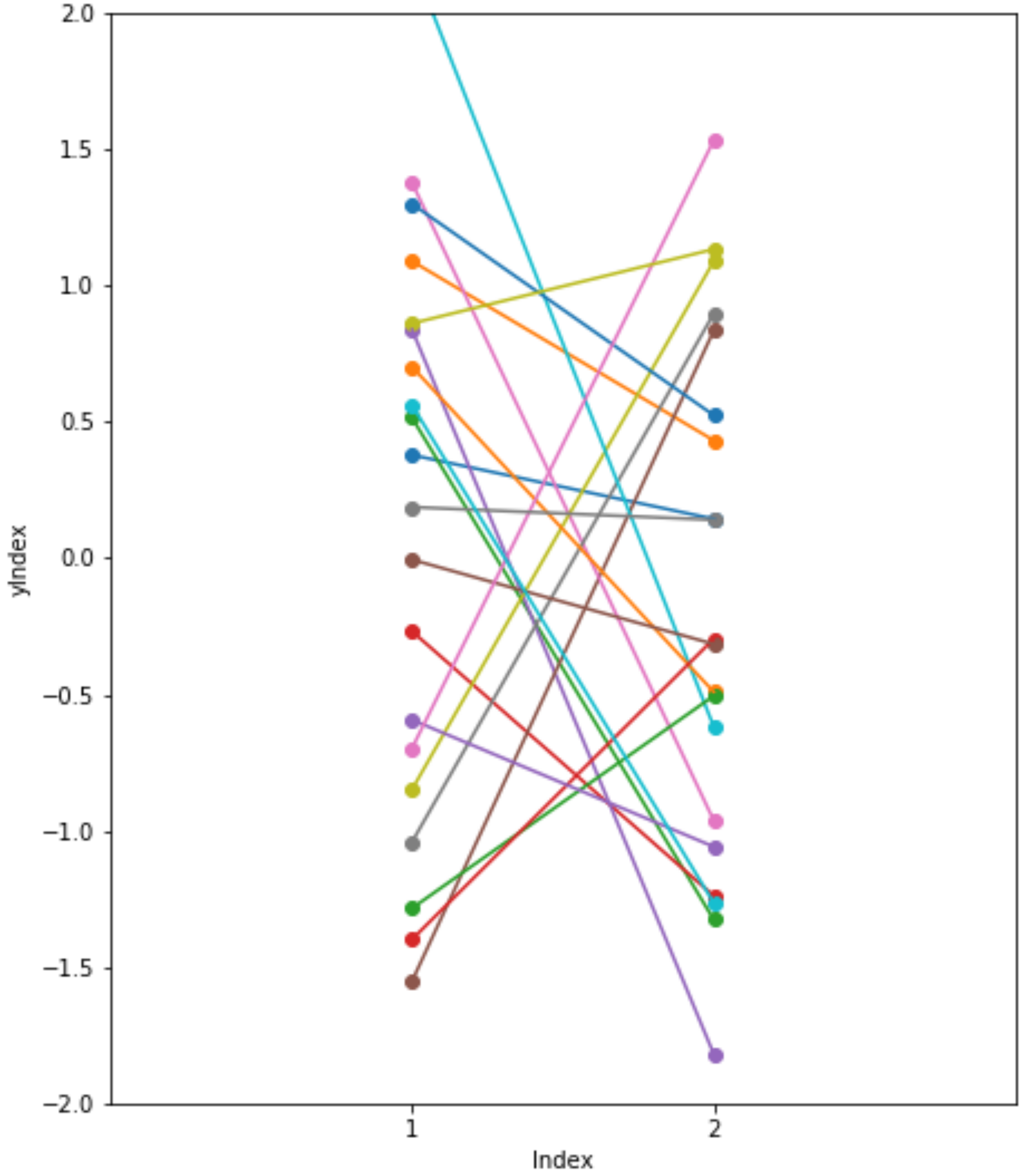
VISUALISATION





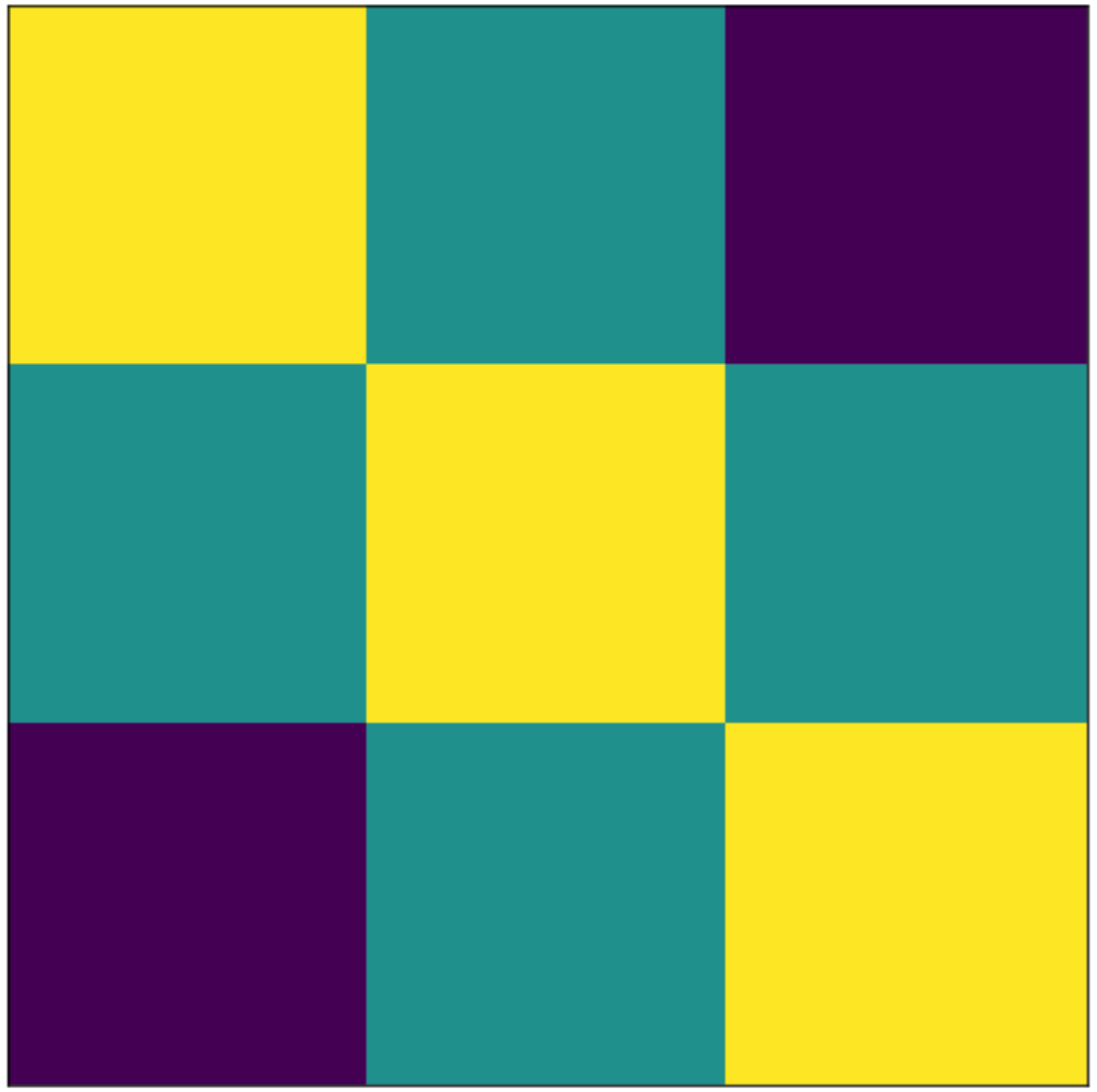
VISUALISATION





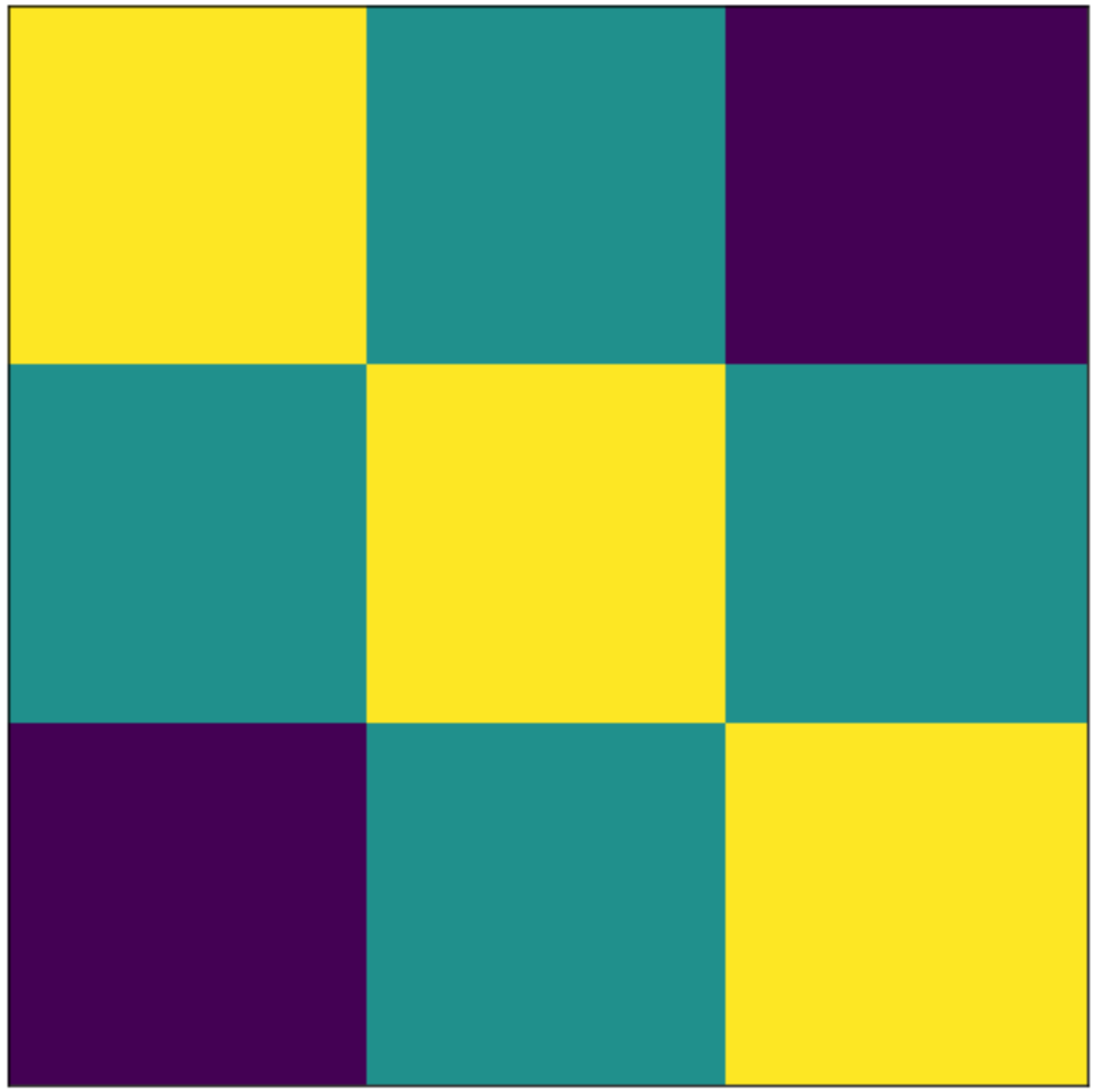
Visualisation

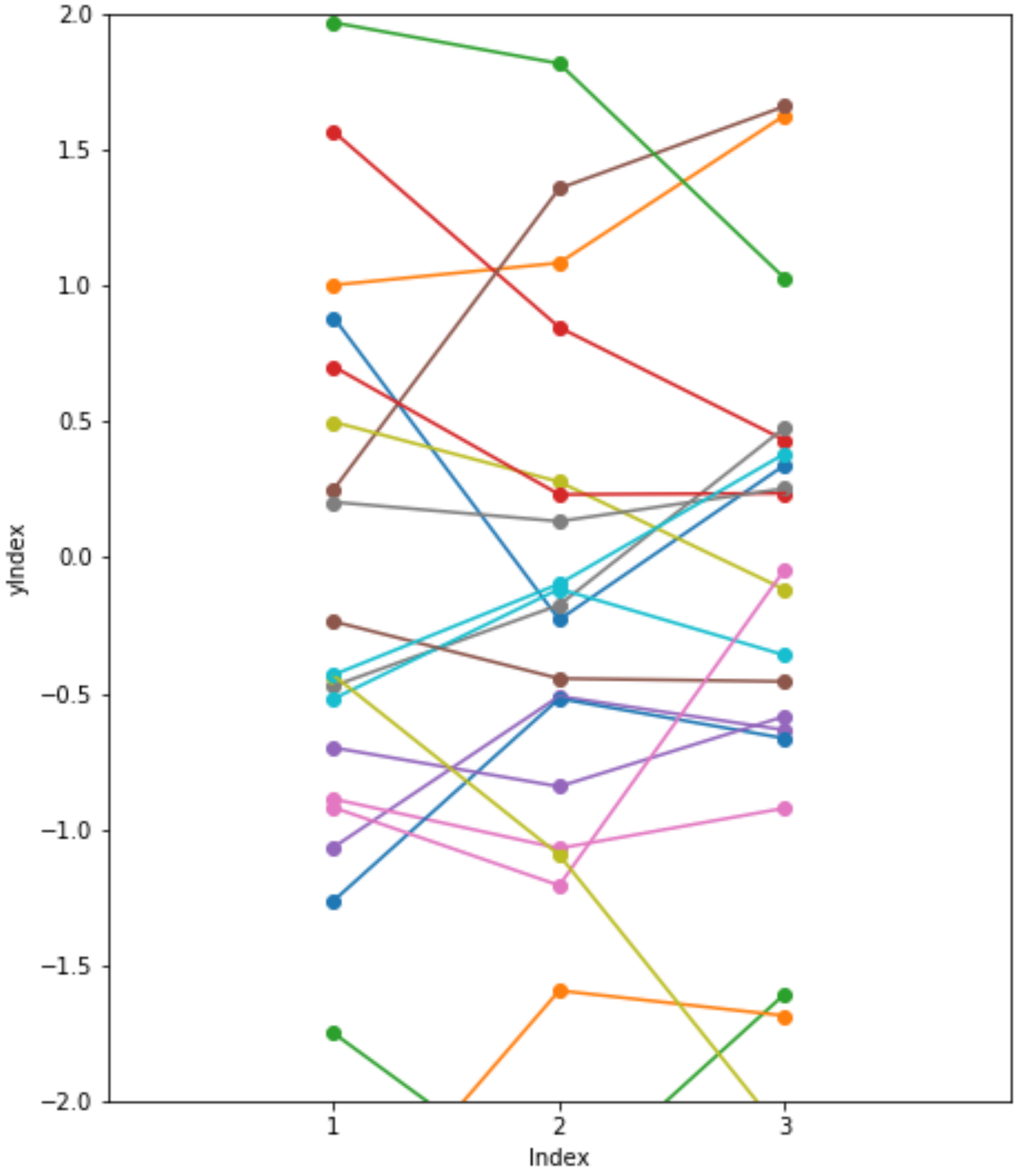
$$\boldsymbol{\Sigma} = \begin{bmatrix} 1.0 & 0.8 & 0.6 \\ 0.8 & 1.0 & 0.8 \\ 0.6 & 0.8 & 1.0 \end{bmatrix} \longrightarrow$$



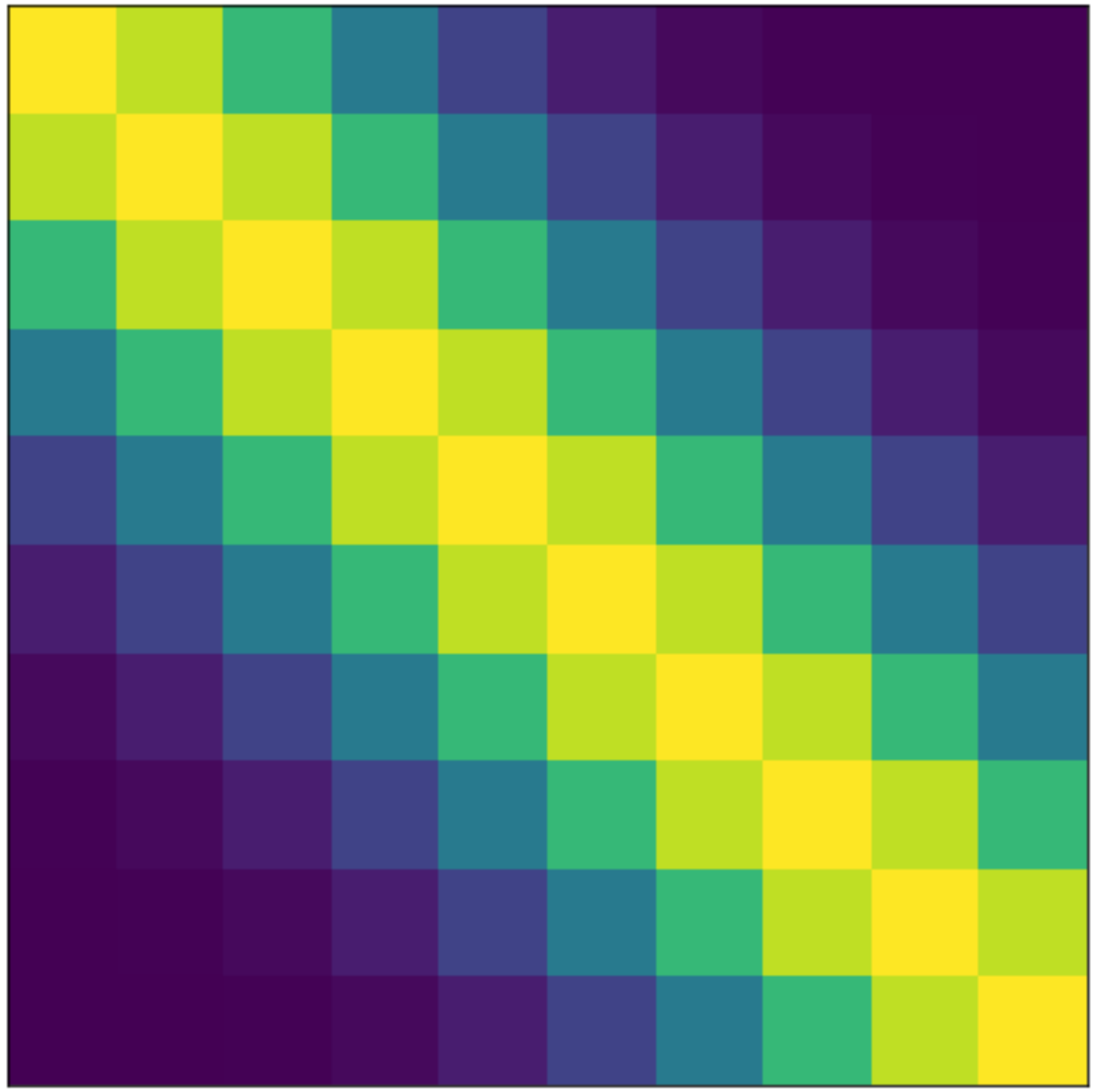
VISUALISATION

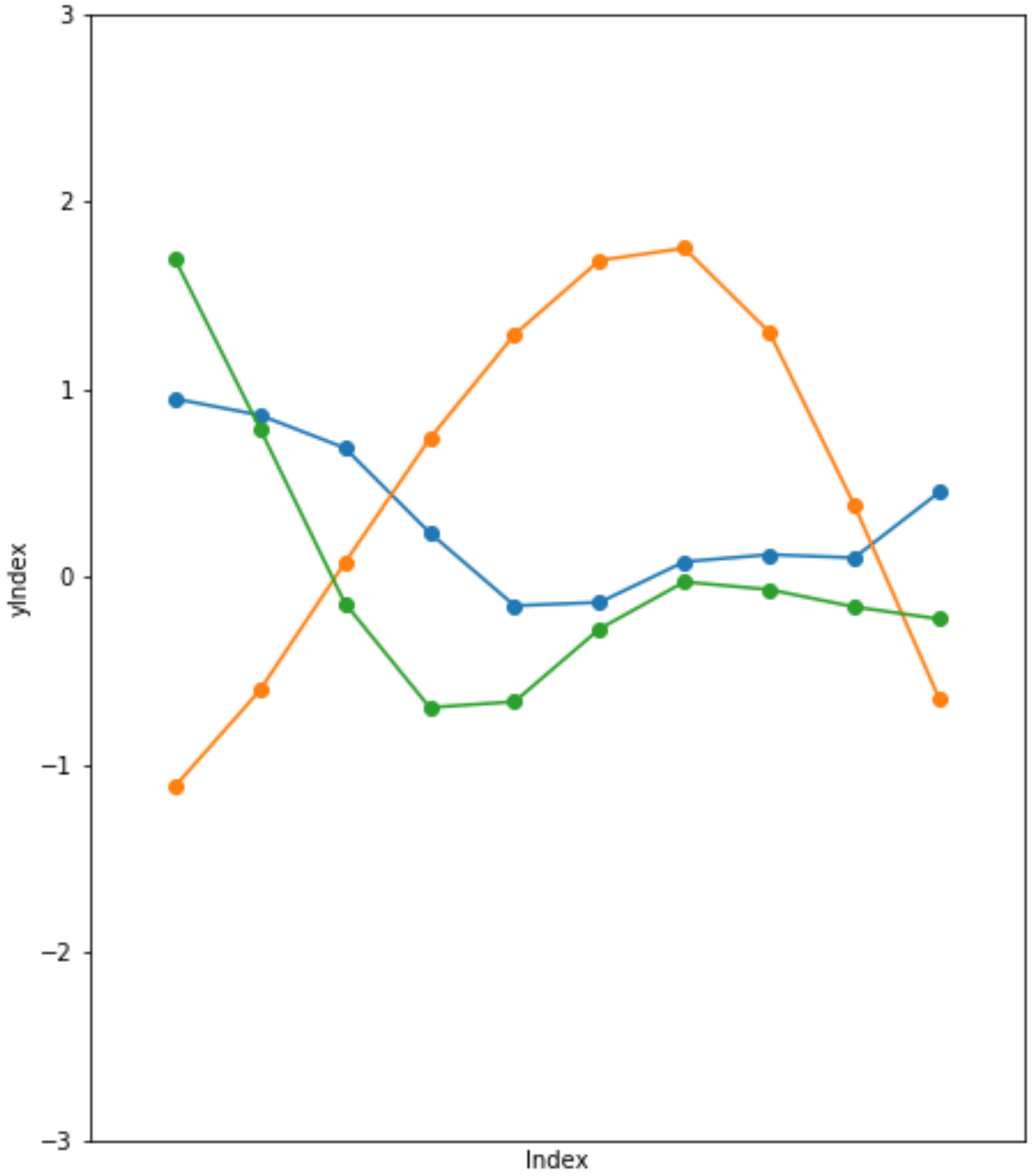
$$\boldsymbol{\Sigma} = \begin{bmatrix} 1.0 & 0.8 & 0.6 \\ 0.8 & 1.0 & 0.8 \\ 0.6 & 0.8 & 1.0 \end{bmatrix}$$



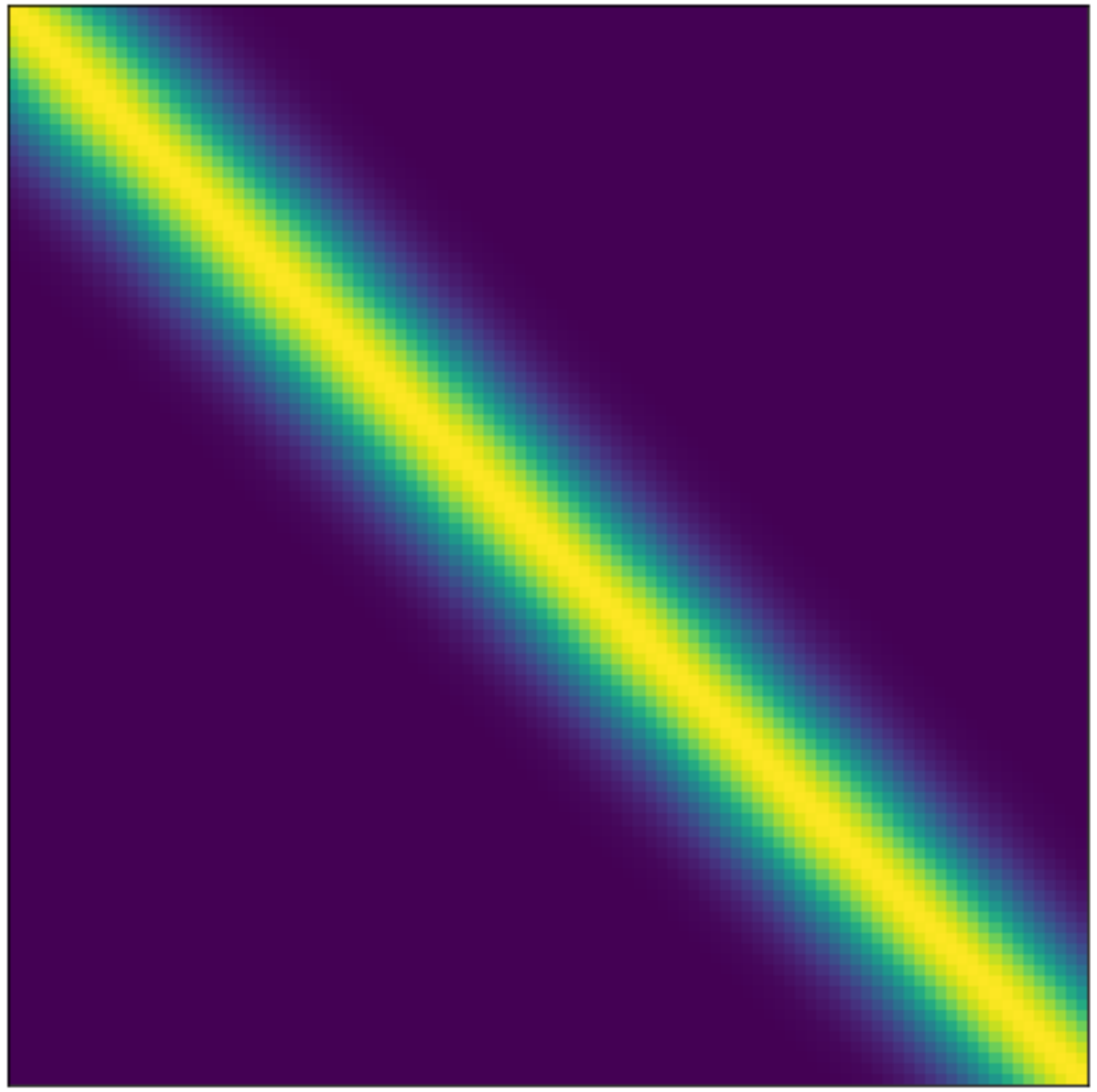


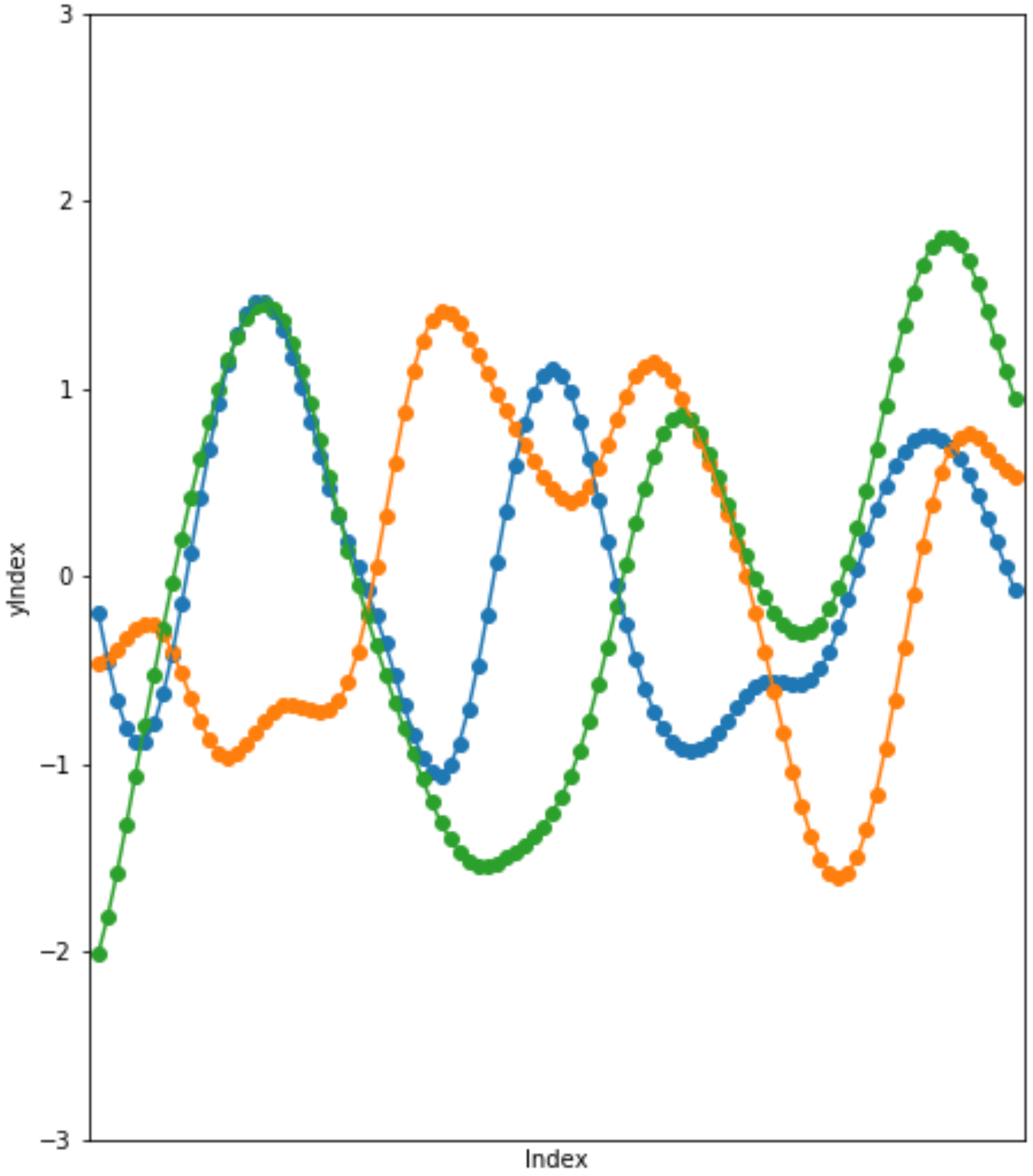
VISUALISATION





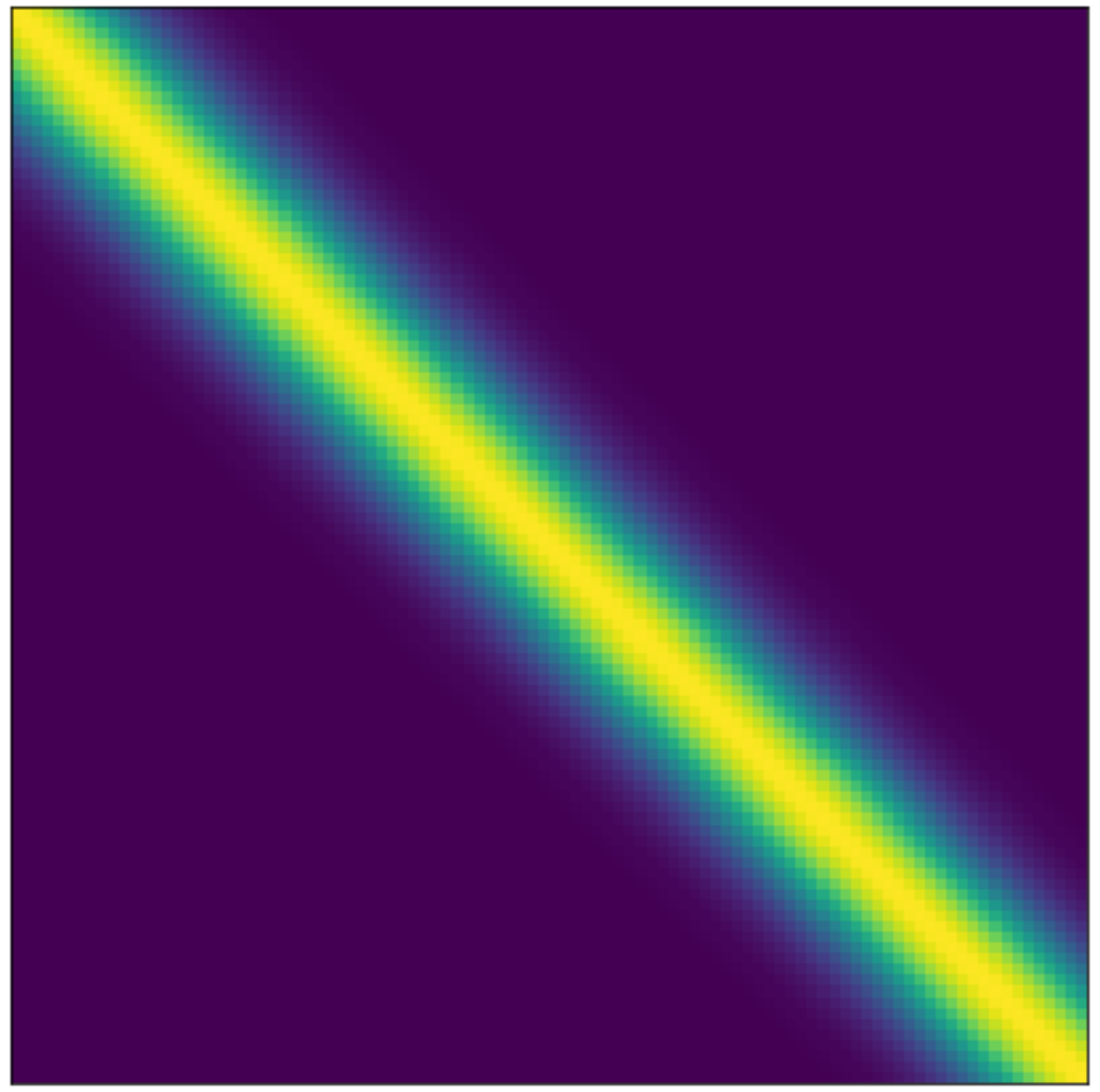
Visualisation

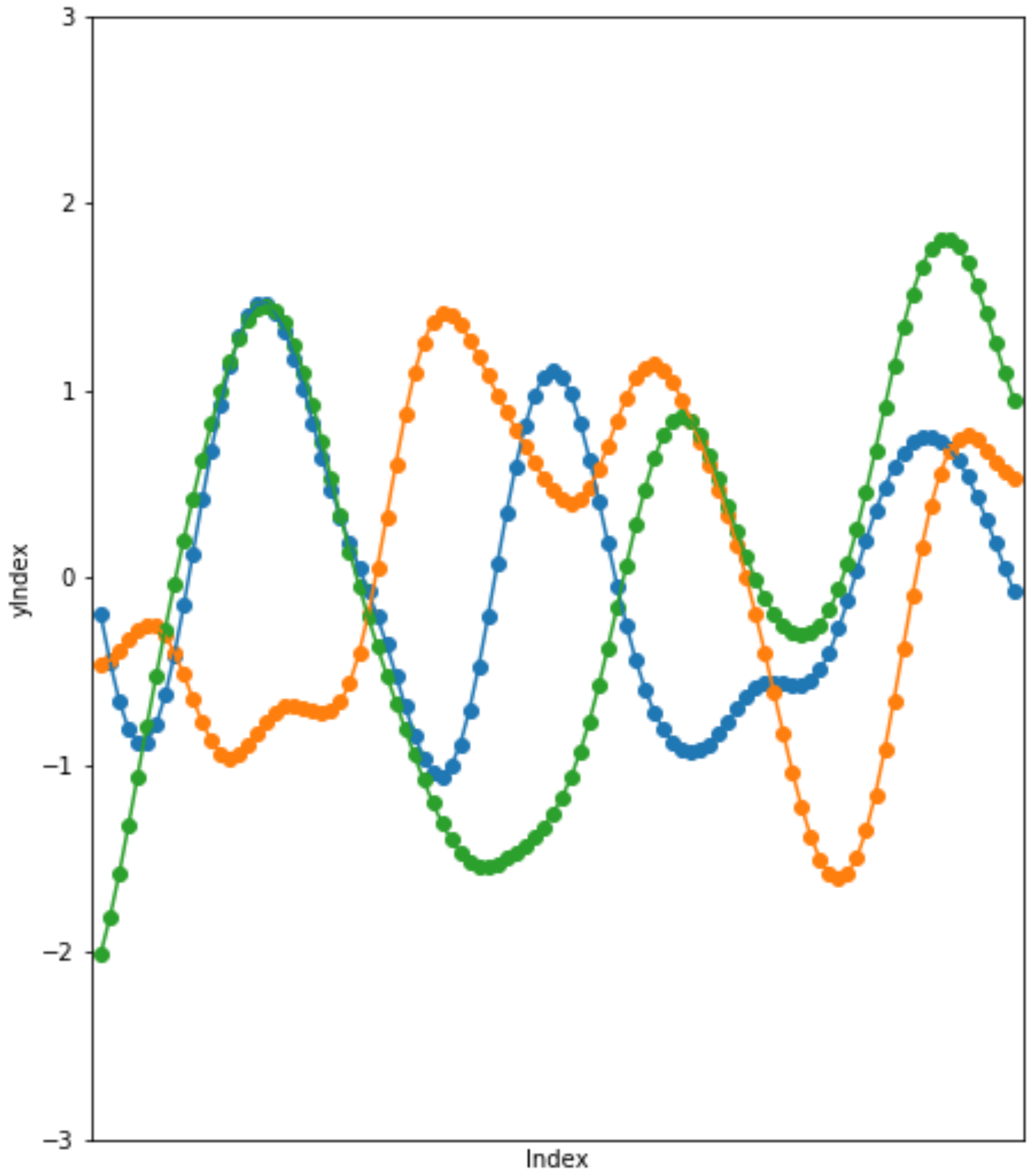




Definition of a GP

Loosely - A multivariate Gaussian of infinite length





Definition

Not so loosely:

f is a Gaussian process if $f(t) = [f(t_1), \ldots, f(t_n)]'$ has a multivariate normal distribution for all $t = [t_1, \ldots, t_n]'$:

$f(t) \sim \mathcal{N}(m(t), K(t,t))$

Where $m(t)$ is any function that maps any t index to a real value.

$K(t,t)$ is the kernel, or covariance, function. Must be a positive semidefinite matrix.

$K(t,t) = \{k(t_i, t_j)\}_{i,j=1\ldots n}$

So $k(t_i, t_j)$ is a function that we can define.

Every finite subset of the domain t has a multivariate normal $f(t) \sim \mathcal{N}(m(t), K(t,t))$.

Session 3 - Correlation in the data	
Section 1 ~30 Mins	Motivation - Correlation in the data
Section 2 ~40 Mins	An introduction to a Gaussian Process
Section 2.1 ~40 Mins	Applying Gaussian Processes
Section 3 ~30 Mins	An introduction to Celerite

Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})d\mathbf{w}}$$

NOTE: FORM FROM RW06

Applying Gaussian Processes

Set up a GP

The Mean function: $m(t) = 0$

The kernal function: $k(t_i, t_j) = \sigma^2 \exp\left(-\frac{1}{2l^2}|t_i - t_j|^2\right)$

$K(t, t) = \{k(t_i, t_j)\}_{i,j=1\ldots n}$

$f(t) \sim \mathcal{GP}(m(t), K(t, t))$

Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})d\mathbf{w}}$$

Kernel Choice: SE, more later

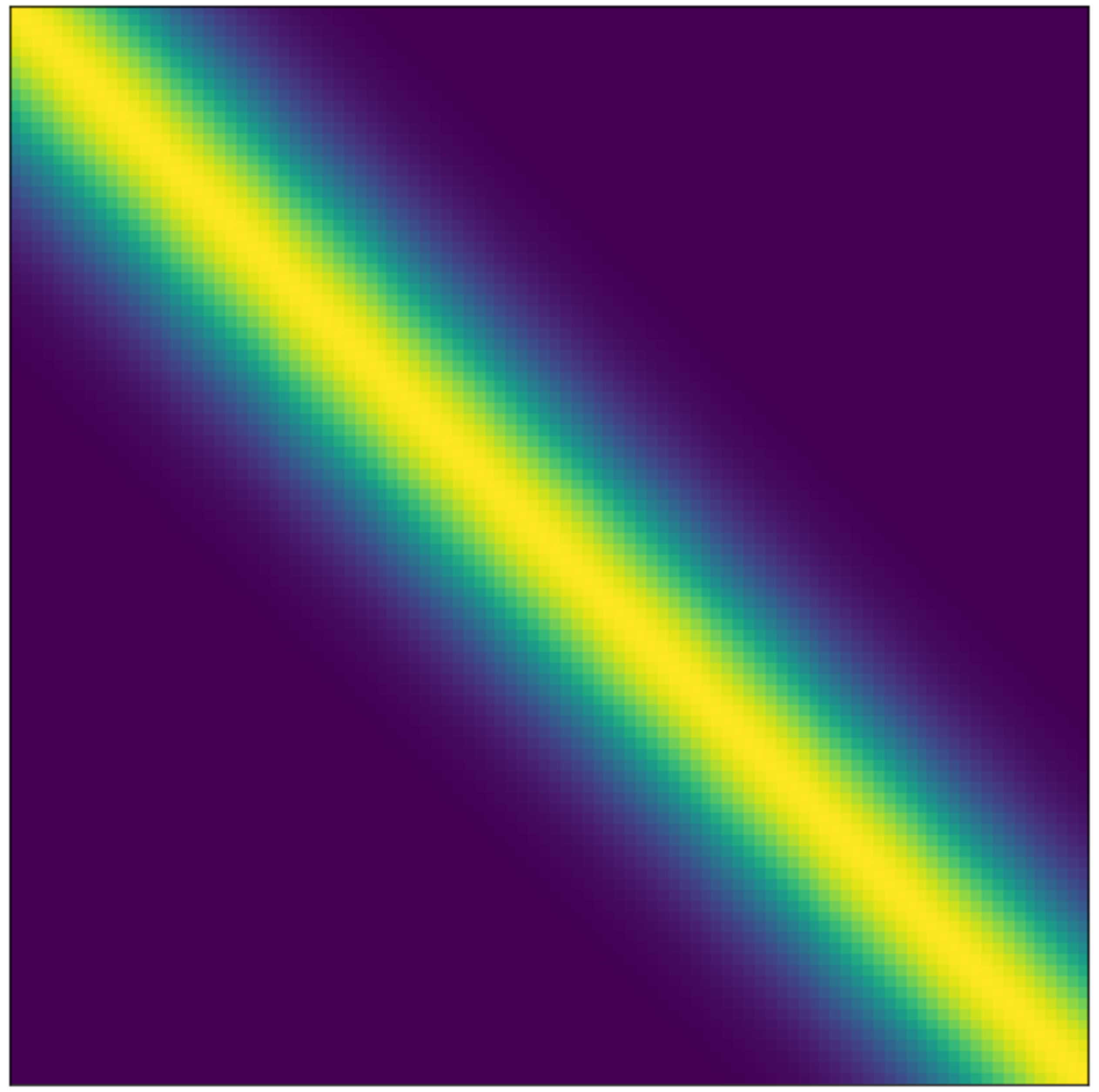
Set up a GP

The Mean function: $m(t) = 0$

The kernal function: $k(t_i, t_j) = \sigma^2 \exp\left(-\frac{1}{2l^2}|t_i - t_j|^2\right)$

$K(t, t) = \{k(t_i, t_j)\}_{i,j=1...n}$

$f(t) \sim \mathcal{GP}(m(t), K(t, t))$



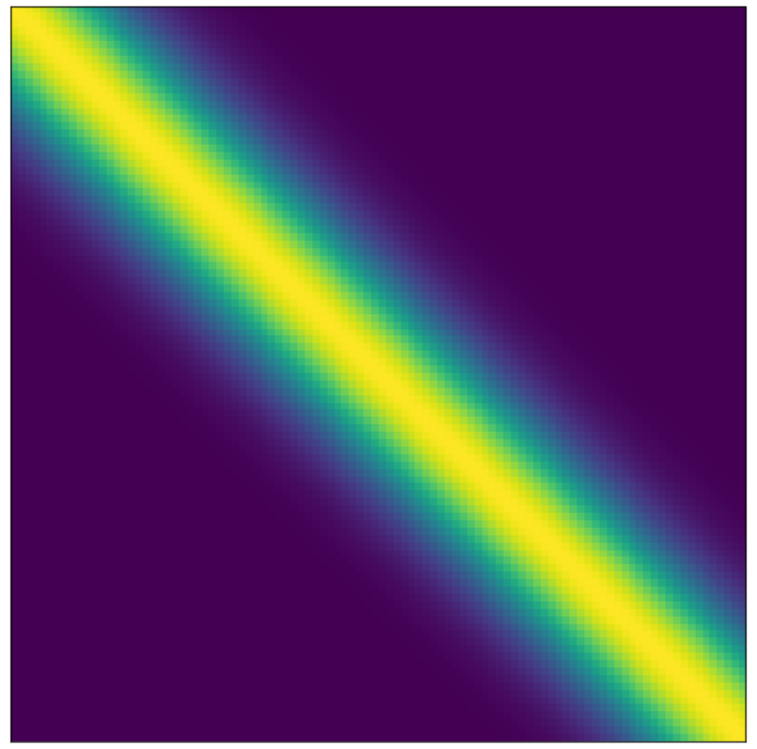
Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w}}$$

KERNEL CHOICE: SE, MORE LATER

Applying Gaussian Processes



Baye's rule

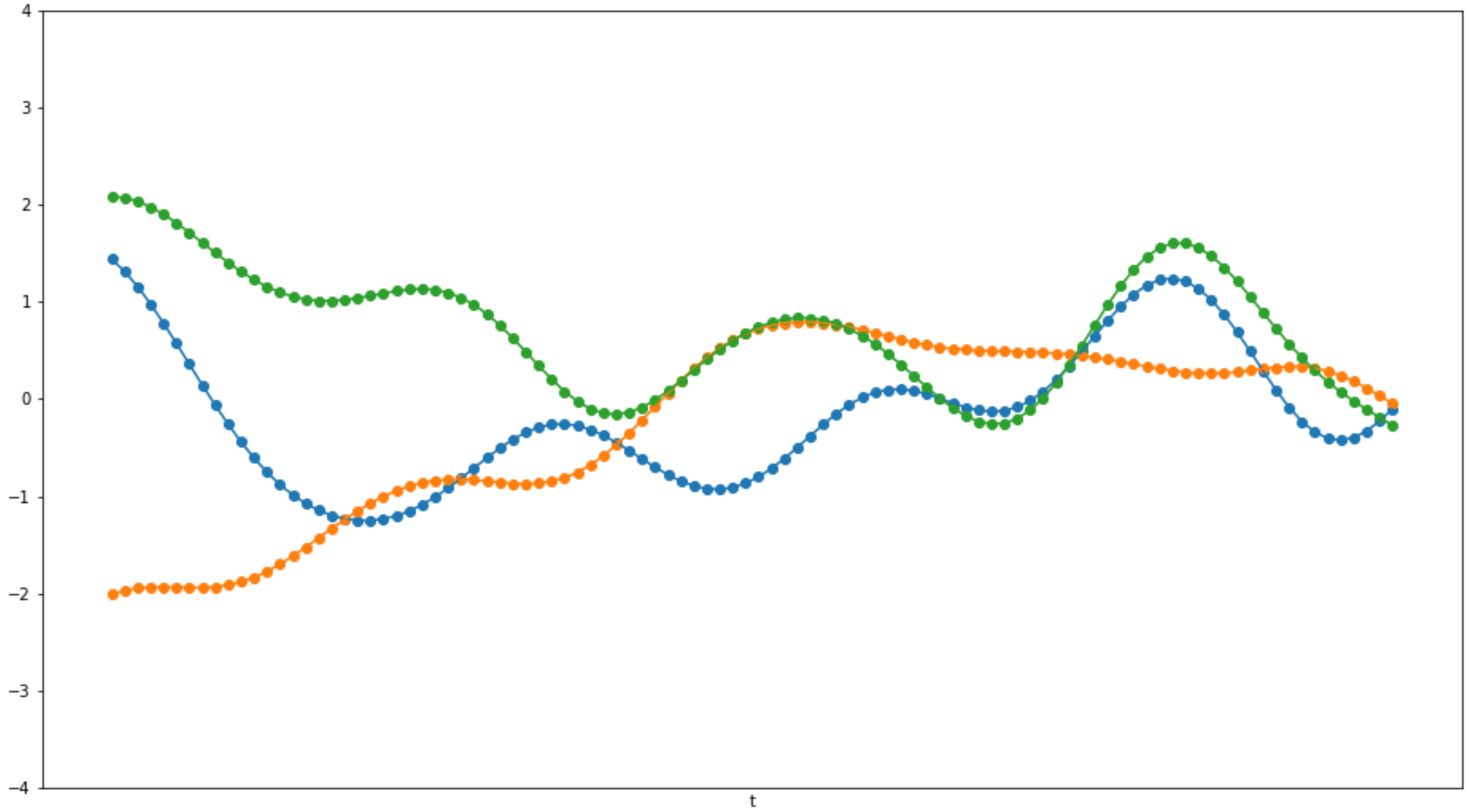
$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})d\mathbf{w}}$$

Kernel choice:

- Functional form of the kernel defines the types of functions you will get.
- Choose your kernel wisely.
- This SE kernel has two parameters.
- This is the point at which you encode your prior knowledge.

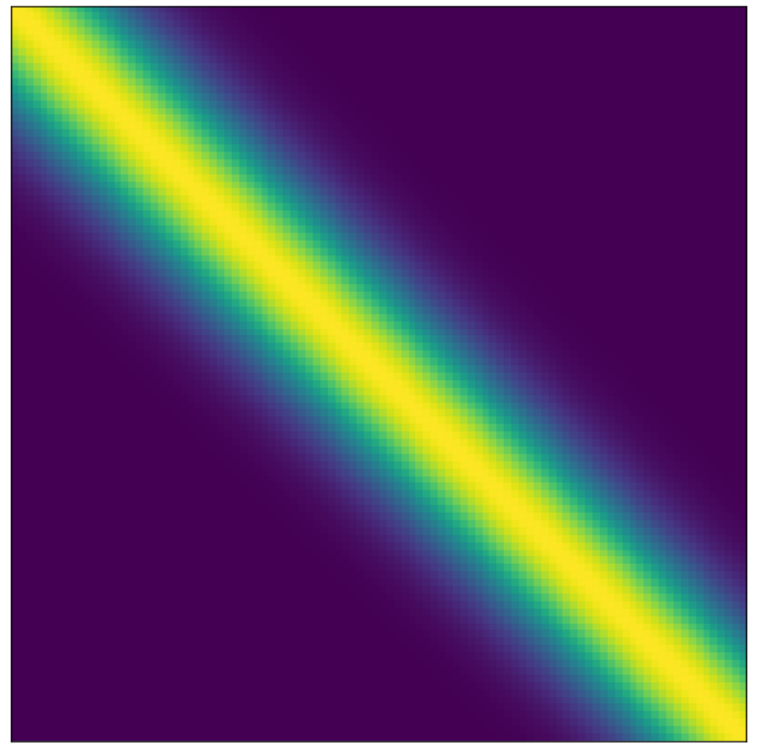
Applying Gaussian Processes



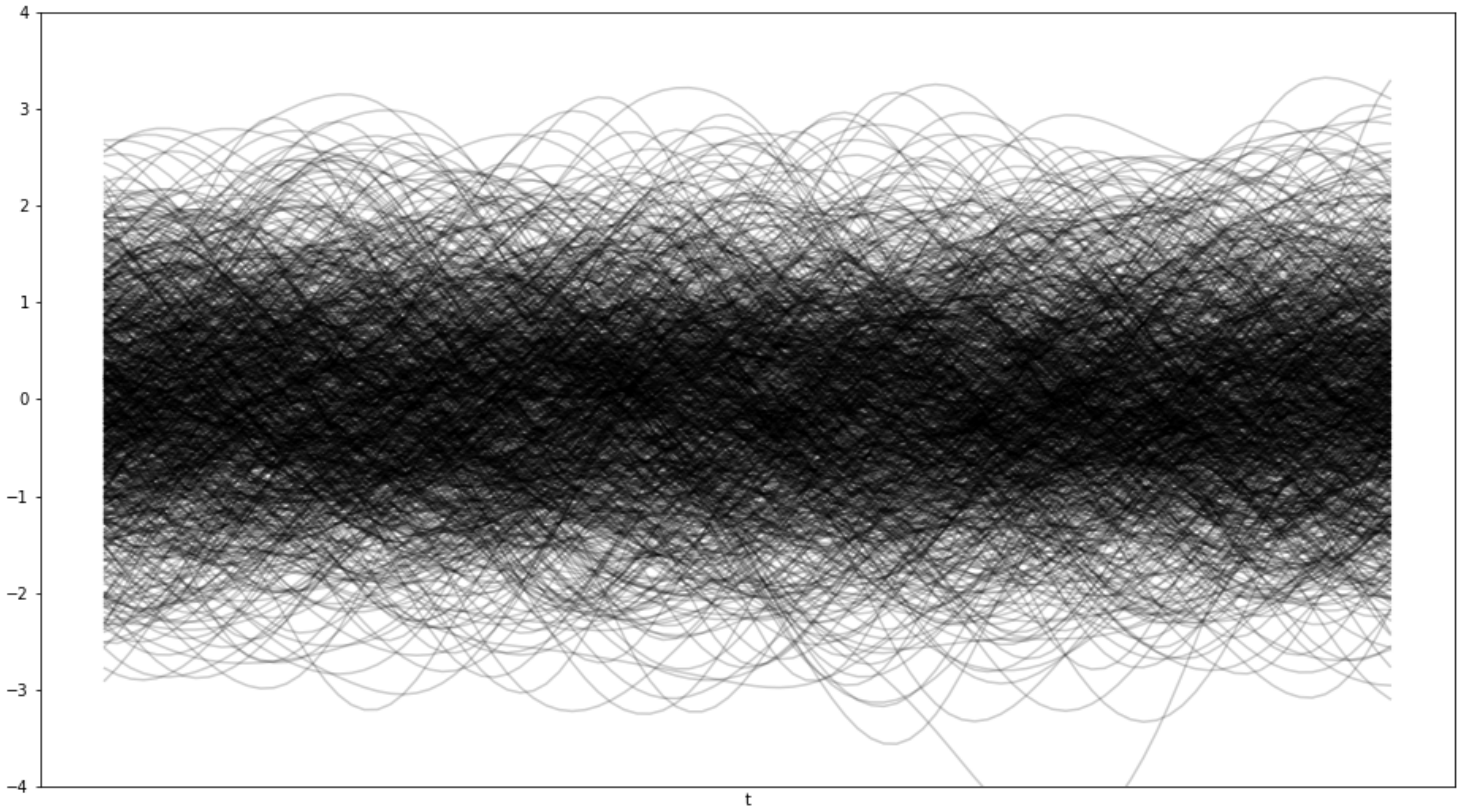
Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w}}$$



Applying Gaussian Processes

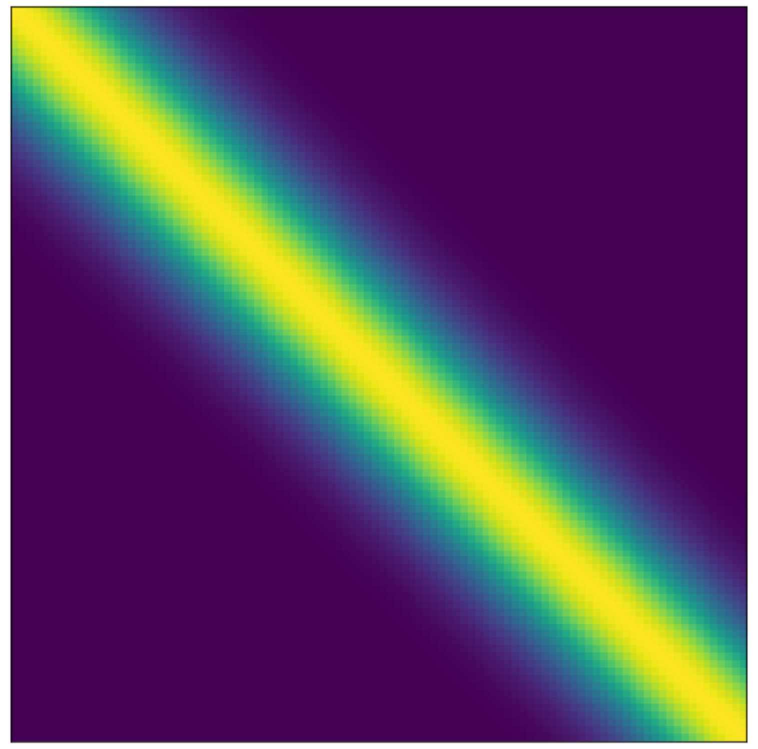


The Prior - For this kernel

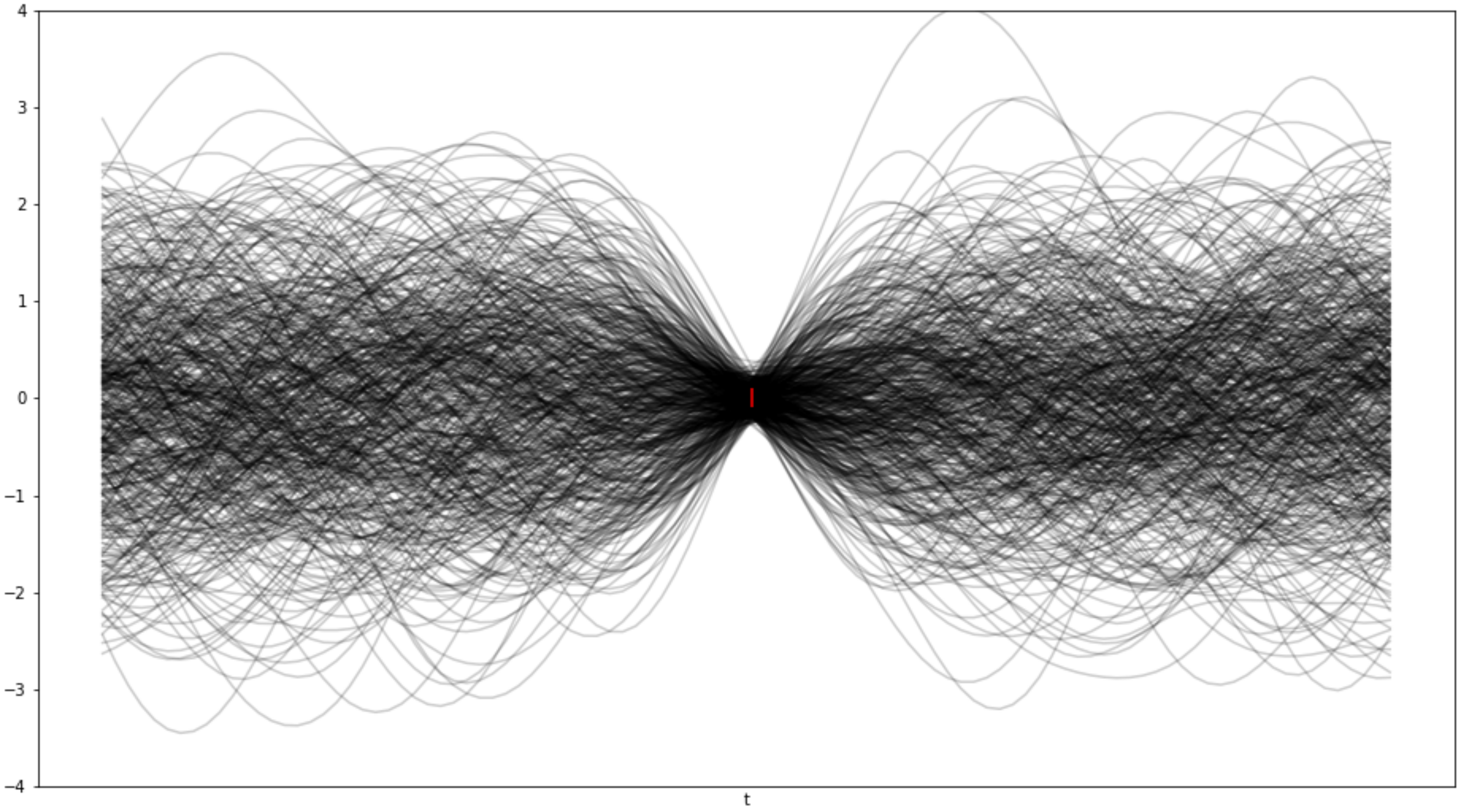
Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})d\mathbf{w}}$$



Applying Gaussian Processes

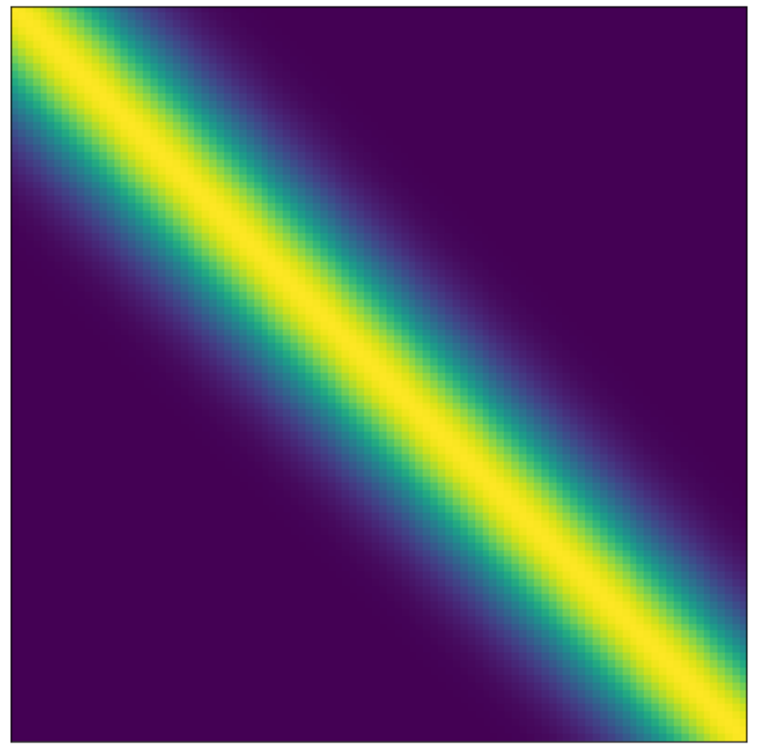


Posterior

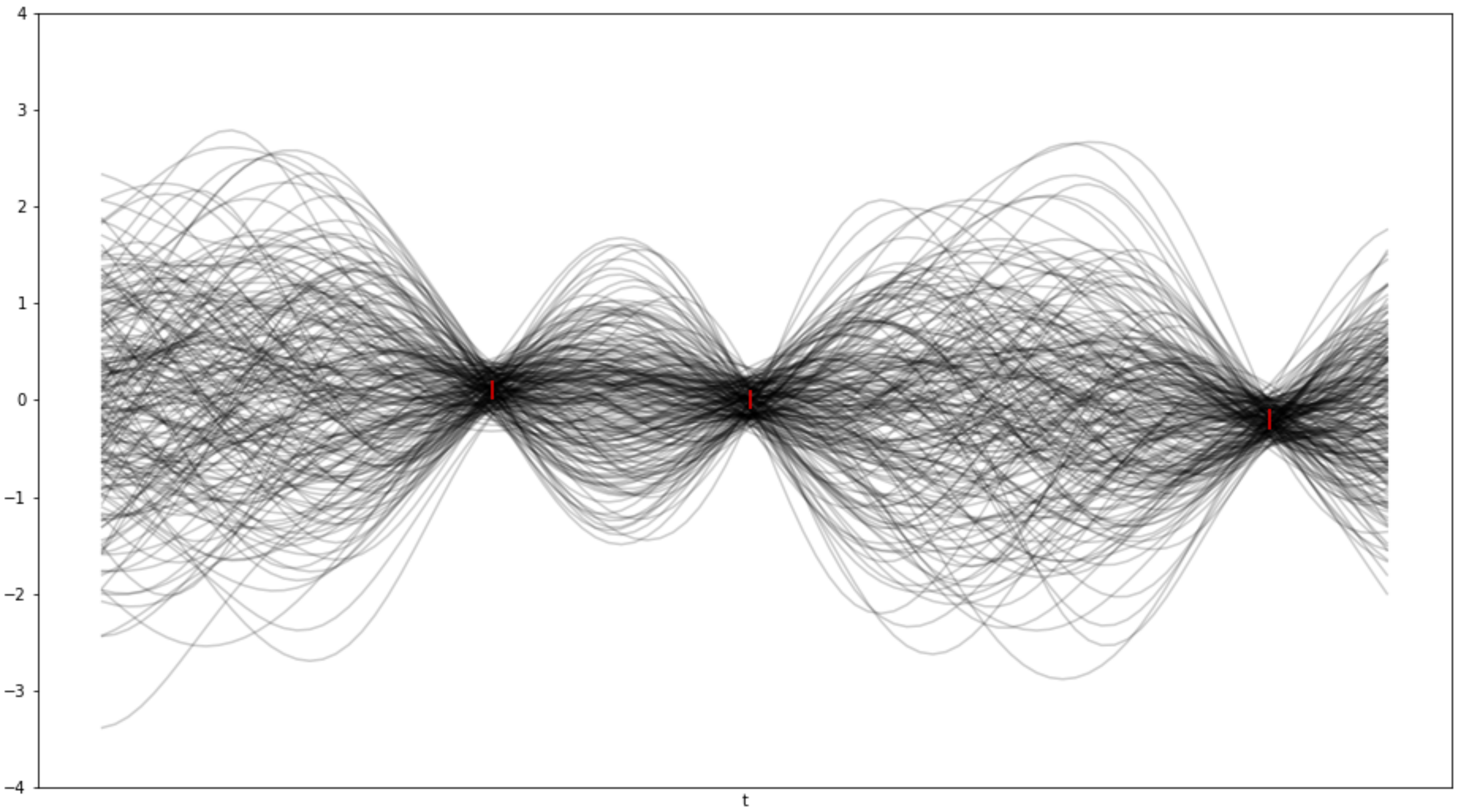
Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w}}$$



Applying Gaussian Processes

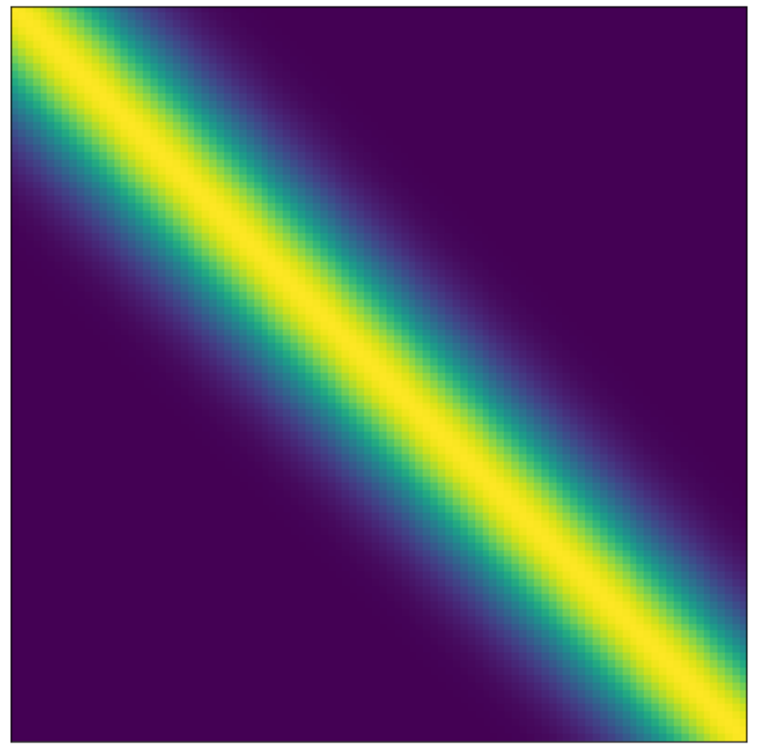


Posterior

Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w}}$$



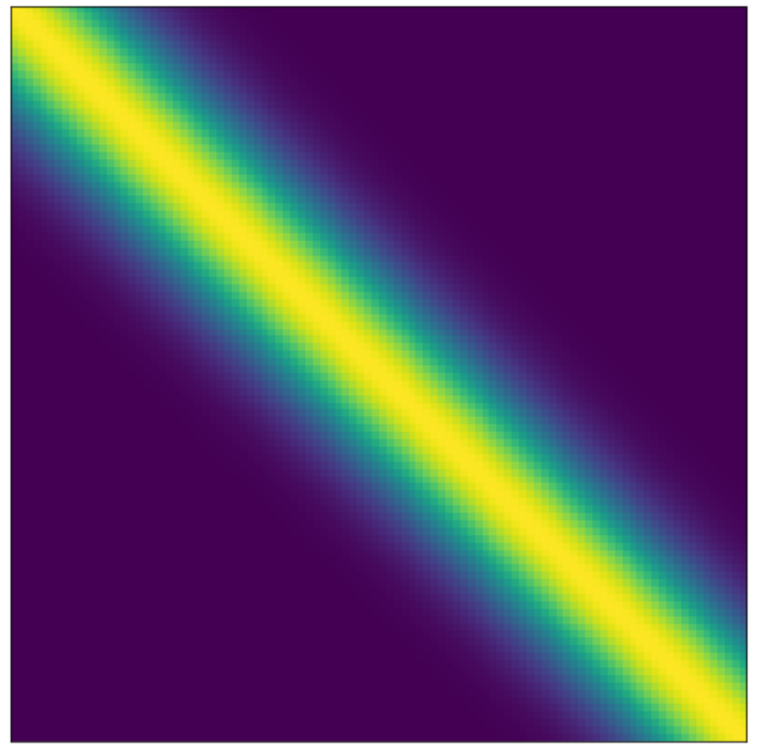
Applying a Multivariate Normal Prior

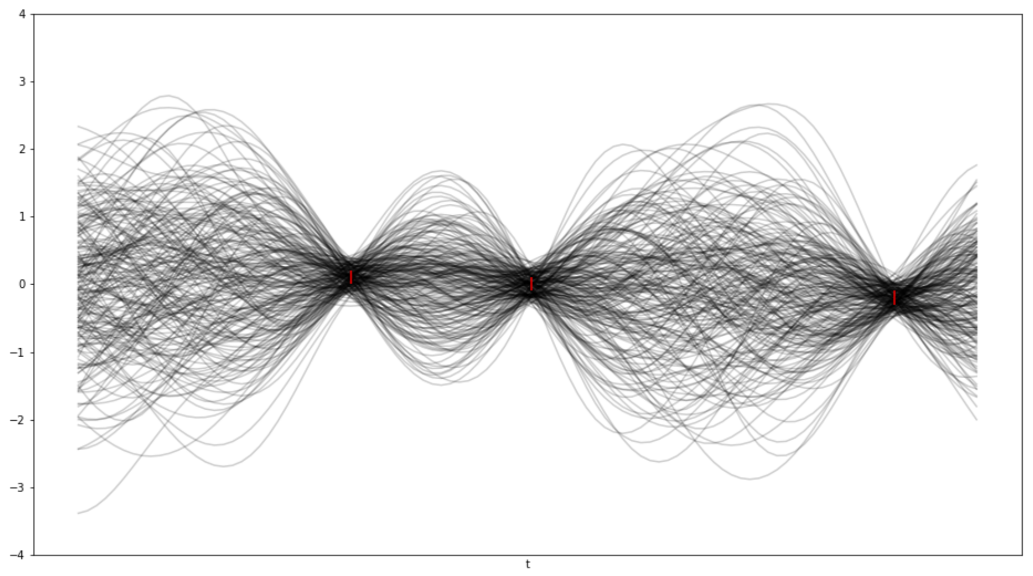
```
n = 100
mu = np.zeros(n)
Sigma = np.zeros([n, n])
l = 10
for i in range(n):
    for j in range(n):
        Sigma[i, j] = np.exp(-0.5 * np.abs(i - j)**2 / l**2)
fig, ax = plt.subplots(figsize=[16,9])

x = np.array([50, 30, 90])
y = np.array([0.0, 0.1, -0.2])
yerr = np.array([0.1, 0.1, 0.1])
for i in range(24000):
    p = np.random.multivariate_normal(mu, Sigma, 1)
    L = 1.0 / (2.0 * np.pi * yerr**2)**0.5 * np.exp(-0.5 * (p[0][x] - y)**2 / yerr**2)
    if np.random.rand() < np.prod(L):
        ax.plot(np.arange(len(p[0])) + 1, [n for n in p[0]], 'k-', alpha=0.2)
ax.errorbar(x+1, y, yerr=yerr, c='r', zorder=99, linestyle='none')
ax.set_xticks([])
ax.set_ylim([-4, 4])
ax.set_xlabel('t')
fig.savefig('posterior2_SE.png')
```

Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

$$p(\mathbf{w}|\mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})d\mathbf{w}}$$





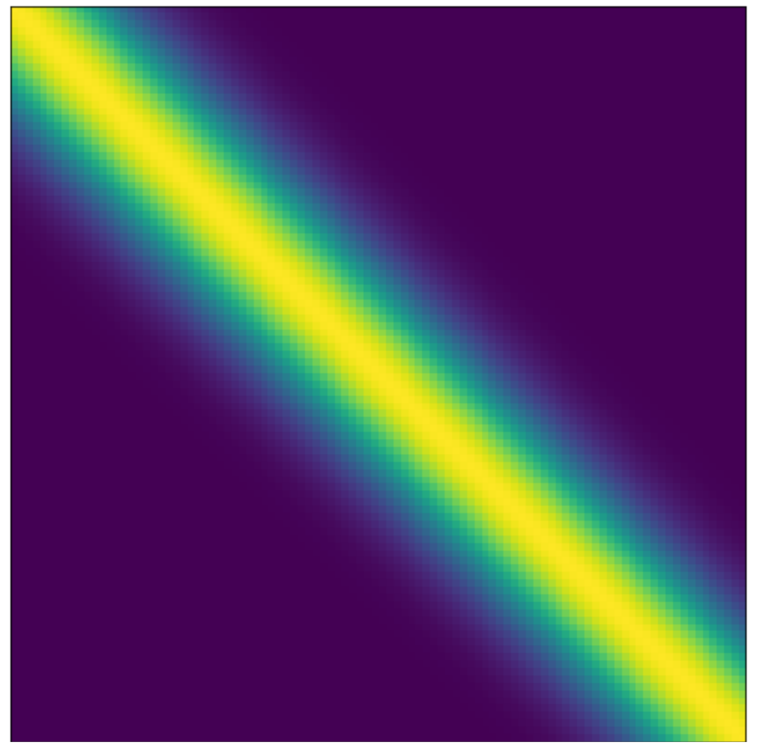
Applying Gaussian Processes

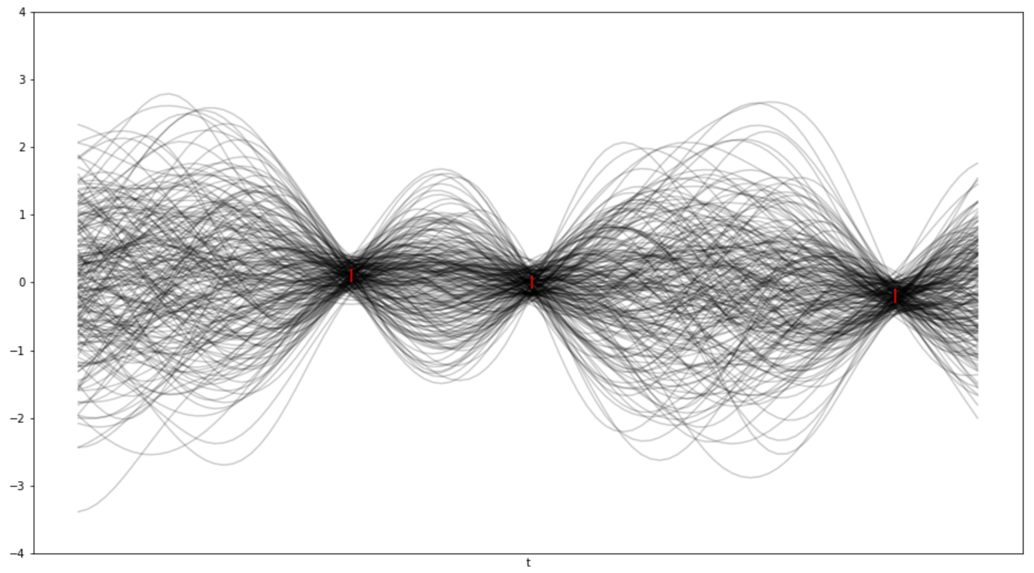
Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

Here comes the magic:

What if there was a way to evaluate all the functions for a given prior!






```
n = 100
mu = np.zeros(n)
Sigma = np.zeros([n, n])
l = 10
for i in range(n):
    for j in range(n):
        Sigma[i, j] = np.exp(-0.5 * np.abs(i - j)**2 / l**2)
fig, ax = plt.subplots(figsize=[16,9])

x = np.array([50, 30, 90])
y = np.array([0.0, 0.1, -0.2])
yerr = np.array([0.1, 0.1, 0.1])
for i in range(24000):
    p = np.random.multivariate_normal(mu, Sigma, 1)
    L = 1.0 / (2.0 * np.pi * yerr**2)**0.5 * np.exp(-0.5 * (p[0][x] - y)**2 / yerr**2)
    if np.random.rand() < np.prod(L):
        ax.plot(np.arange(len(p[0])) + 1, [n for n in p[0]], 'k-', alpha=0.2)
ax.errorbar(x+1, y, yerr=yerr, c='r', zorder=99, linestyle='none')
ax.set_xticks([])
ax.set_ylim([-4, 4])
ax.set_xlabel('t')
fig.savefig('posterior2_SE.png')
```

Applying Gaussian Processes

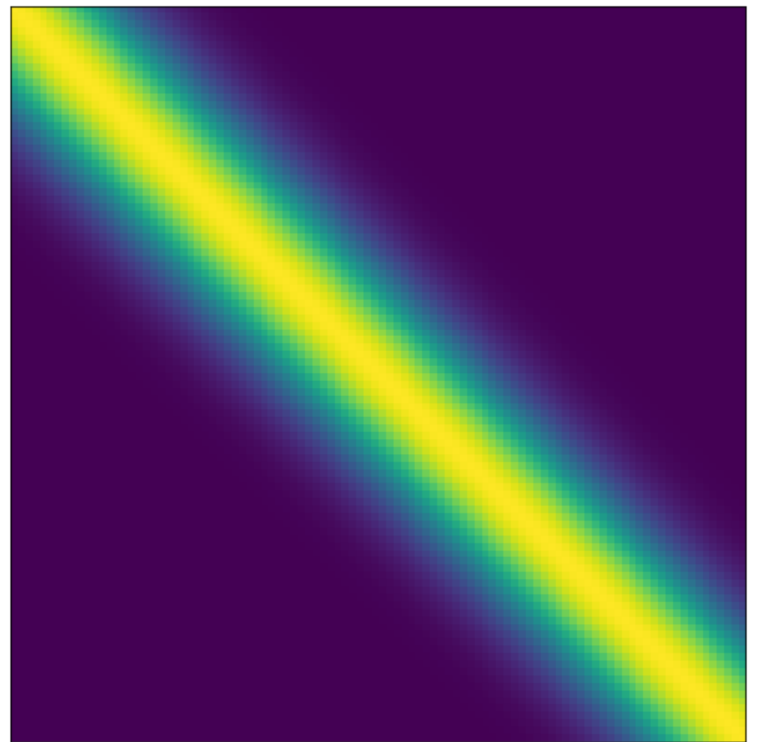
Here comes the magic:

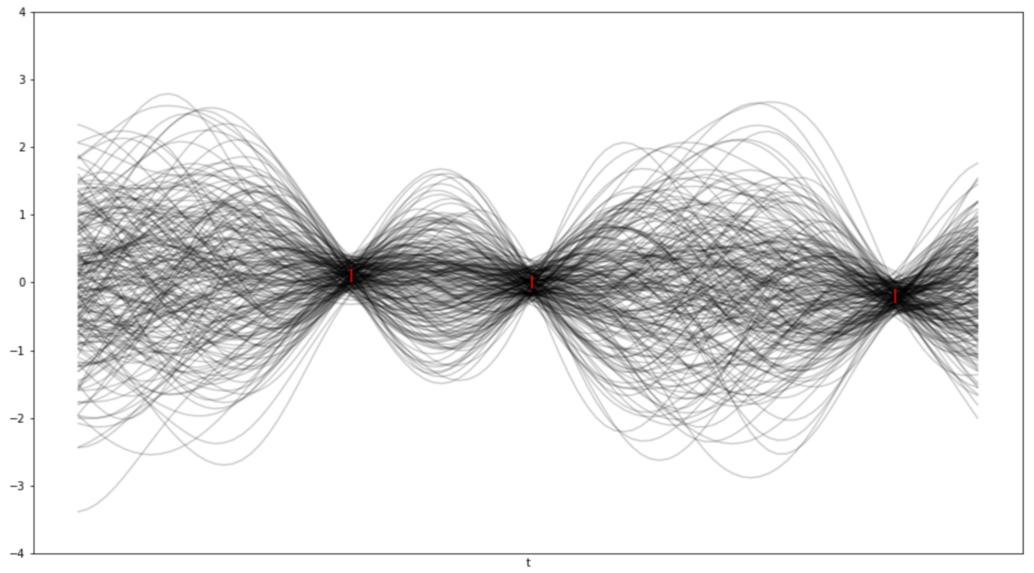
What if there was a way to evaluate all the functions for a given prior!

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{f}, X)\, p(\mathbf{f}|X)\, d\mathbf{f}$$

Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$






```
n = 100
mu = np.zeros(n)
Sigma = np.zeros([n, n])
l = 10
for i in range(n):
    for j in range(n):
        Sigma[i, j] = np.exp(-0.5 * np.abs(i - j)**2 / l**2)
fig, ax = plt.subplots(figsize=[16,9])

x = np.array([50, 30, 90])
y = np.array([0.0, 0.1, -0.2])
yerr = np.array([0.1, 0.1, 0.1])
for i in range(24000):
    p = np.random.multivariate_normal(mu, Sigma, 1)
    L = 1.0 / (2.0 * np.pi * yerr**2)**0.5 * np.exp(-0.5 * (p[0][x] - y)**2 / yerr**2)
    if np.random.rand() < np.prod(L):
        ax.plot(np.arange(len(p[0])) + 1, [n for n in p[0]], 'k-', alpha=0.2)
ax.errorbar(x+1, y, yerr=yerr, c='r', zorder=99, linestyle='none')
ax.set_xticks([])
ax.set_ylim([-4, 4])
ax.set_xlabel('t')
fig.savefig('posterior2_SE.png')
```

Formal justification

Consider normal regression problems (non-linear)

$$y(\mathbf{t}) = f(\mathbf{t}) + \epsilon\sigma_y$$

$$p(\epsilon) = \mathcal{N}(0, 1)$$

Here we place a prior over the non-linear function:

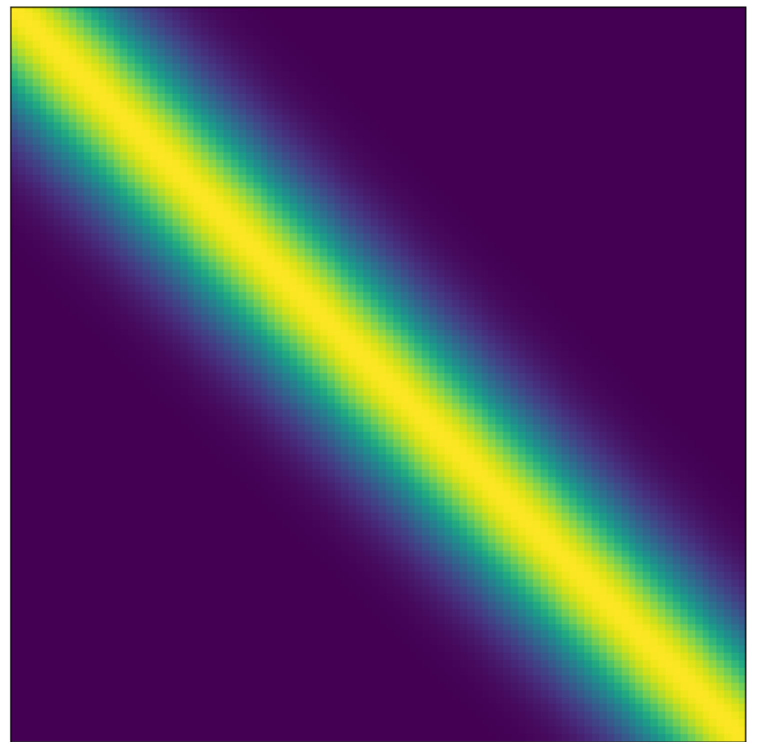
$$p(f(\mathbf{t})|\theta) = \mathcal{GP}(0, K(\mathbf{t}, \mathbf{t}'))$$

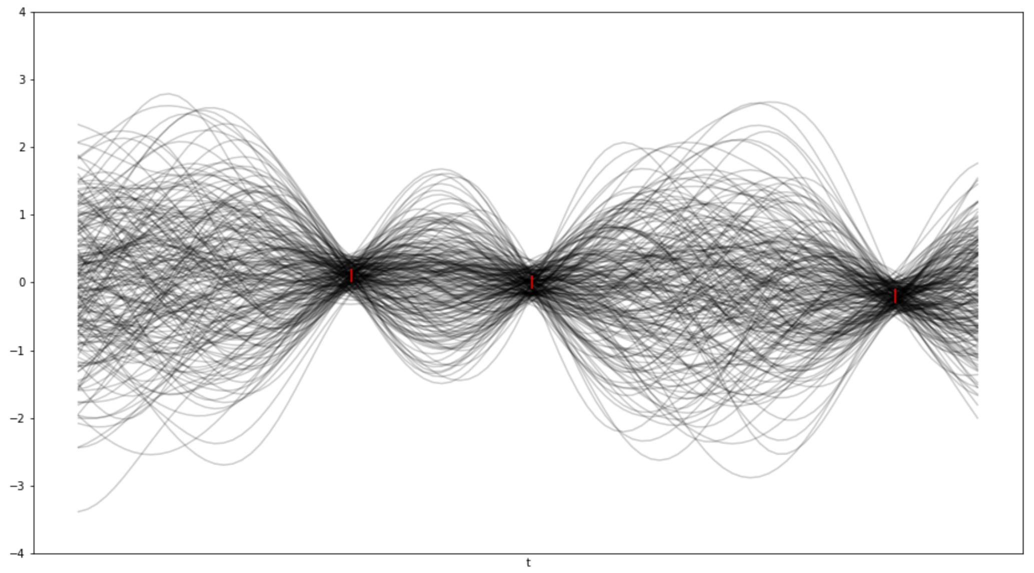
Now, the sum of Gaussian variables is a Gaussian. This results in a GP over $y(t)$.

$$p(y(\mathbf{t})|\theta) = \mathcal{GP}(0, K(\mathbf{t}, \mathbf{t}') + \mathbf{I}\sigma_y^2)$$

Baye's rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$






```python
n = 100
mu = np.zeros(n)
Sigma = np.zeros([n, n])
l = 10
for i in range(n):
    for j in range(n):
        Sigma[i, j] = np.exp(-0.5 * np.abs(i - j)**2 / l**2)
fig, ax = plt.subplots(figsize=[16,9])

x = np.array([50, 30, 90])
y = np.array([0.0, 0.1, -0.2])
yerr = np.array([0.1, 0.1, 0.1])
for i in range(24000):
    p = np.random.multivariate_normal(mu, Sigma, 1)
    L = 1.0 / (2.0 * np.pi * yerr**2)**0.5 * np.exp(-0.5 * (p[0][x] - y)**2 / yerr**2)
    if np.random.rand() < np.prod(L):
        ax.plot(np.arange(len(p[0])) + 1, [n for n in p[0]], 'k-', alpha=0.2)
ax.errorbar(x+1, y, yerr=yerr, c='r', zorder=99, linestyle='none')
ax.set_xticks([])
ax.set_ylim([-4, 4])
ax.set_xlabel('t')
fig.savefig('posterior2_SE.png')
```

Applying Gaussian Processes - The Likelihood Functions

$$p(\mathbf{y_1}, \mathbf{y_2}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

Applying Gaussian Processes - The Likelihood Functions

$$p(\mathbf{y_1}, \mathbf{y_2}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{c} \end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}\right)$$

$$A = \begin{bmatrix} \mathrm{var}(y_{1,0}) & \mathrm{cov}(y_{1,0}, y_{1,1}) & \dots & \mathrm{cov}(y_{1,0}, y_{1,n}) \\ \mathrm{cov}(y_{1,1}, y_{1,0}) & \mathrm{var}(y_{1,1}) & \dots & \mathrm{cov}(y_{1,1}, y_{1,n}) \\ \dots & \dots & \dots & \dots \\ \mathrm{cov}(y_{1,n}, y_{1,0}) & \dots & \dots & \mathrm{var}(y_{1,n}) \end{bmatrix}$$

Applying Gaussian Processes - The Likelihood Functions

$$p(\mathbf{y_1}, \mathbf{y_2}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C\end{bmatrix}\right)$$

$$A = \begin{bmatrix} \mathrm{var}(y_{1,0}) & \mathrm{cov}(y_{1,0}, y_{1,1}) & \dots & \mathrm{cov}(y_{1,0}, y_{1,n}) \\ \mathrm{cov}(y_{1,1}, y_{1,0}) & \mathrm{var}(y_{1,1}) & \dots & \mathrm{cov}(y_{1,1}, y_{1,n}) \\ \dots & \dots & \dots & \dots \\ \mathrm{cov}(y_{1,n}, y_{1,0}) & \dots & \dots & \mathrm{var}(y_{1,n}) \end{bmatrix}$$

$$C = \begin{bmatrix} \mathrm{var}(y_{2,0}) & \mathrm{cov}(y_{2,0}, y_{2,1}) & \dots & \mathrm{cov}(y_{2,0}, y_{2,m}) \\ \mathrm{cov}(y_{2,1}, y_{2,0}) & \mathrm{var}(y_{2,1}) & \dots & \mathrm{cov}(y_{2,1}, y_{2,m}) \\ \dots & \dots & \dots & \dots \\ \mathrm{cov}(y_{2,m}, y_{2,0}) & \dots & \dots & \mathrm{var}(y_{2,m}) \end{bmatrix}$$

Applying Gaussian Processes - The Likelihood Functions

$$p(\mathbf{y_1}, \mathbf{y_2}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C\end{bmatrix}\right)$$

$$A = \begin{bmatrix} \mathrm{var}(y_{1,0}) & \mathrm{cov}(y_{1,0}, y_{1,1}) & \dots & \mathrm{cov}(y_{1,0}, y_{1,n}) \\ \mathrm{cov}(y_{1,1}, y_{1,0}) & \mathrm{var}(y_{1,1}) & \dots & \mathrm{cov}(y_{1,1}, y_{1,n}) \\ \dots & \dots & \dots & \dots \\ \mathrm{cov}(y_{1,n}, y_{1,0}) & \dots & \dots & \mathrm{var}(y_{1,n}) \end{bmatrix}$$

$$B = \begin{bmatrix} \mathrm{cov}(y_{1,0}, y_{2,0}) & \mathrm{cov}(y_{1,0}, y_{2,1}) & \dots & \mathrm{cov}(y_{1,0}, y_{2,n}) \\ \mathrm{cov}(y_{1,1}, y_{2,0}) & \mathrm{cov}(y_{1,1}, , y_{2,1}) & \dots & \mathrm{cov}(y_{1,1}, y_{2,n}) \\ \dots & \dots & \dots & \dots \\ \mathrm{cov}(y_{1,m}, y_{2,0}) & \dots & \dots & \mathrm{cov}(y_{1,m}, y_{2,n}) \end{bmatrix}$$

$$C = \begin{bmatrix} \mathrm{var}(y_{2,0}) & \mathrm{cov}(y_{2,0}, y_{2,1}) & \dots & \mathrm{cov}(y_{2,0}, y_{2,m}) \\ \mathrm{cov}(y_{2,1}, y_{2,0}) & \mathrm{var}(y_{2,1}) & \dots & \mathrm{cov}(y_{2,1}, y_{2,m}) \\ \dots & \dots & \dots & \dots \\ \mathrm{cov}(y_{2,m}, y_{2,0}) & \dots & \dots & \mathrm{var}(y_{2,m}) \end{bmatrix}$$

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

or

$$\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

$$p(\mathbf{y}_1) = \int p(\mathbf{y}_2, \mathbf{y}_1)\, \mathrm{d}\mathbf{y}_2$$

so

Applying Gaussian Processes - The Likelihood Functions

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

or

$$\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

$$p(\mathbf{y}_1) = \int p(\mathbf{y}_2, \mathbf{y}_1)\, \mathrm{d}\mathbf{y}_2$$

so

$$p(\mathbf{y}_1) = \mathcal{N}(\mathbf{a}, A).$$

Applying Gaussian Processes - Making Predictions

$$p(\mathbf{y}_2|\mathbf{y}_1) = \frac{p(\mathbf{y}_1,\mathbf{y}_2)}{p(\mathbf{y}_1)}$$

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

$$p(\mathbf{y}_2|\mathbf{y}_1) = \mathcal{N}(\mathbf{c} + BA^{-1}(\mathbf{y}_1 - \mathbf{a}), C - BA^{-1}B^T)$$

$$p(\mathbf{y}_2|\mathbf{y}_1) = \mathcal{N}(BA^{-1}(\mathbf{y}_1), C - BA^{-1}B^T)$$

Applying Gaussian Processes - Making Predictions

$$p(\mathbf{y}_2|\mathbf{y}_1) = \frac{p(\mathbf{y}_1,\mathbf{y}_2)}{p(\mathbf{y}_1)}$$

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a} \\ \mathbf{c}\end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}\right)$$

$$p(\mathbf{y}_2|\mathbf{y}_1) = \mathcal{N}(\mathbf{c} + BA^{-1}(\mathbf{y}_1 - \mathbf{a}), C - BA^{-1}B^T)$$

$$p(\mathbf{y}_2|\mathbf{y}_1) = \mathcal{N}(BA^{-1}(\mathbf{y}_1), C - BA^{-1}B^T)$$

A linear operator on y1

Applying Gaussian Processes - Making Predictions

$$p(\mathbf{y}_2|\mathbf{y}_1) = \frac{p(\mathbf{y}_1,\mathbf{y}_2)}{p(\mathbf{y}_1)}$$

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

$$p(\mathbf{y}_2|\mathbf{y}_1) = \mathcal{N}(\mathbf{c} + BA^{-1}(\mathbf{y}_1 - \mathbf{a}), C - BA^{-1}B^T)$$

$$p(\mathbf{y}_2|\mathbf{y}_1) = \mathcal{N}(BA^{-1}(\mathbf{y}_1), C - BA^{-1}B^T)$$

A linear operator on y1

Your Prior

Constraint from new information

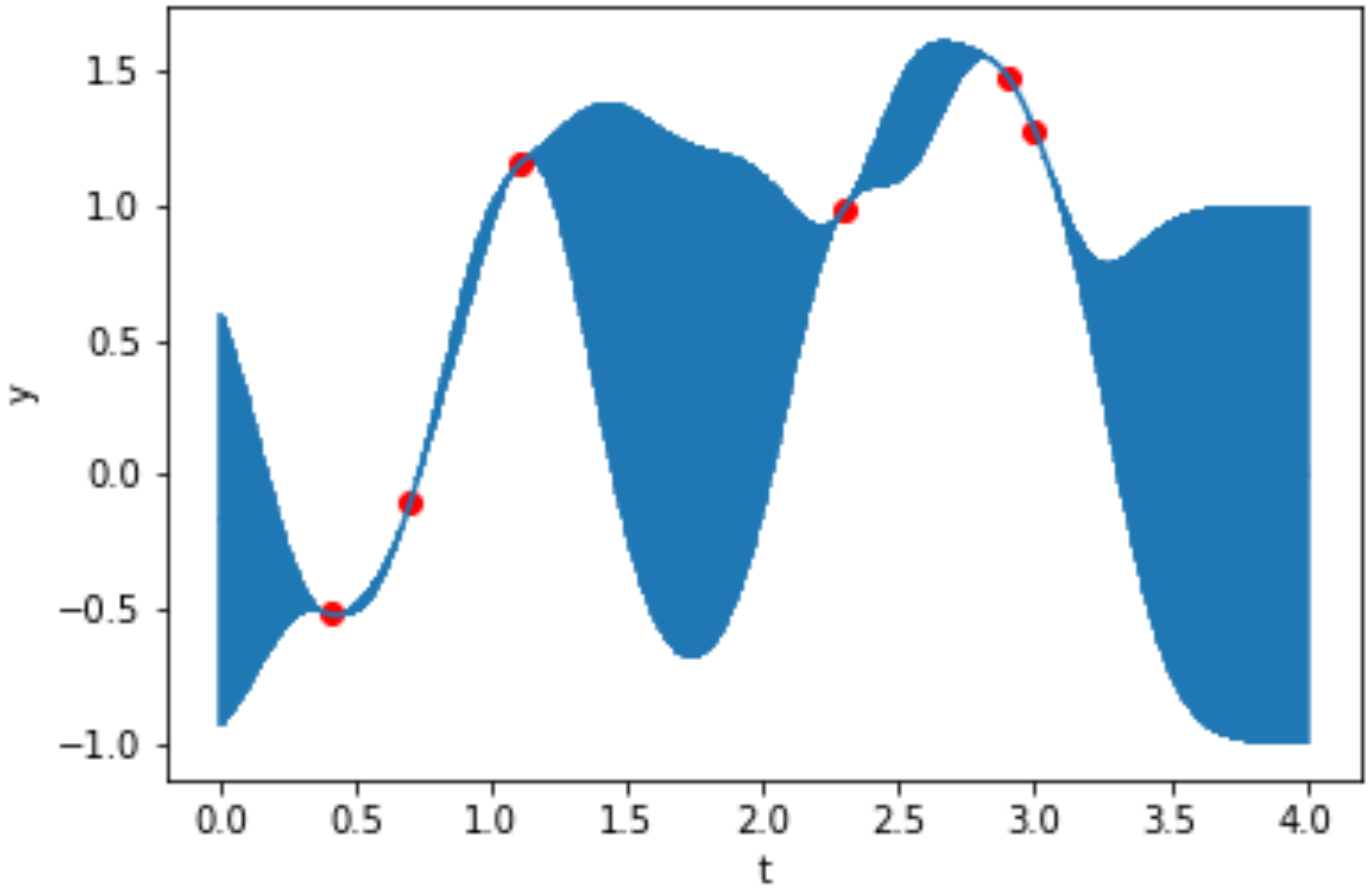
Applying Gaussian Processes - Making Predictions

$$p(\mathbf{y}_2|\mathbf{y}_1) = \frac{p(\mathbf{y}_1,\mathbf{y}_2)}{p(\mathbf{y}_1)}$$

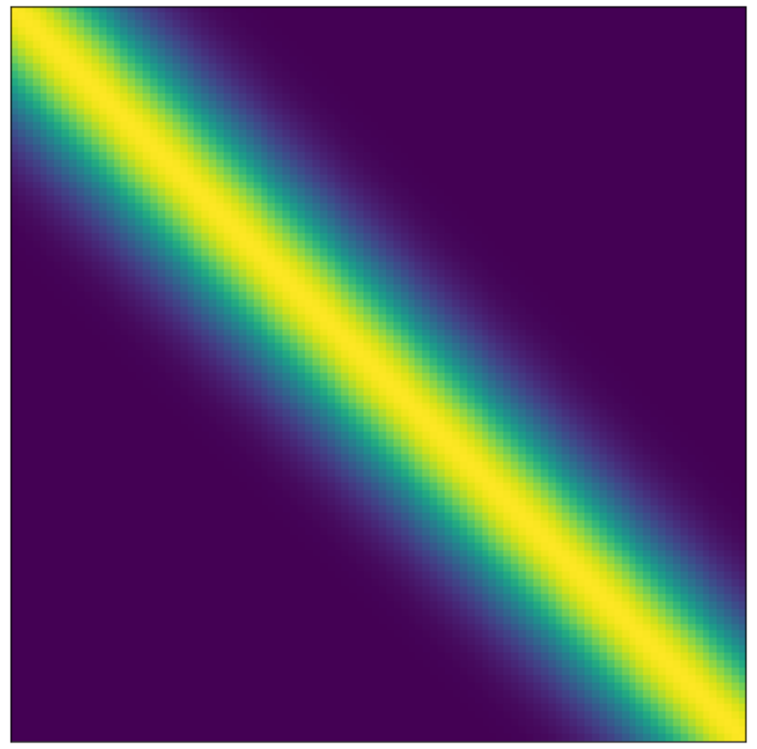
$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

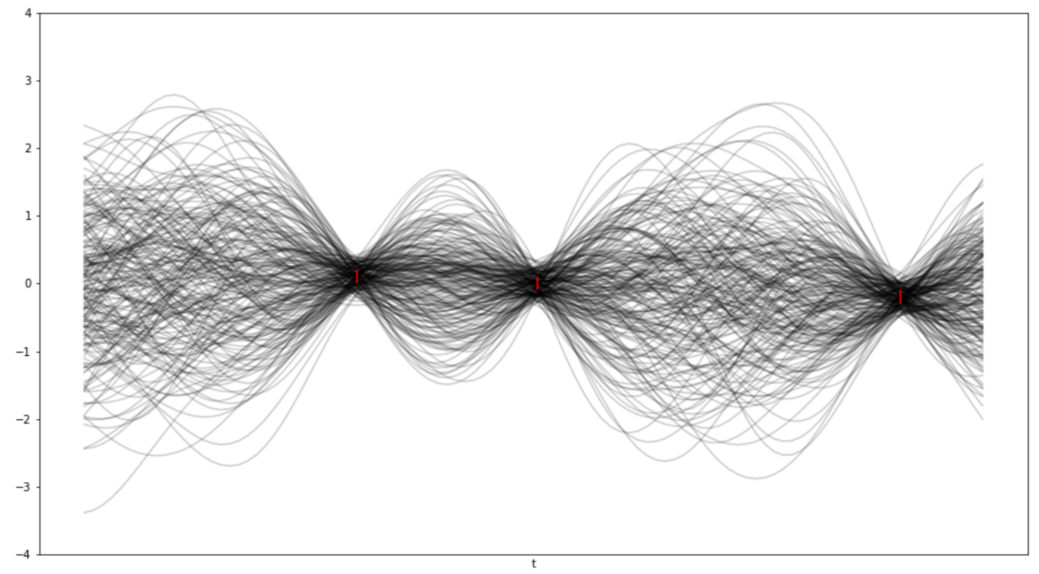
$$p(\mathbf{y}_2|\mathbf{y}_1) = \mathcal{N}(\mathbf{c} + BA^{-1}(\mathbf{y}_1 - \mathbf{a}), C - BA^{-1}B^T)$$

$$p(\mathbf{y}_2|\mathbf{y}_1) = \mathcal{N}(BA^{-1}(\mathbf{y}_1), C - BA^{-1}B^T)$$

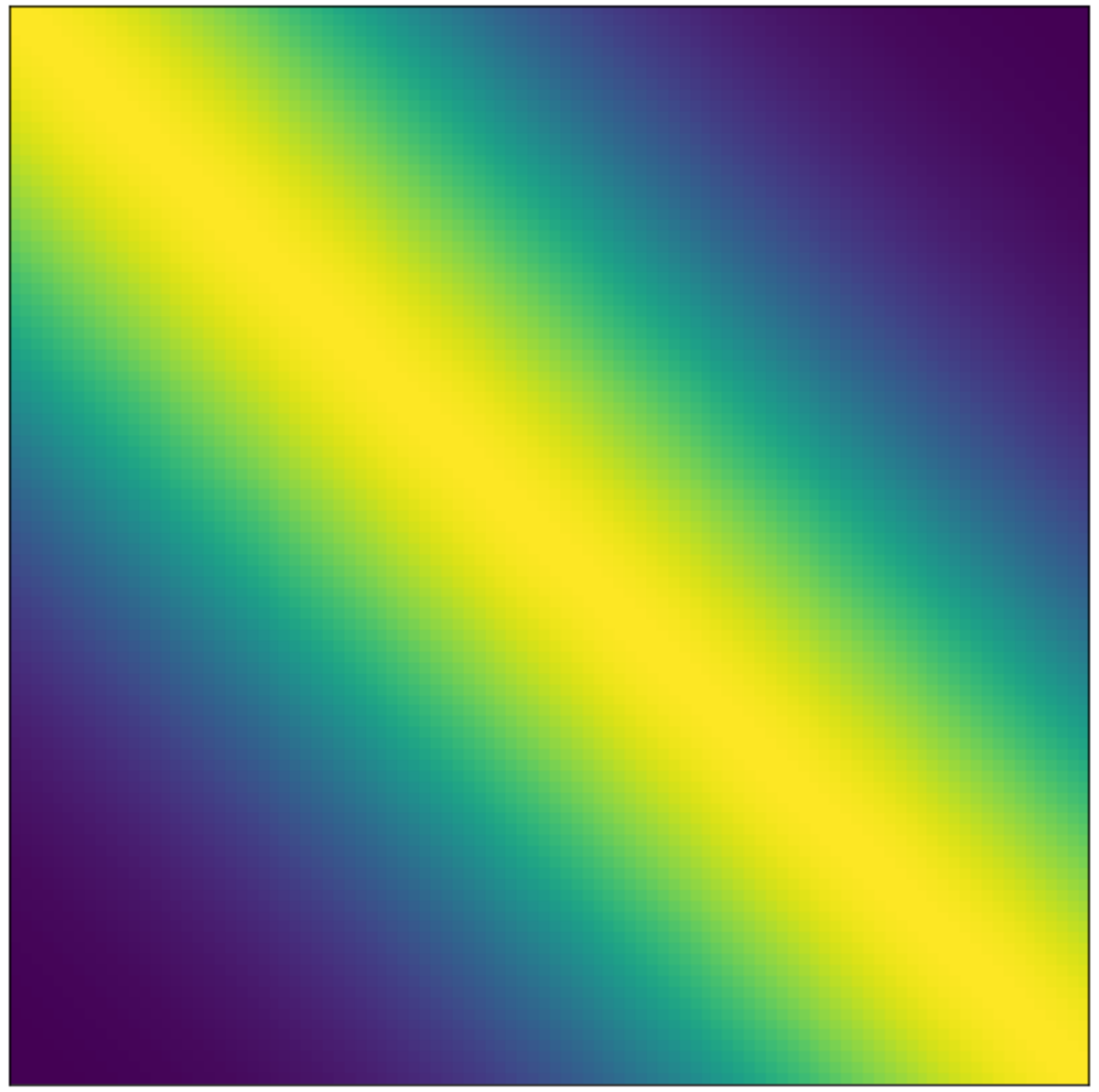


Applying Gaussian Processes

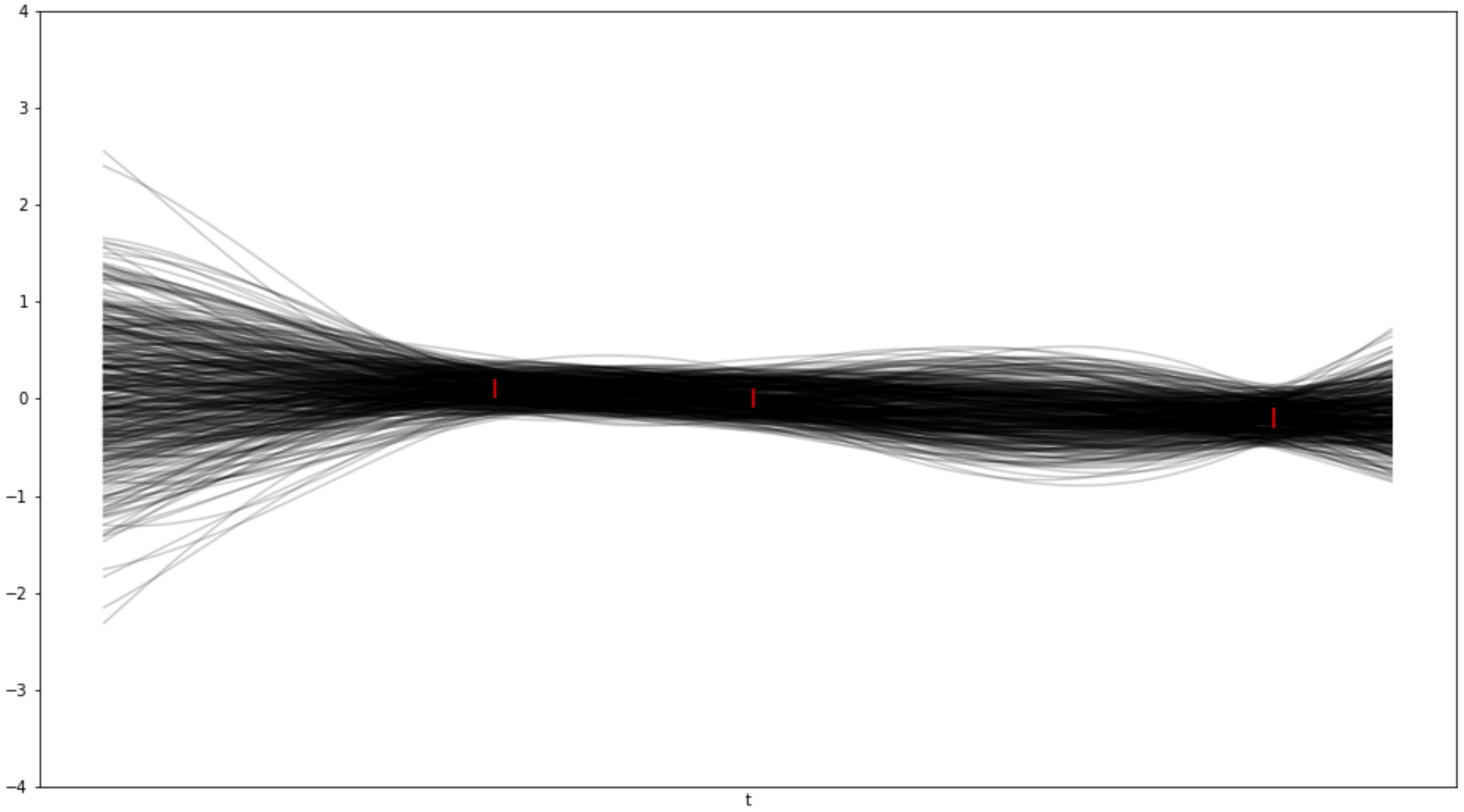




Kernel Choice: What if I triple the value of l?

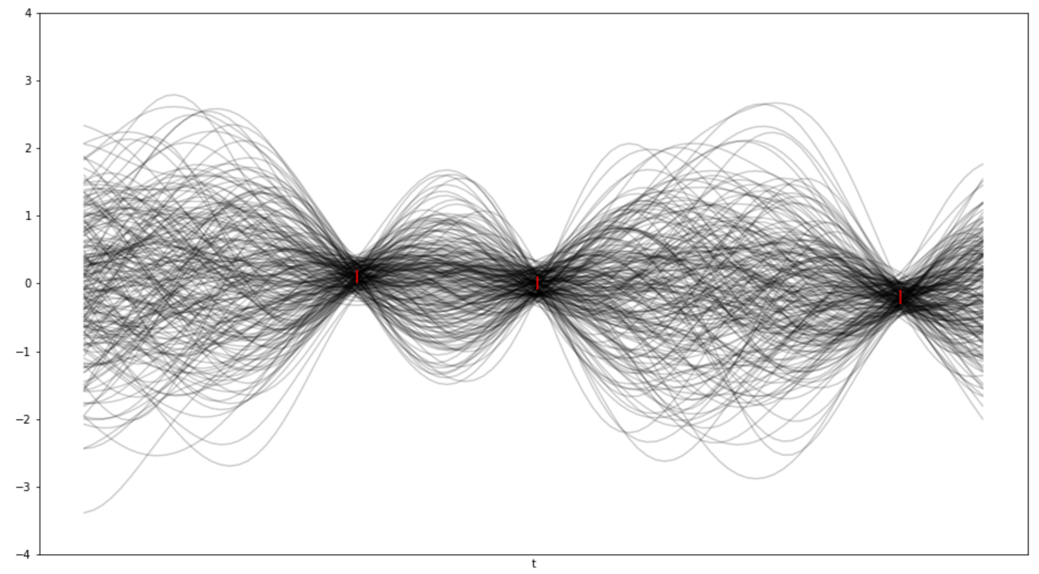


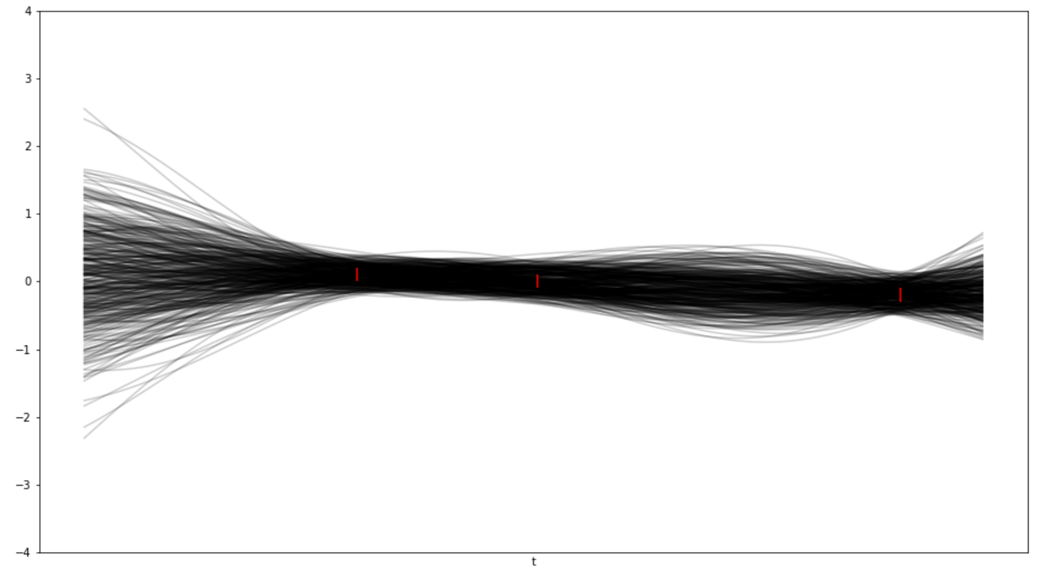
Prior



Posterior

Applying Gaussian Processes



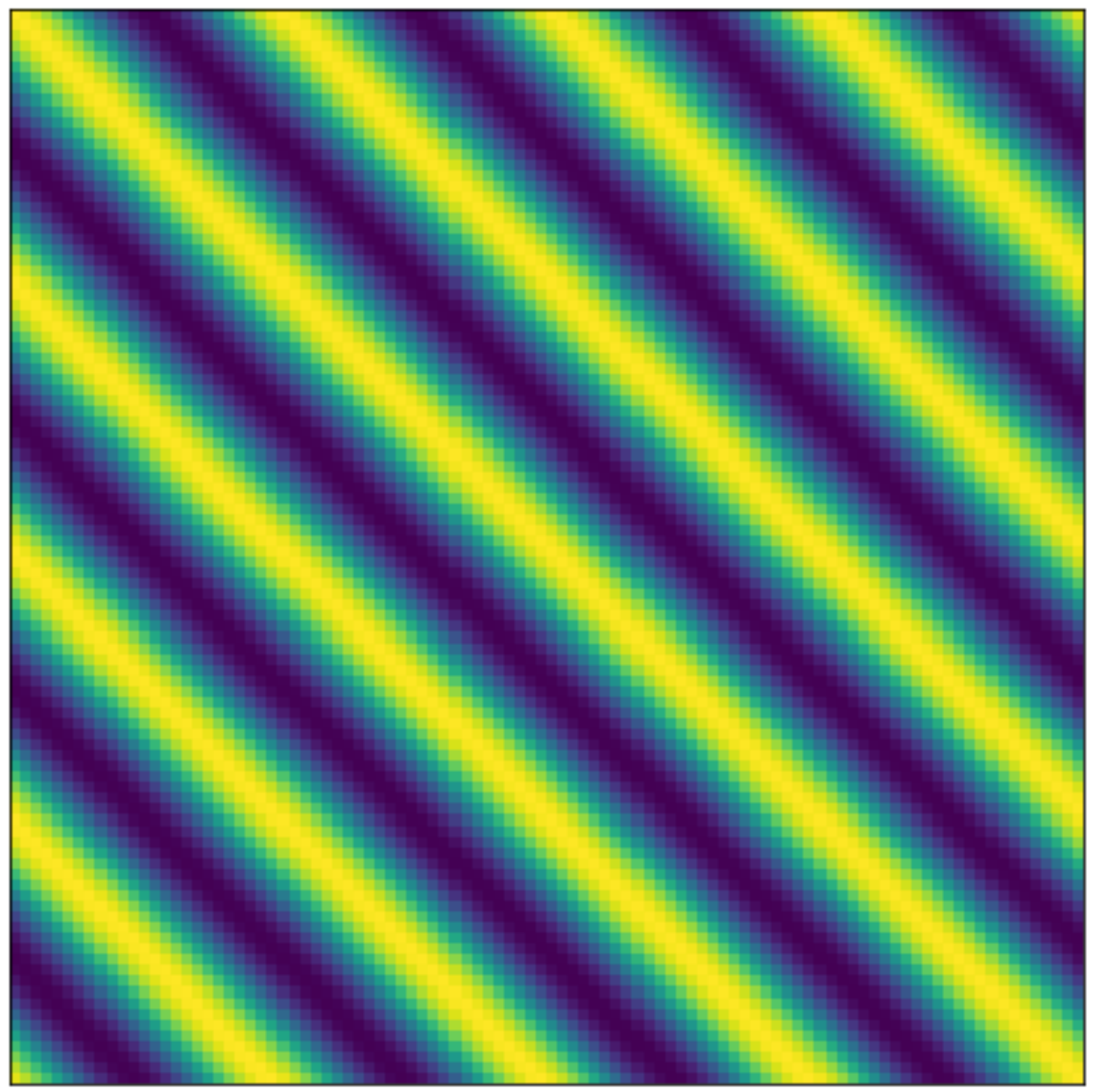


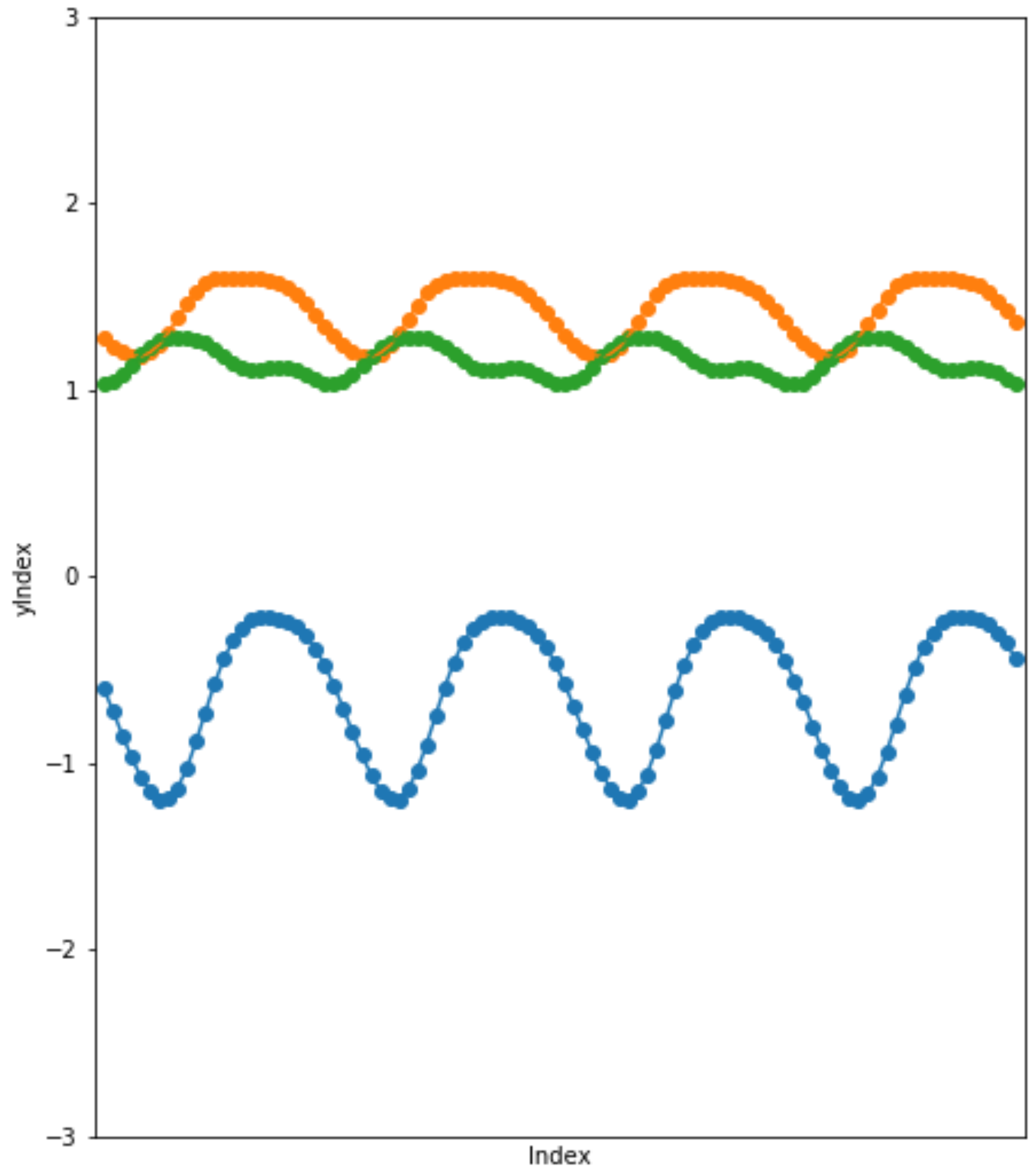
Hyperparameters

- Kernels have parameters which we will call hyperparameters.
- Hyperparameters can have a big impact.
- Want to estimate hyperparameters!

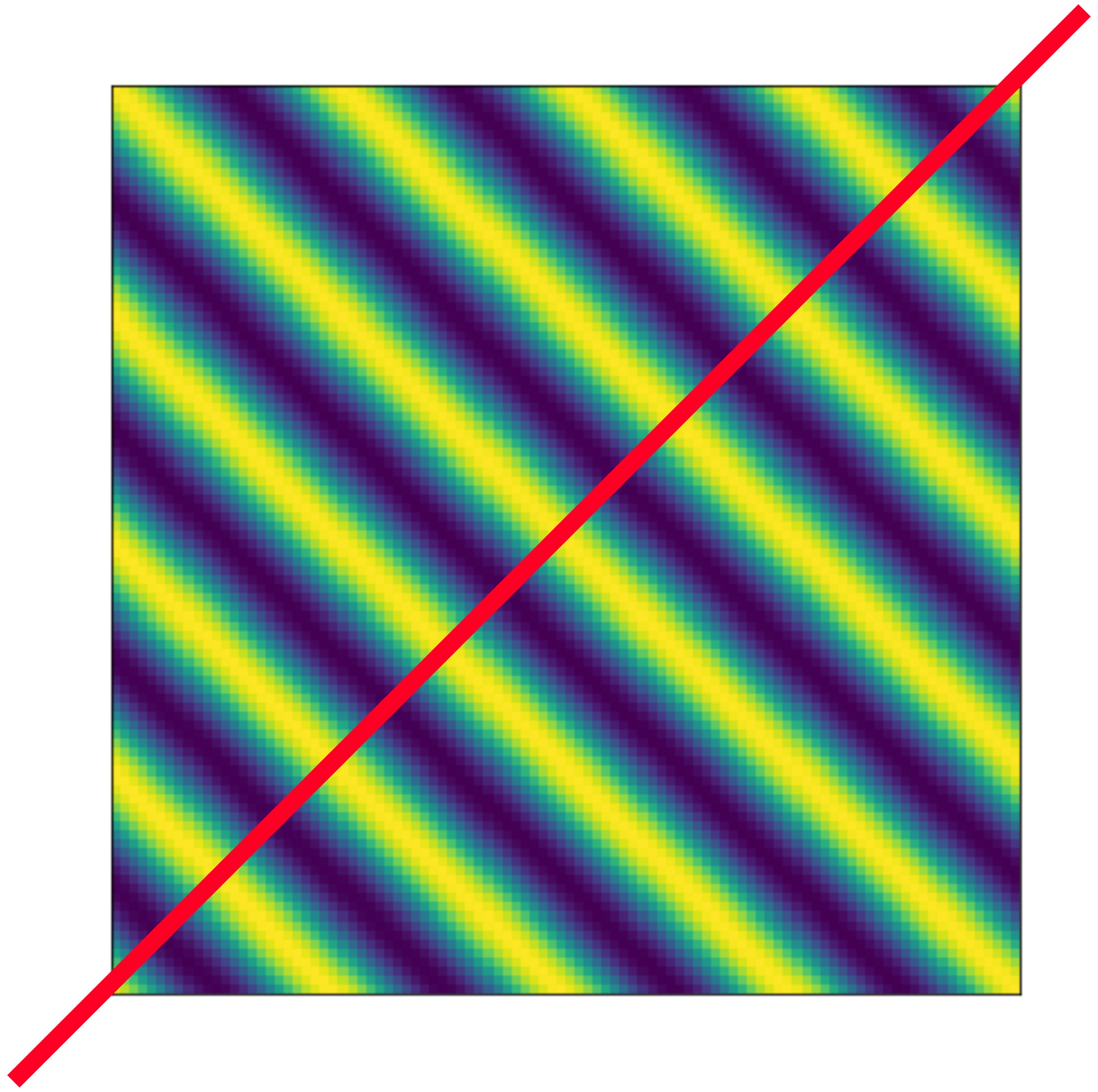
$$\arg\max_{\theta} \log p(\mathbf{y}|\theta)$$

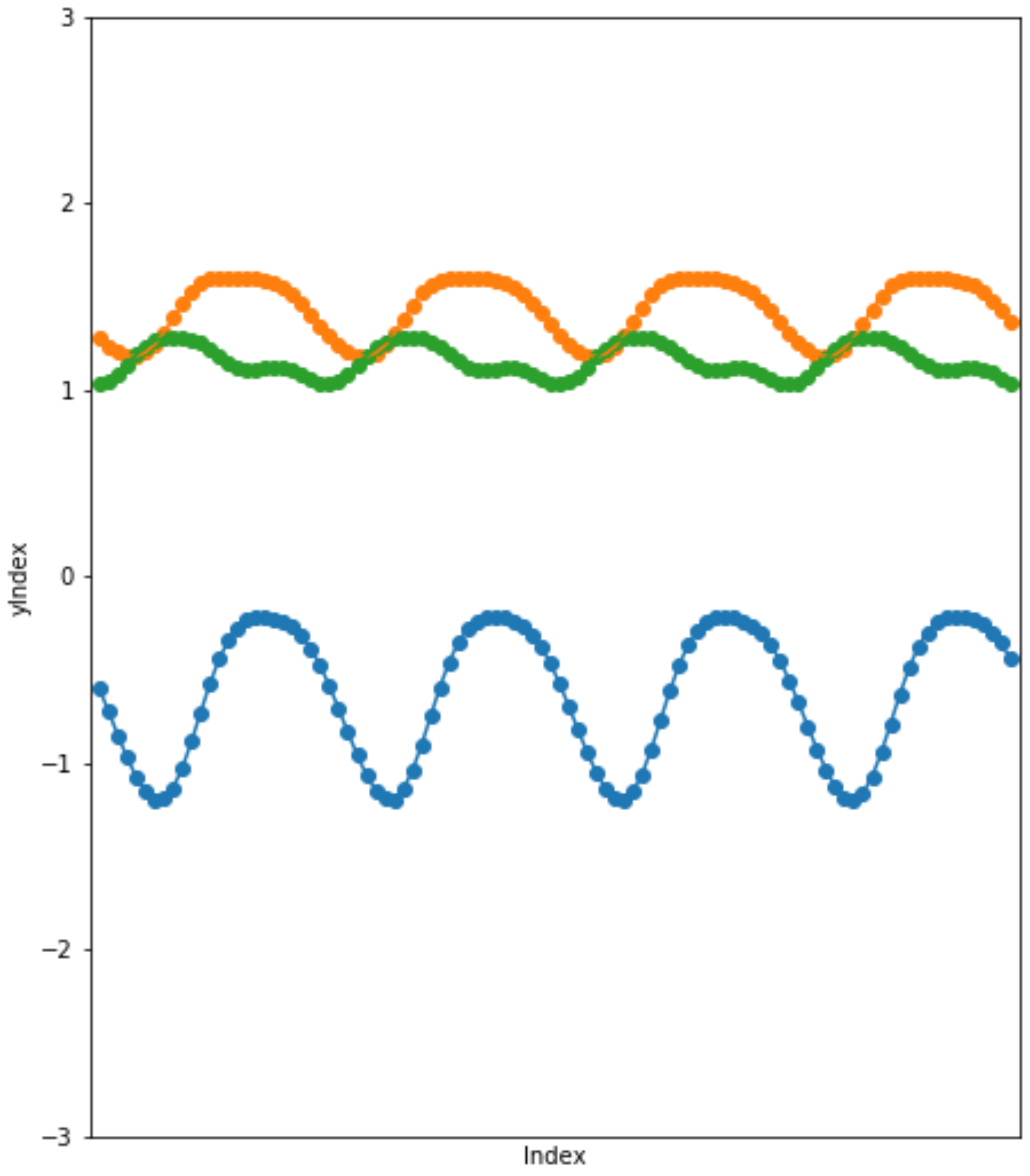
Applying Gaussian Processes



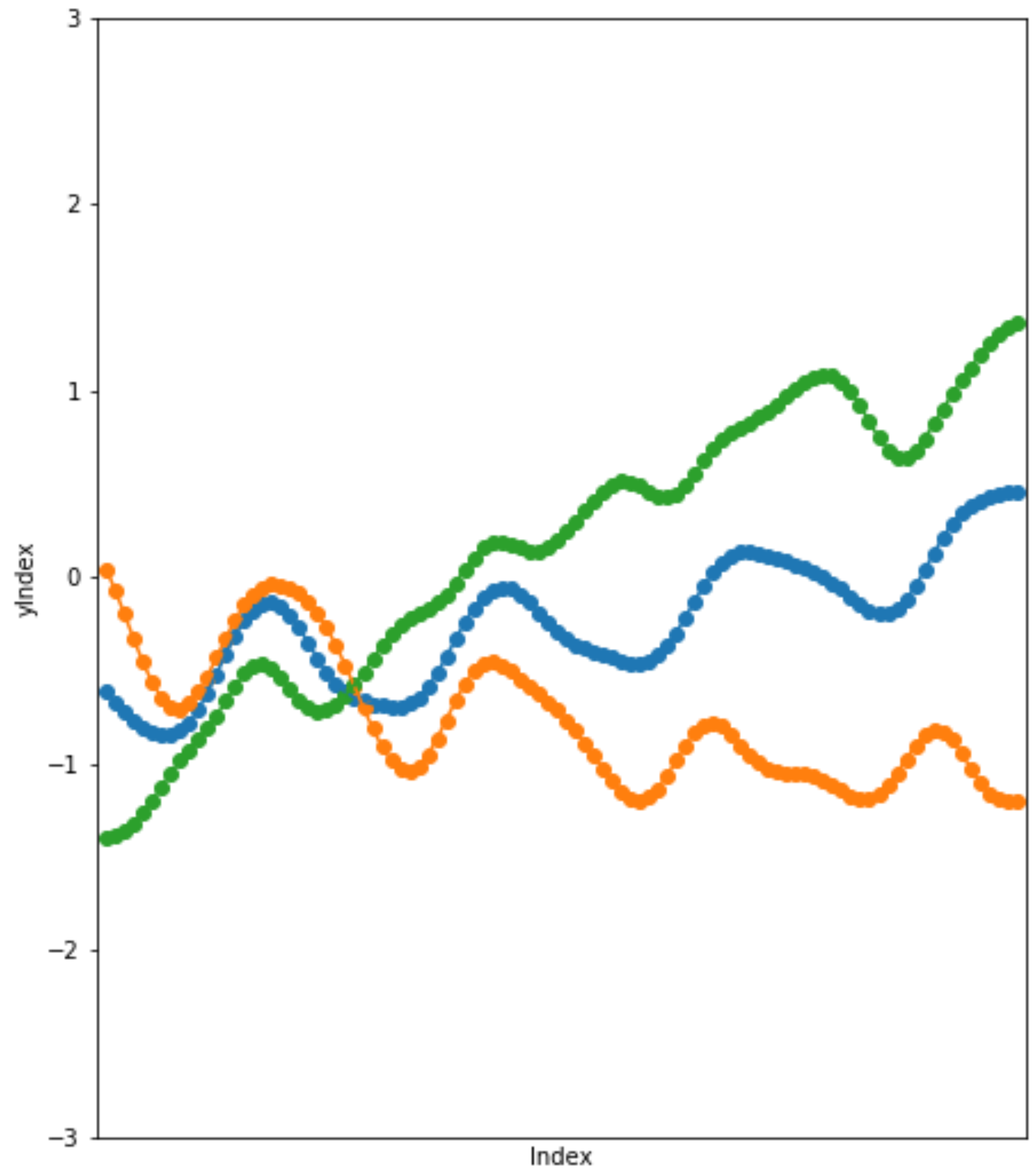


Applying Gaussian Processes





APPLYING GAUSSIAN PROCESSES



Session 3 - Correlation in the data	
Section 1 ~30 Mins	Motivation - Correlation in the data
Section 2 ~40 Mins	An introduction to a Gaussian Process
Section 2.1 ~40 Mins	Applying Gaussian Processes
Section 3 ~30 Mins	An introduction to Celerite

CELERITE

celerite

> **celerite** \se.le.ʁi.te\ *noun, archaic literary*
> A scalable method for Gaussian Process regression. From French *célérité*.

https://celerite.readthedocs.io/en/stable/

https://github.com/dfm/celerite

Celerite

celerite

celerite \se.le.ʁi.te\ *noun, archaic literary*
A scalable method for Gaussian Process regression. From French *célérité*.

A scaleable method for 1D Gaussian Process regression

- *All the good stuff is under the hood.*
- *Celerite is fast $O(N, I^2)$. N is number of data points, I is number of components.*
- *Celerite has Kernels that replicate physical processes.*

Celerite

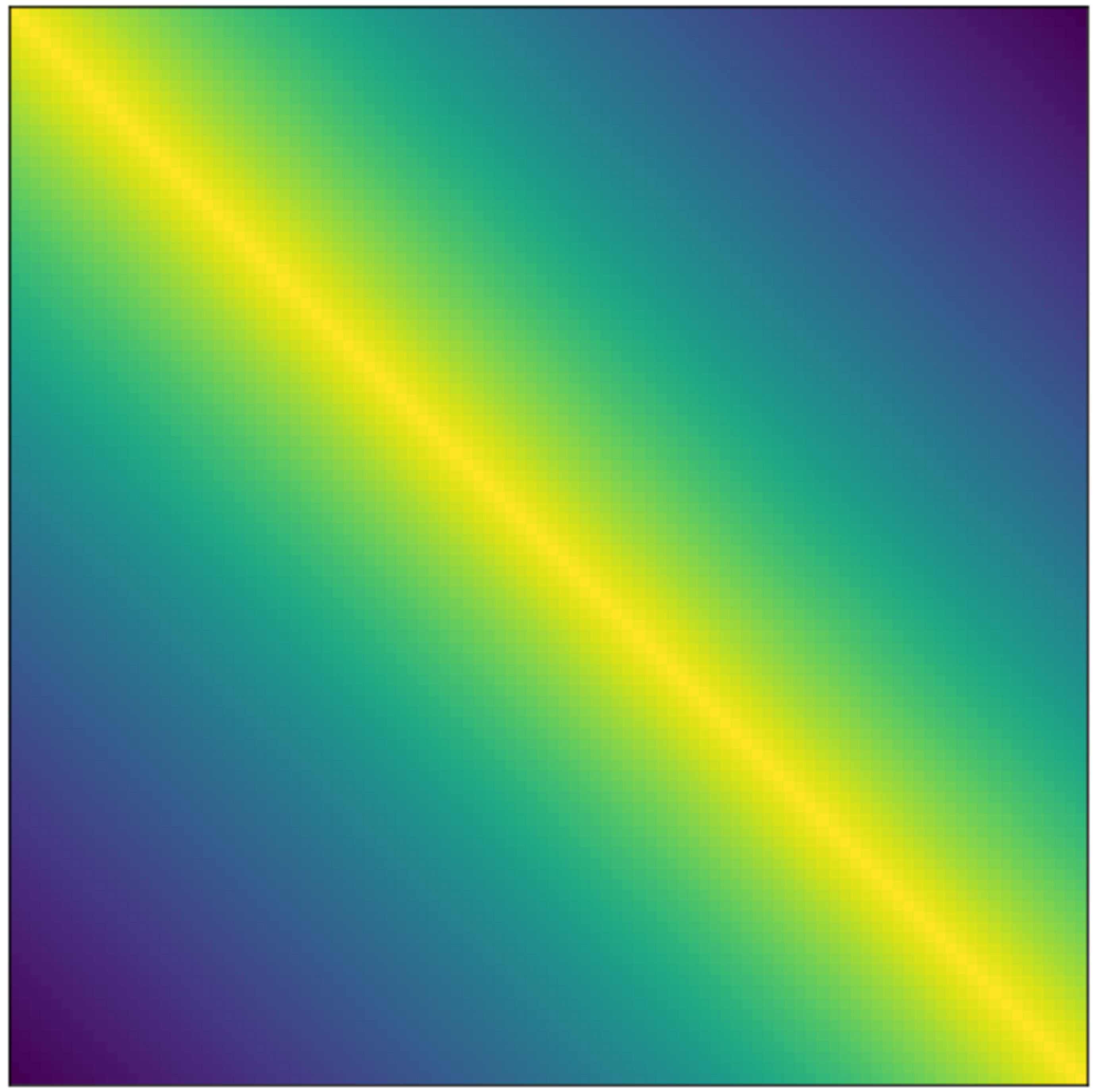
A scaleable method for 1D Gaussian Process regression

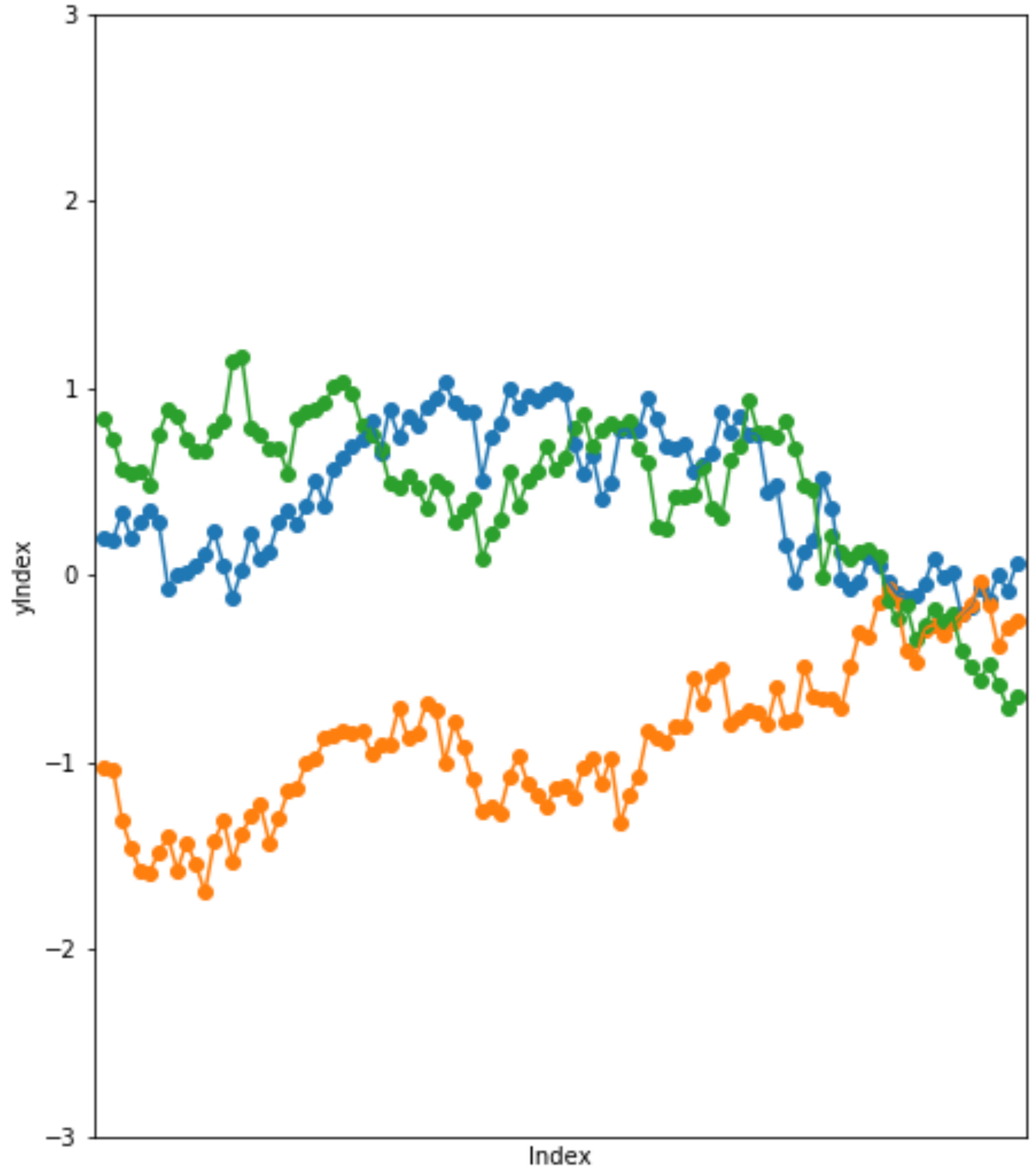
- *Build kernels (sum components)*
- *Use mean model functions*
- *Get likelihood values*
- *Explore parameter space with the algorithm of your choice (e.g., SCIPY, MCMC, emcees …)*

CELERITE - KERNEL EXAMPLES - REAL TERM

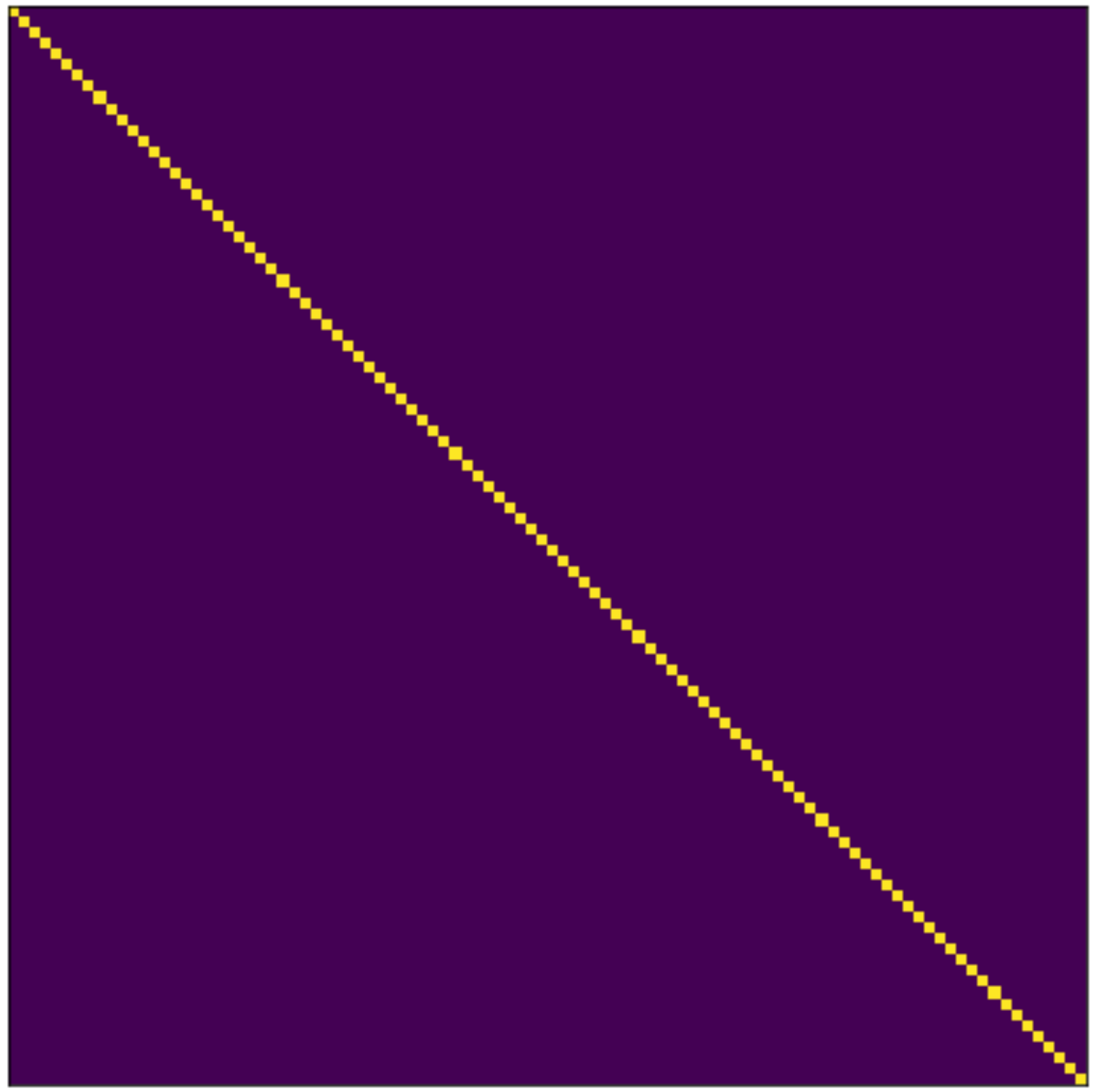
$$k(t_i, t_j) = a \exp(-c\,|t_i - t_j|)$$

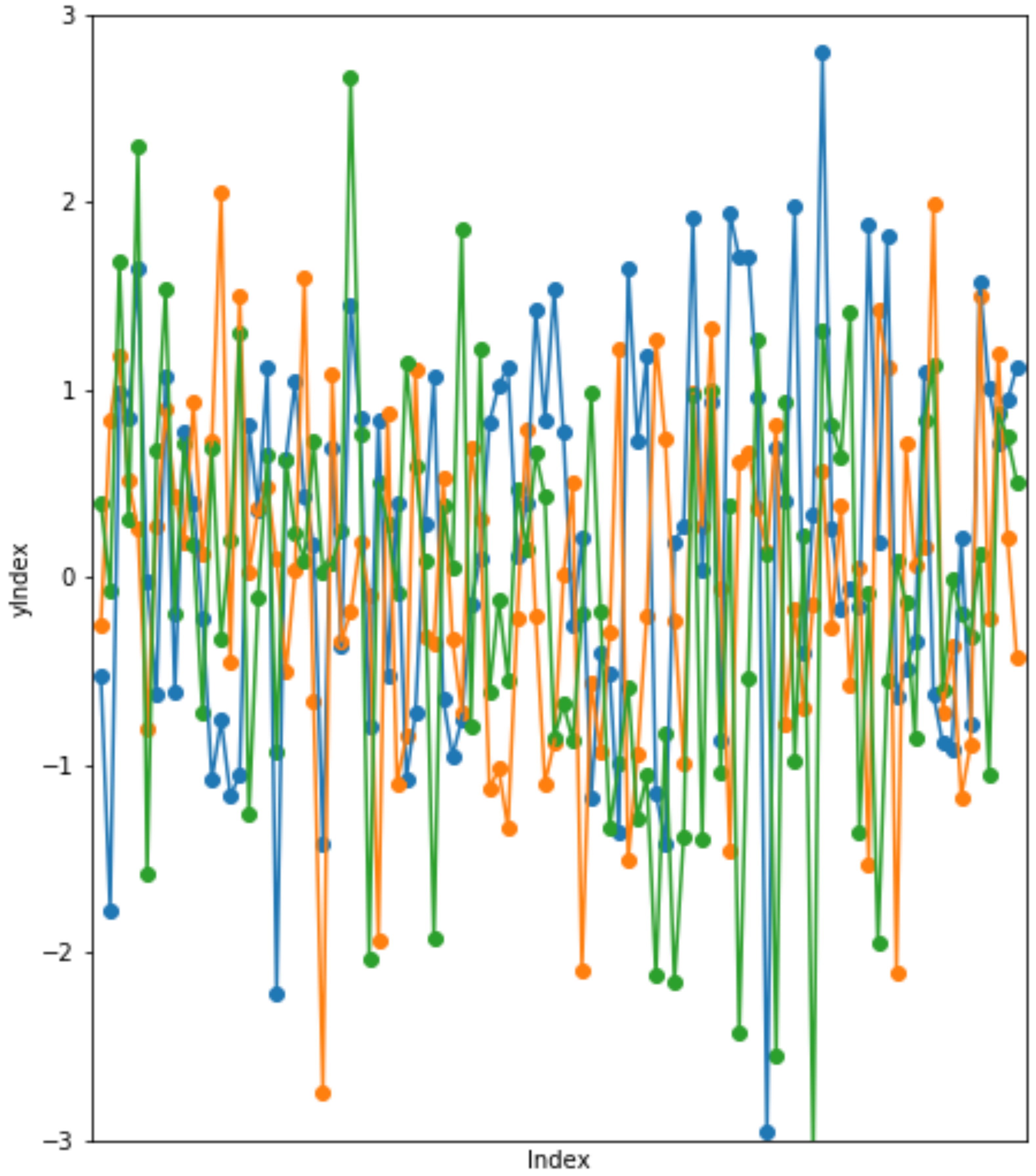
$$a = 1\,,\; c = 0.01$$



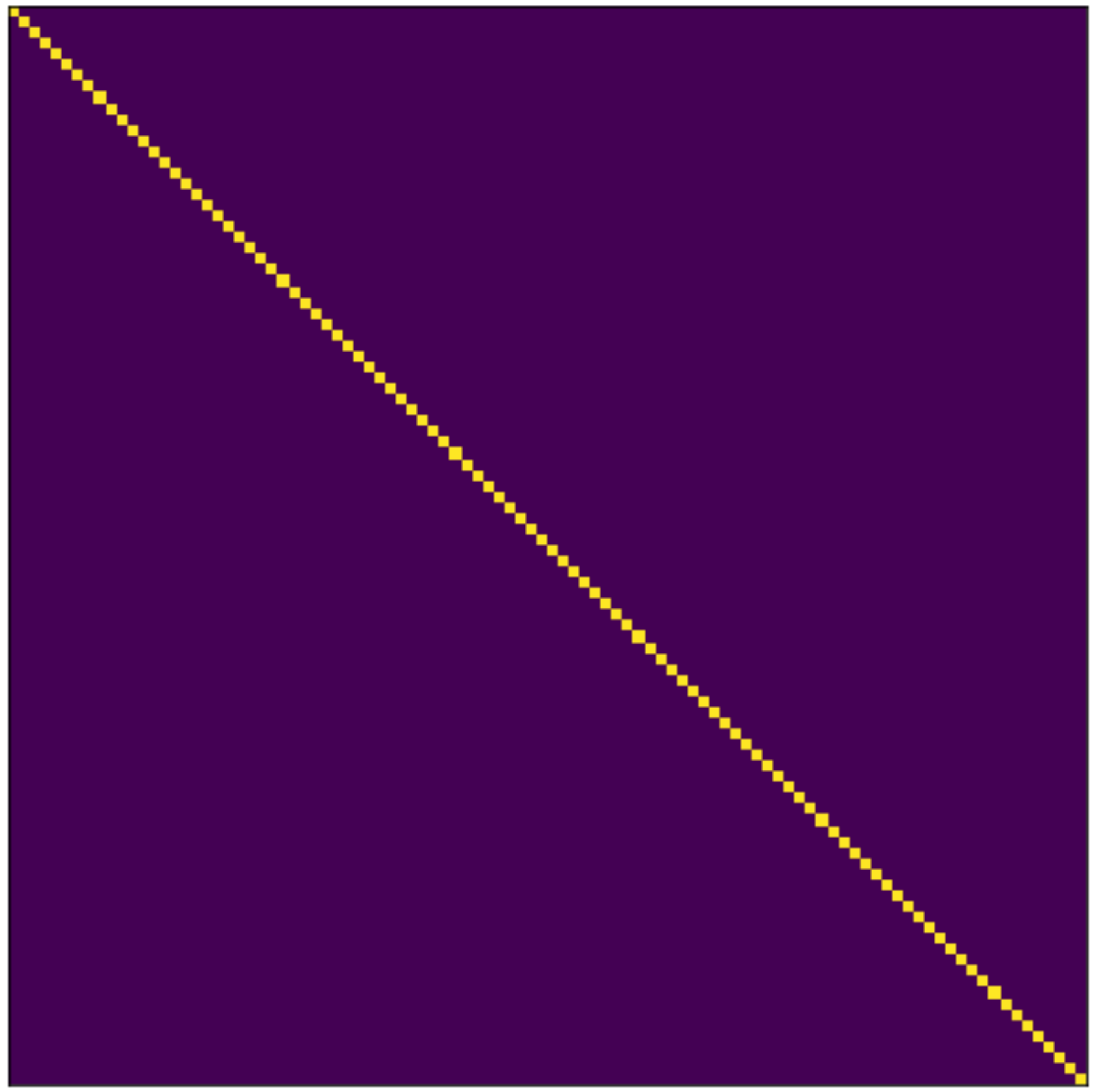


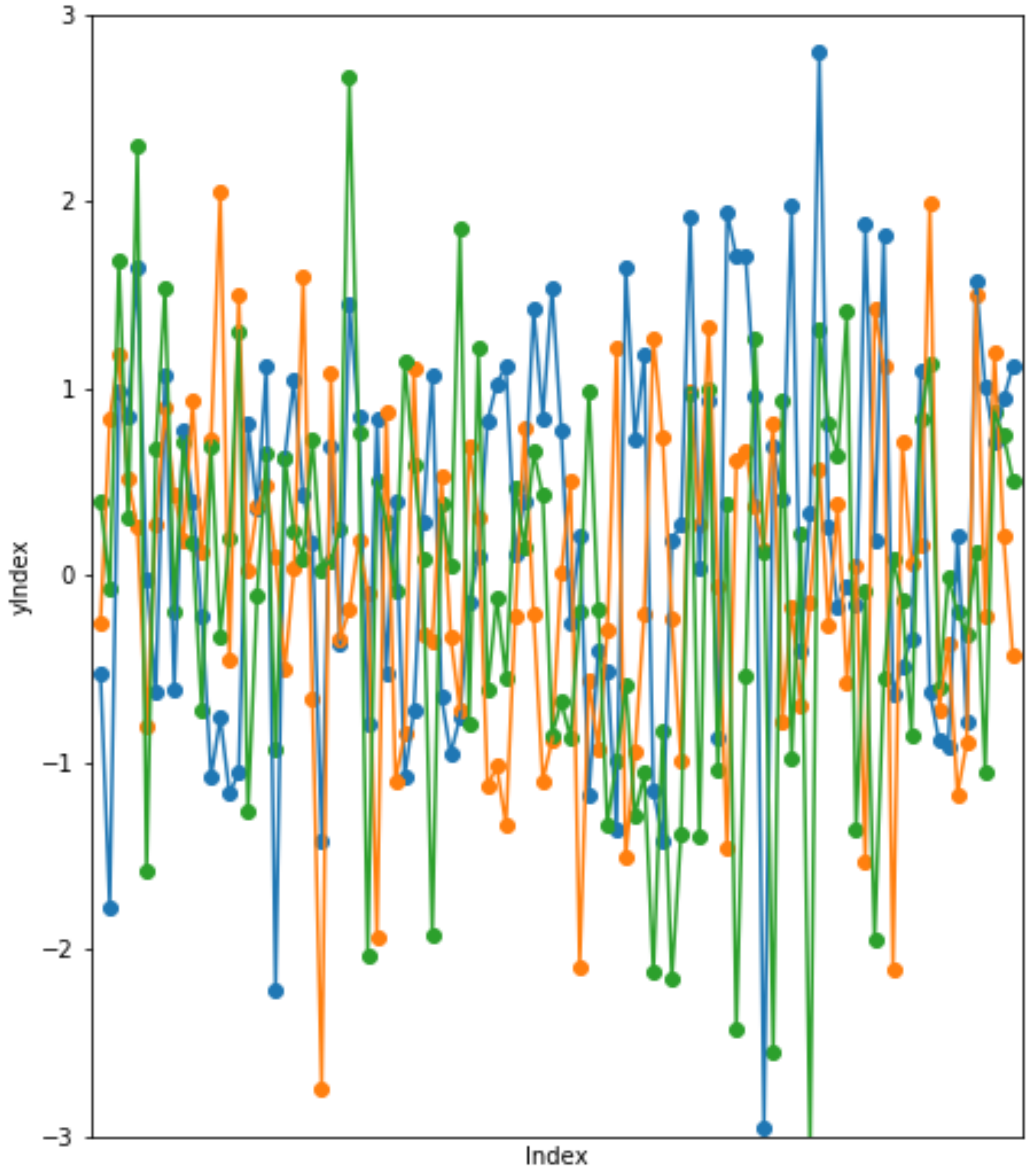
Celerite - Kernel Examples





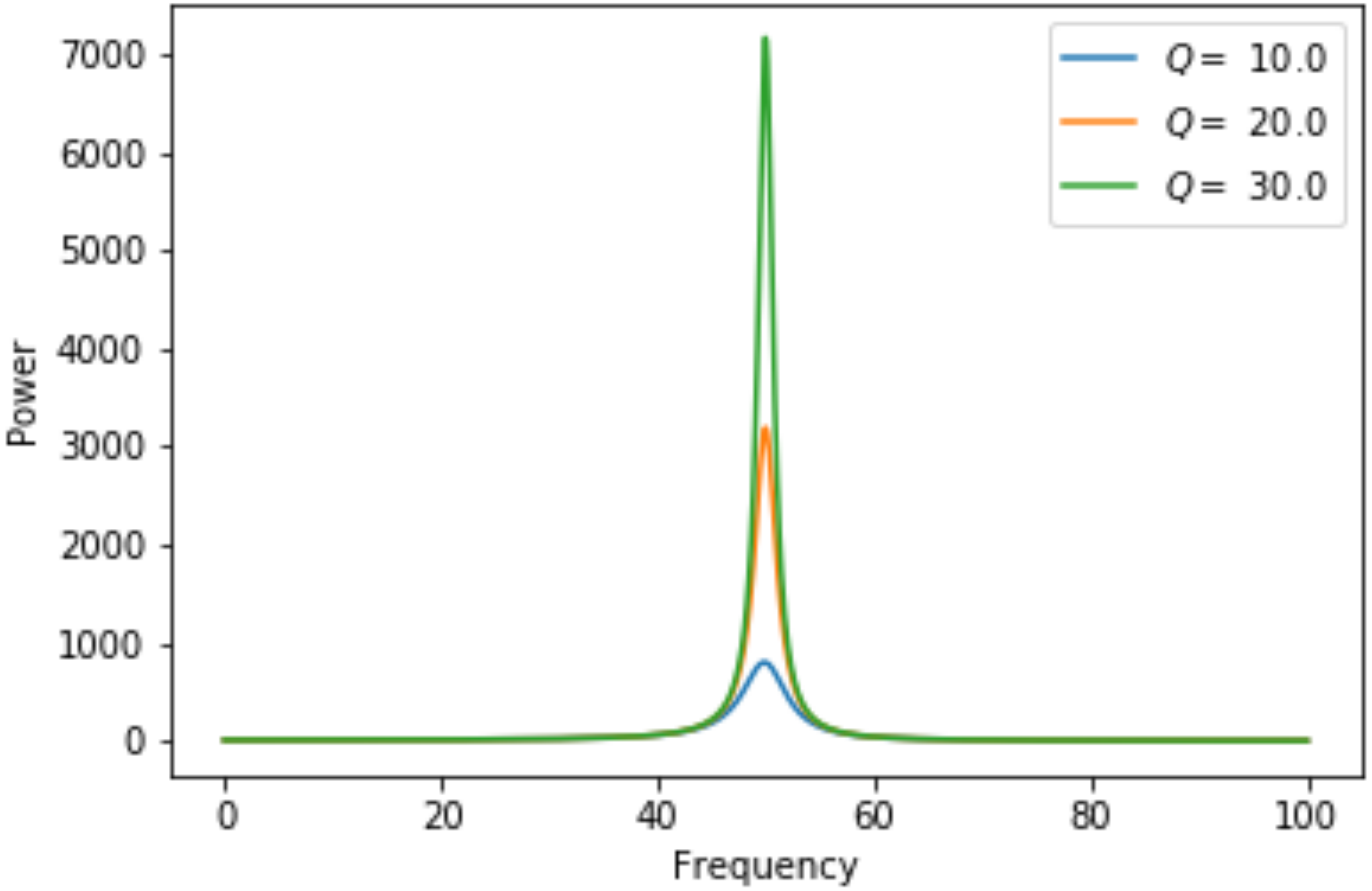
Celerite - Kernel Examples - White Noise (Jitter in Celerite)



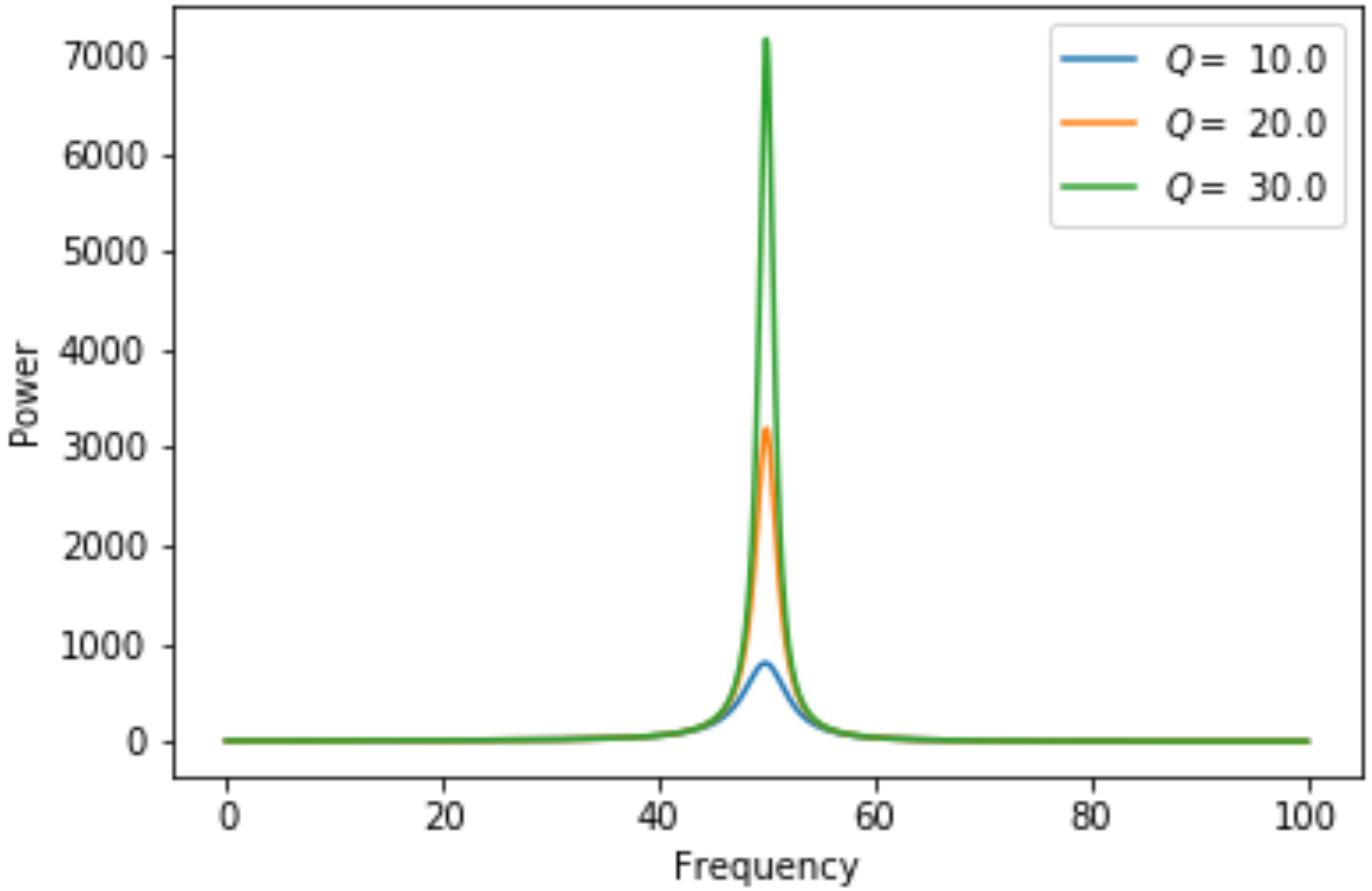


Celerite - SHO terms

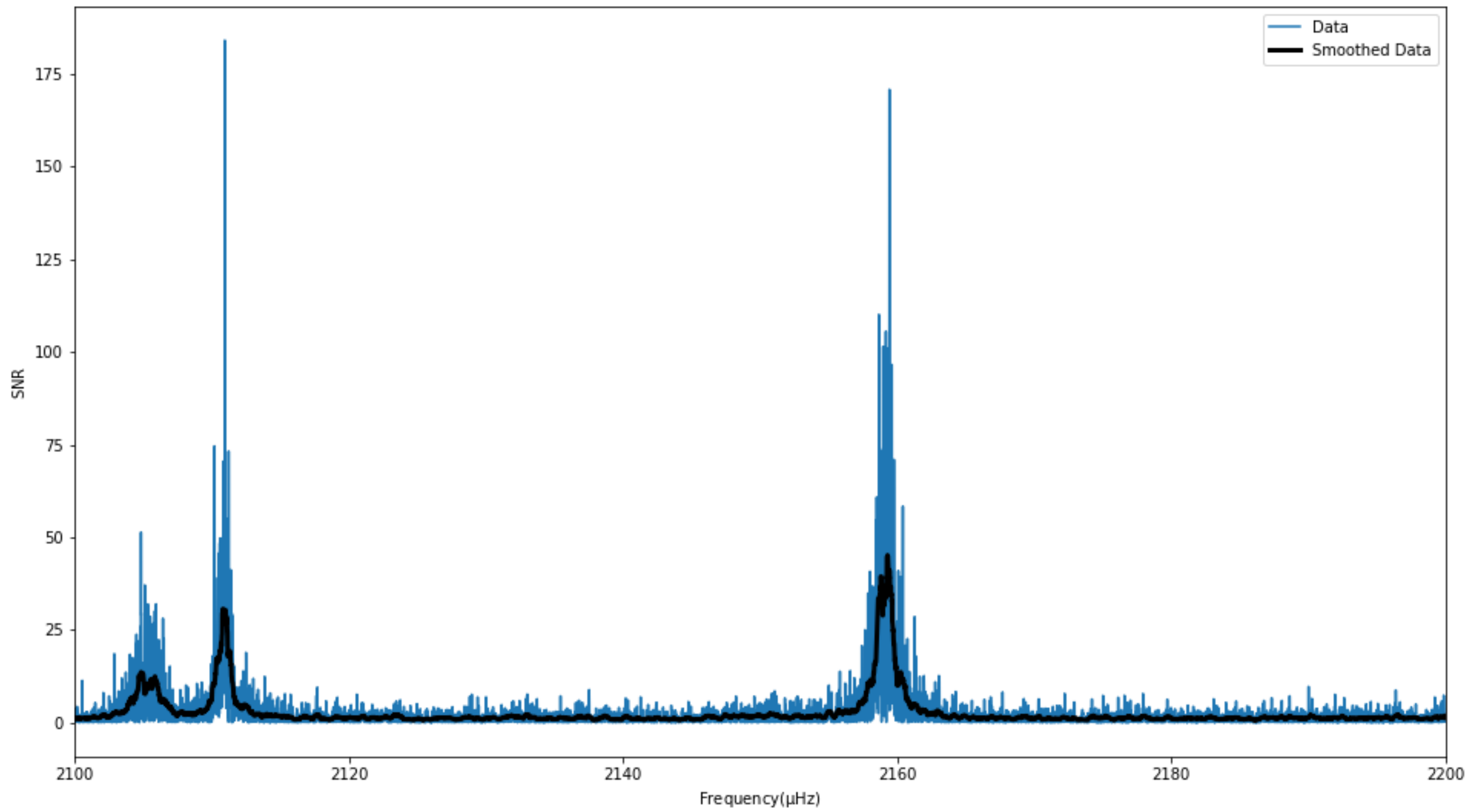
$$S(\omega) = \sqrt{\frac{2}{\pi}} \frac{S_0 \omega_0^4}{(\omega^2 - \omega_0^2)^2 + \omega_0^2 \omega^2 / Q^2}$$



Celerite - SHO terms



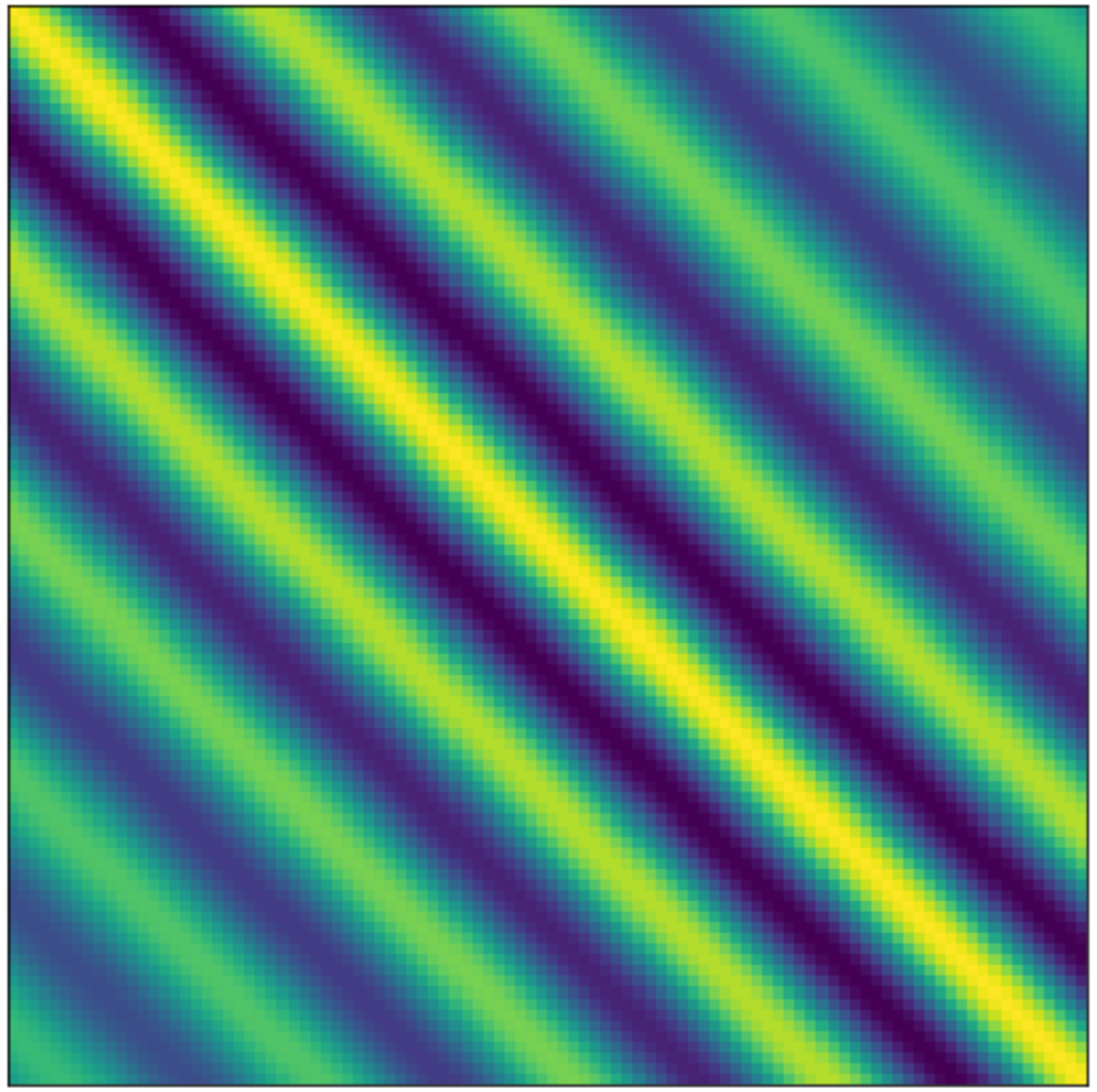
SNR

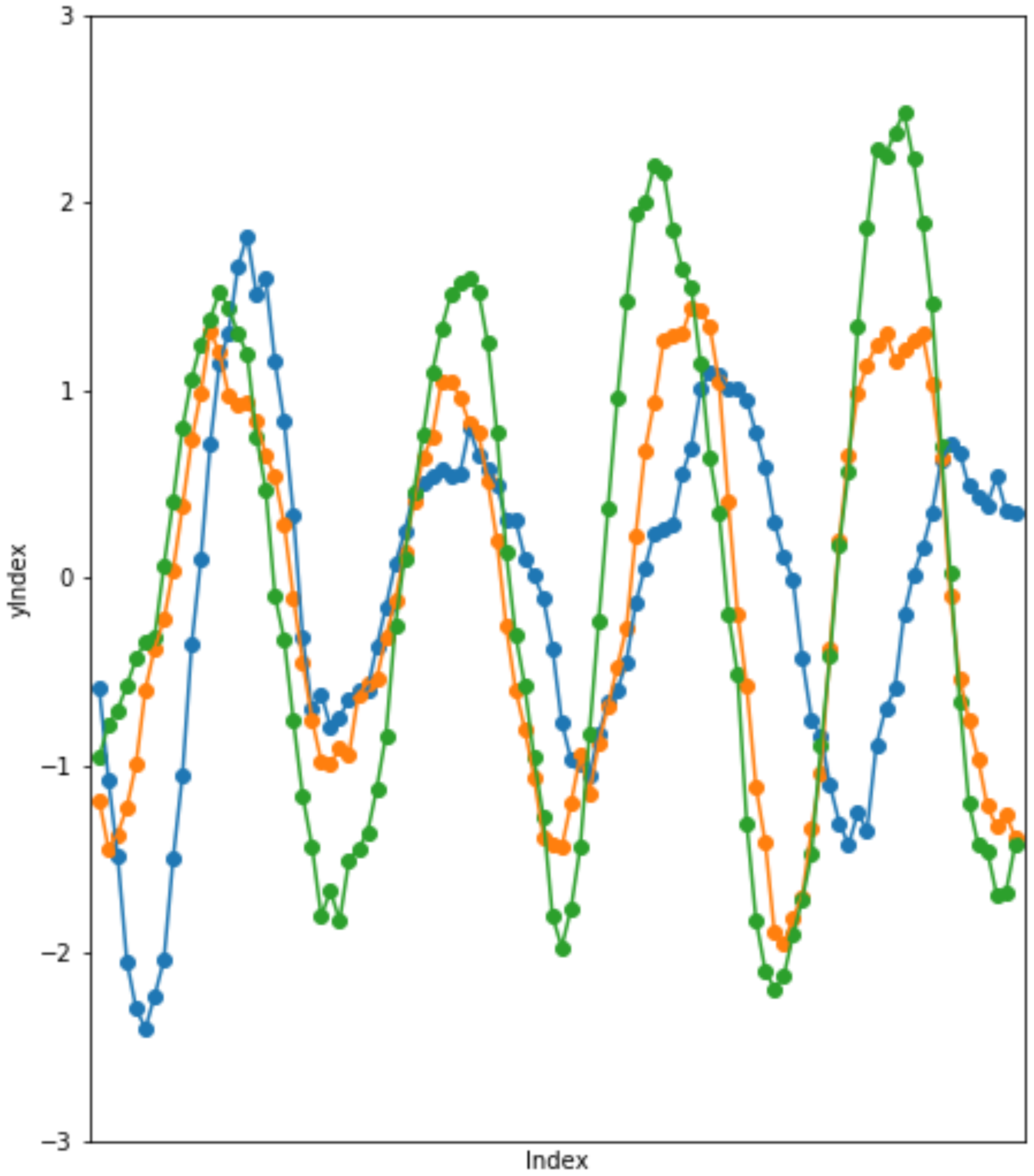


Frequency

Celerite - SHO terms

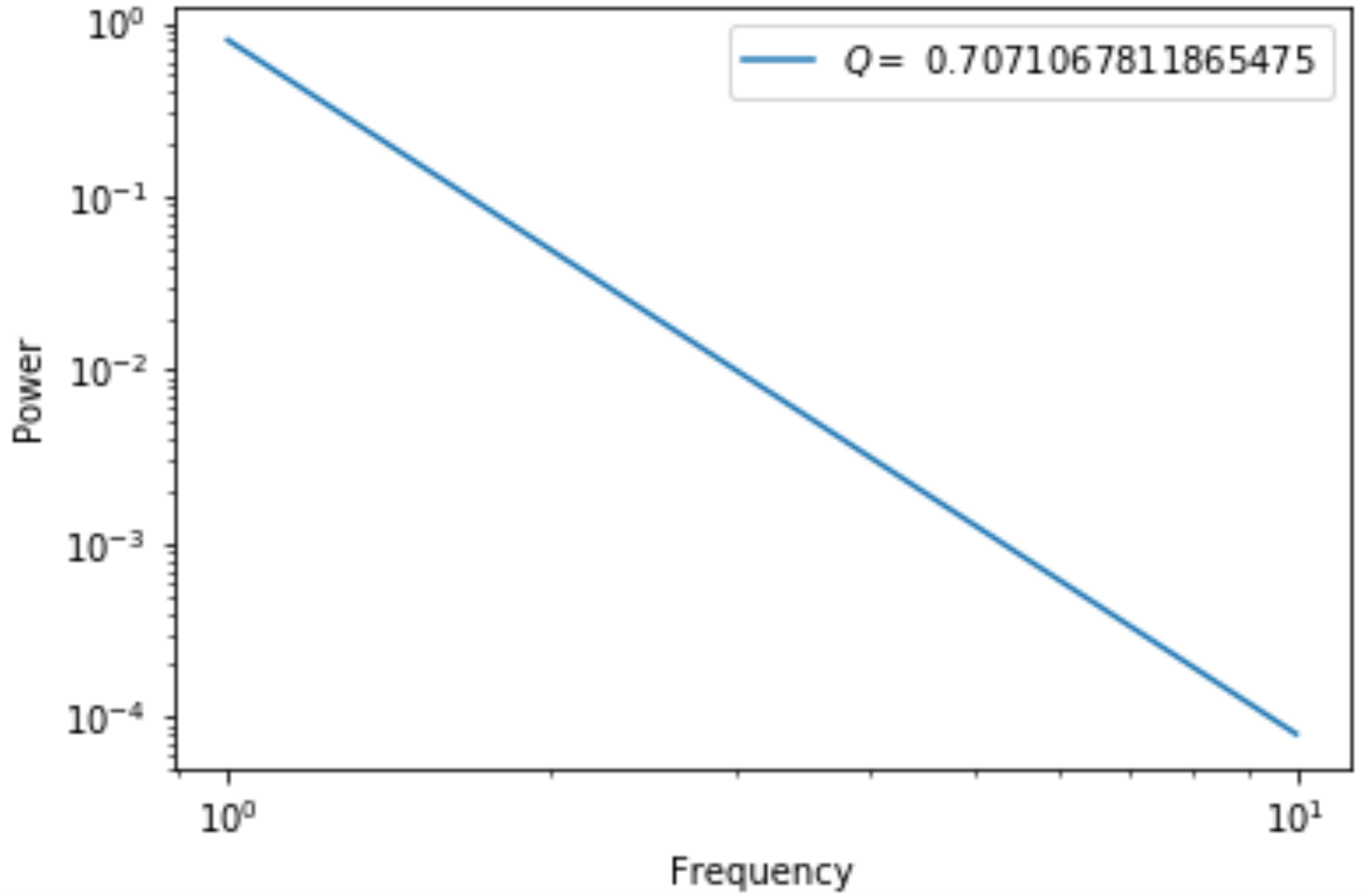
$$S(\omega) = \sqrt{\frac{2}{\pi}} \frac{S_0 \omega_0^4}{(\omega^2 - \omega_0^2)^2 + \omega_0^2 \omega^2 / Q^2}$$





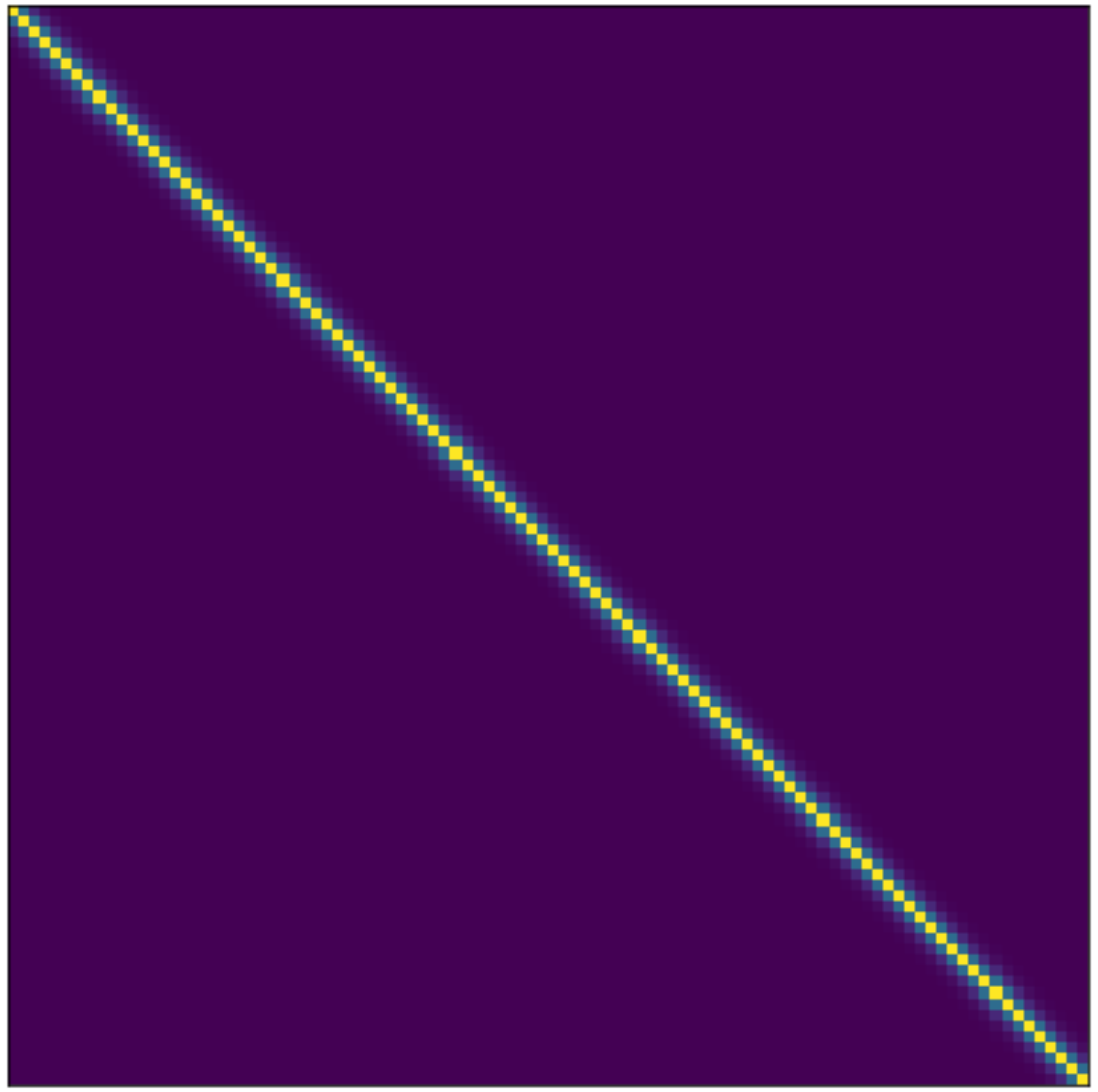
CELERITE - SHO TERMS

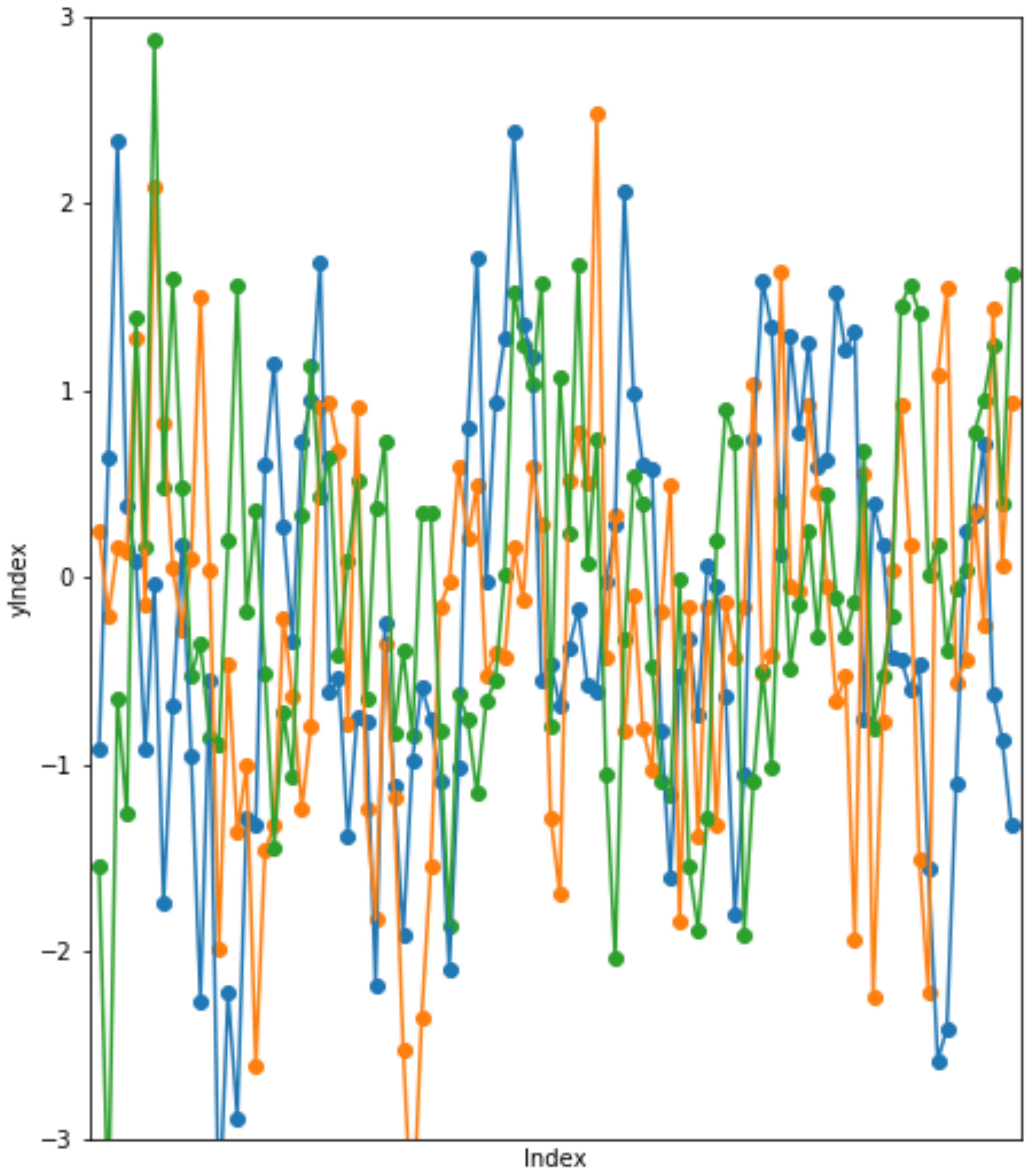
$$S(\omega) = \sqrt{\frac{2}{\pi}} \frac{S_0 \omega_0^4}{(\omega^2 - \omega_0^2)^2 + \omega_0^2 \omega^2 / Q^2}$$



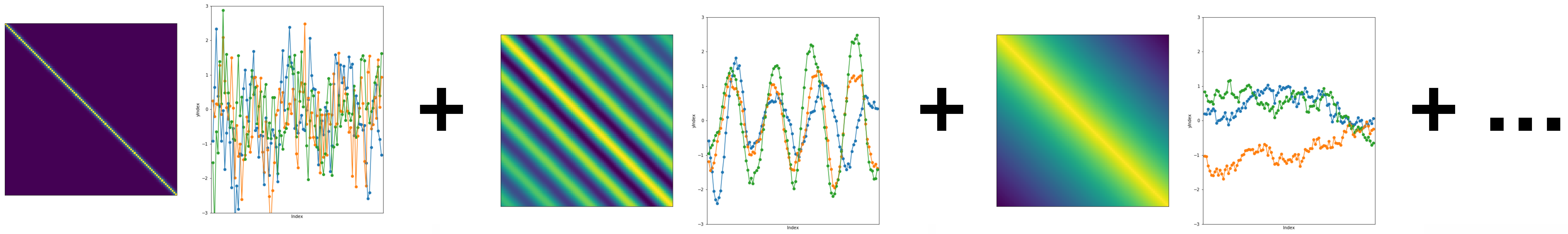
CELERITE - SHO TERMS

$$S(\omega) = \sqrt{\frac{2}{\pi}} \frac{S_0 \omega_0^4}{(\omega^2 - \omega_0^2)^2 + \omega_0^2 \omega^2 / Q^2}$$

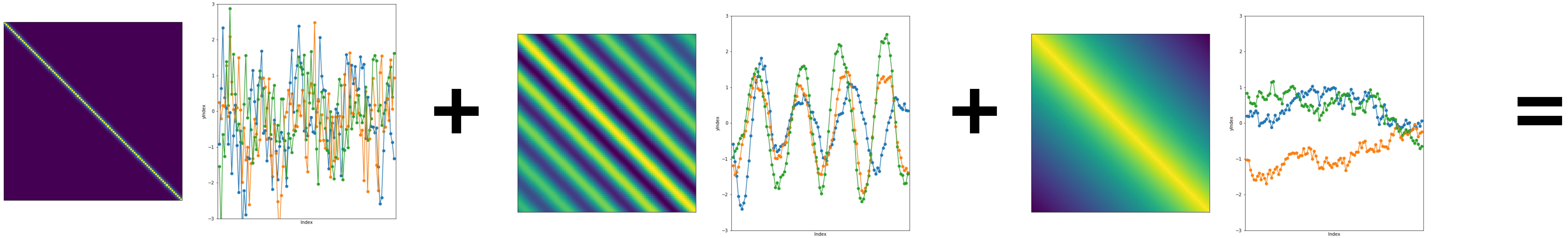




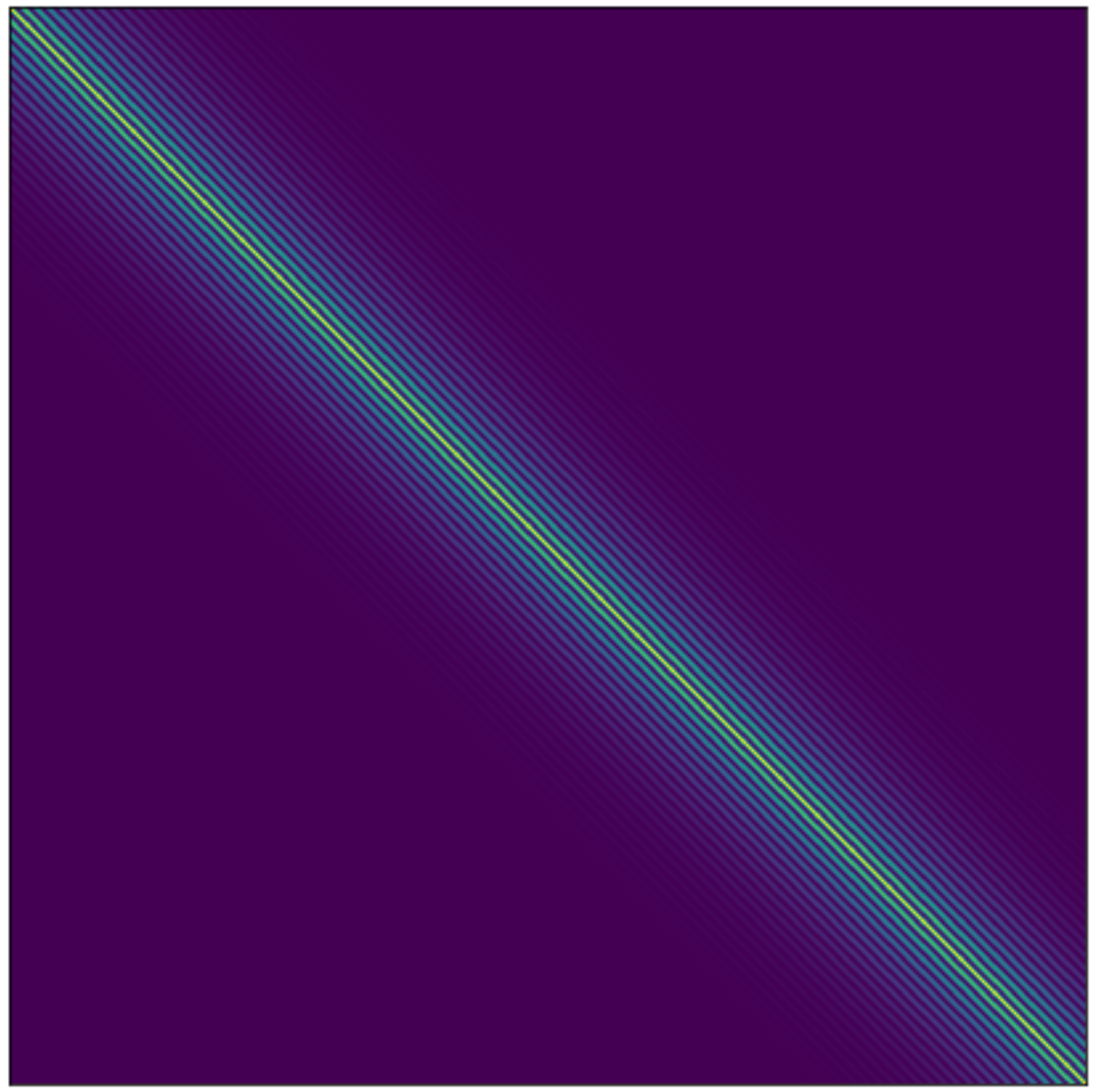
Celerite - Term addition

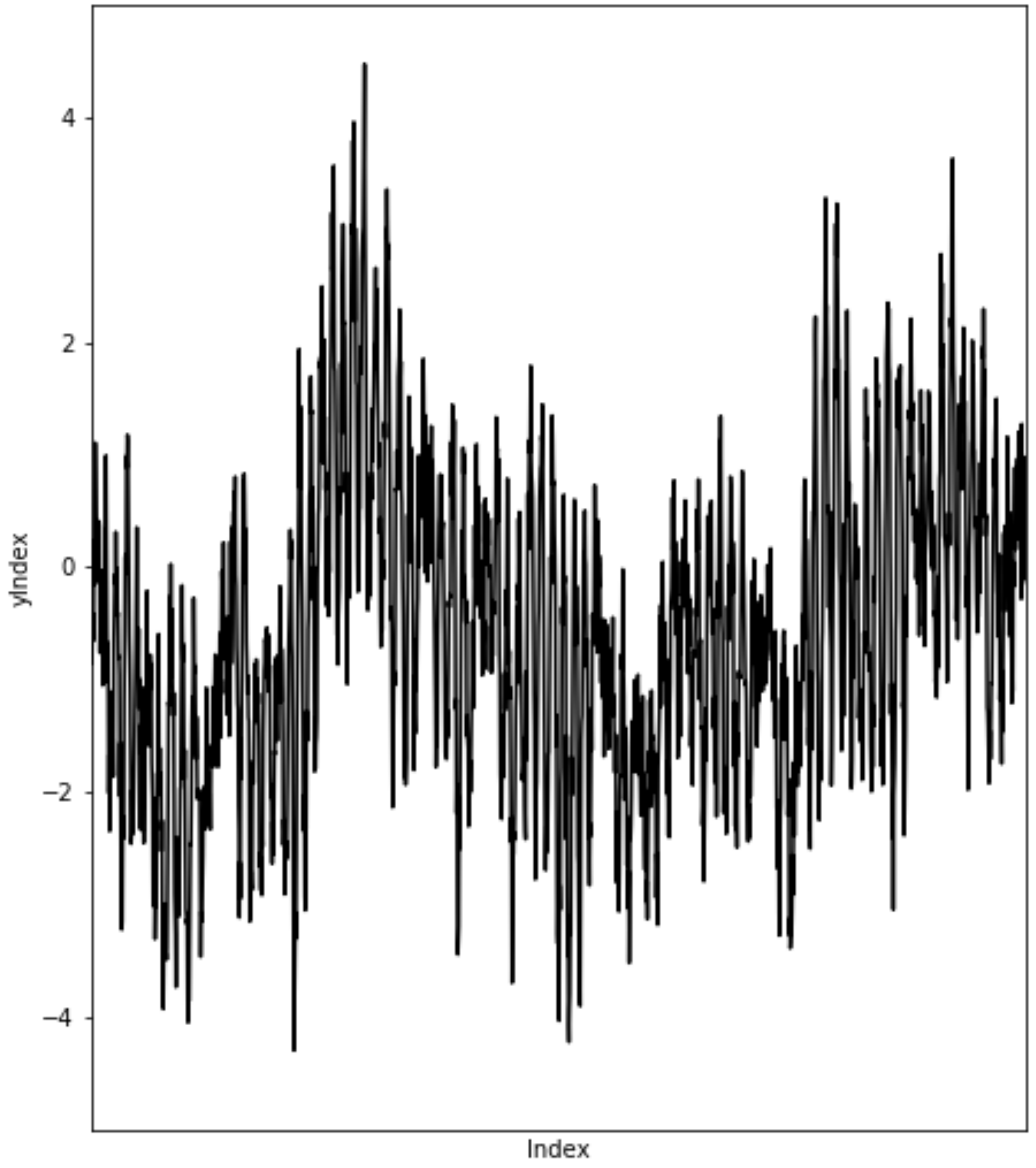


Celerite - Term addition



Ignore the scale NxM issue …





Celerite - Mean Function Examples

Mean functions are useful for deterministic functions

- *Planet RV signal*
- *Planet Transit Signal*
- *Very high Q pulsations*
- *Alien (deliberate) communication signals*
- *…*

A SCALEABLE METHOD FOR 1D GAUSSIAN PROCESS REGRESSION

- *GET LIKELIHOOD VALUES*

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

or

$$\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{c}\end{bmatrix}, \begin{bmatrix}A & B\\ B^T & C\end{bmatrix}\right)$$

$$p(\mathbf{y}_1) = \int p(\mathbf{y}_2, \mathbf{y}_1)\, \mathrm{d}\mathbf{y}_2$$

so

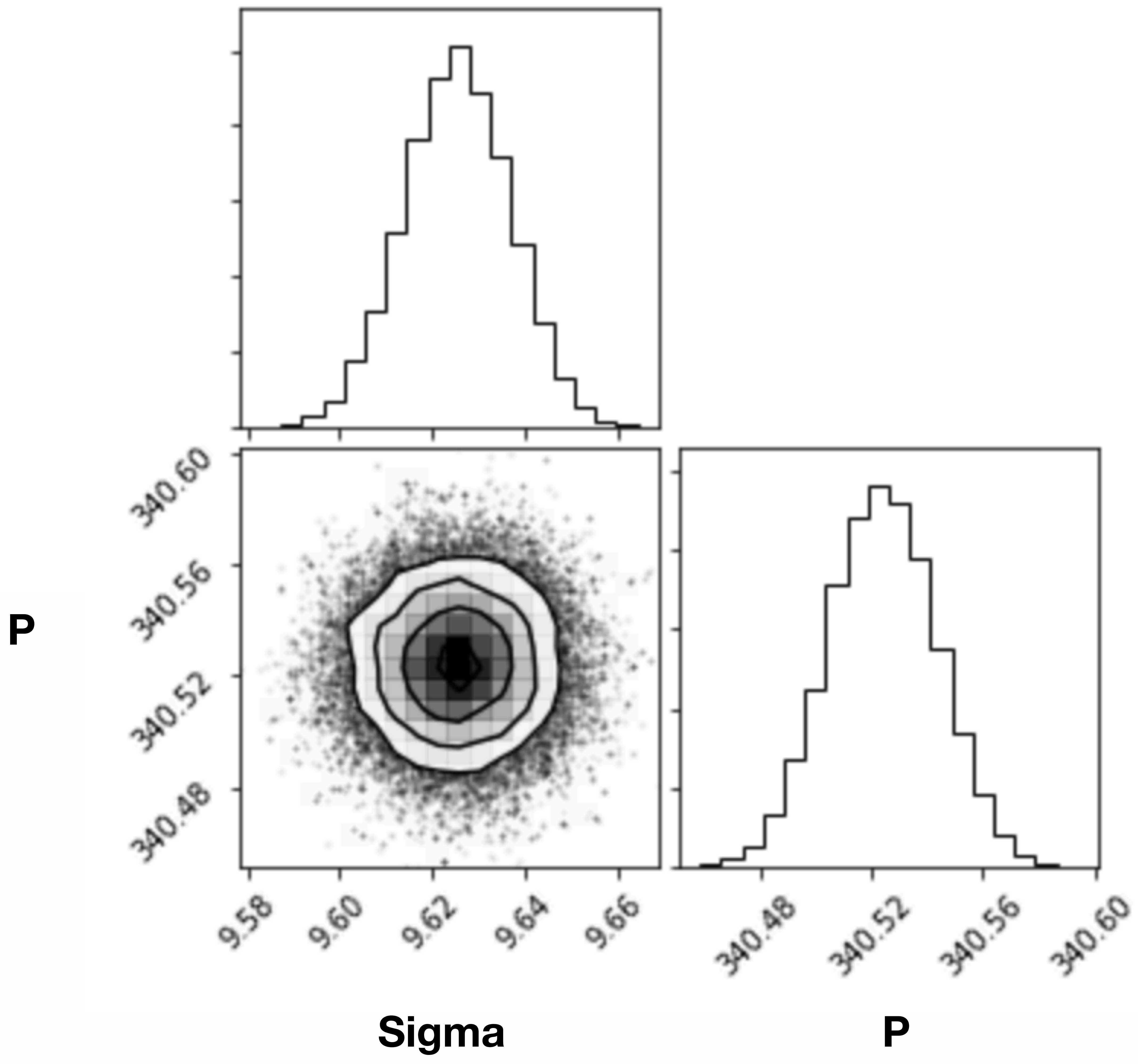
$$p(\mathbf{y}_1) = \mathcal{N}(\mathbf{a}, A).$$

Celerite - Likelihood Calls

A scaleable method for 1D Gaussian Process regression

- *Get likelihood values*
- *Celerite includes a basic prior functionality (e.g., bounds)*
- *You might want to apply more elaborate priors*

Celerite - Exploring the parameter space



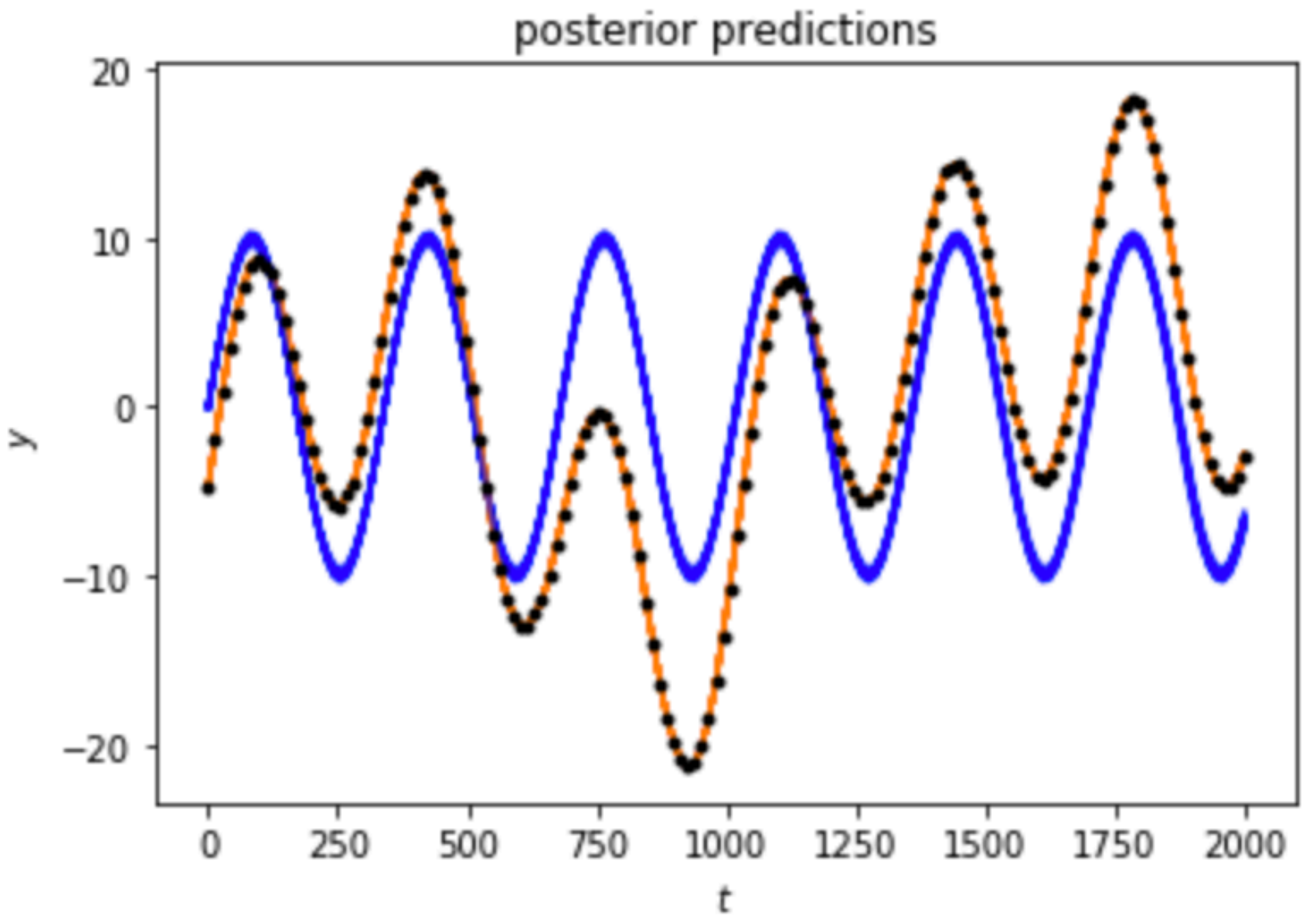
Various ways to explore the parameter space

- MLE
- MCMC
- Nested Sampling
- Integration
- ...

Celerite - Making Predictions

A scaleable method for 1D Gaussian Process regression

- Celerite includes a predict method
- Couple predict with your posterior inference
- Decide what you want to show (i.e., including or excluding observational uncertainty)



Session 3 - Correlation in the data	
Section 1 ~30 Mins	Motivation - Correlation in the data
Section 2 ~40 Mins	An introduction to a Gaussian Process
Section 2.1 ~40 Mins	Applying Gaussian Processes
Section 3 ~30 Mins	An introduction to Celerite

Key Take Aways From This Week/Session:

- Noise can correlated as a result of physical processes
- Don't neglect correlation
- A GP is an extension of a multivariate normal distribution
- A GP has very interesting properties that we can use to model correlated noise
- There are a bunch of GP software implementations but one that ties in nicely with a physical interpretation is Celerite