Gaussian Process Models

Lecture 14

Dr. Colin Rundel

Multivariate Normal

Multivariate Normal Distribution

For an n-dimension multivate normal distribution with covariance $\boldsymbol{\Sigma}$ (positive semidefinite) can be written as

$$\underset{n\times 1}{\boldsymbol{y}} \sim N(\underset{n\times 1}{\boldsymbol{\mu}}, \underset{n\times n}{\boldsymbol{\Sigma}})$$

$$\begin{pmatrix} y_1 \\ \vdots \\ ys_n \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}, \begin{pmatrix} \rho_{11}\sigma_1\sigma_1 & \cdots & \rho_{1n}\sigma_1\sigma_n \\ \vdots & \ddots & \vdots \\ \rho_{n1}\sigma_n\sigma_1 & \cdots & \rho_{nn}\sigma_n\sigma_n \end{pmatrix}\right)$$

Density

For the n dimensional multivate normal given on the last slide, its density is given by

$$(2\pi)^{-n/2} \ \det(\boldsymbol{\Sigma})^{-1/2} \ \exp\left(-\frac{1}{2}\underset{1\times n}{(\boldsymbol{y}-\boldsymbol{\mu})'}\underset{n\times n}{\boldsymbol{\Sigma}^{-1}}\underset{n\times 1}{(\boldsymbol{y}-\boldsymbol{\mu})}\right)$$

and its log density is given by

$$-\frac{n}{2}\log 2\pi - \frac{1}{2}\log\det(\boldsymbol{\Sigma}) - -\frac{1}{2}\underset{1\times n}{(\boldsymbol{y}-\boldsymbol{\mu})'}\underset{n\times n}{\boldsymbol{\Sigma}^{-1}}\underset{n\times 1}{(\boldsymbol{y}-\boldsymbol{\mu})}$$

Sampling

To generate draws from an n-dimensional multivate normal with mean $\underset{\mathrm{n}\times 1}{\boldsymbol{\mu}}$ and covariance matrix $\underset{\mathrm{n}\times\mathrm{n}}{\boldsymbol{\Sigma}}$,

- Find a matrix $\underset{\mathrm{n}\times\mathrm{n}}{\boldsymbol{A}}$ such that $\boldsymbol{\Sigma} = \boldsymbol{A}\,\boldsymbol{A}^{\mathrm{t}}$
  - most often we use $\boldsymbol{A} = \mathrm{Chol}(\boldsymbol{\Sigma})$ where $\boldsymbol{A}$ is a lower triangular matrix.
- Draw n iid unit normals, $\mathrm{N}(0, 1)$, as $\underset{\mathrm{n}\times 1}{\boldsymbol{z}}$
- Obtain multivariate normal draws using

$$\underset{\mathrm{n}\times 1}{\boldsymbol{y}} = \underset{\mathrm{n}\times 1}{\boldsymbol{\mu}} + \underset{\mathrm{n}\times\mathrm{n}}{\boldsymbol{A}}\;\underset{\mathrm{n}\times 1}{\boldsymbol{z}}$$

Bivariate Examples

$$\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

Marginal distributions

Proposition - For an n-dimensional multivate normal with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, any marginal or conditional distribution of the y's will also be (multivariate) normal.

For a univariate marginal distribution,

$$\mathrm{y_i} = \mathrm{N}(\boldsymbol{\mu}_\mathrm{i},\, \boldsymbol{\Sigma}_\mathrm{ii})$$

For a bivariate marginal distribution,

$$\boldsymbol{y}_\mathrm{ij} = \mathrm{N}\left(\begin{pmatrix}\boldsymbol{\mu}_\mathrm{i} \\ \boldsymbol{\mu}_\mathrm{j}\end{pmatrix},\, \begin{pmatrix}\boldsymbol{\Sigma}_\mathrm{ii} & \boldsymbol{\Sigma}_\mathrm{ij} \\ \boldsymbol{\Sigma}_\mathrm{ji} & \boldsymbol{\Sigma}_\mathrm{jj}\end{pmatrix}\right)$$

For a k-dimensional marginal distribution,

$$\boldsymbol{y}_{\mathrm{i},\ldots,\mathrm{k}} = \mathrm{N}\left(\begin{pmatrix} \boldsymbol{\mu}_{\mathrm{i}} \\ \vdots \\ \boldsymbol{\mu}_{\mathrm{k}} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{\mathrm{ii}} & \cdots & \boldsymbol{\Sigma}_{\mathrm{ik}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{\mathrm{ki}} & \cdots & \boldsymbol{\Sigma}_{\mathrm{kk}} \end{pmatrix}\right)$$

$\mathrm{i}, \ldots, \mathrm{k}$ do not need to be ordered in any particular way

Conditional Distributions

If we partition the n-dimensions into two pieces such that $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2)^t$ then

$$\underset{n\times 1}{\boldsymbol{y}} \sim \mathrm{N}\left(\underset{n\times 1}{\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}}, \underset{n\times n}{\begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}} \right)$$

$$\underset{k\times 1}{\boldsymbol{y}_1} \sim \mathrm{N}(\underset{k\times 1}{\boldsymbol{\mu}_1}, \underset{k\times k}{\boldsymbol{\Sigma}_{11}})$$

$$\underset{n-k\times 1}{\boldsymbol{y}_2} \sim \mathrm{N}(\underset{n-k\times 1}{\boldsymbol{\mu}_2}, \underset{n-k\times n-k}{\boldsymbol{\Sigma}_{22}})$$

then the conditional distributions are given by

$$\boldsymbol{y}_1 \mid \boldsymbol{y}_2 = \boldsymbol{a} \sim \mathrm{N}(\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\, \boldsymbol{\Sigma}_{22}^{-1}\, (\boldsymbol{a} - \boldsymbol{\mu}_2),\; \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\, \boldsymbol{\Sigma}_{22}^{-1}\, \boldsymbol{\Sigma}_{21})$$

$$\boldsymbol{y}_2 \mid \boldsymbol{y}_1 = \boldsymbol{b} \sim \mathrm{N}(\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\, \boldsymbol{\Sigma}_{11}^{-1}\, (\boldsymbol{b} - \boldsymbol{\mu}_1),\; \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\, \boldsymbol{\Sigma}_{11}^{-1}\, \boldsymbol{\Sigma}_{12})$$

Gaussian Processes

From Shumway,

> A process, $\boldsymbol{y} = \{\mathrm{y(t)} \; : \; \mathrm{t} \in \mathrm{T}\}$, is said to be a Gaussian process if all possible finite dimensional vectors $\boldsymbol{y} = (\mathrm{y}_{\mathrm{t}_1}, \mathrm{y}_{\mathrm{t}_2}, \ldots, \mathrm{y}_{\mathrm{t}_\mathrm{n}})^{\mathrm{t}}$, for every collection of time points $\mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_\mathrm{n}$, and every positive integer n, have a multivariate normal distribution.

So far we have only looked at examples of time series where T is discete (and evenly spaces & contiguous), it turns out things get a lot more interesting when we explore the case where T is defined on a *continuous* space (e.g. $\mathbb{R}$ or some subset of $\mathbb{R}$).

Gaussian Process Regression

Parameterizing a Gaussian Process

Imagine we have a Gaussian process defined such that
$\boldsymbol{y} = \{y(t) \; : \; t \in [0,1]\}$,

- We now have an uncountably infinite set of possible t's and $y(t)$s.
- We will only have a (small) finite number of observations $y(t_1), \ldots, y(t_n)$ with which to say something useful about this infinite dimensional process.
- The unconstrained covariance matrix for the observed data can have up to $n(n+1)/2$ unique values*
- Necessary to make some simplifying assumptions:
  - Stationarity
  - Simple parameterization of Σ

Covariance Functions

More on these next week, but for now some common examples

Exponential covariance:

$$\Sigma(y_t, y_{t'}) = \sigma^2 \exp\Big(-|t-t'| \; l \Big)$$

Squared exponential covariance (Gaussian):

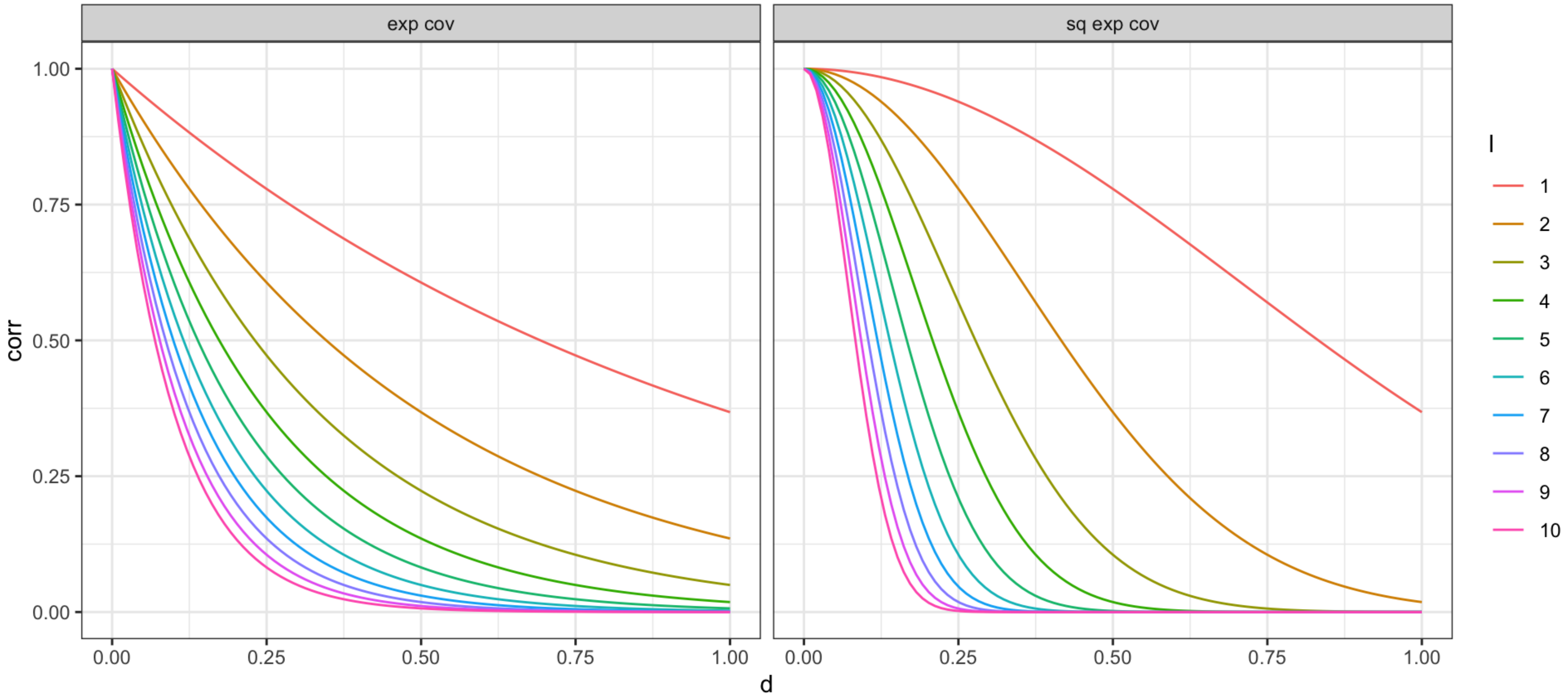
$$\Sigma(y_t, y_{t'}) = \sigma^2 \exp\Big(-(|t-t'| \; l)^2 \Big)$$

Powered exponential covariance ($p \in (0,2]$):

$$\Sigma(y_t, y_{t'}) = \sigma^2 \exp\Big(-(|t-t'| \; l)^p \Big)$$

Covariance Function - Correlation Decay

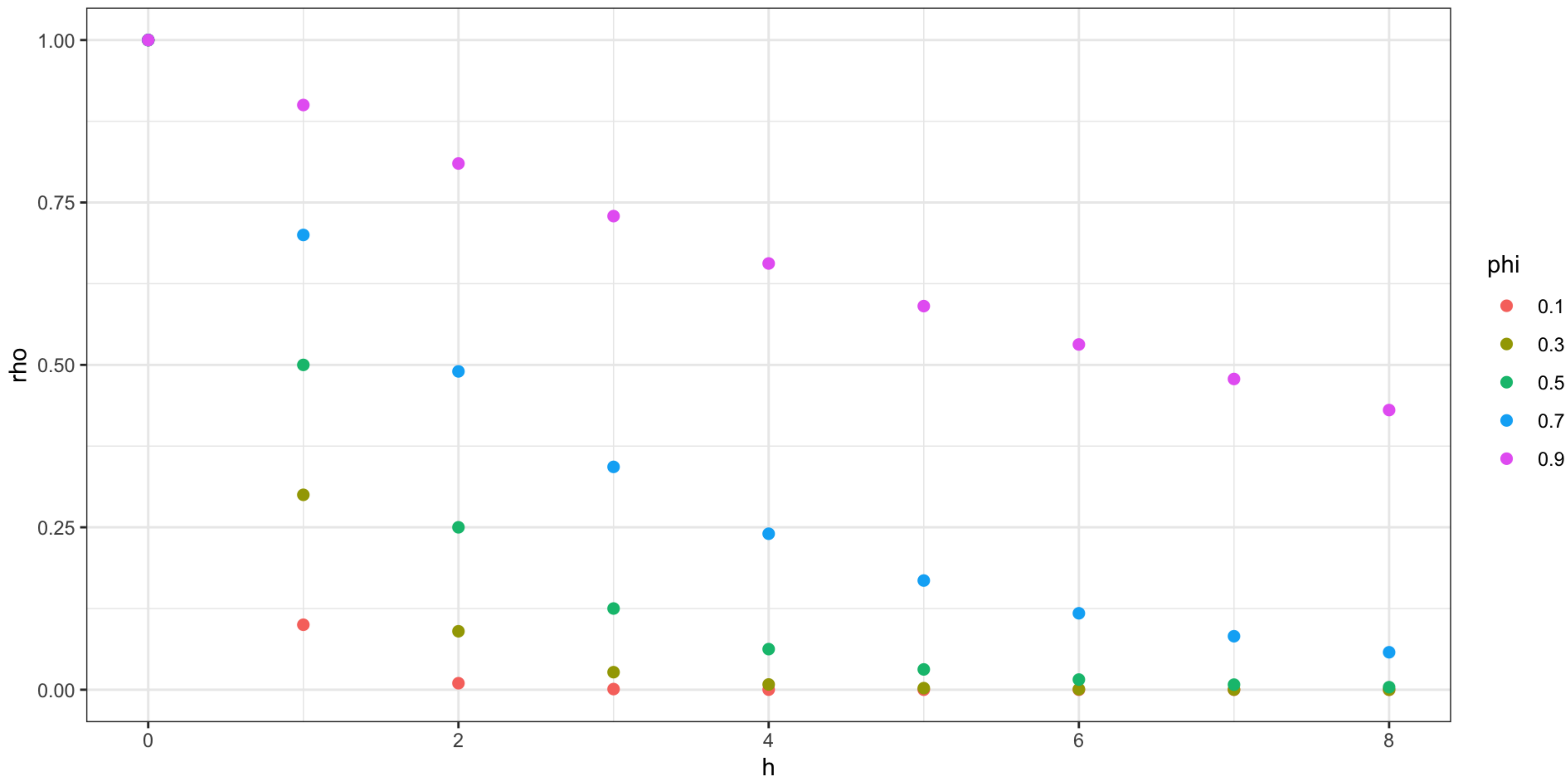
Letting $\sigma^2 = 1$ and trying different values of the length scale l,



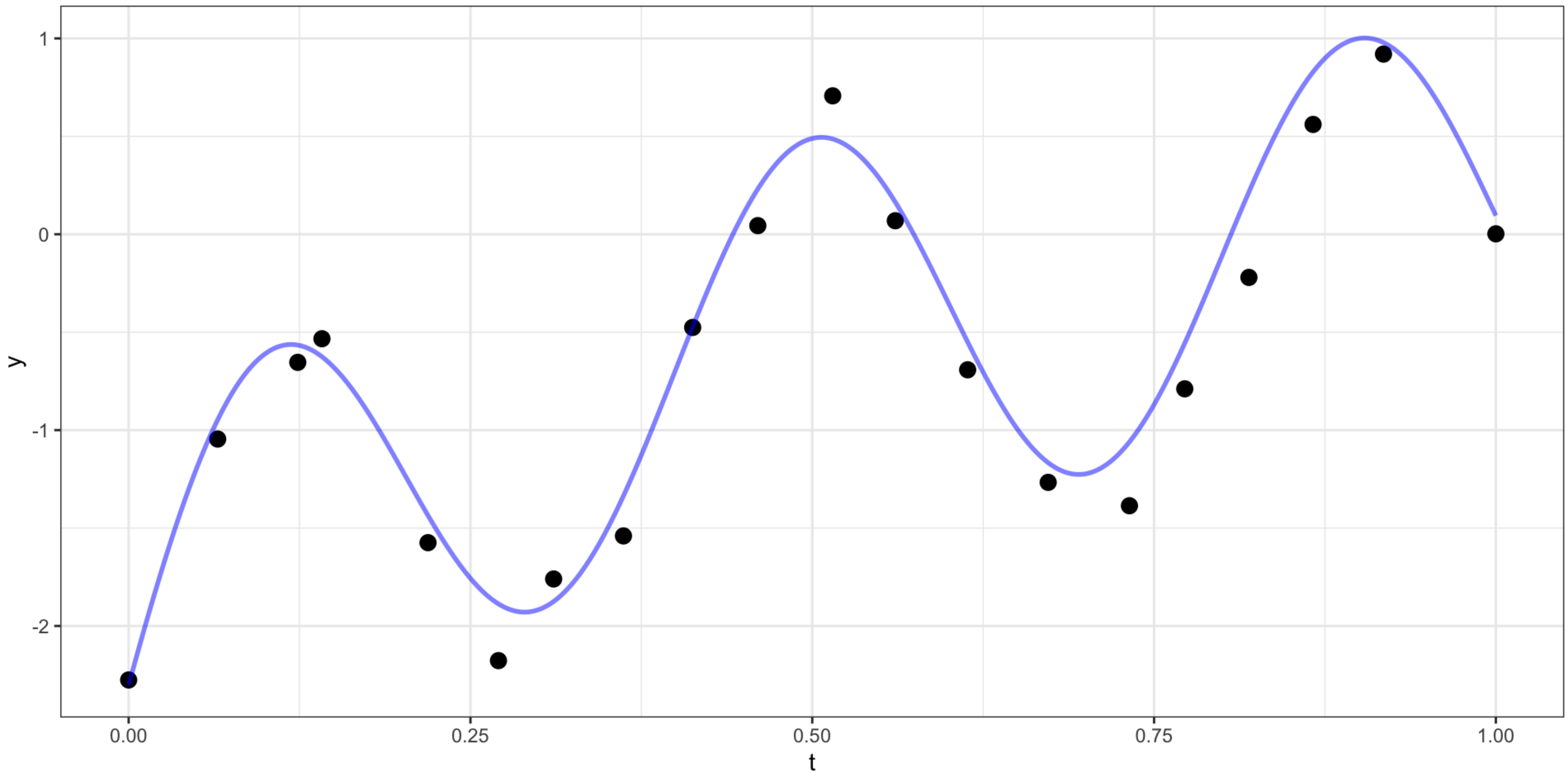
Correlation Decay - AR(1)

Recall that for a stationary AR(1) process: $\gamma(h) = \sigma_w^2 \phi^{|h|}$ and $\rho(h) = \phi^{|h|}$

we can draw a somewhat similar picture about the decay of ρ as a function of distance.


rho
1.00
0.75
0.50
0.25
0.00
0
2
4
6
8
h
phi
0.1
0.3
0.5
0.7
0.9

Example



Prediction

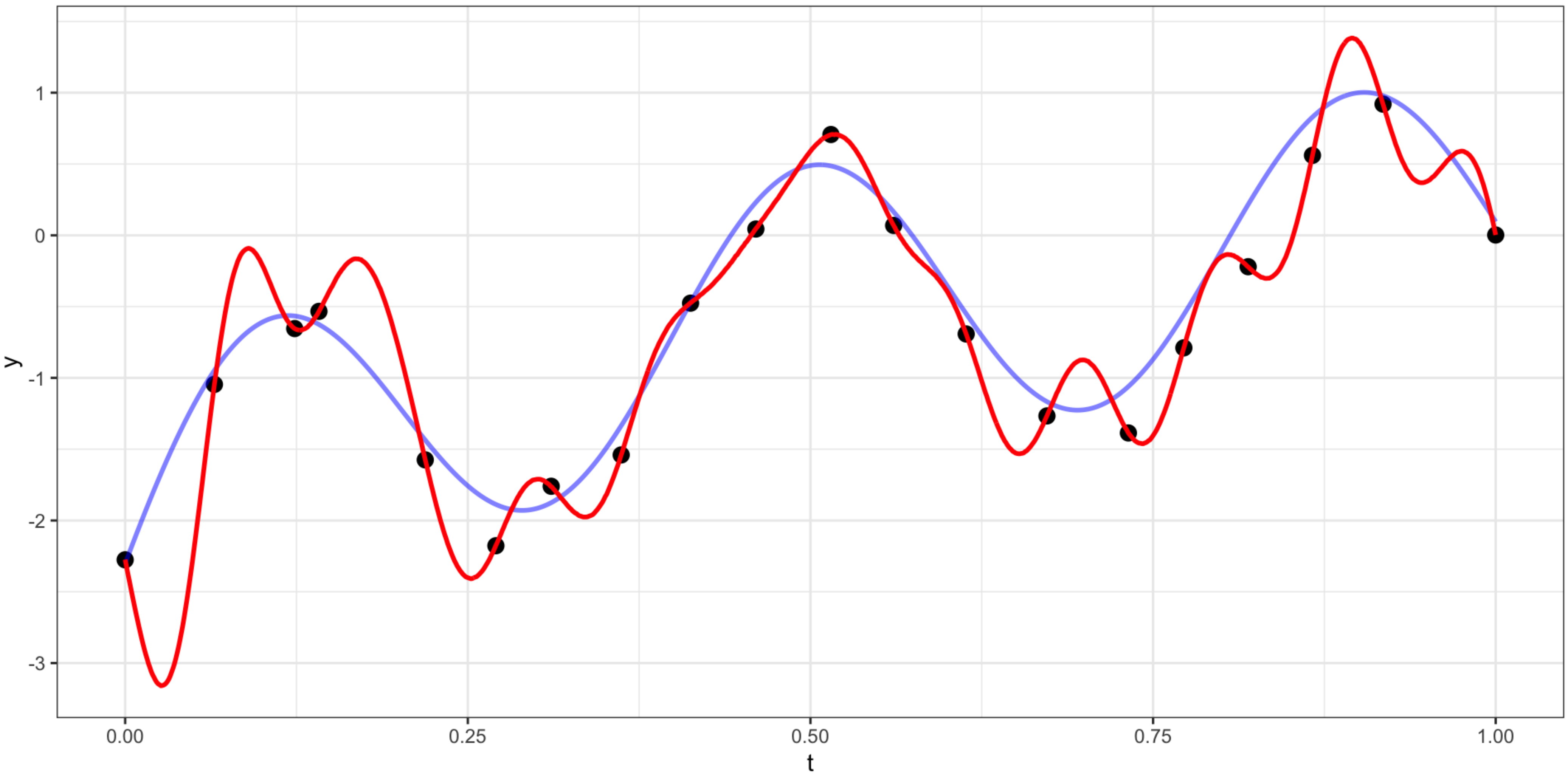
Our example has 20 observations which we would like to use as the basis for predicting $y(t)$ at other values of t (say a regular sequence of values from 0 to 1).

For now lets use a square exponential covariance with $\sigma^2 = 10$ and $l = 15$

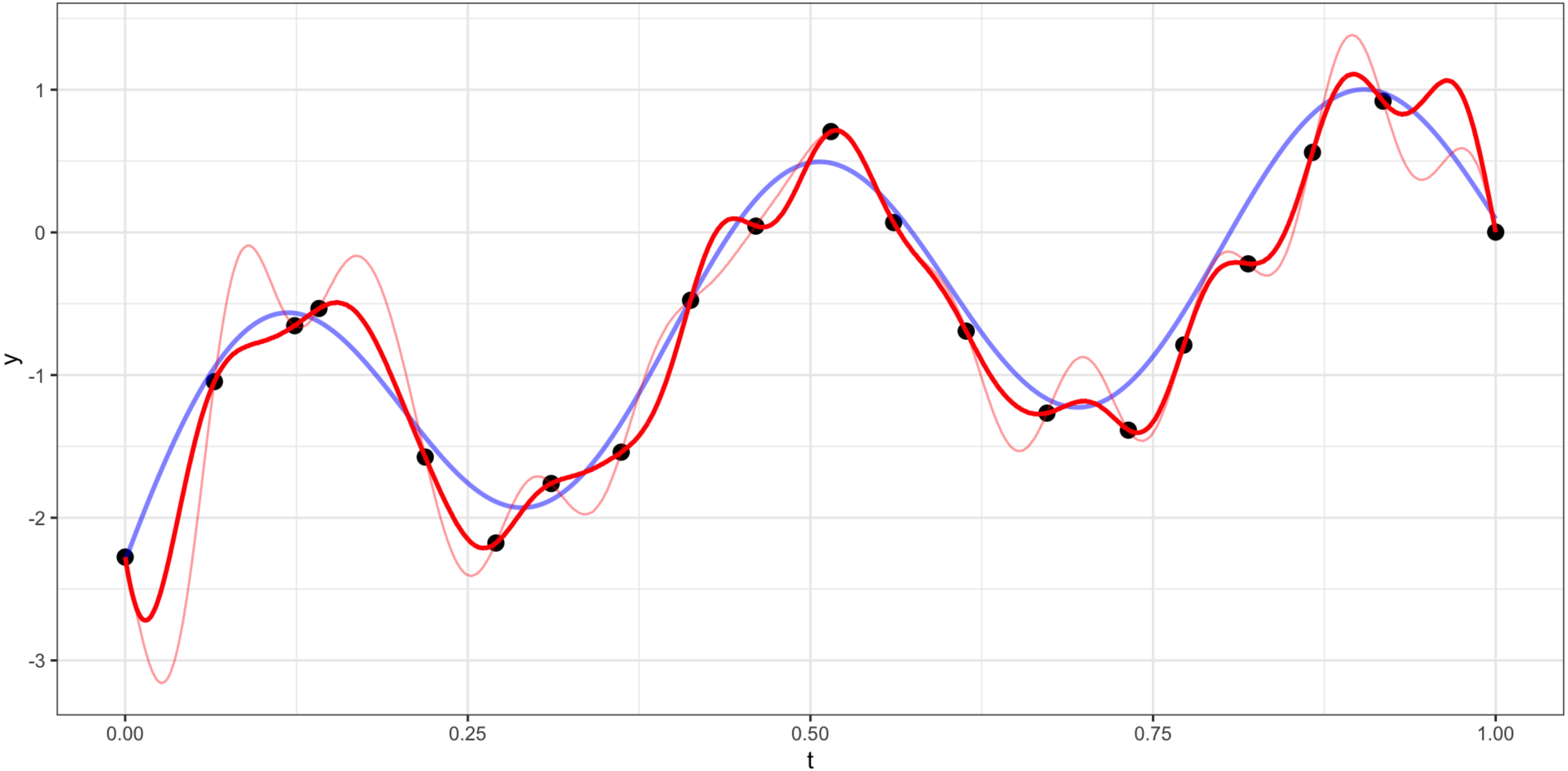
We therefore want to sample from $\boldsymbol{y}_{\text{pred}} | \boldsymbol{y}_{\text{obs}}$.

$$\boldsymbol{y}_{\text{pred}} \mid \boldsymbol{y}_{\text{obs}} = \boldsymbol{y} \sim N(\boldsymbol{\Sigma}_{\text{po}} \, \boldsymbol{\Sigma}_{\text{obs}}^{-1} \, \boldsymbol{y}, \; \boldsymbol{\Sigma}_{\boldsymbol{pred}} - \boldsymbol{\Sigma}_{\text{po}} \, \boldsymbol{\Sigma}_{\text{pred}}^{-1} \, \boldsymbol{\Sigma}_{\text{op}})$$

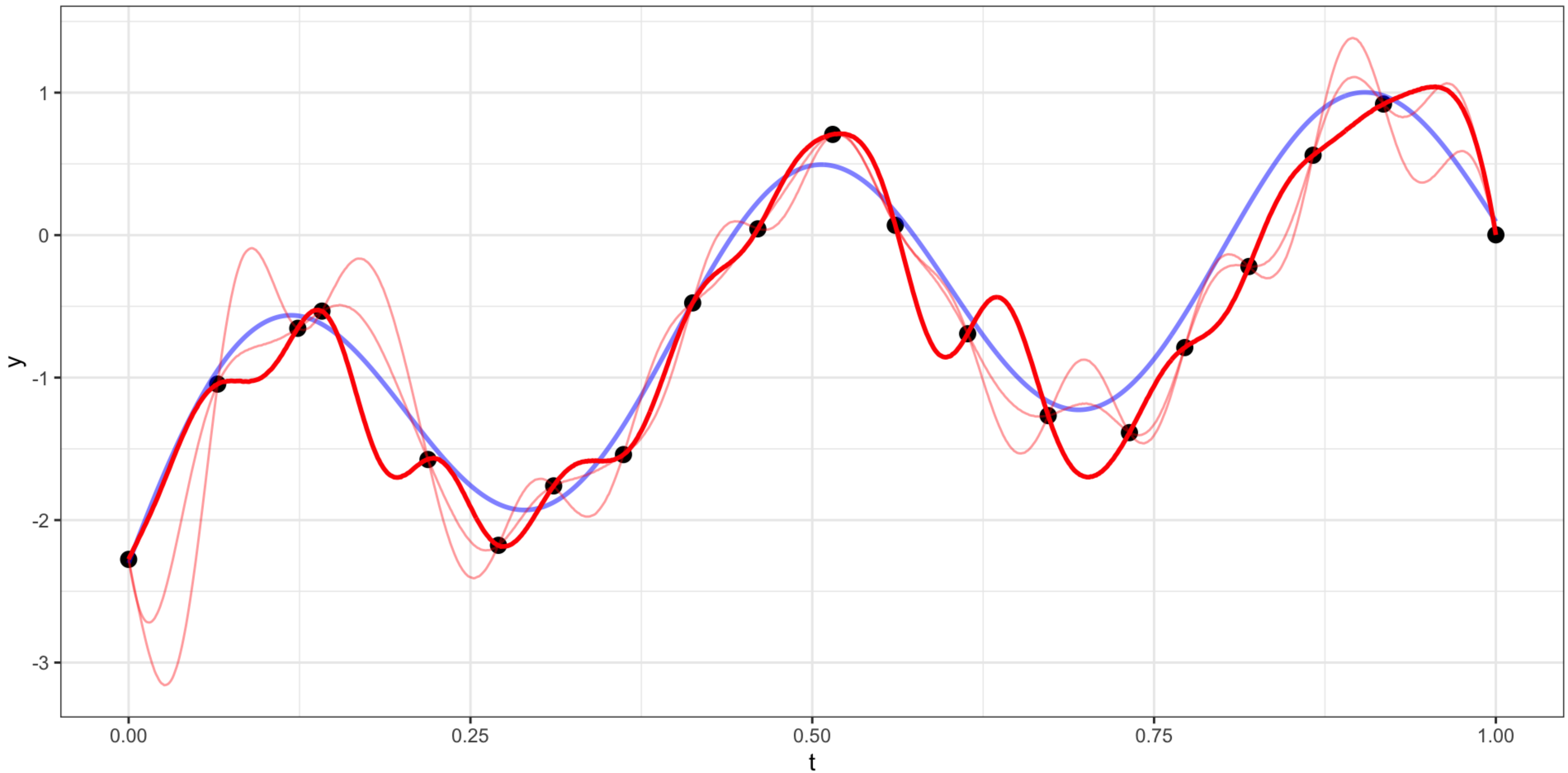
Draw 1



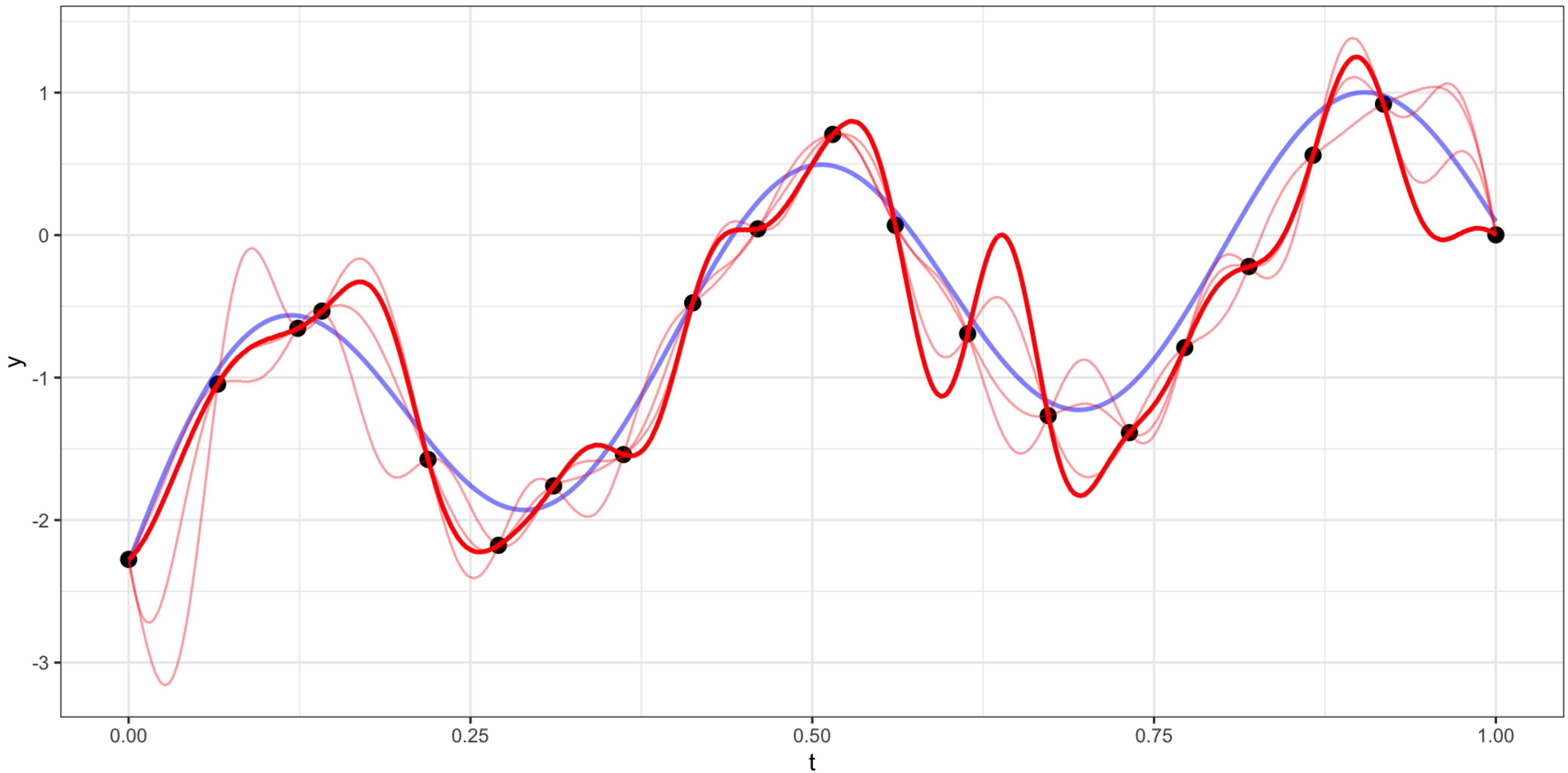
Draw 2



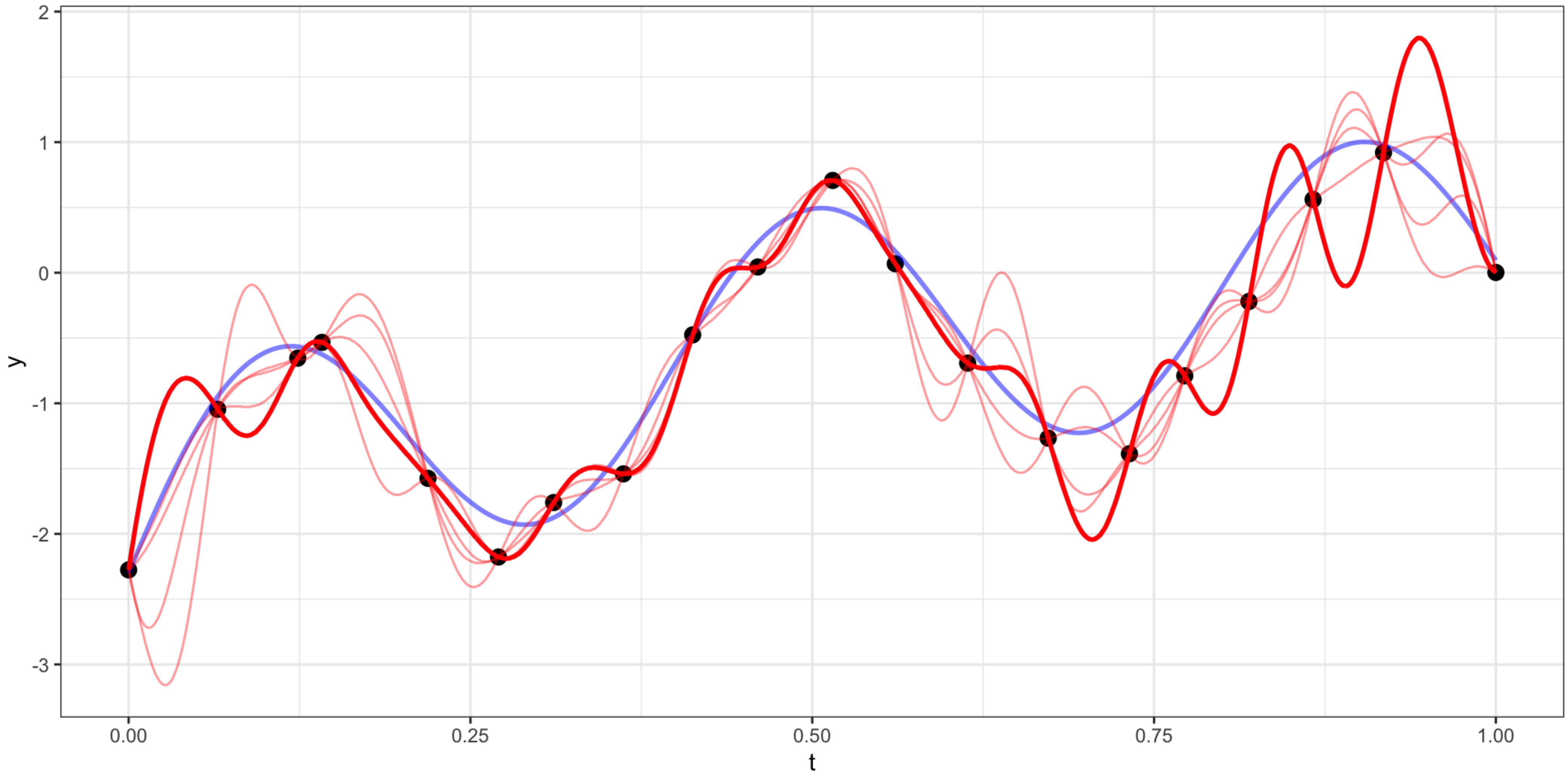
Draw 3



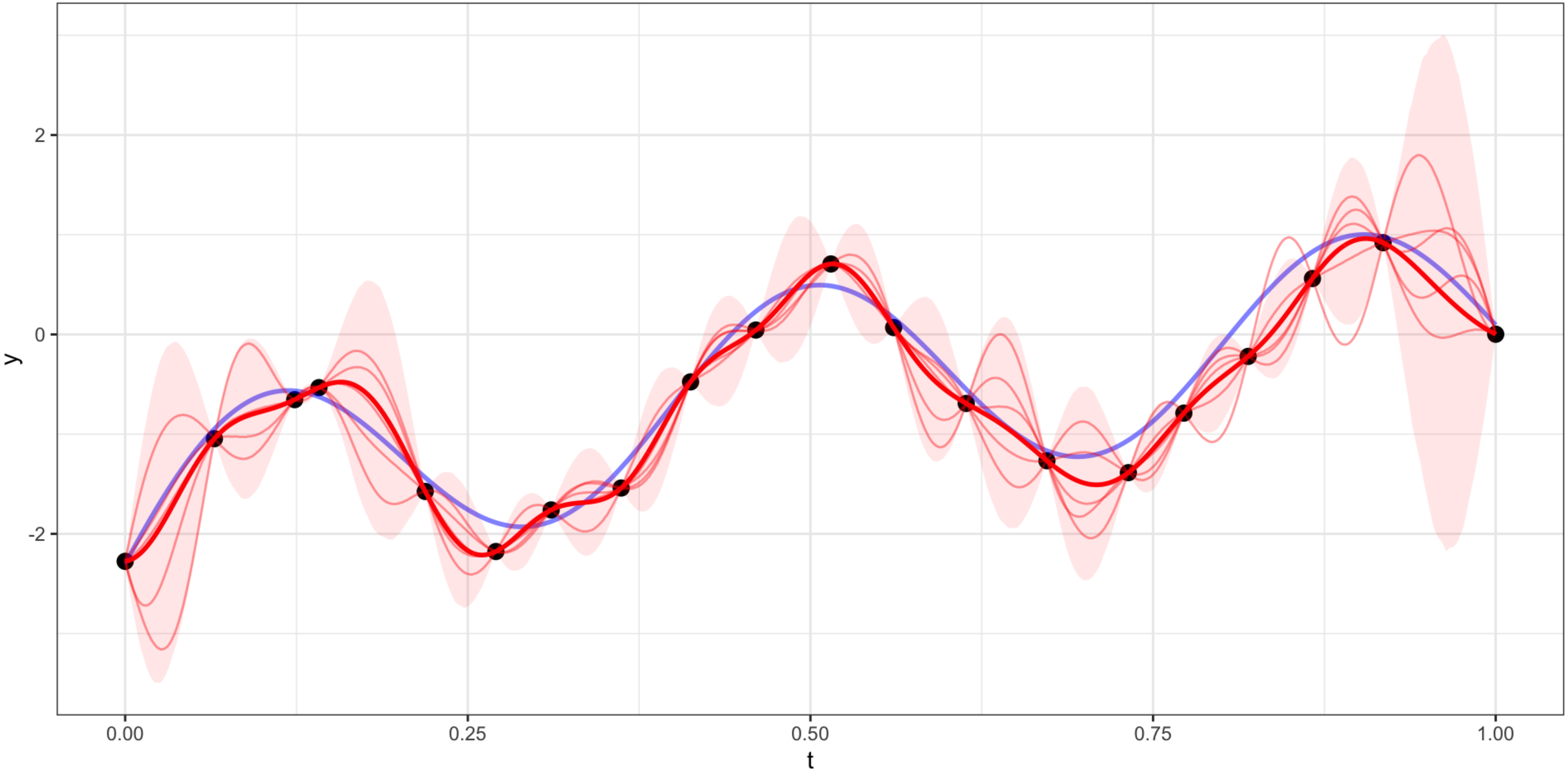
Draw 4



Draw 5

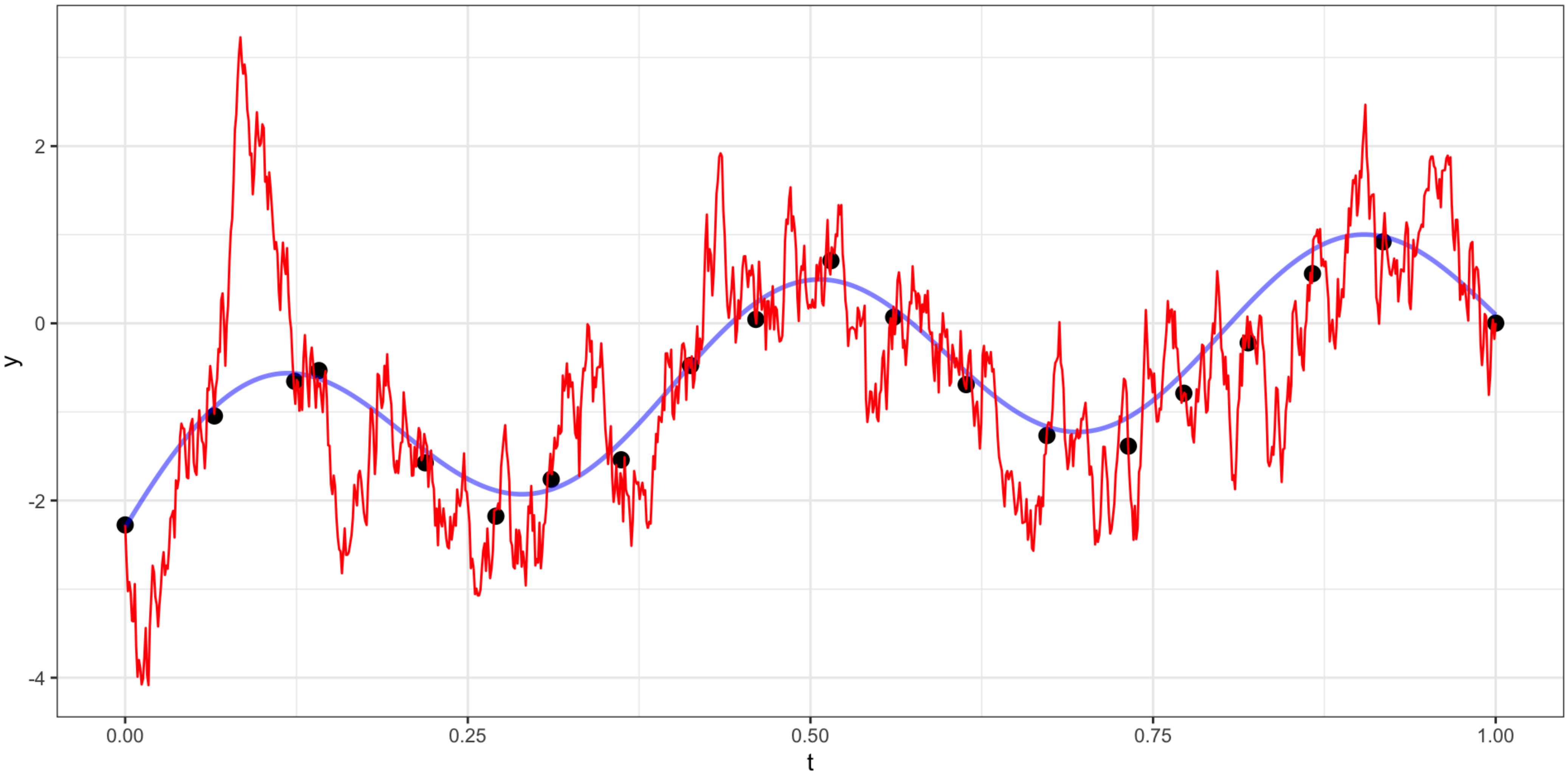


Many draws later

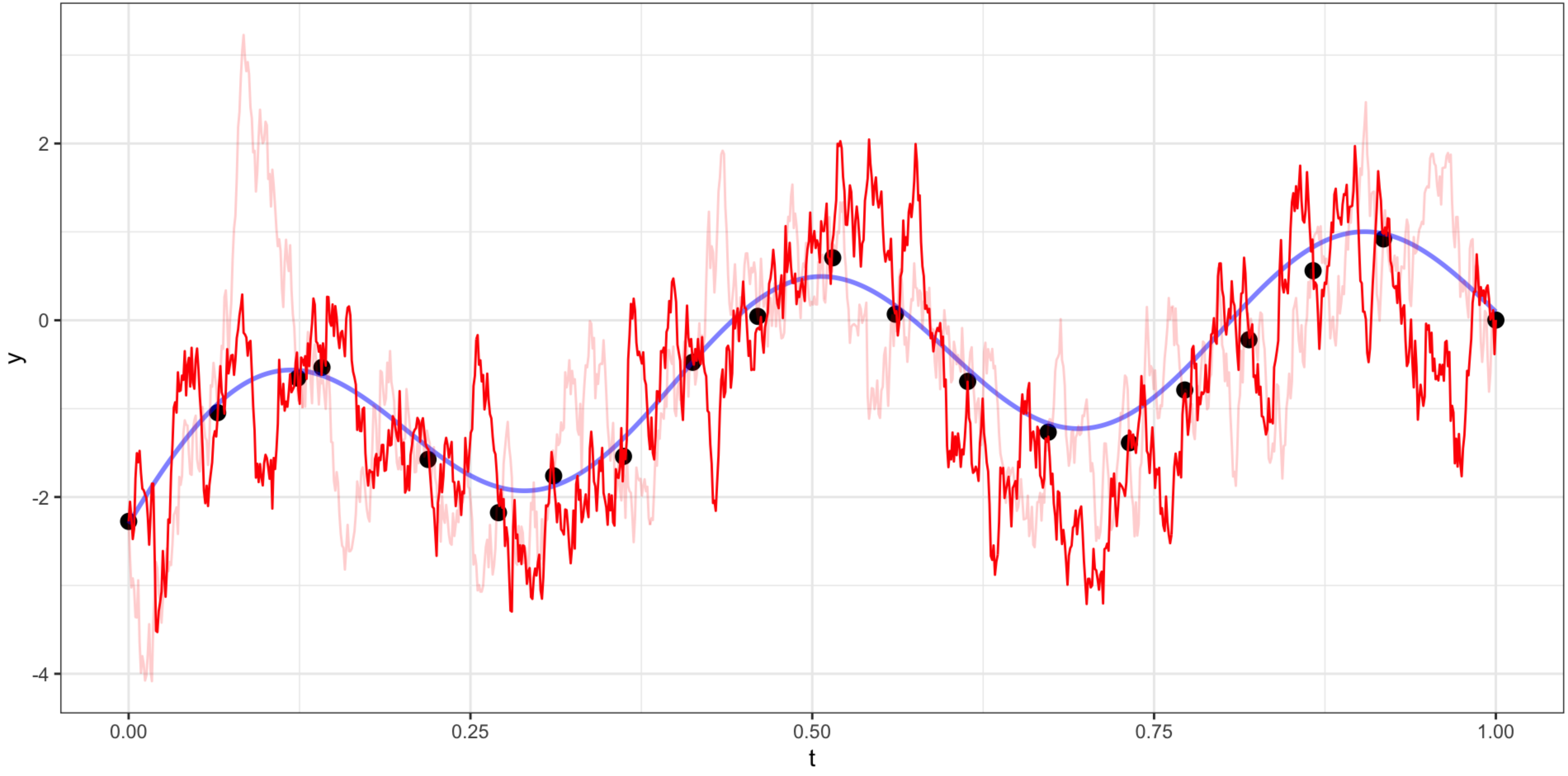


Exponential Covariance

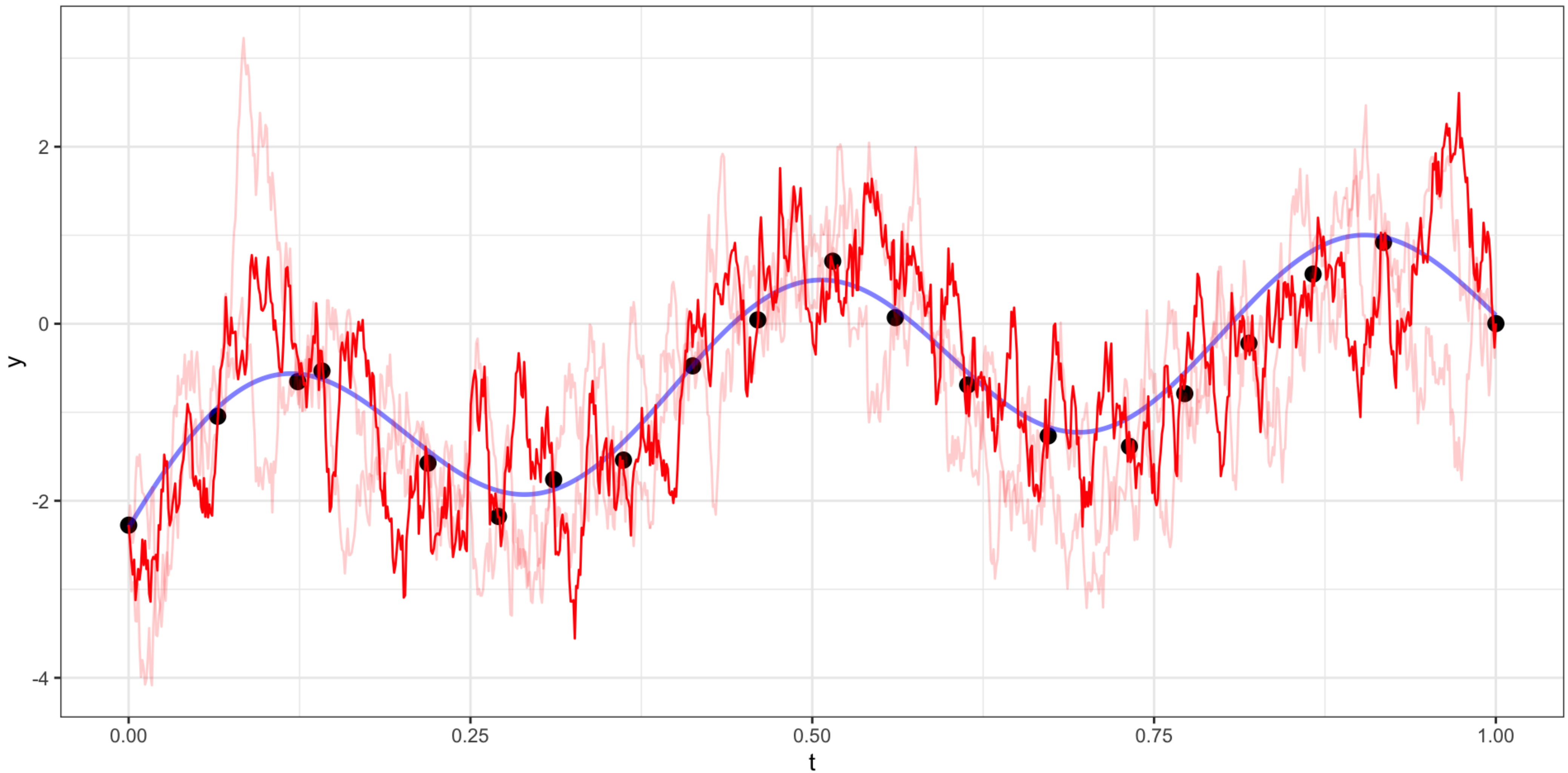
Where $\sigma = 10, l = \sqrt{15}$,



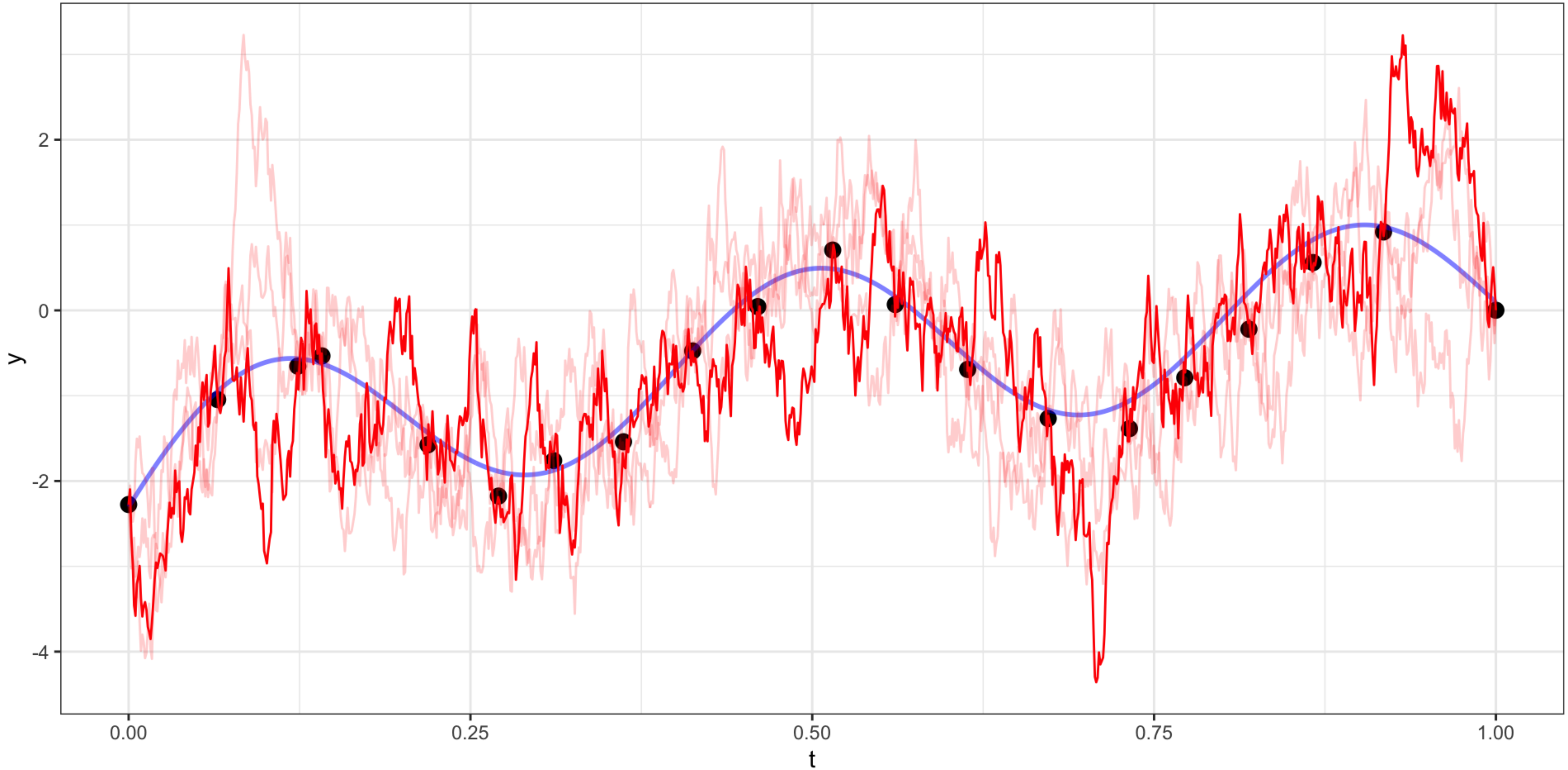
Exponential Covariance - Draw 2



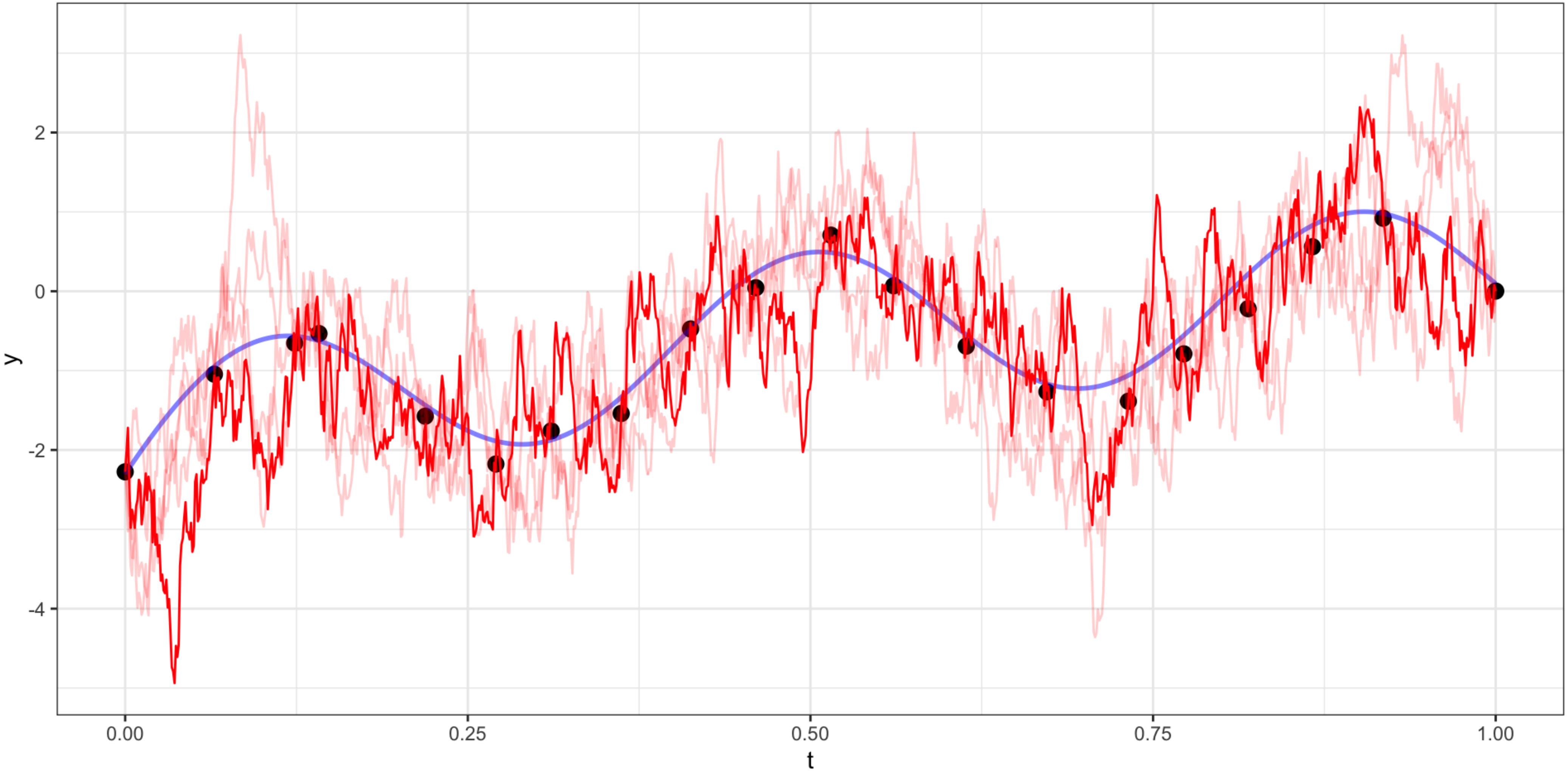
Exponential Covariance - Draw 3



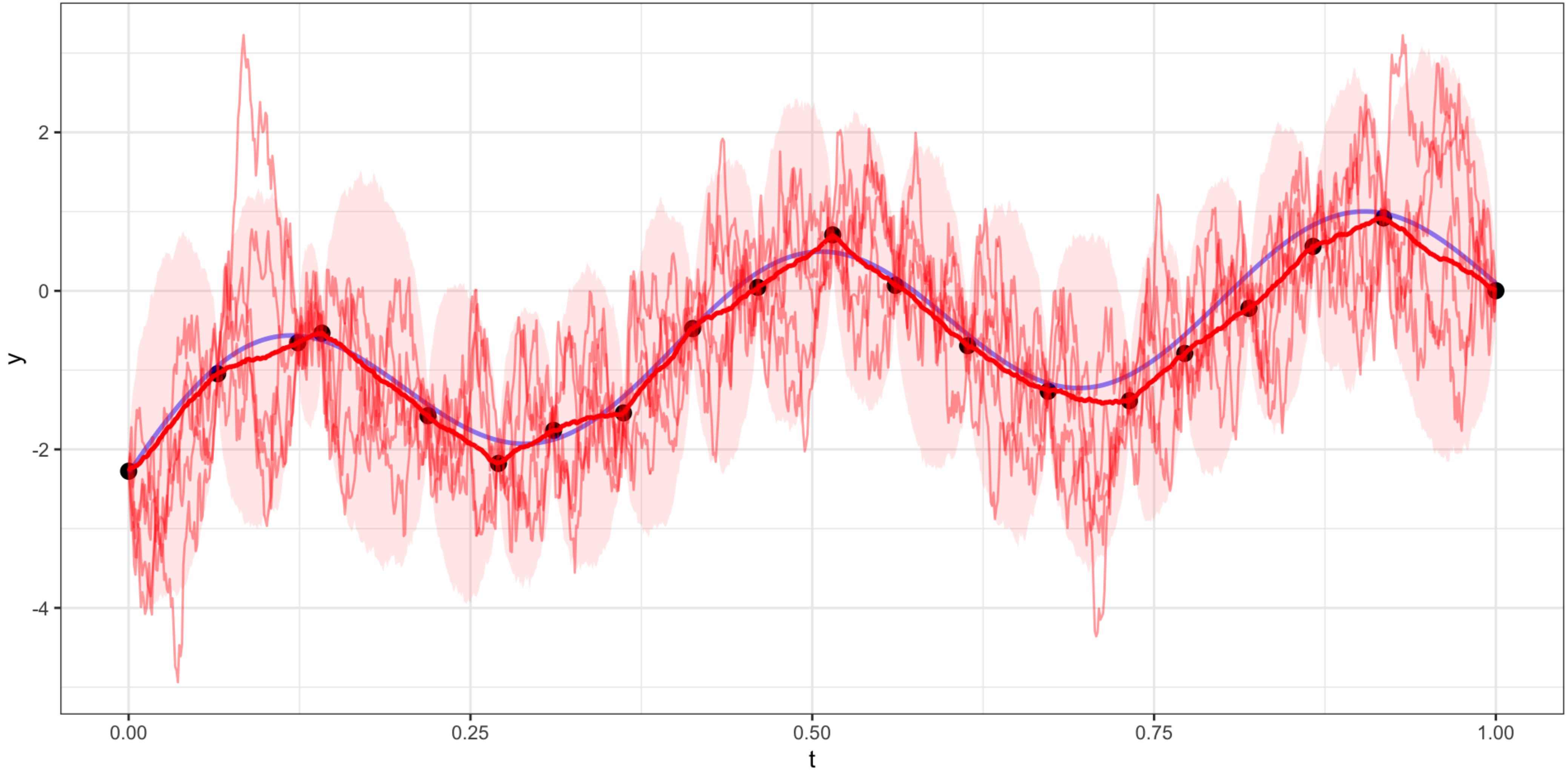
Exponential Covariance - Draw 4



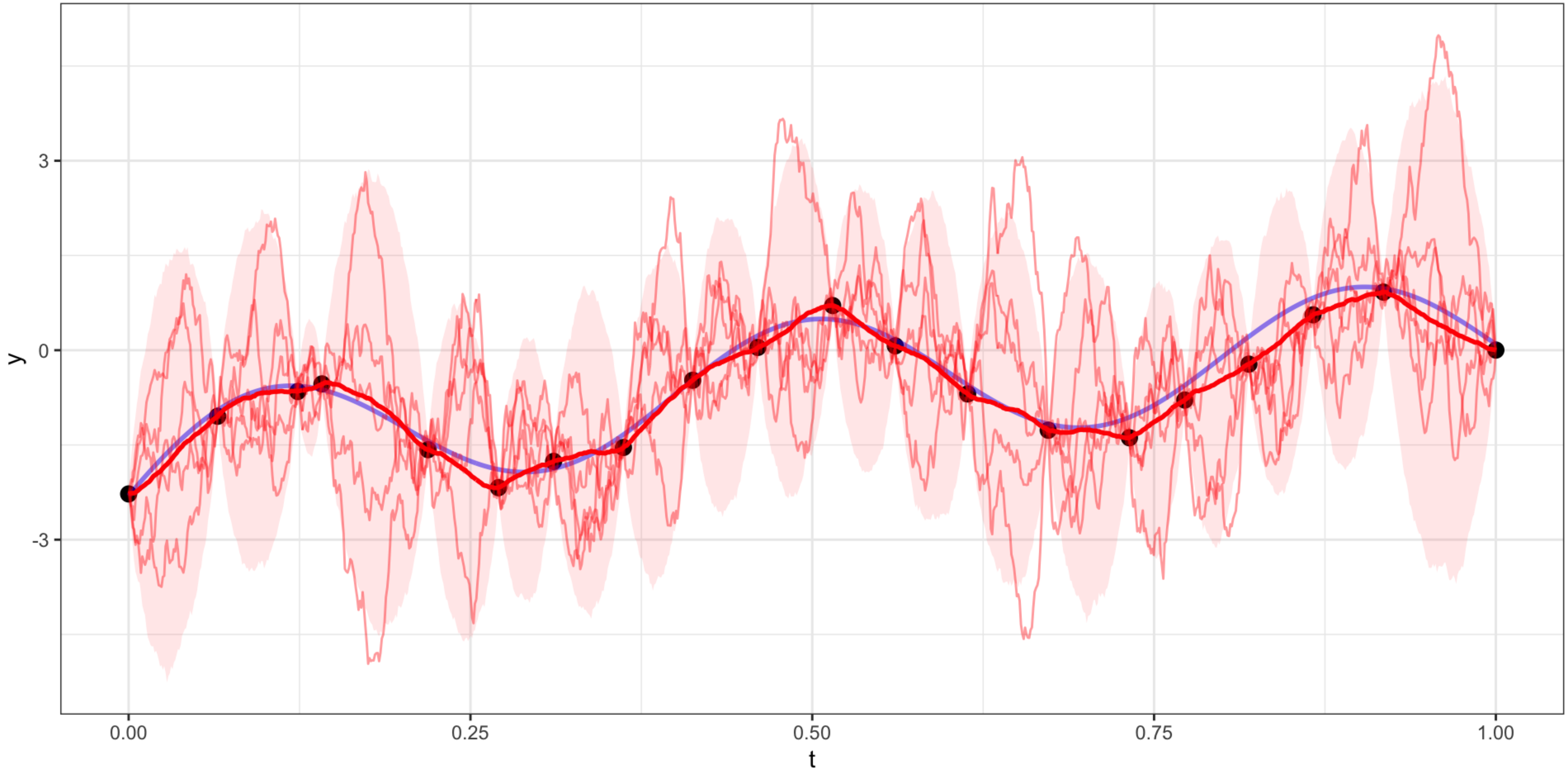
Exponential Covariance - Draw 5



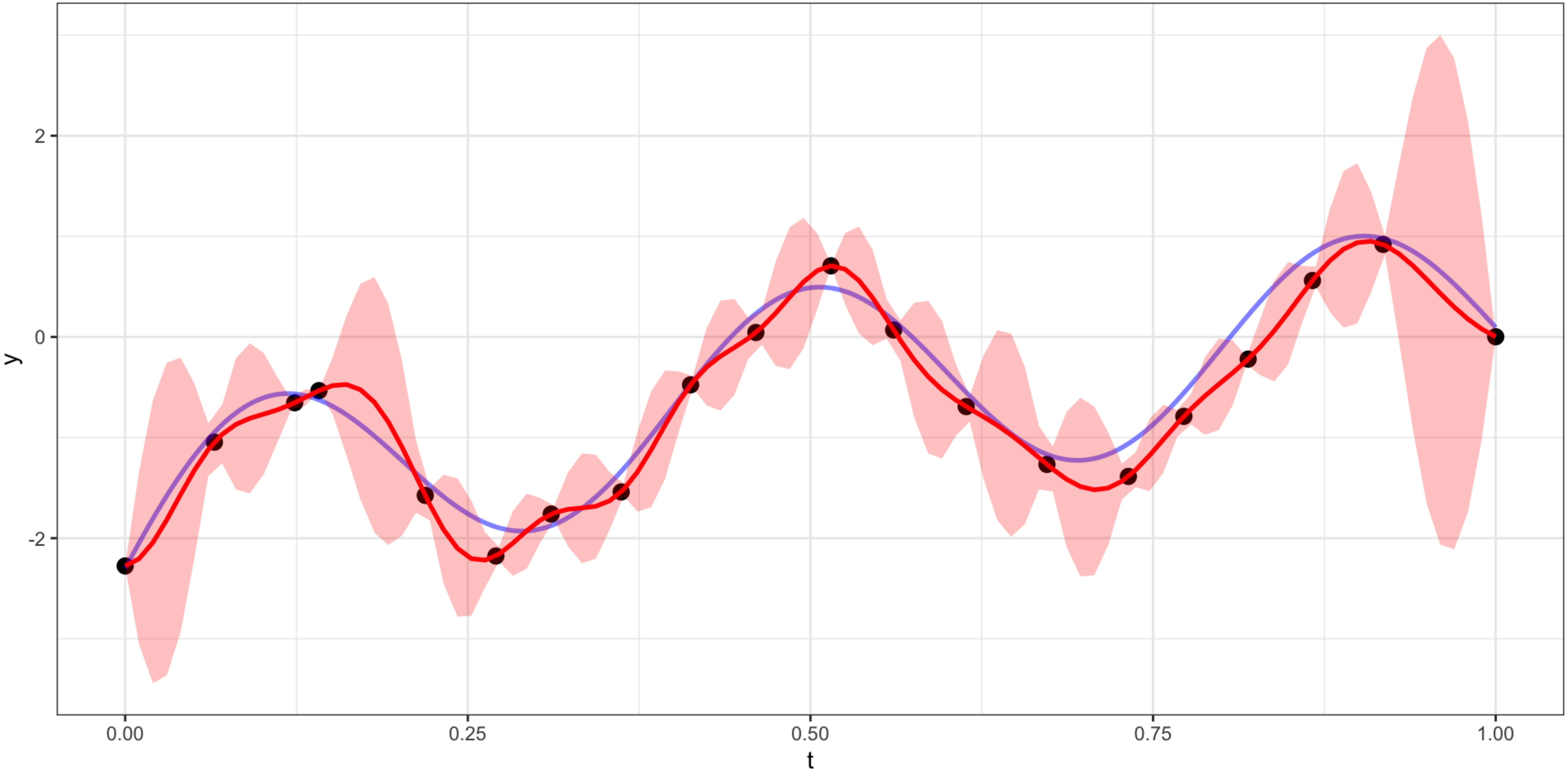
Exponential Covariance - Variability



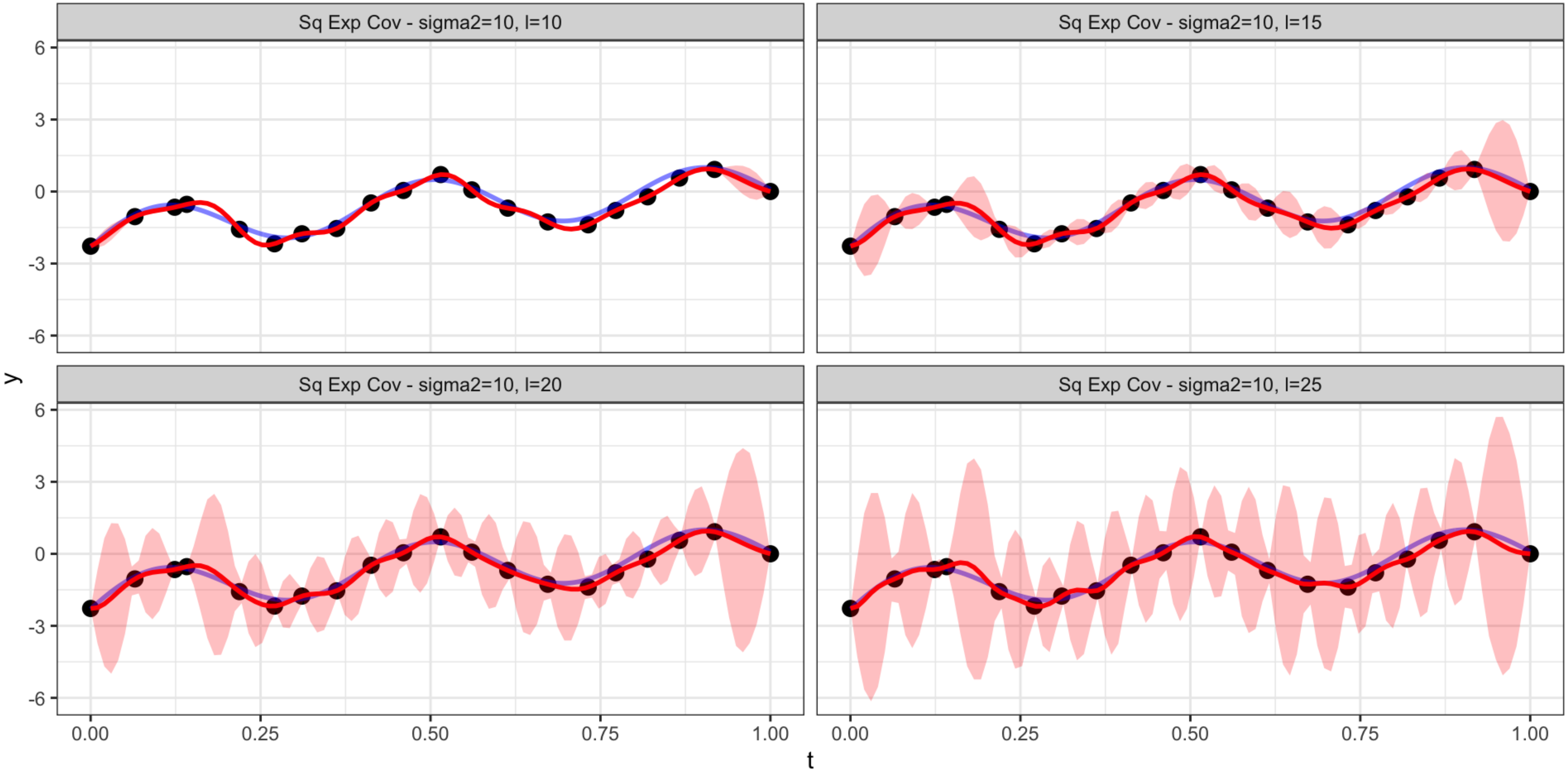
Powered Exponential Covariance ($p = 1.5$)



Back to the square exponential



Changing the range (l)



Effective Range

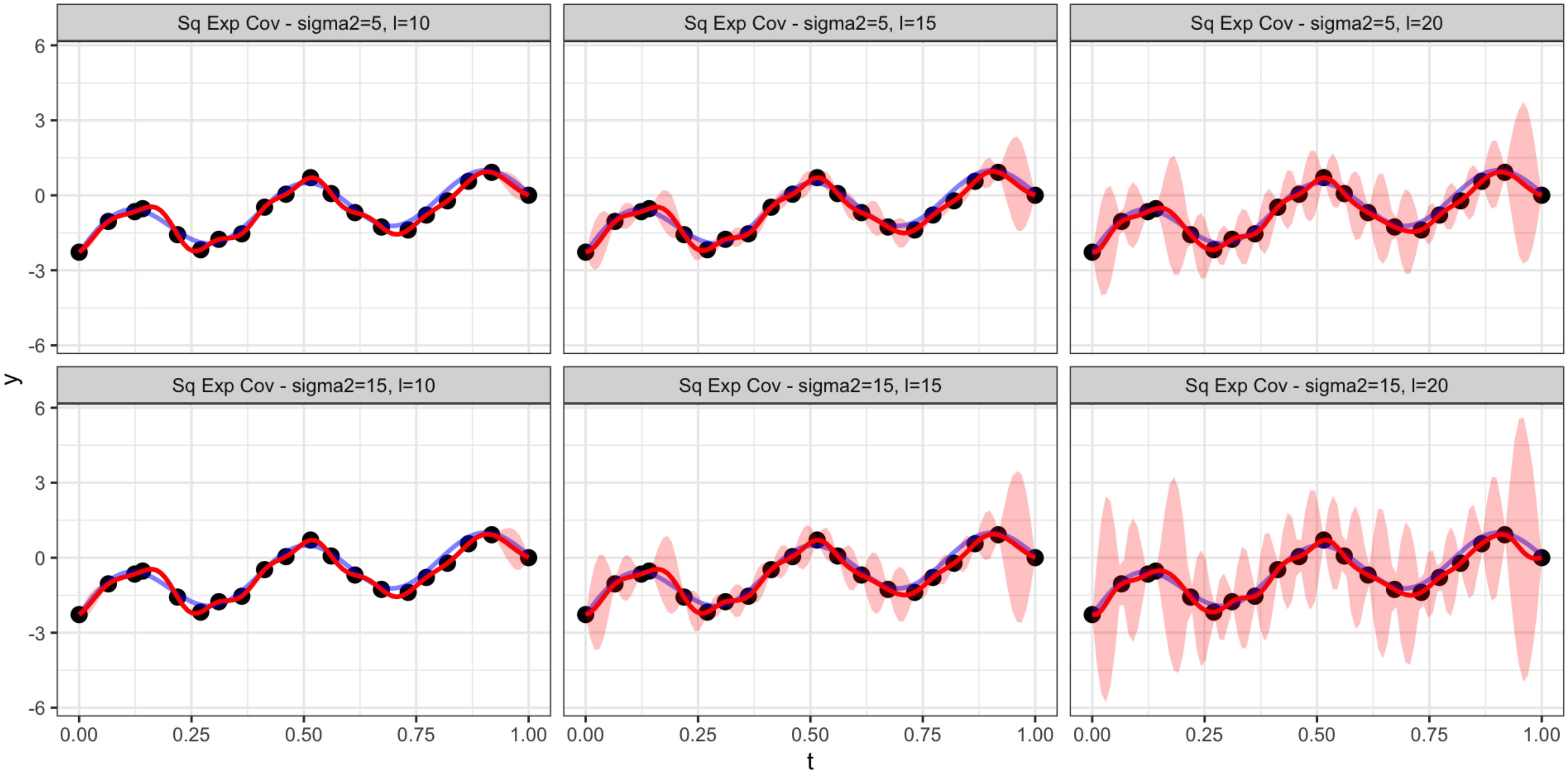
For the square exponential covariance

$$\mathrm{Cov}(d) = \sigma^2 \exp\left(-(l \cdot d)^2\right)$$
$$\mathrm{Corr}(d) = \exp\left(-(l \cdot d)^2\right)$$

we would like to know, for a given value of l, beyond what distance apart must observations be to have a correlation less than 0.05?

$$\begin{aligned}
\exp\left(-(l \cdot d)^2\right) &< 0.05 \\
-(l \cdot d)^2 &< \log 0.05 \\
l \cdot d &< \sqrt{3} \\
d &< \sqrt{3}/l
\end{aligned}$$

Changing the scale (σ^2)



Fitting w/ BRMS

```r
library(brms)
gp = brm(y ~ gp(t), data=d, cores=4, refresh=0)
```

```r
summary(gp)
```

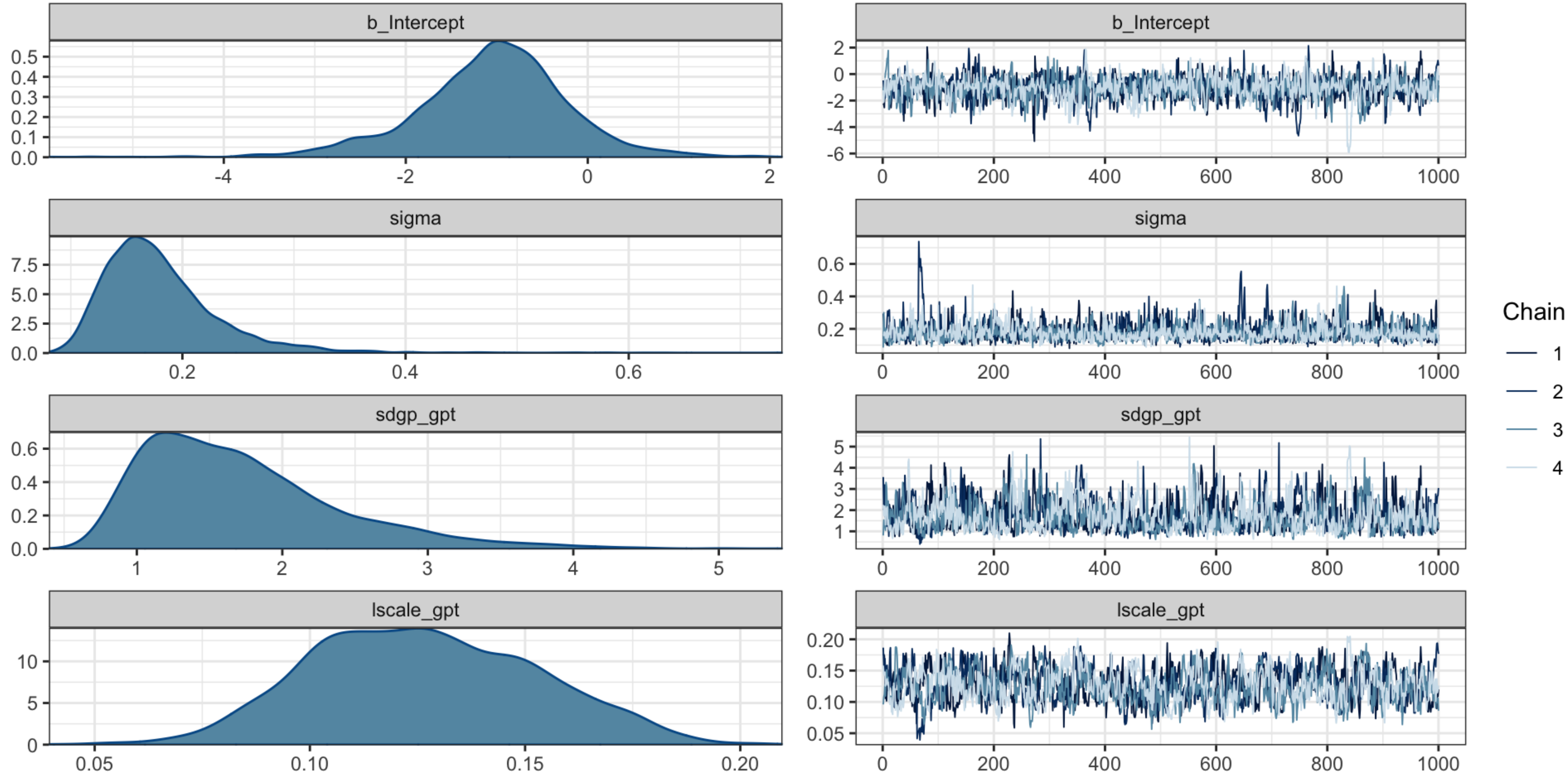
```
 Family: gaussian 
  Links: mu = identity; sigma = identity 
Formula: y ~ gp(t) 
   Data: d (Number of observations: 20) 
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Gaussian Process Terms: 
            Estimate Est.Error l-95% CI u-95% CI
sdgp(gpt)       1.72      0.69     0.81     3.47
lscale(gpt)     0.13      0.03     0.08     0.18
            Rhat Bulk_ESS Tail_ESS
sdgp(gpt)   1.00      536     1211
lscale(gpt) 1.00      453      976

Population-Level Effects:
```

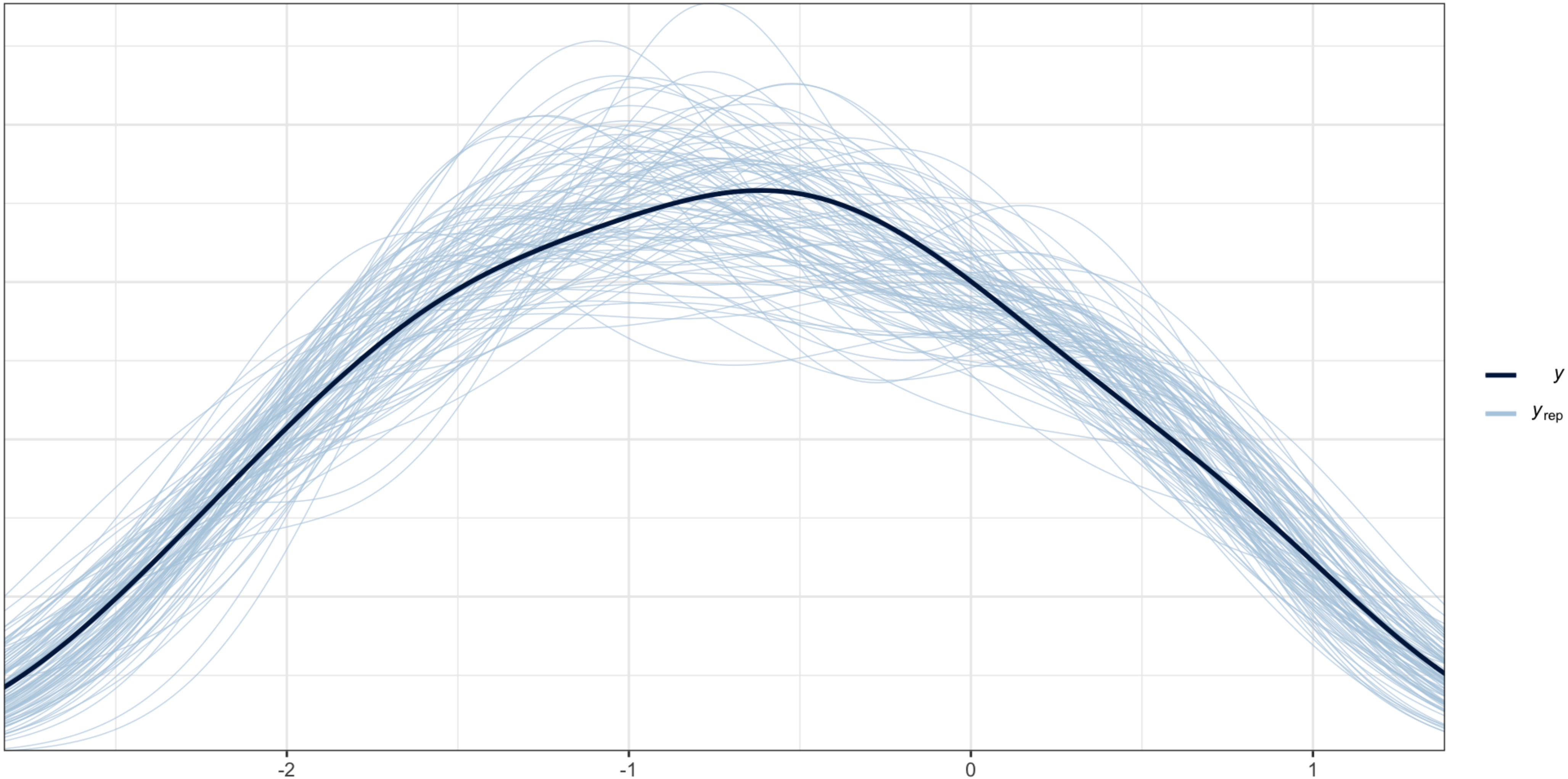
Trace plots

```r
plot(gp)
```

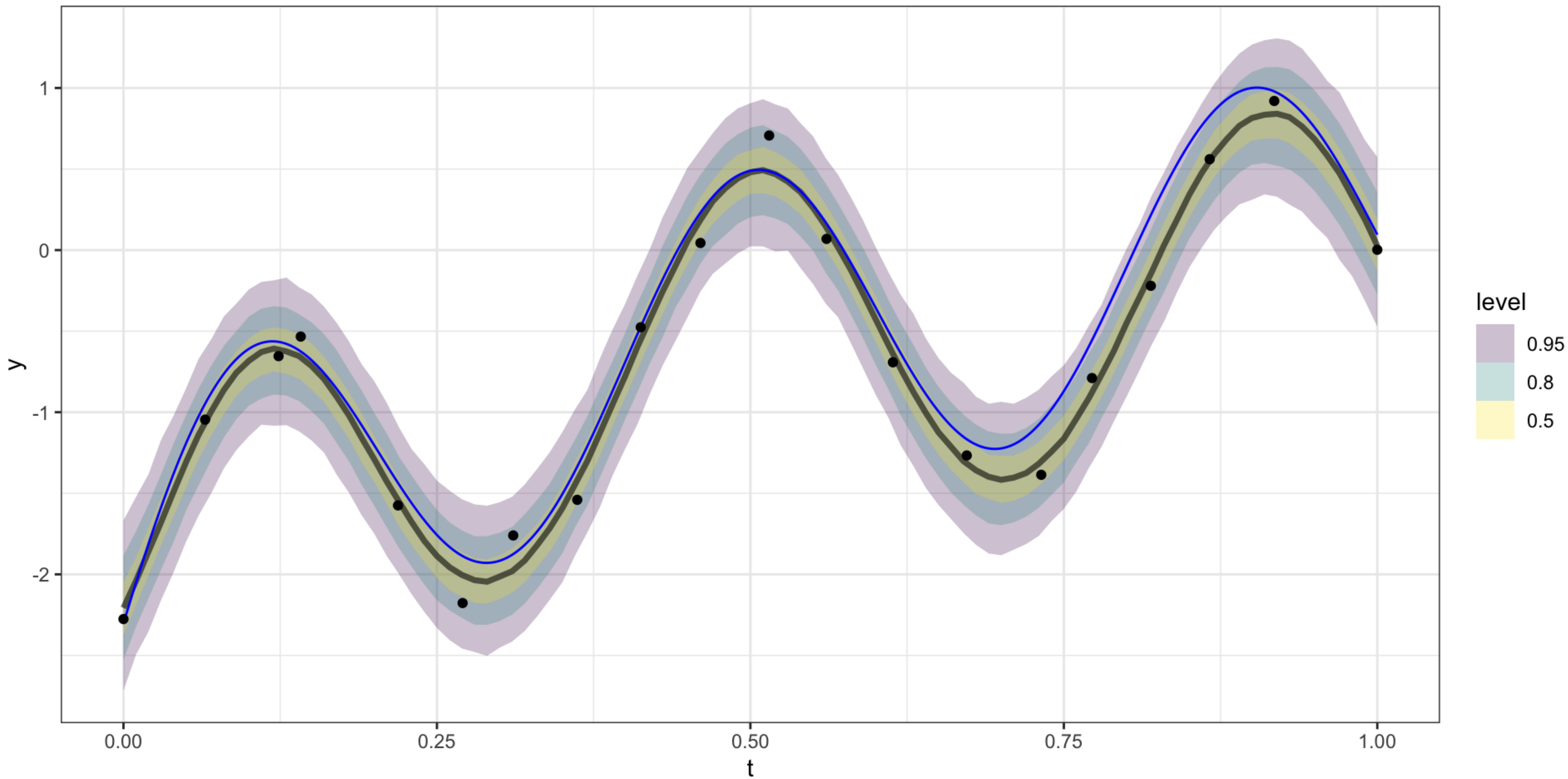


PP Checks

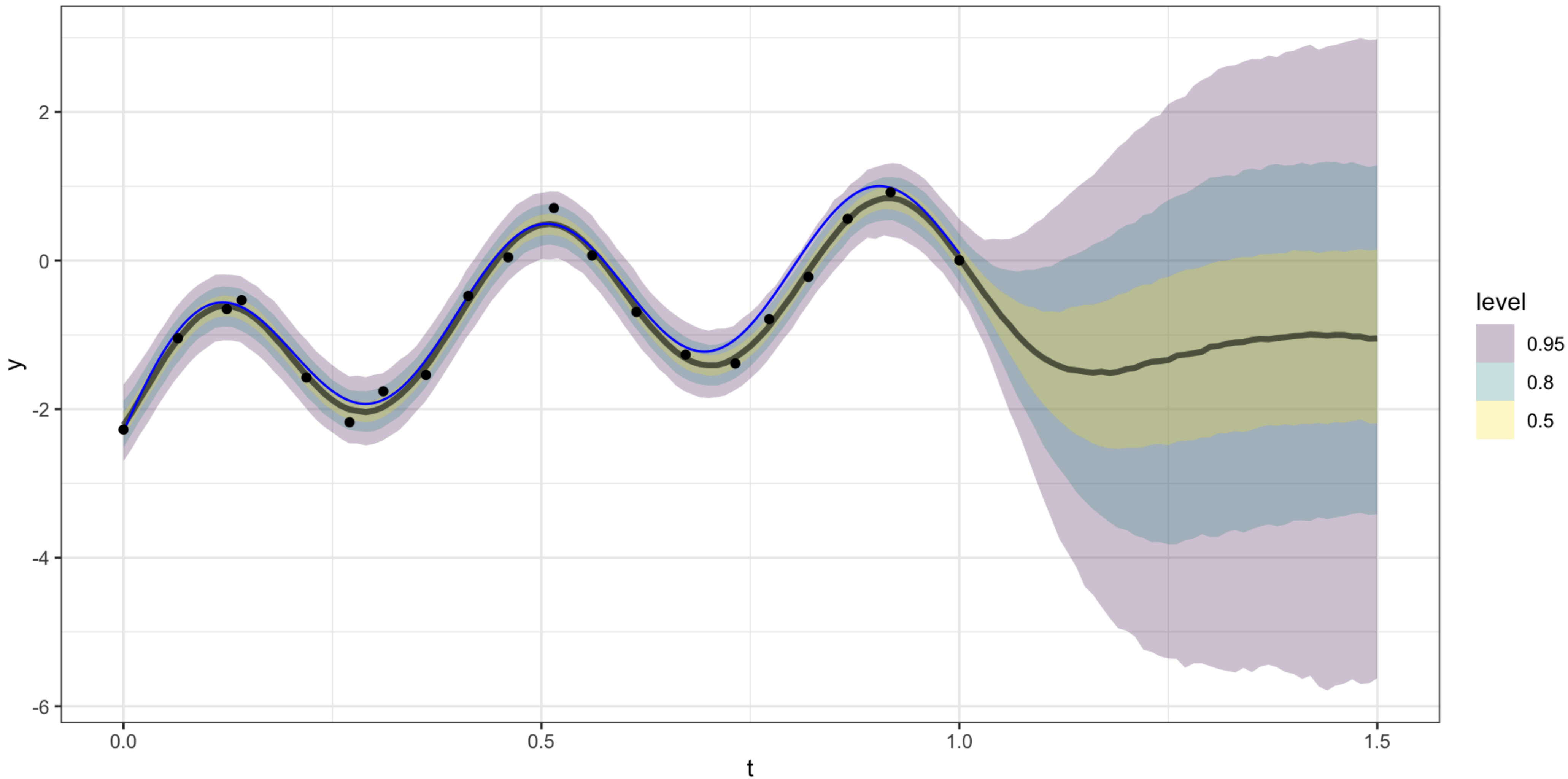
```
pp_check(gp, ndraws = 100)
```



Model predictions



Forecasting



Stan code

```r
gp %>%
  brms::stancode()
```

```
// generated with brms 2.18.0
functions {
  /* compute a latent Gaussian process
   * Args:
   *   x: array of continuous predictor values
   *   sdgp: marginal SD parameter
   *   lscale: length-scale parameter
   *   zgp: vector of independent standard normal variables
   * Returns:
   *   a vector to be added to the linear predictor
   */
  vector gp(data vector[] x, real sdgp, vector lscale, vector zgp) {
    int Dls = rows(lscale);
    int N = size(x);
    matrix[N, N] cov;
    if (Dls == 1) {
```