Gaussian Process Models Part 2

Lecture 14

Dr. Colin Rundel

EDA and GPs

Variogram

When fitting a Gaussian process model, it is often difficult to fit the covariance parameters (they are often correlated and hard to identify).

Today we will discuss some EDA approaches for getting a sense of the values for the scale, lengthscale / effective range and nugget parameters.

From the spatial modeling literature the typical approach is to examine an *empirical variogram*, first we will define the *theoretical variogram* and its connection to the covariance.

Variogram & semivariogram

Variogram

$$2\gamma(t_i, t_j) = \text{Var}\left(y(t_i) - y(t_j)\right)$$

SemiVariogram

$$\gamma(t_i, t_j) = \frac{1}{2}\text{Var}\left(y(t_i) - y(t_j)\right)$$

Properties of the Variogram / Semivariogram

- both are non-negative

$$\gamma(t_i, t_j) \geq 0$$

- both are equal to 0 at distance 0

$$\gamma(t_i, t_i) = 0$$

- both are symmetric -

$$\gamma(t_i, t_j) = \gamma(t_j, t_i)$$

- if observations are independent

$$2\gamma(t_i, t_j) = \text{Var}\left(y(t_i)\right) + \text{Var}\left(y(t_j)\right) \qquad \text{for all } i \neq j$$

- if the process *is not* stationary

$$2\gamma(t_i, t_j) = \text{Var}\left(y(t_i)\right) + \text{Var}\left(y(t_j)\right) - 2\,\text{Cov}\left(y(t_i), y(t_j)\right)$$

- if the process *is* stationary

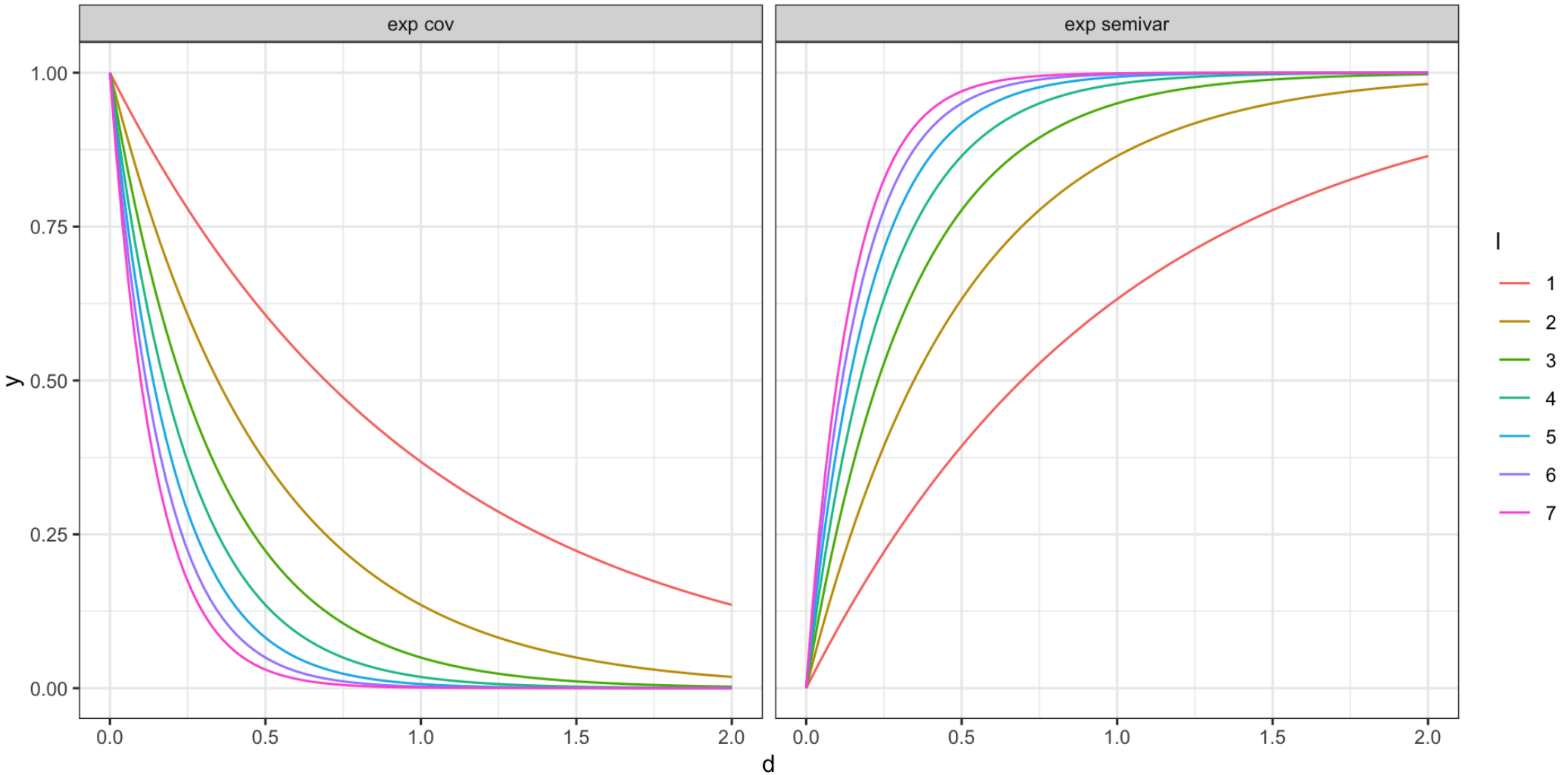
$$2\,\gamma(t_i, t_j) = 2\,\text{Var}\left(y(t_i)\right) - 2\,\text{Cov}\left(y(t_i), y(t_j)\right)$$

Connection to Covariance

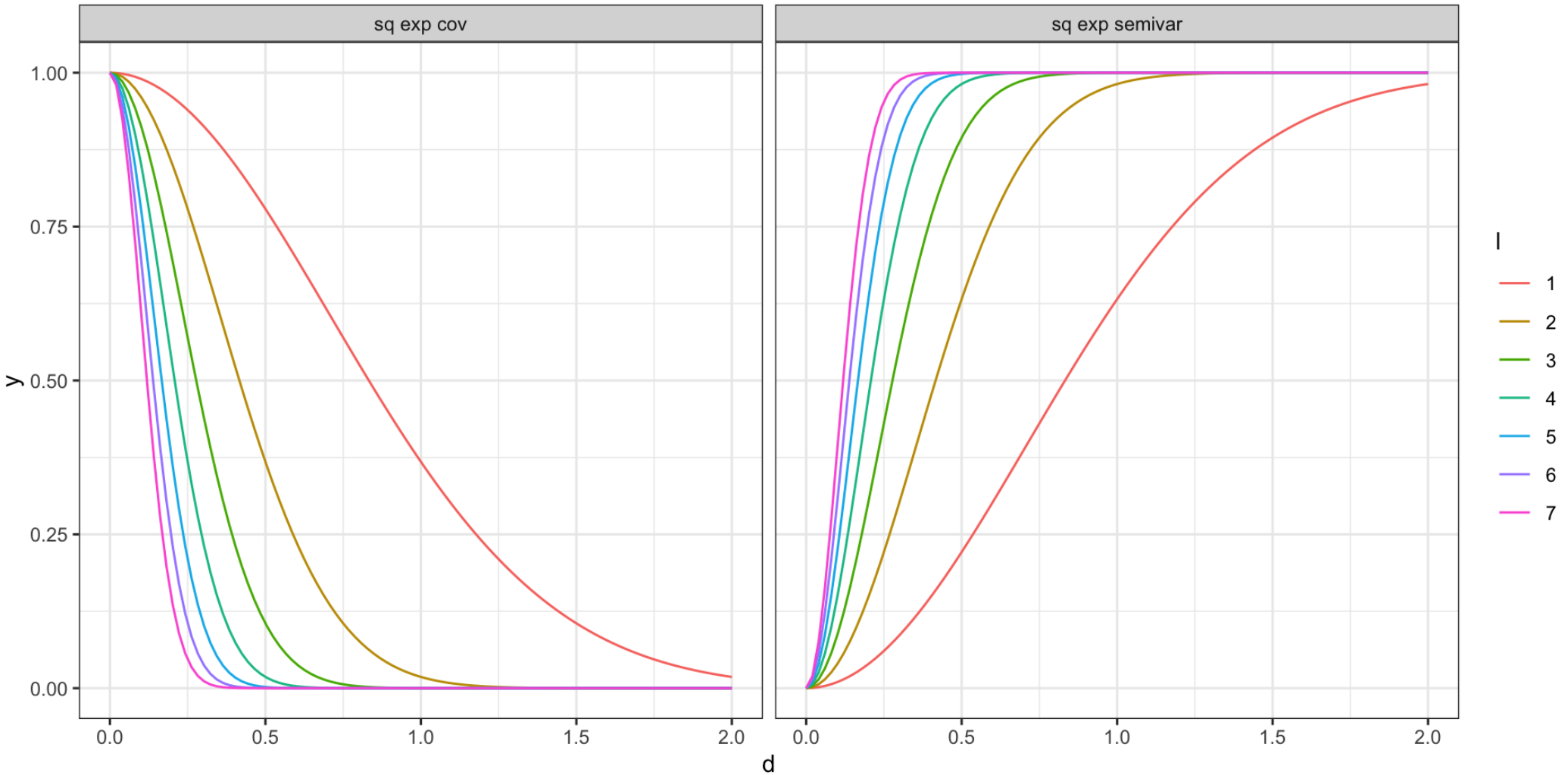
Assuming a squared exponential covariance structure and stationarity,

$$
\begin{aligned}
2\gamma(t_i, t_j) &= 2\text{Var}\Big(y(t_i)\Big) - 2\,\text{Cov}\Big(y(t_i), y(t_j)\Big) \\
\gamma(t_i, t_j) &= \text{Var}\Big(y(t_i)\Big) - \text{Cov}\Big(y(t_i), y(t_j)\Big) \\
&= \sigma^2 - \sigma^2 \exp\Big(-(|t_i - t_j|\,l)^2\Big)
\end{aligned}
$$

Covariance vs Semivariogram - Exponential



Covariance vs Semivariogram - Sq. Exp.



Nugget variance

Very often in the real world we will observe that $\gamma(t_i, t_i) = 0$ is not true - there will be an initial discontinuity in the semivariogram at $|t_i - t_j| = 0$.

Why is this?

We can think about Gaussian process regression in the following way,

$$y(t) = \mu(t) + w(t) + \epsilon(t)$$

where

$$\begin{aligned} \boldsymbol{\mu}(\boldsymbol{t}) &= \boldsymbol{X}\boldsymbol{\beta} \\ \boldsymbol{w}(\boldsymbol{t}) &\sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}) \\ \epsilon(t) &\overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma_w^2) \end{aligned}$$

Implications

With the inclusion of the $\epsilon(t)$ terms in the model we now have,

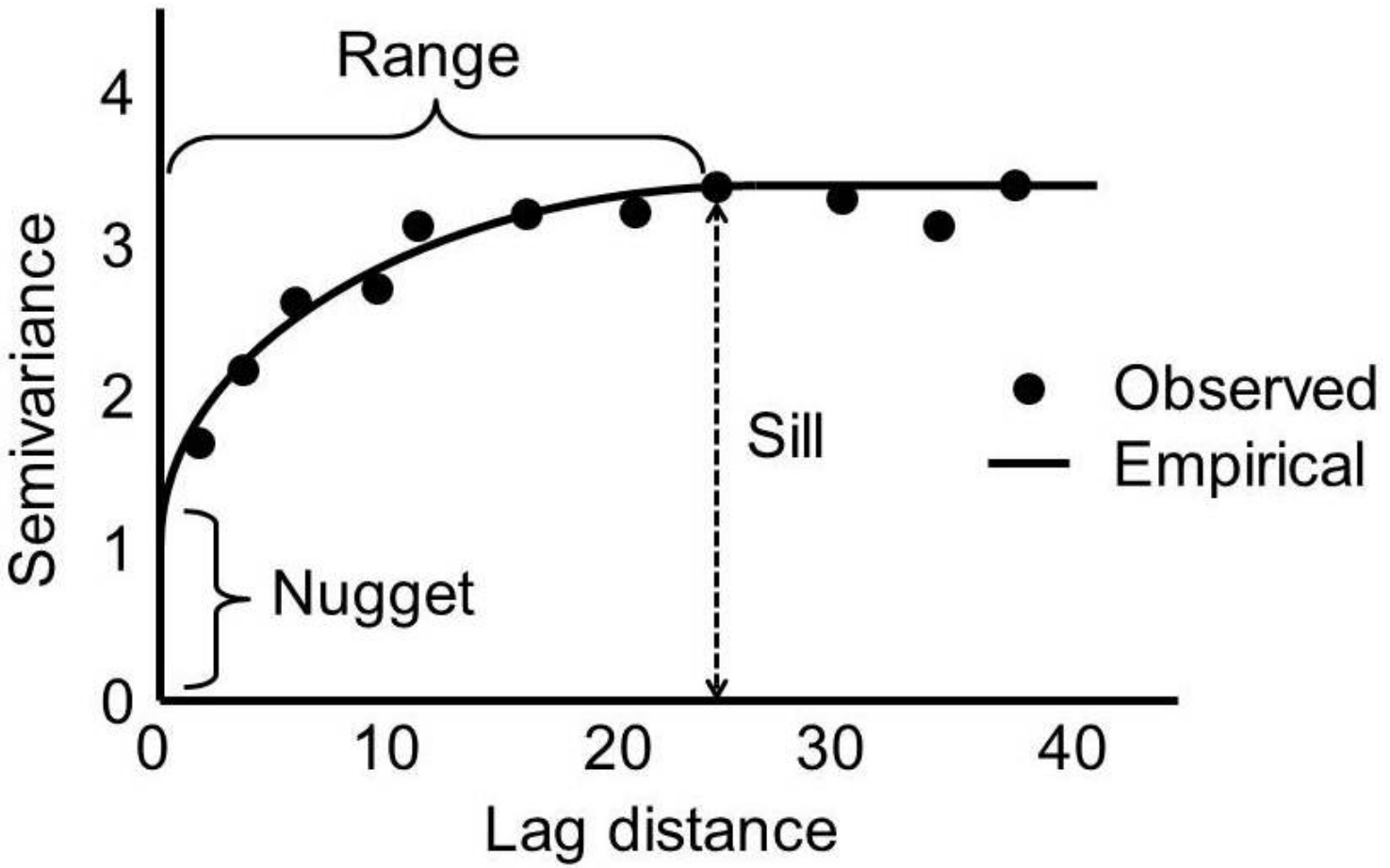
$$\begin{aligned}\text{Var}\Big(y(t_i)\Big) &= \sigma_w^2 + \Sigma_{ii} \\ \text{Cov}\Big(y(t_i), y(t_j)\Big) &= \Sigma_{ij}\end{aligned}$$

Therefore, for a squared exponential covariance model with a nugget component the semivariogram is given by,

$$\gamma(t_i, t_j) = \left(\sigma^2 + \sigma_w^2\right) - \sigma^2 \exp\left(-(|t_i - t_j|\, l)^2\right)$$

$$\begin{aligned}\gamma(t_i, t_i) &= \left(\sigma^2 + \sigma_w^2\right) - \sigma^2 \exp\left(-(|t_i - t_j|\, l)^2\right) \\ &= \left(\sigma^2 + \sigma_w^2\right) - \sigma^2 \exp(0) = \sigma_w^2\end{aligned}$$

Semivariogram features



From https://www.intechopen.com/chapters/39857

Empirical Semivariogram

We will assume that our process of interest is stationary, in which case we will parameterize the semivariagram in terms of $d = |t_i - t_j|$.

Empirical Semivariogram:

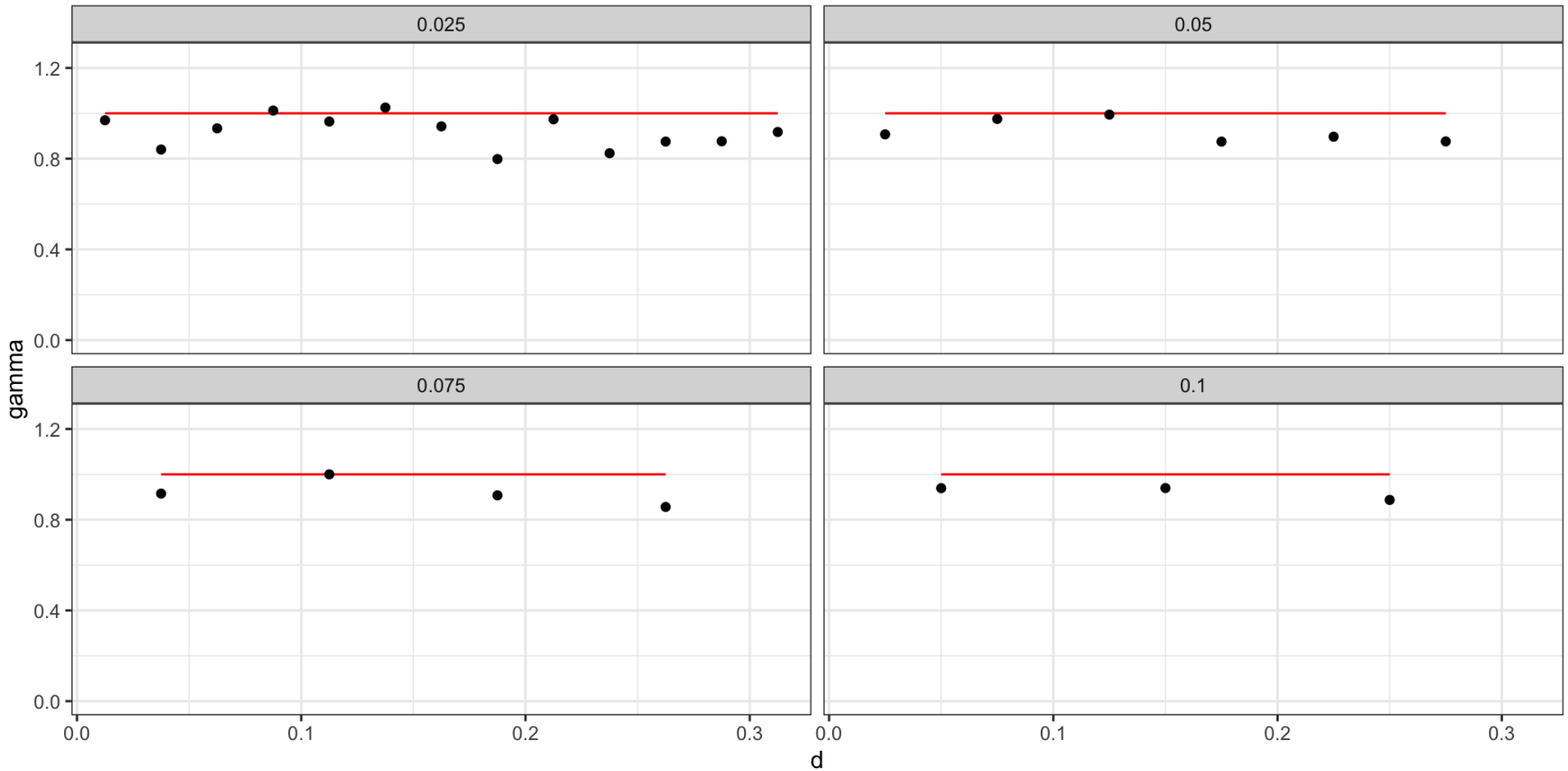
$$\hat{\gamma}(d) = \frac{1}{2\,N(d)} \sum_{|t_i - t_j| \in (d-\epsilon, d+\epsilon)} (y(t_i) - y(t_j))^2$$

. . .

Practically, for any data set with n observations there are $\binom{n}{2} + n$ possible data pairs to examine. Each individually is not very informative, so we aggregate into bins and calculate the empirical semivariogram for each bin.

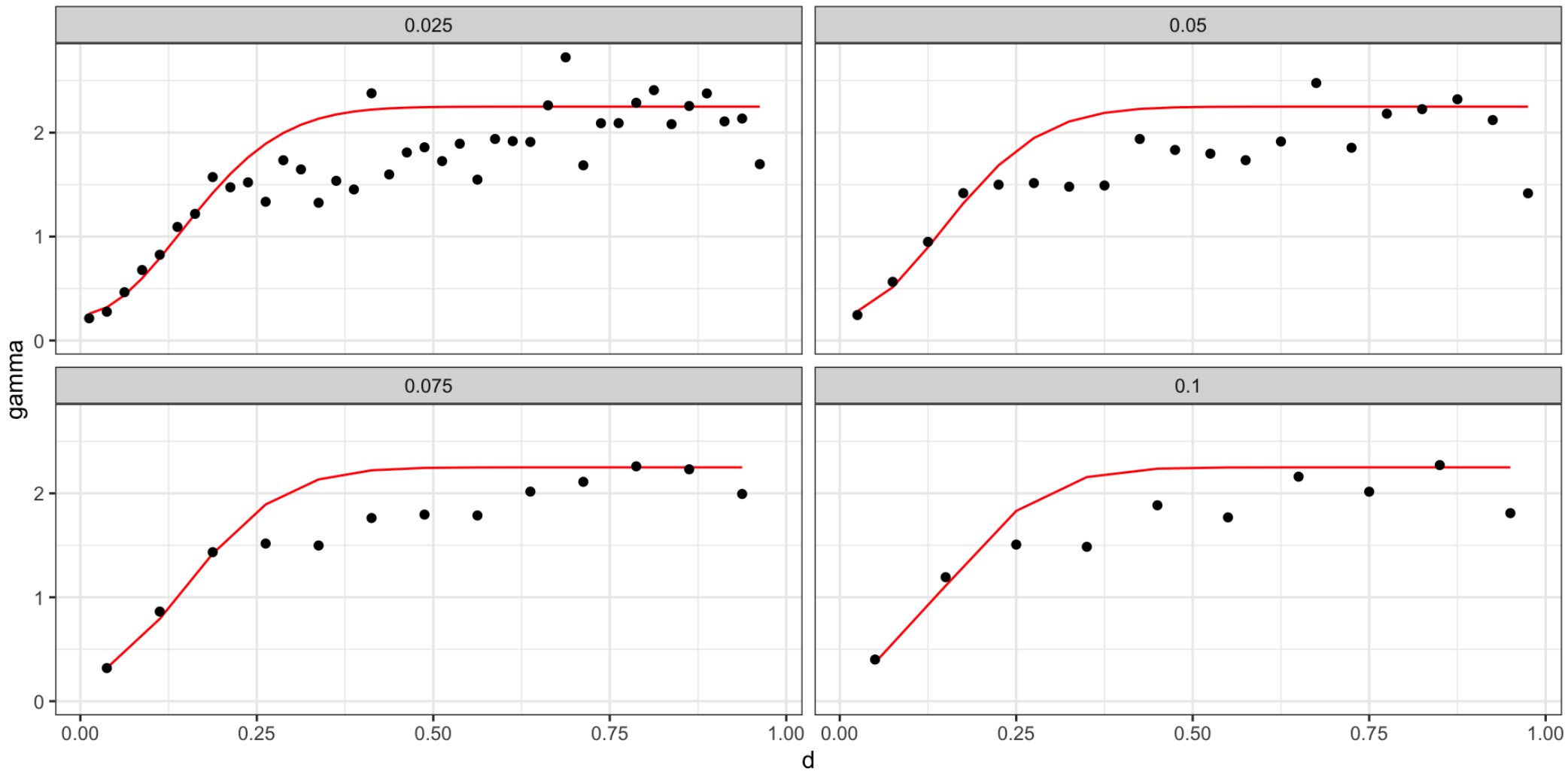
Empirical semivariogram of white noise

Where $\sigma_w^2 = 1$,



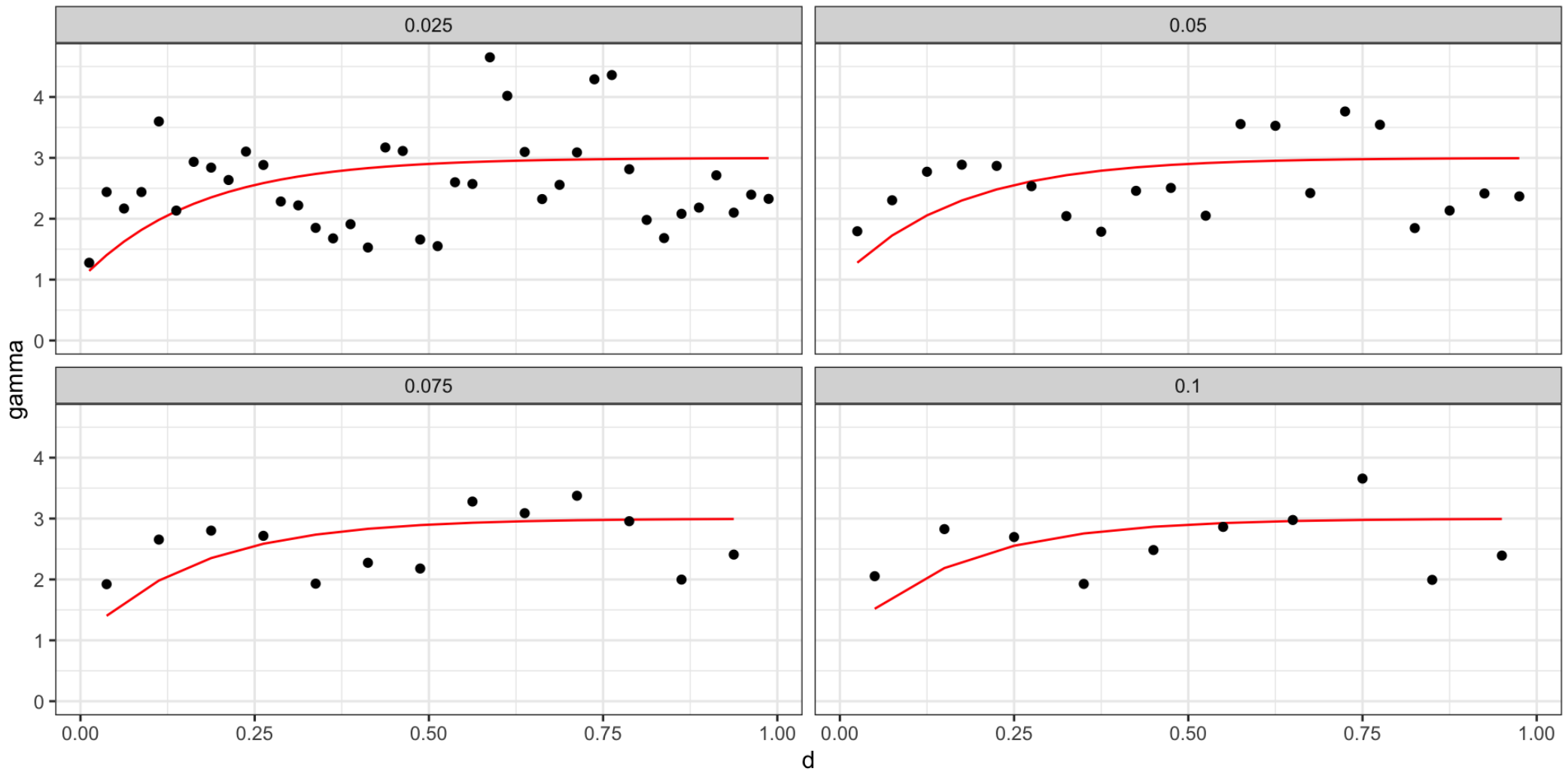
Empirical Variogram of GP w/ Sq Exp cov

Where $\sigma^2 = 2, l = 5$, and $\sigma_w^2 = 0.25$,

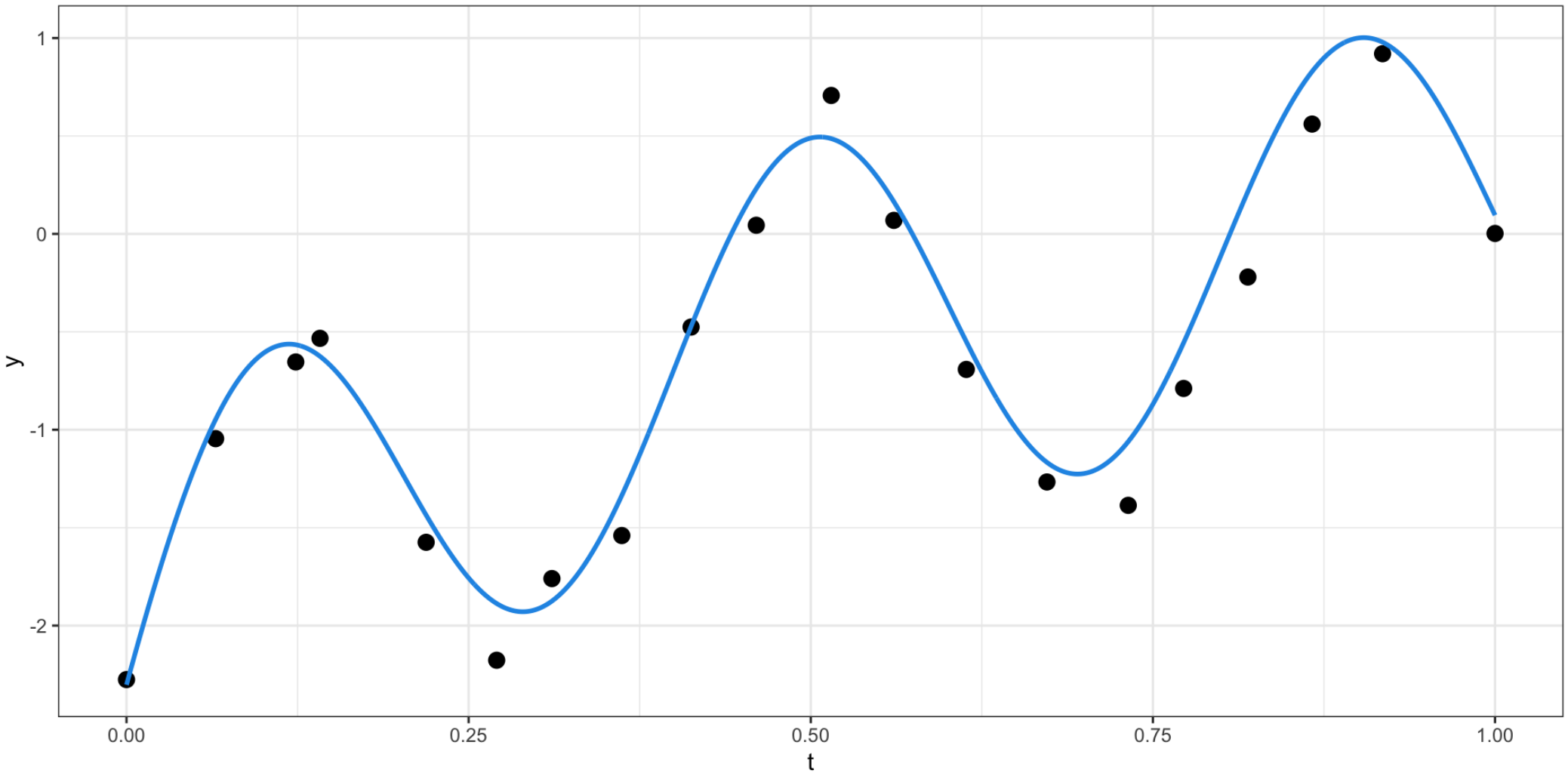


Empirical Variogram of GP w/ Exp cov

Where $\sigma^2 = 2, l = 6$, and $\sigma_w^2 = 1$,

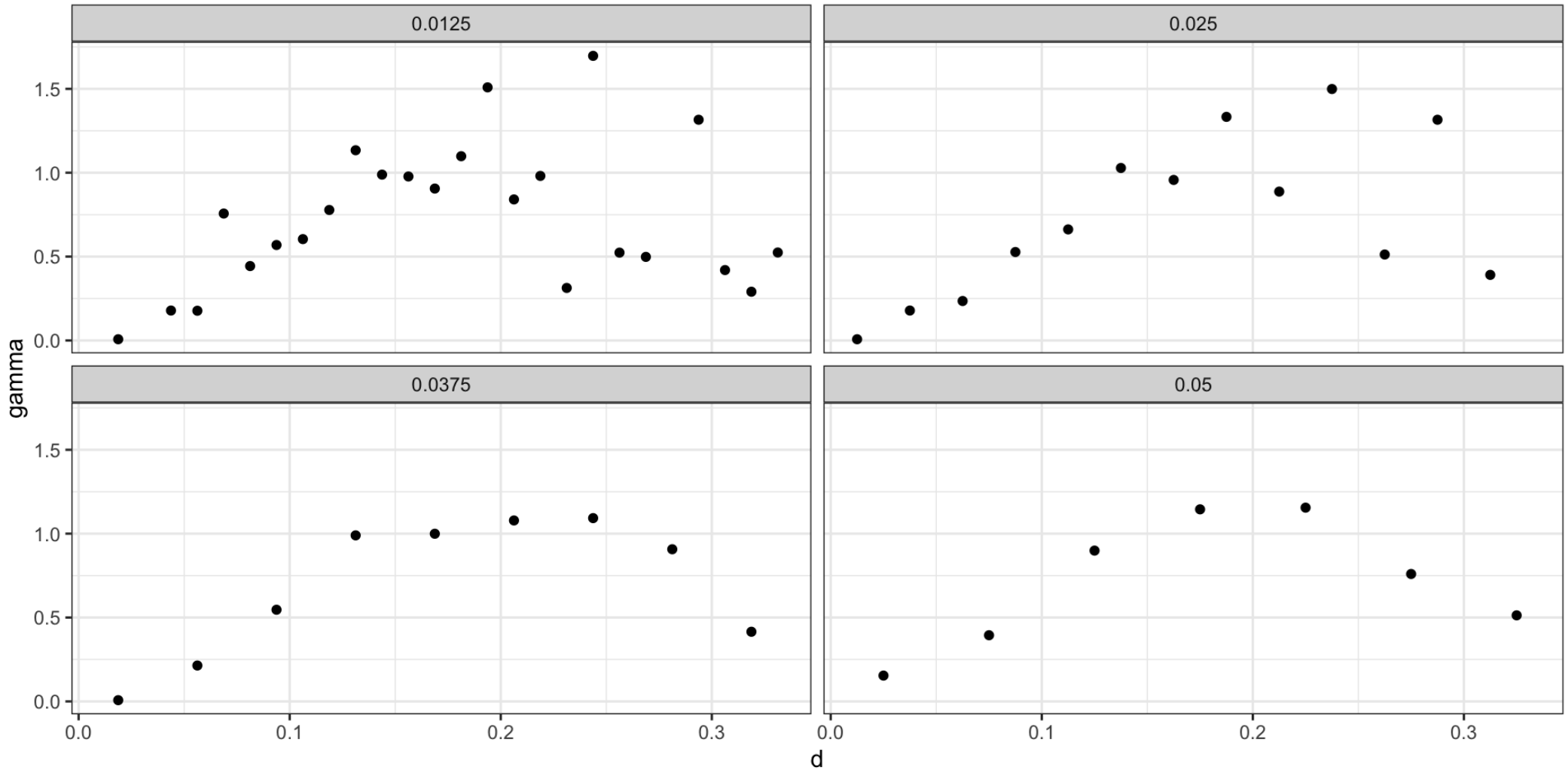


From last time

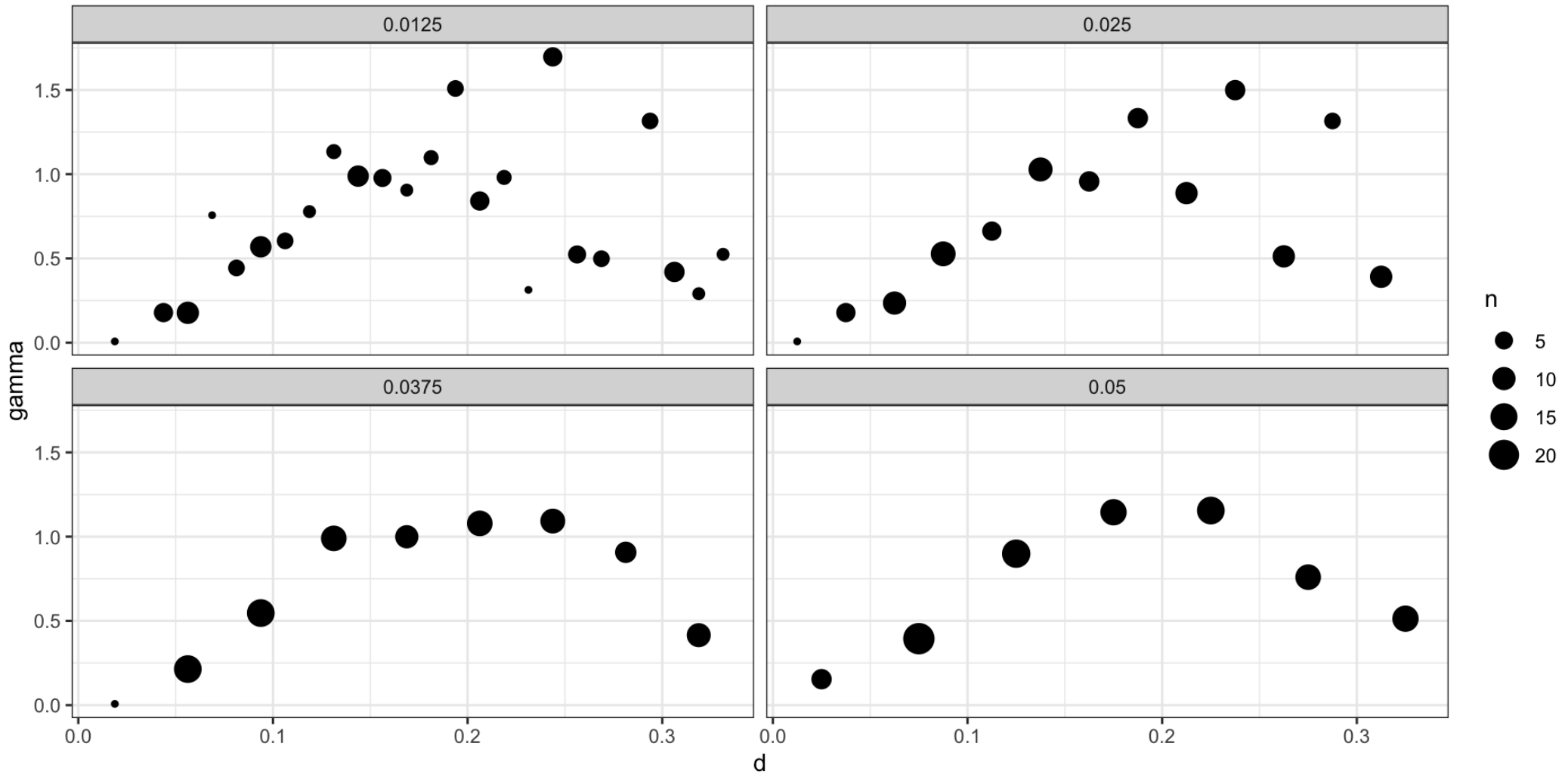


Empirical semivariogram - no bins / cloud

Empirical semivariogram (binned)



Empirical semivariogram (binned w/ size)



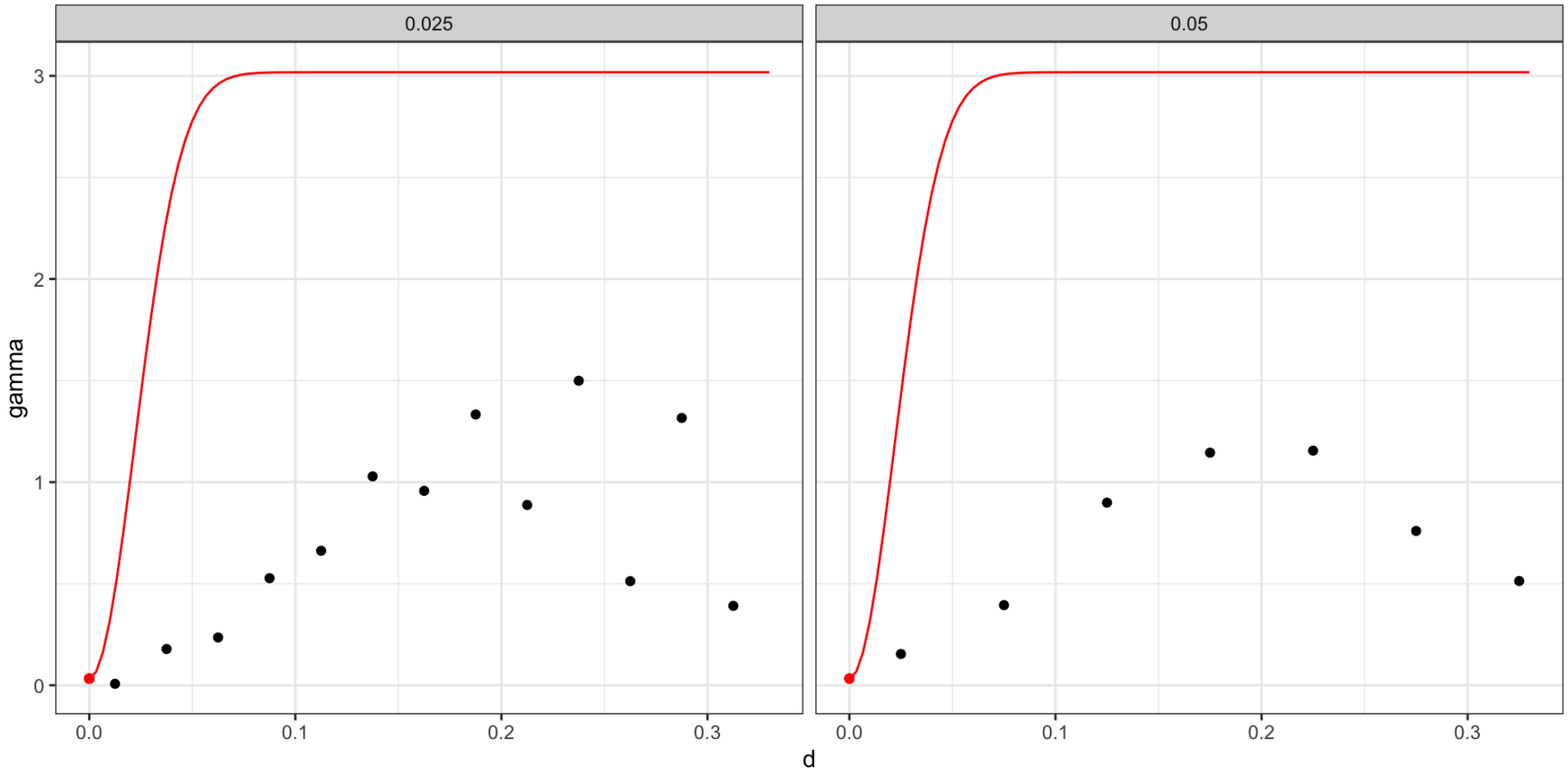
Theoretical vs empirical semivariogram

After fitting the model last time we came up with a posterior mean of $\sigma^2 = 2.99, l = 31.78$, and σ_w^2 = 0.03 for a square exponential covariance.

$$\text{Cov}(d) = \sigma^2 \exp\left(-(d\,l)^2\right) + \sigma_w^2 \mathbf{1}_{h=0}$$

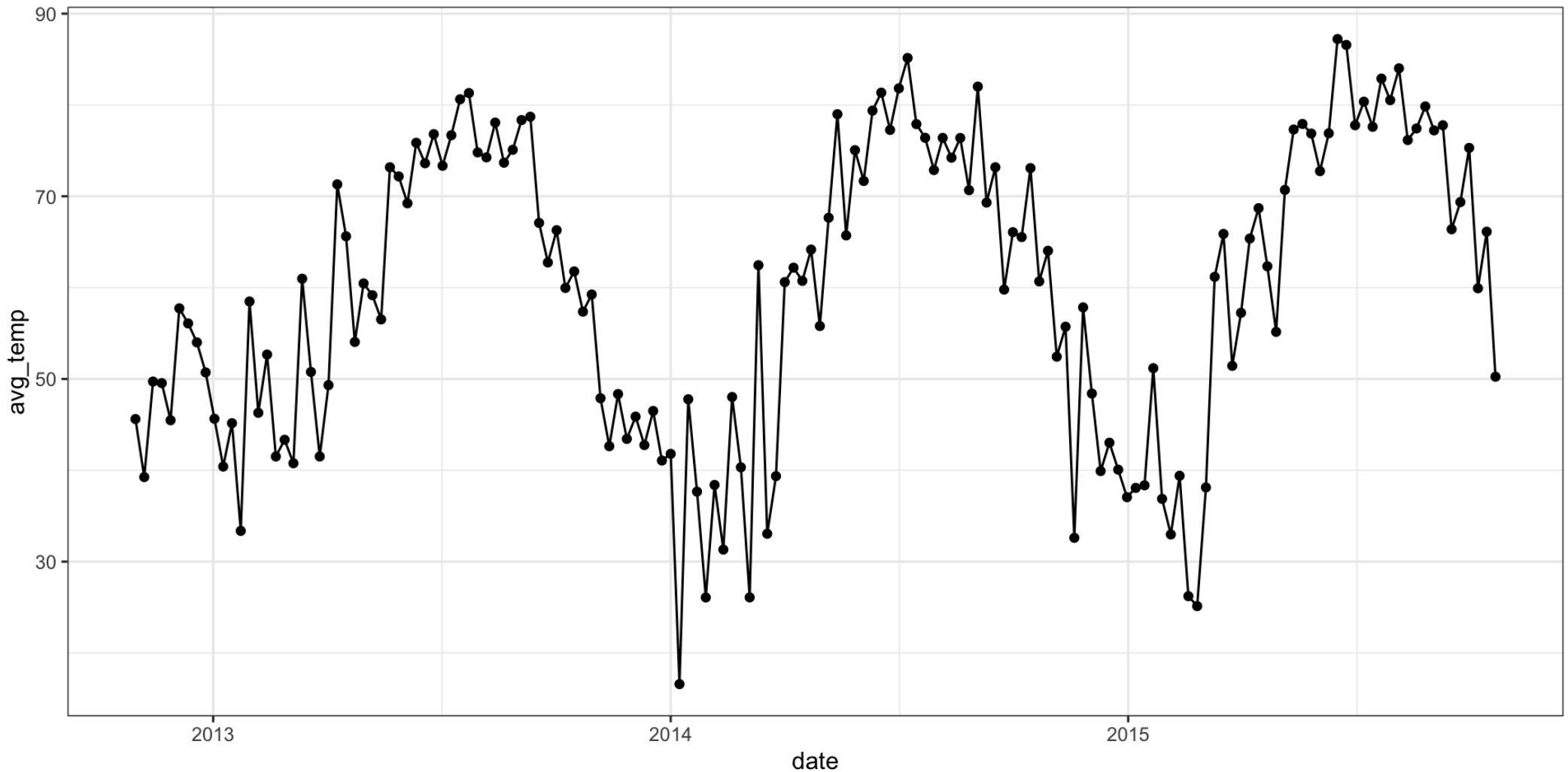
$$\gamma(h) = (\sigma^2 + \sigma_w^2) - \sigma^2 \exp\left(-(h\,l)^2\right)$$

$$= (2.9850975 + 0.032637) - 2.9850975 \exp\left(-(31.7775906\,h)^2\right)$$

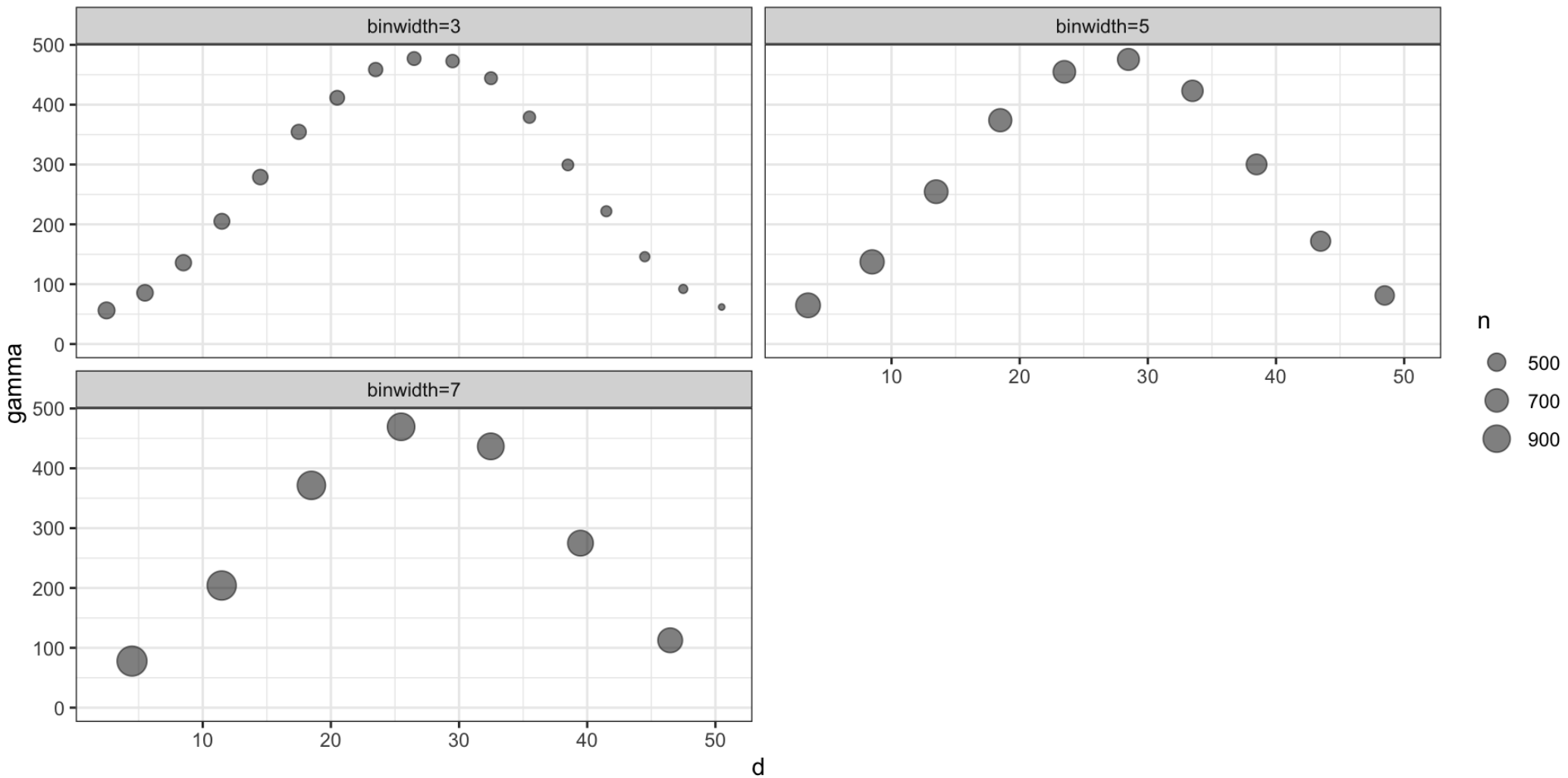

0.025
0.05
gamma
d
0
1
2
3
0.0
0.1
0.2
0.3

Durham Average Daily Temperature

Temp Data



Empirical semivariogram



Model

What does the model we are trying to fit actually look like?

$$y(t) = \mu(t) + w(t) + \epsilon(t)$$

where

$$\begin{aligned} \boldsymbol{\mu}(\boldsymbol{t}) &= \beta_0 \\ \boldsymbol{w}(\boldsymbol{t}) &\sim \square\square(0, \boldsymbol{\Sigma}) \\ \epsilon(t) &\sim \square(0, \sigma_w^2) \end{aligned}$$

$$\{\boldsymbol{\Sigma}\}_{ij} = \text{Cov}(t_i, t_j) = \sigma^2 \exp(-(|t_i - t_j| \, l)^2)$$

BRMS Model

```
library(brms)
( m = brm(
    avg_temp ~ 1 + gp(week), data=temp,
    cores = 4, refresh=0
) )
```

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: avg_temp ~ 1 + gp(week)
   Data: temp (Number of observations: 156)
  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 Rhat Bulk_ESS Tail_ESS
sdgp(gpweek)       10.64      5.41     4.20    16.01 3.82        4       15
lscale(gpweek)      1.26      1.20     0.06     2.69 4.34        4       NA

Population-Level Effects:
```

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	6.32	7.27	-1.28	16.12	3.13	5	30

BRMS Alternatives

The BRMS model (and hence Stan) took between 5-10 minutes (per chain) to attempt to fit the model and failed spectacularly.

We could potentially improve things by tweaking the priors and increasing iterations but this won't solve the slowness issue.

The stop gap work around - using spBayes

- Interface is old and clunky (inputs and outputs)
- Designed for spatial GPs
- Super fast (~10 seconds for 20k iterations)
- `dukestm` has a wrapper, called `gplm()` to make the interface / usage not as terrible

Fitting the model

```
m = gplm(
  avg_temp~1,
  data = temp, coords = "week",
  cov_model = "gaussian",
  starting=list(
    "phi"=sqrt(3)/4, "sigma.sq"=1, "tau.sq"=1
  ),
  priors=list(
    "phi.unif"=c(sqrt(3)/52, sqrt(3)/1),
    "sigma.sq.ig"=c(2, 1),
    "tau.sq.ig"=c(2, 1)
  ),
  thin=10
)
```

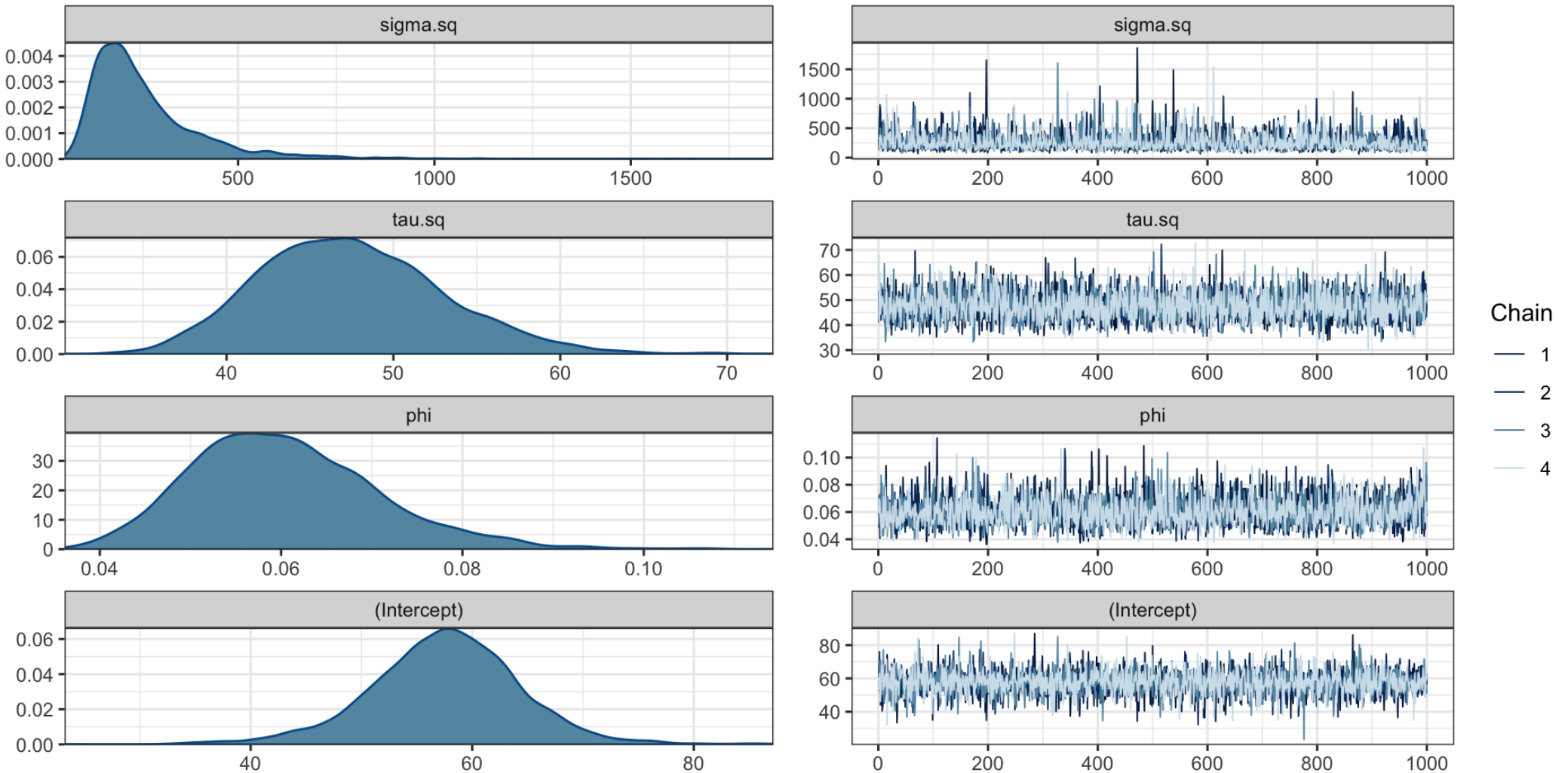
Model results

```
1 m
```

```
# A gplm model (spBayes spLM) with 4 chains, 4 variables, and 4000
iterations.
# A tibble: 4 × 10
  variable        mean   median        sd       mad       q5      q95  rhat
ess_bulk
  <chr>          <num>    <num>     <num>     <num>    <num>    <num> <num>
<num>
1 sigma.sq    263.     221.     150.     98.2     1.15e+2 5.54e+2  1.00
3075.
2 tau.sq       47.5     47.1      5.54    5.50    3.91e+1 5.72e+1  1.00
3704.
3 phi           0.0605   0.0595   0.0104  0.00997 4.55e-2 7.92e-2  1.00
3097.
4 (Intercept)  57.6     57.7      6.80    6.15    4.63e+1 6.81e+1  1.00
3955.
# ℹ 1 more variable: ess_tail <num>
```

Parameter posteriors

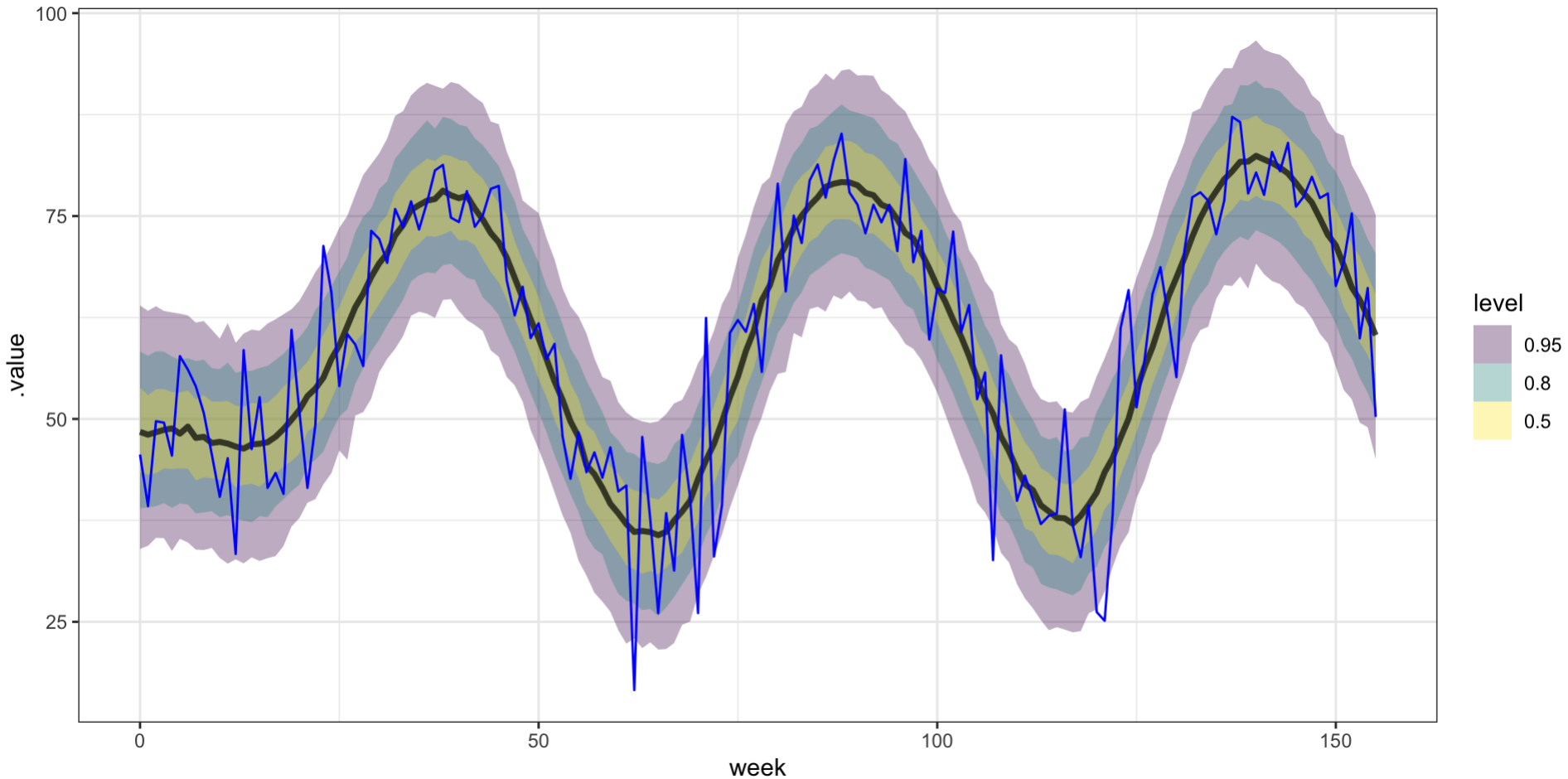
```
1 plot(m)
```



Fitted model

```
predict(m, newdata = tibble(week=1:(3*52)-1+1e-6), coords = "week") |>
  tidybayes::gather_draws(y[i]) |>
  mutate(week = i-1) |>
  filter(.chain == 1, week <= 3*52) |>
  ggplot(aes(x=week, y=.value)) +
    tidybayes::stat_lineribbon(alpha=0.33) +
    geom_line(data=temp, aes(y=avg_temp), color="blue")
```

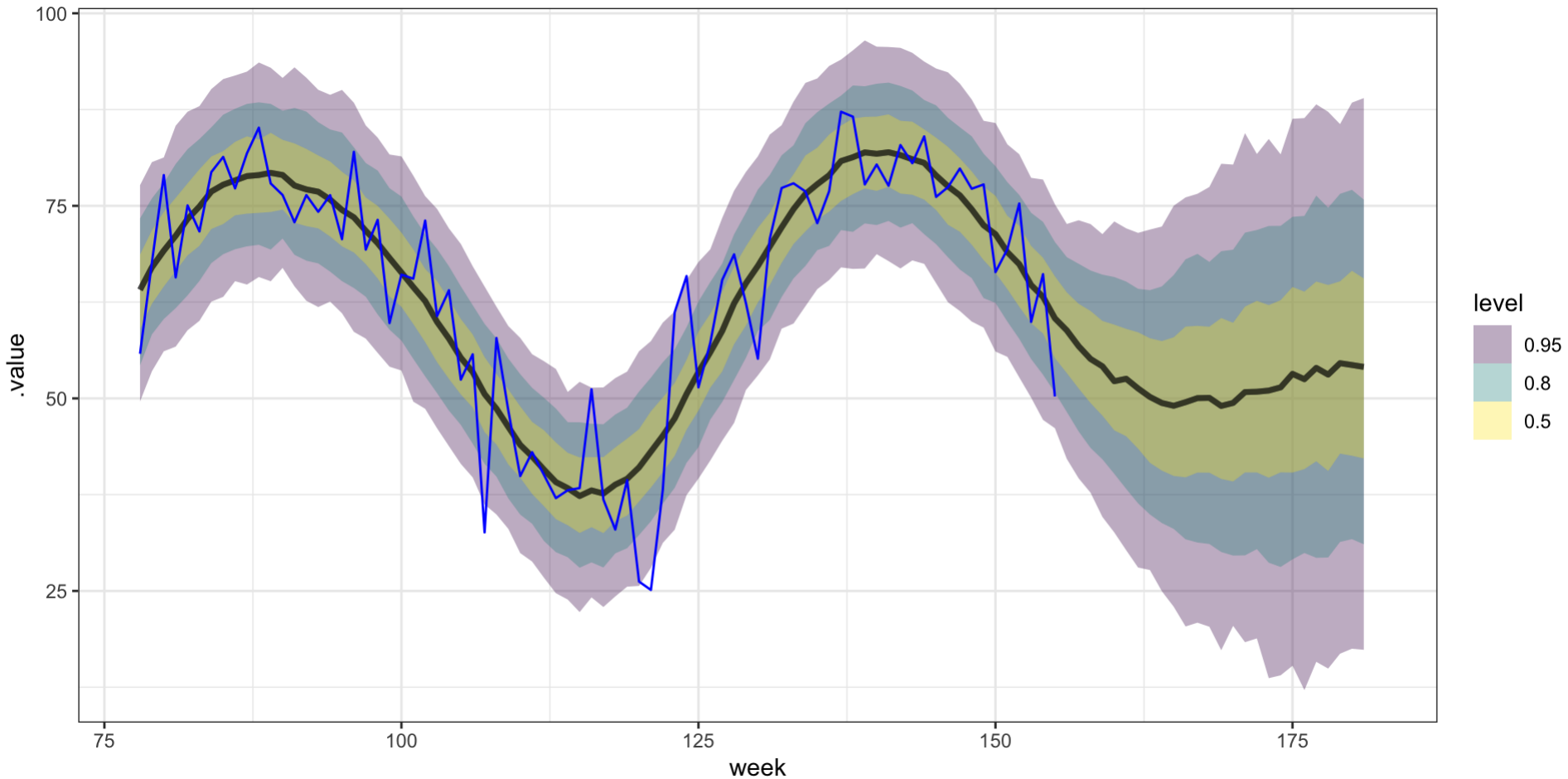
Fitted model



Forecasting

```
predict(m, newdata = tibble(week=1:(3.5*52)-1+1e-6), coords = "week") |>
  tidybayes::gather_draws(y[i]) |>
  mutate(week = i-1) |>
  filter(.chain == 1) |>
  ggplot(aes(x=week, y=.value)) +
    tidybayes::stat_lineribbon(alpha=0.33) +
    geom_line(data=temp, aes(y=avg_temp), color="blue") +
    xlim(1.5*52, 3.5*52)
```

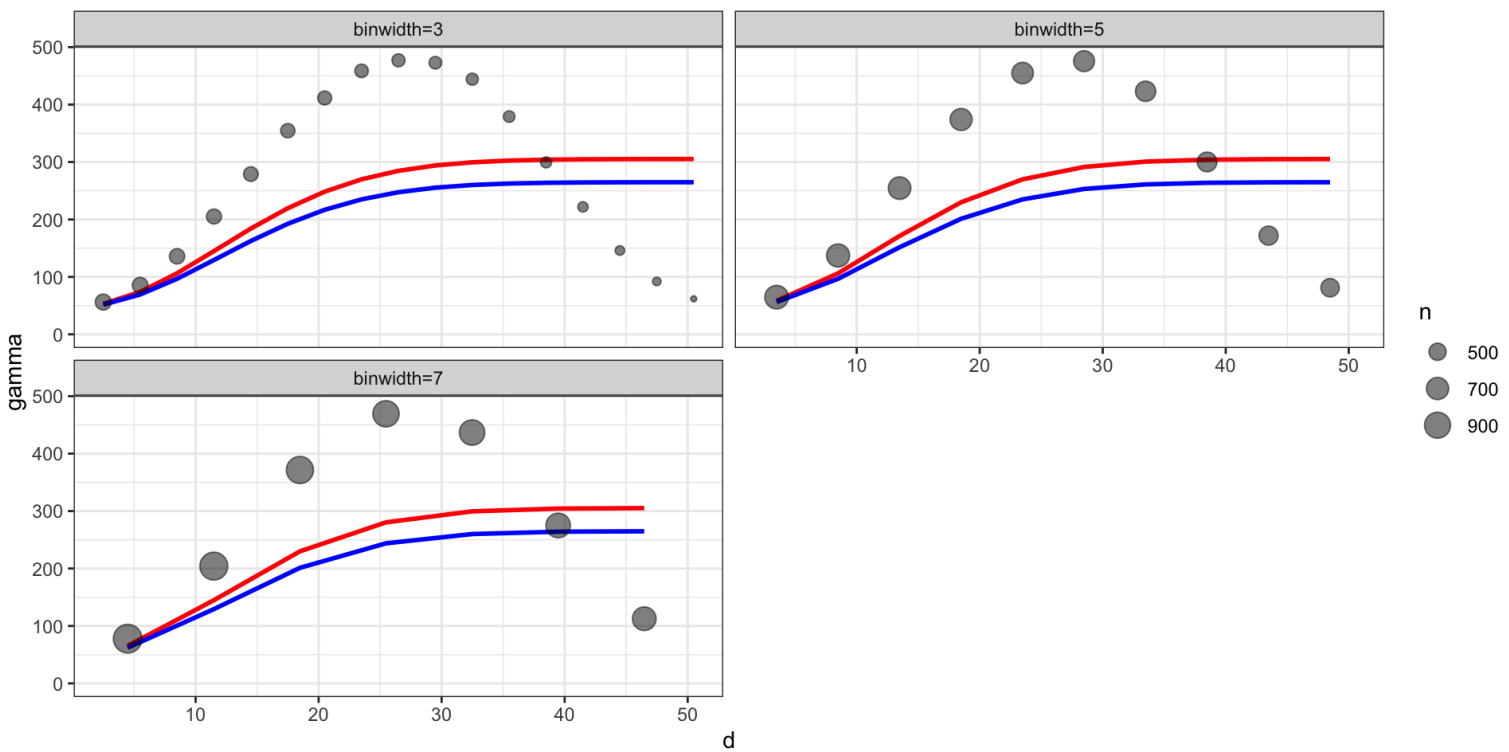
Forecasting



Empirical semivariogram vs. model

From the model summary we have the following,

- *posterior means*: $\sigma^2 = 258, \sigma_w^2 = 47.3, 1 = 0.06$
- *posterior medians*: $\sigma^2 = 218, \sigma_w^2 = 46.9, 1 = 0.06$



Exponential model

Fitting the model

```
m = gplm(
  avg_temp~1,
  data = temp, coords = "week",
  cov_model = "exponential",
  starting=list(
    "phi"=3/4, "sigma.sq"=1, "tau.sq"=1
  ),
  priors=list(
    "phi.unif"=c(3/52, 3/1),
    "sigma.sq.ig"=c(2, 1),
    "tau.sq.ig"=c(2, 1)
  ),
  thin=10
)
```

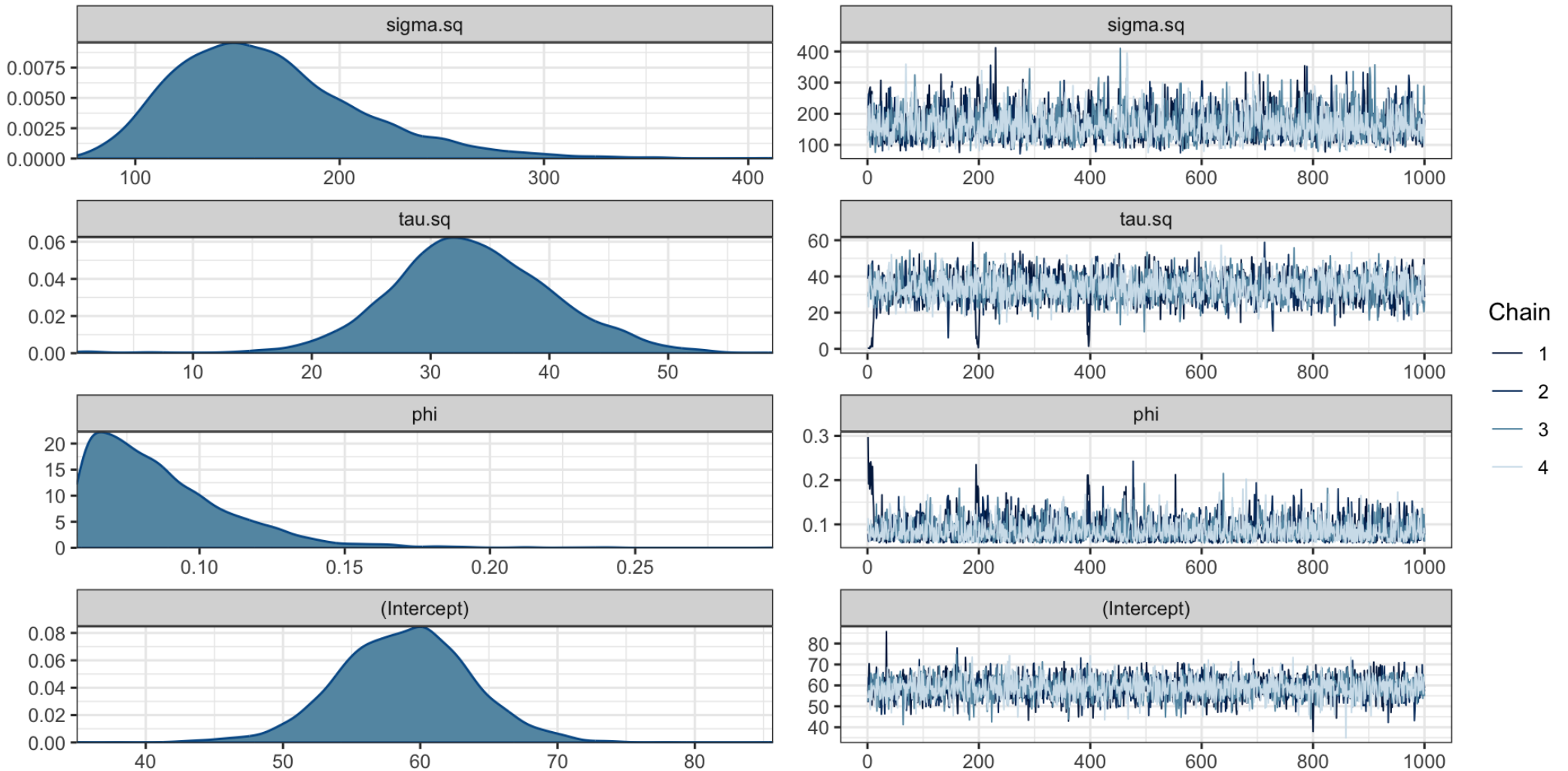
Model results

```
1 m
```

```
# A gplm model (spBayes spLM) with 4 chains, 4 variables, and 4000
iterations.
# A tibble: 4 × 10
  variable        mean   median      sd     mad        q5      q95  rhat
ess_bulk
  <chr>          <num>    <num>   <num>   <num>     <num>    <num> <num>
<num>
1 sigma.sq    165.     159.     46.1    42.7    103.     251.      1.00
2871.
2 tau.sq       33.5     33.2     6.97    6.48    22.7     45.3     1.00
2766.
3 phi           0.0865   0.0809  0.0244  0.0210   0.0597   0.132   1.00
2858.
4 (Intercept)  58.8     59.0     4.80    4.68    51.2     66.7     1.00
3910.
# ℹ 1 more variable: ess_tail <num>
```

Parameter posteriors

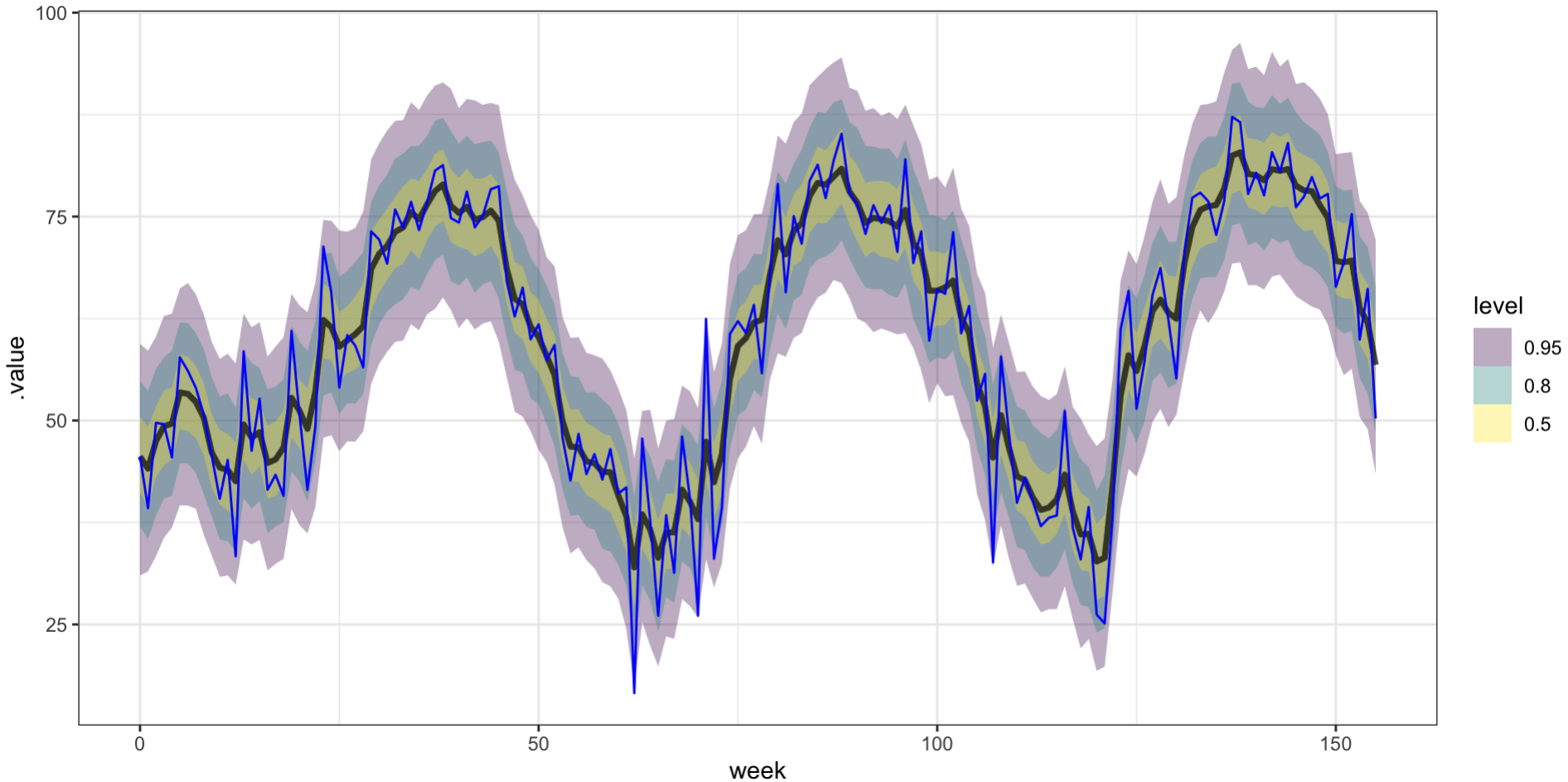
```
1  plot(m)
```



Fitted model

```
predict(m, newdata = tibble(week=1:(3*52)-1+1e-6), coords = "week") |>
  tidybayes::gather_draws(y[i]) |>
  mutate(week = i-1) |>
  filter(.chain == 1, week <= 3*52) |>
  ggplot(aes(x=week, y=.value)) +
    tidybayes::stat_lineribbon(alpha=0.33) +
    geom_line(data=temp, aes(y=avg_temp), color="blue")
```

Fitted model



Forecasting

```
predict(m, newdata = tibble(week=1:(3.5*52)-1+1e-6), coords = "week") |>
  tidybayes::gather_draws(y[i]) |>
  mutate(week = i-1) |>
  filter(.chain == 1) |>
  ggplot(aes(x=week, y=.value)) +
    tidybayes::stat_lineribbon(alpha=0.33) +
    geom_line(data=temp, aes(y=avg_temp), color="blue") +
    xlim(1.5*52, 3.5*52)
```

Forecasting