

The Phylogenetic Ornstein-Uhlenbeck process

J.P. Meagher

10 October 2017

Key to modelling the evolution of continuous characteristics is the Ornstein-Uhlenbeck process. This process can be approached from either a Gaussian process or a Stochastic Differential Equation perspective. I begin by coming at them from GPs.

```
library(tidyverse)
library(magrittr)
```

Gaussian Processes

In order to get a feel for the OU process and how it fits with other Gaussian processes generate a series of Gaussian processes from some gaussian noise.

Gaussian Noise

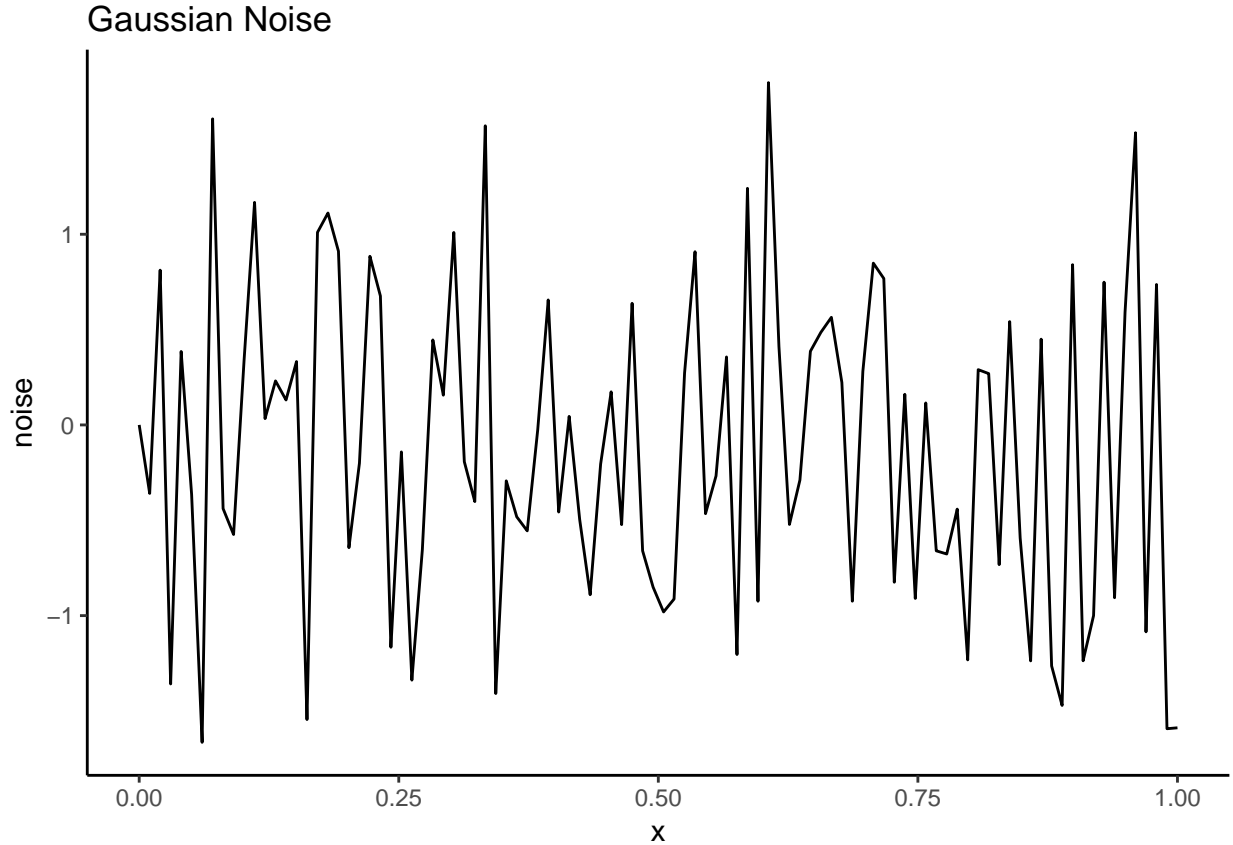
Gaussian noise can be thought of as the Gaussian process defined by the kernel

$$k_N(\mathbf{x}, \mathbf{x}') = \sigma^2 \delta(\mathbf{x}, \mathbf{x}')$$

```
x <- seq(0,1, length.out = 100)
y <- x %>% length %>% rnorm
y[1] <- 0

df <- data.frame(x= x, noise = y)

df %>%
  ggplot() +
  geom_line(aes(x = x, y = noise)) +
  theme_classic() +
  labs(
    title = 'Gaussian Noise'
  )
```



Kernel is the general name given to any function of two arguments mapping the inputs \mathbf{x} and \mathbf{x}' into \mathbf{R} .

Squared Exponential

The next standard example in the GP literature is the GP with a squared exponential kernel. Define $r = |\mathbf{x} - \mathbf{x}'|$. Then

$$k_{SE}(r) = \sigma^2 \exp\left(-\frac{r^2}{2\ell^2}\right),$$

where ℓ is the characteristic length-scale.

A kernel that depends only on $\mathbf{x} - \mathbf{x}'$ is stationary. Furthermore, a kernel depending only on r is isotropic and these kernels are known as radial basis functions (RBF).

```
se_k <- function(x, s = 1, l = 1){
  r <- dist(x) %>% as.matrix
  k <- (s^2)*exp(((r^2))/(2*(l^2)))) +
    x %>% length %>% diag %>% multiply_by(1e-10)

  return(k)
}

gp_draw <- function(x, noise, kernel = se_k, ...){
  k <- kernel(x, ...)
  chol_k <- chol(k)
```

```

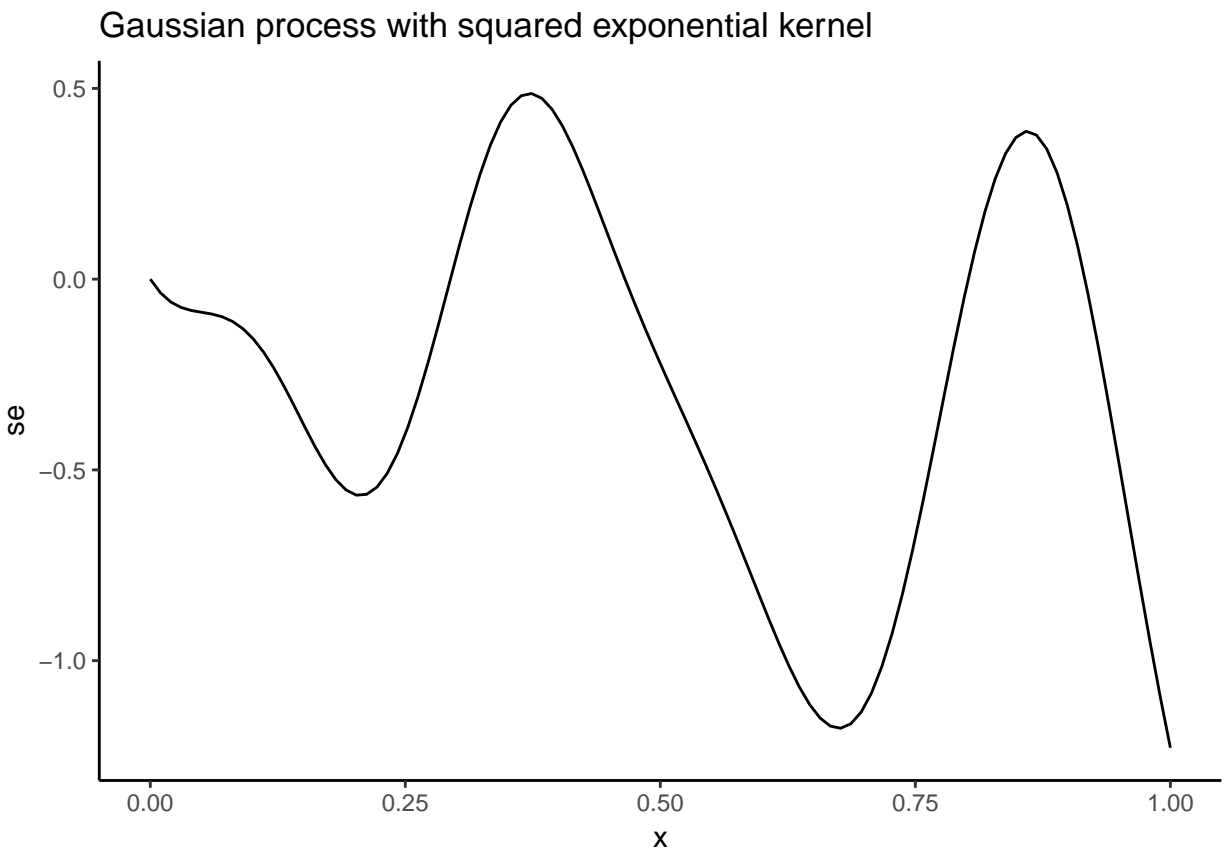
draw <- t(chol_k) %*% noise

return(draw)
}

df <- df %>%
  mutate(se = gp_draw(x= x, noise = noise, l = 0.1) %>% c)

df %>%
  ggplot() +
  geom_line(aes(x = x, y = se)) +
  theme_classic() +
  labs(
    title = 'Gaussian process with squared exponential kernel'
  )

```



Note that there are some computational tricks required for Gaussian processes. Some extra weight must be placed on the diagonal of the covariance matrix to ensure the positive definiteness. This extra weight is what produces the not quite smooth function above. Also note that when optimising the log of the hyperparameters should be used.

Matern

The Matern class of covariance functions are defined by the kernel

$$k_{\text{matern}}(r) = \frac{2^{1-v}}{\Gamma(v)} \left(\frac{\sqrt{2v}r}{\ell} \right)^v K_v \left(\frac{\sqrt{2v}r}{\ell} \right)$$

where v, ℓ are positive, $\Gamma(\cdot)$ is the gamma function and K_v is the modified Bessel function.

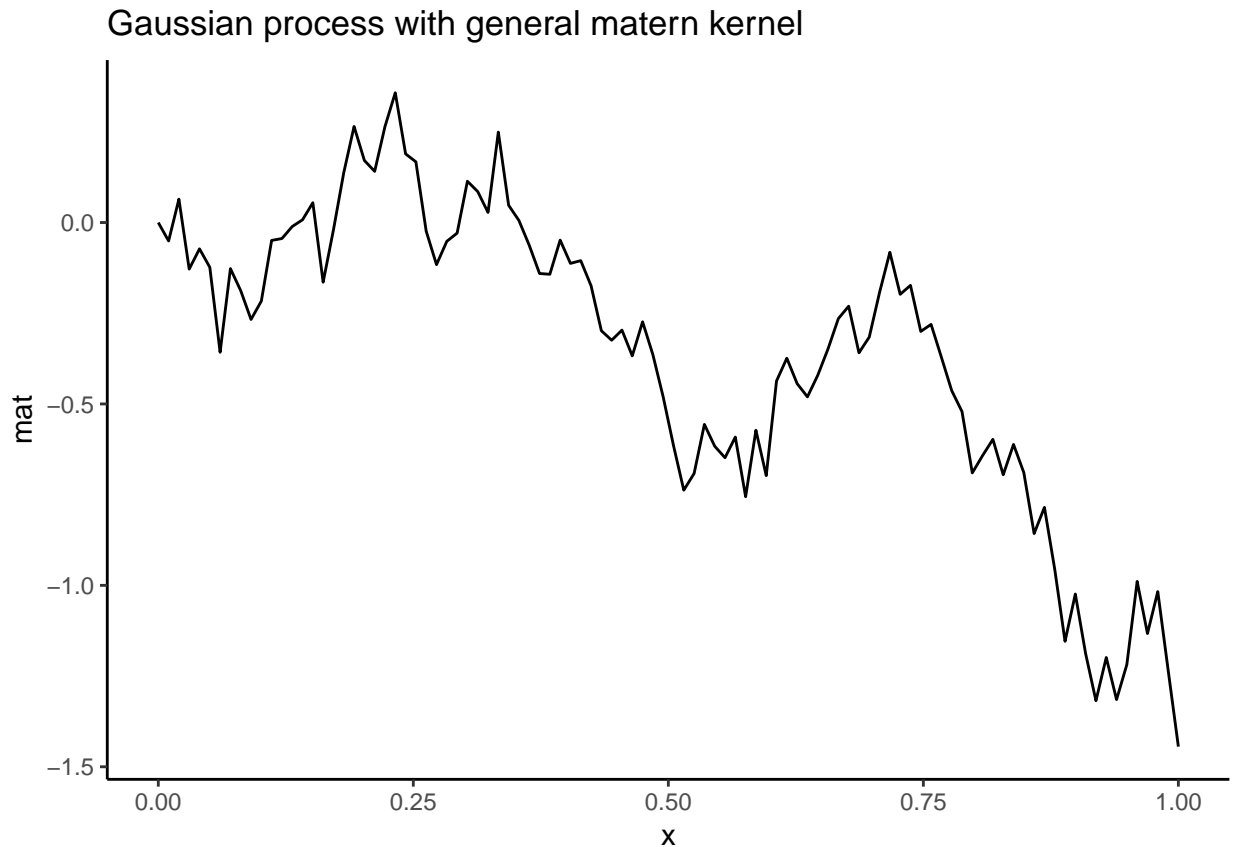
```
matern_k <- function(x, s = 1, l = 1, ups = 1/2){
  r <- dist(x) %>% as.matrix %>% add(1e-100)

  k <- (s^2)*
    ((2^(1 - ups)) / gamma(ups))*
    ((sqrt(2*ups)*(r/l))^ups)*
    besselK(sqrt(2*ups)*(r/l), ups) +
    x %>% length %>% diag %>% multiply_by(1e-10)

  return(k)
}

df <- df %>%
  mutate(mat = gp_draw(x= x, noise = noise, kernel = matern_k, l = 1) %>% c)

df %>%
  ggplot() +
  geom_line(aes(x = x, y = mat)) +
  theme_classic() +
  labs(
    title = 'Gaussian process with general matern kernel'
  )
```



Ornstein-Uhlenbeck

The Ornstein-Uhlenbeck kernel is a special case of the Matern class when $\nu = 1/2$.

$$k_{OU}(r) = \sigma \exp\left(-\frac{r}{\ell}\right)$$

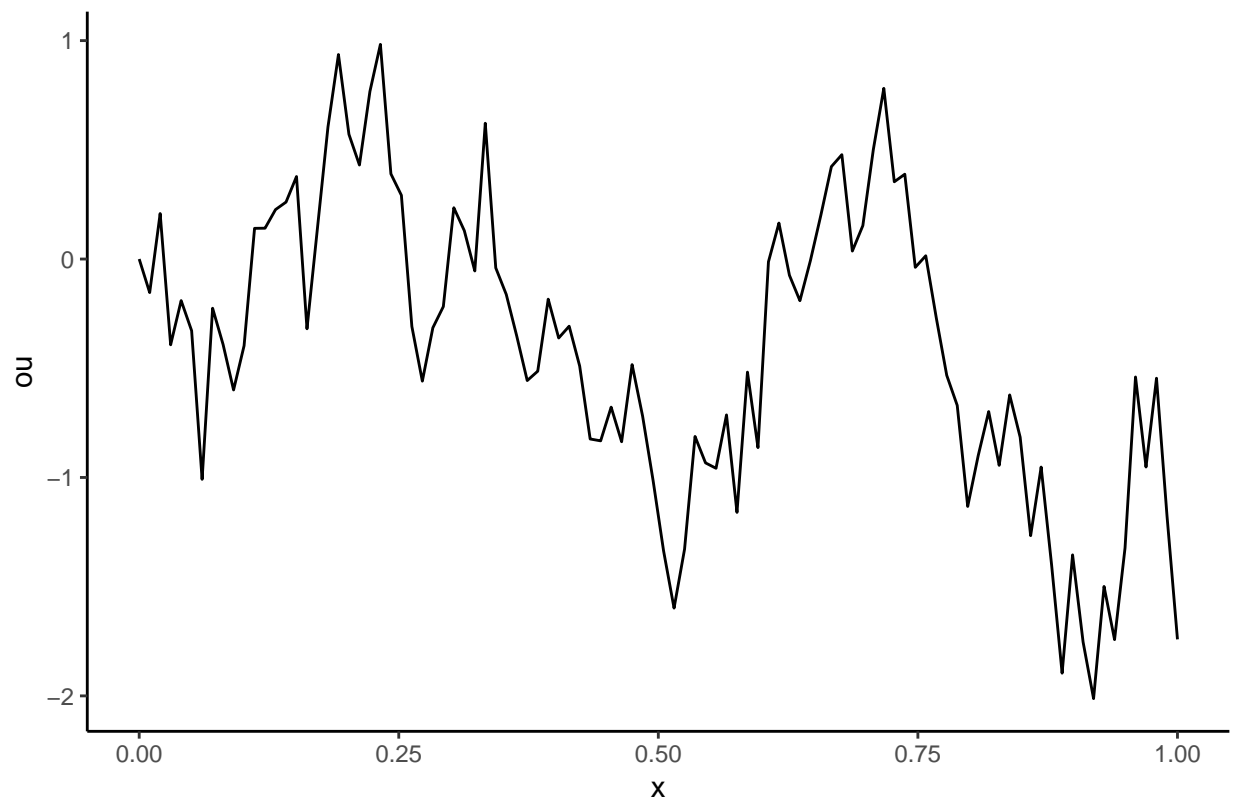
```
ou_k <- function(x, s = 1, l = 1){
  r <- dist(x) %>% as.matrix %>% abs
  k <- (s^2)*exp(-r/l) +
    x %>% length %>% diag %>% multiply_by(1e-10)

  return(k)
}

df <- df %>%
  mutate(ou = gp_draw(x, noise, ou_k, l = 0.1)%>% c)

df %>%
  ggplot() +
  geom_line(aes(x = x, y = ou)) +
  theme_classic() +
  labs(
    title = 'Gaussian process with Ornstein-Uhlenbeck kernel'
  )
}
```

Gaussian process with Ornstein–Uhlenbeck kernel



Rational Quadratic

The Rational Quadratic Covariance Function is defined by the kernel

$$k_{RQ}(r) = \left(1 + \frac{r^2}{2\alpha\ell^2}\right)^{-\alpha}$$

```
rq_k <- function(x, s = 1, l = 1, alpha = 1){
  r <- dist(x) %>% as.matrix %>% abs
  k <- (s^2)*
    (1 + ((r^2) / (2*alpha*(l^2))))^(-alpha) +
    x %>% length %>% diag %>% multiply_by(1e-10)

  return(k)
}

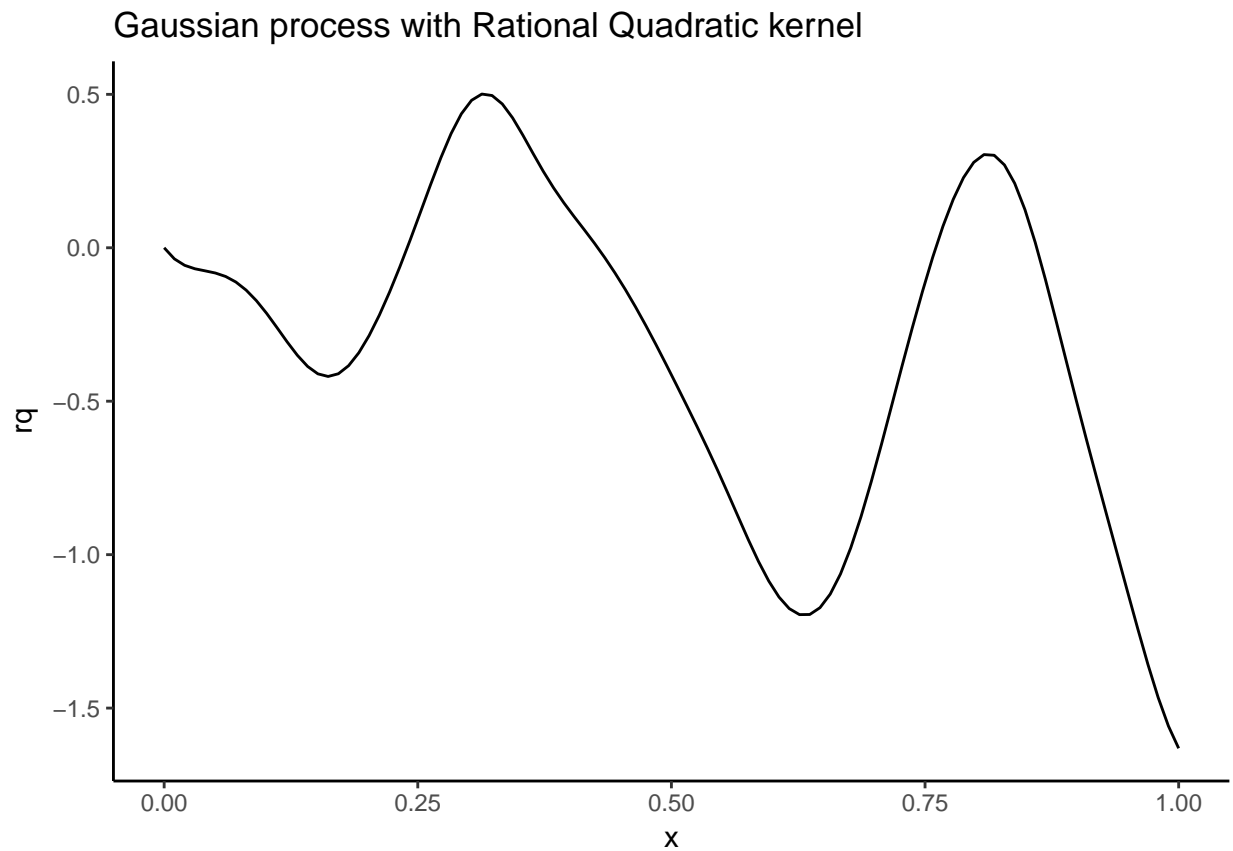
df <- df %>%
  mutate(rq = gp_draw(x, noise, rq_k, l = 0.1, alpha = 3)%>% c)

df %>%
  ggplot() +
  geom_line(aes(x = x, y = rq)) +
  theme_classic() +
  labs(
```

```

title = 'Gaussian process with Rational Quadratic kernel'
)

```



Weiner Process

The Wiener process can be defined by the non-stationary kernel

$$k(x, x') = \min(x, x')$$

```

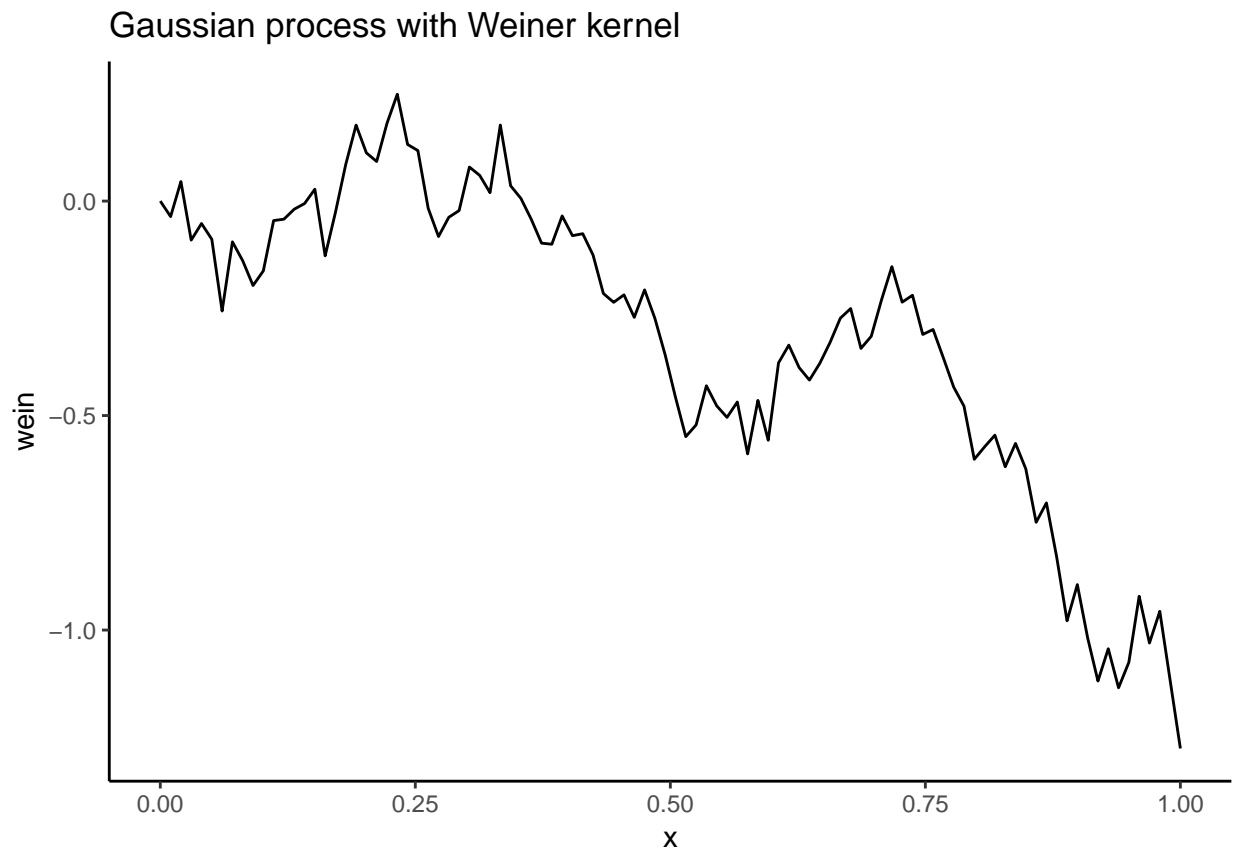
wein_k <- function(x, s = 1){
  k <- matrix(nrow = x %>% length, ncol = x %>% length)
  for(i in seq_along(x)){
    for(j in seq_along(x)){
      k[i,j] <- min(x[i], x[j])
    }
  }
  k <- (s^2)*k + x %>% length %>% diag %>% multiply_by(1e-10)

  return(k)
}

df <- df %>%
  mutate(wein = gp_draw(x, noise, wein_k)%>% c)

```

```
df %>%
  ggplot() +
  geom_line(aes(x = x, y = wein)) +
  theme_classic() +
  labs(
    title = 'Gaussian process with Weiner kernel'
  )
```



Phylogenetic Gaussian Processes

I want to examine how these processes behave as phylogenetic Gaussian processes.

```
library(ape)

tree <- rcoal(32)

phylo_ou_k <- function(x, s = 1, l = 1){
  r <- dist.nodes(x) %>% as.matrix %>% abs
  k <- (s^2)*exp(-r/l) +
    r %>% nrow %>% diag %>% multiply_by(1e-5)

  return(k)
}
```



```

phylo_wein_k <- function(x, s = 1){
  t <- x %>% node.depth.edgelen
  ca <- tree %>% mrca(full = T)

  k <- matrix(nrow = t %>% length, ncol = t %>% length)
  for(i in seq_along(t)){
    for(j in seq_along(t)){
      k[i,j] <- t[ca[i, j]]
    }
  }
  k <- (s^2)*k + t %>% length %>% diag %>% multiply_by(1e-10)

  return(k)
}

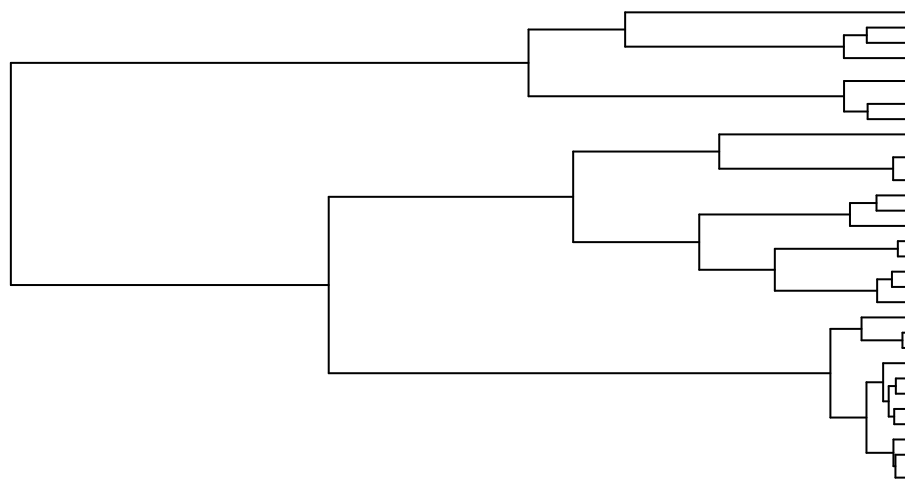
phylo_df <- data.frame(
  x = tree %>% node.depth.edgelen,
  noise = tree %>% Nnode(internal.only = F) %>% rnorm
)

phylo_df <- phylo_df %>%
  mutate(
    br = gp_draw(tree, noise, kernel = phylo_wein_k) %>% c,
    ou = gp_draw(tree, noise, kernel = phylo_ou_k) %>% c
  )

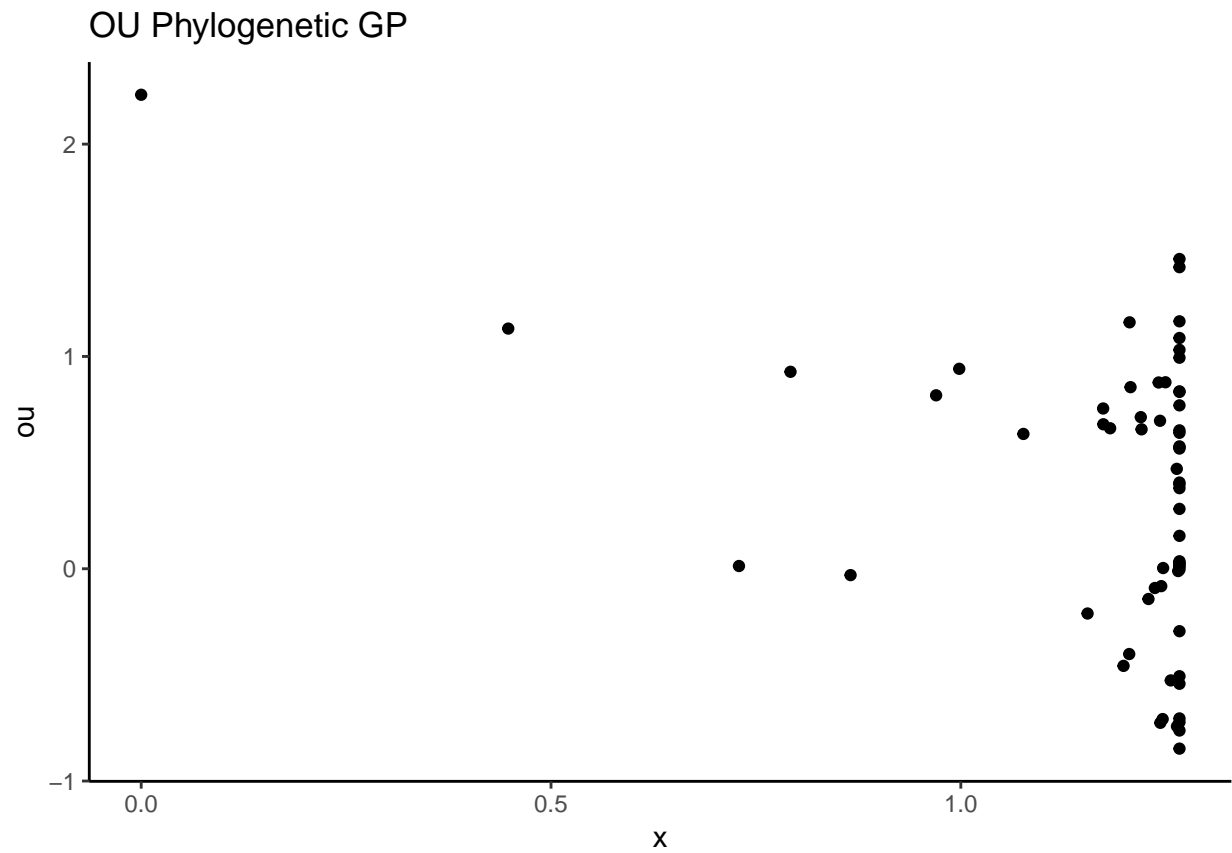
plot.phylo(tree, show.tip.label = F,
  main = 'Random Tree')

```

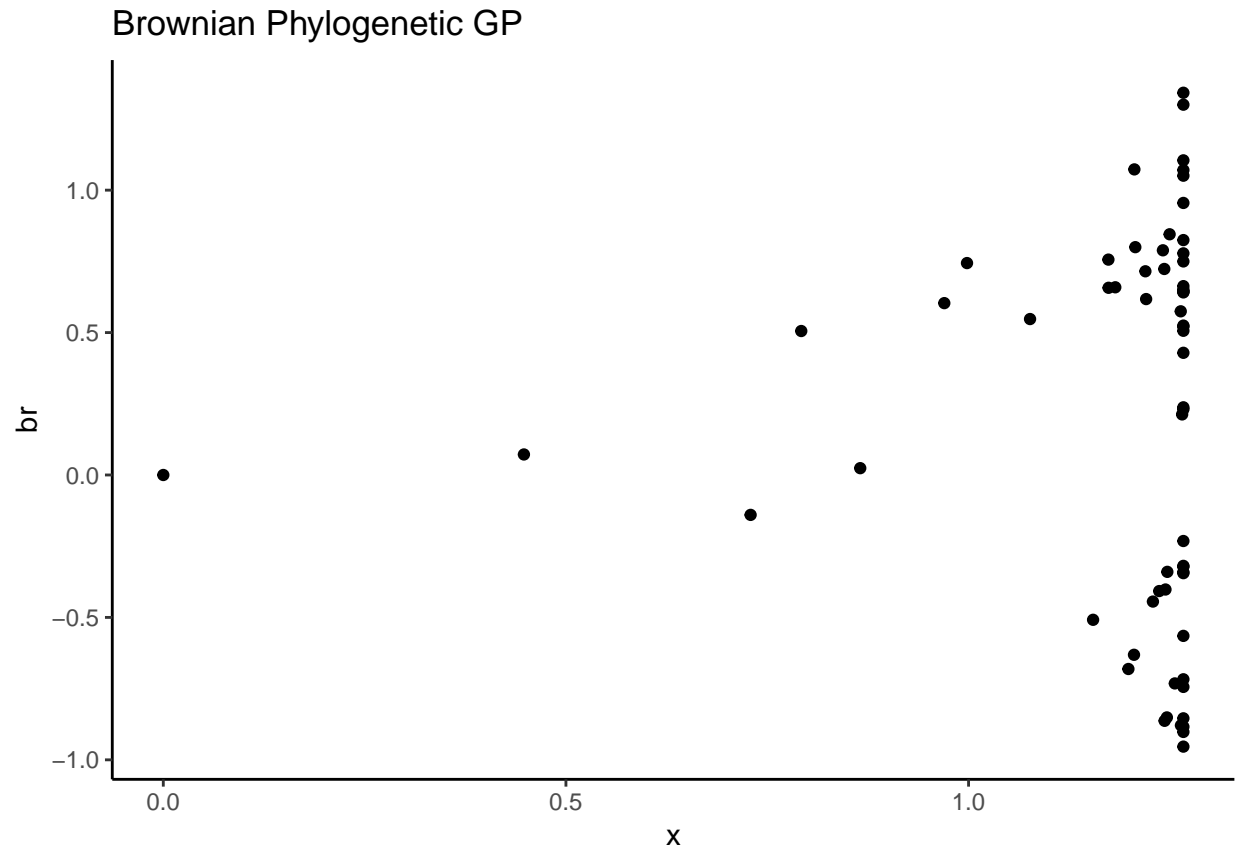
Random Tree



```
phylo_df %>%  
  ggplot() +  
  geom_point(aes(x = x, y = ou)) +  
  theme_classic() +  
  labs(  
    title = 'OU Phylogenetic GP'  
  )
```



```
phylo_df %>%  
  ggplot() +  
  geom_point(aes(x = x, y = br)) +  
  theme_classic() +  
  labs(  
    title = 'Brownian Phylogenetic GP'  
  )
```



The stationary kernel of the Ornstein-Uhlenbeck process ensures that the value observed at every point on the tree remains close to the mean, while the Brownian motion model has much higher variance.