GP Theory

Recap of Conditional Multivariate Normal Distribution Recall the following property of a multivariate normal distribution.

$$\underline{y_1}|\underline{y_2} \sim N\left(X_1\beta + \Sigma_{12}\Sigma_{22}^{-1}\left(\underline{y_2} - X_2\beta\right), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$$

GP Overview Now let's extend this idea to a Gaussian Process (GP). There are two fundamental ideas to a GP.

1. Any finite set of realizations (say $\underline{y_2}$) has a multivariate normal distribution.

2. Conditional on a set of realizations, all other locations (say $\underline{y_1}$) have a conditional normal distribution characterized by the mean, and most importantly the covariance function. Note the dimension of $\underline{y_1}$ can actually be infinite, such as defined on the real line.

The big question is how to we estimate Σ_{12}?

Generally, Σ_{12}, or more specifically the individual elements of Σ_{12}, such as $\sigma i, j$ will be estimated using some idea of distance.

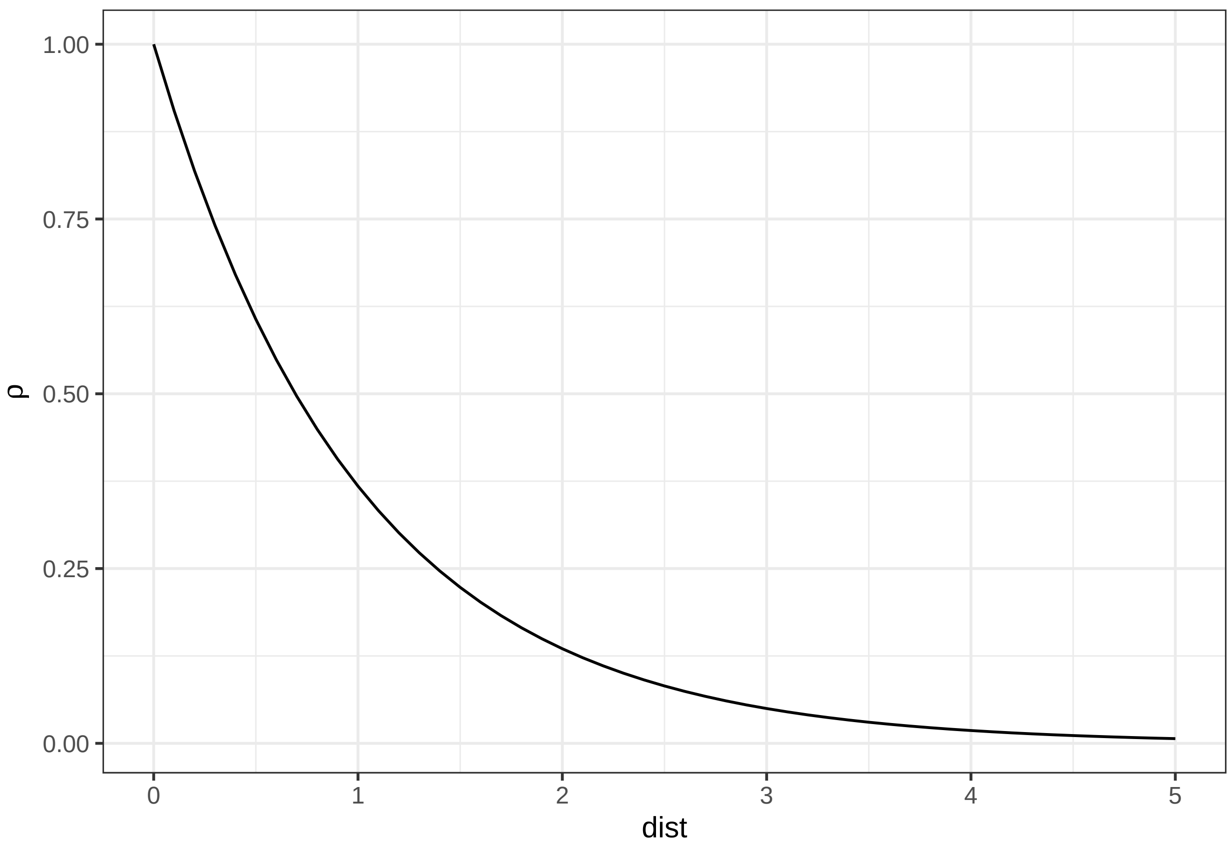
Fundamental idea of spatial statistics is that things close together tend to be similar.

Correlation function Initially, let's consider correlation as a function of distance, in one dimension or on a line.

As a starting point, consider a variant of what is known as the exponential covariance function. First define d as the Euclidean distance between x_1 and x_2, such that $d = \sqrt{(x_i - x_j)^2}$

$$\rho_{i,j} = \exp\left(-d\right)$$

Lets view the exponential correlation as a function of distance between the two points.



Using a correlation function can reduce the number of unknown parameters in a covariance matrix. In an unrestricted case, Σ has $\binom{n}{2} + n$ unknown parameters. However, using a correlation function can reduce the number of unknown parameters substantially, generally less than 4.

Realizations of a Gaussian Process Recall that a process implies an infinite dimensional object. So we can generate a line rather than a discrete set of points. (While in practice the line will in fact be generated with a discrete set of points and then connected.)

For this scenario we will assume a zero-mean GP, with covariance equal to the correlation funciton using $\rho_{i,j} = \exp\left(-d\right)$

```
x <- seq(0, 10, by = .1)
n <- length(x)
d <- plgp::distance(x)
eps <- sqrt(.Machine$double.eps)
H <- exp(-d) + diag(eps, n)
x[1:3]
```

```
## [1] 0.0 0.1 0.2
```

```
H[1:3,1:3]
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.9900498 0.9607894
## [2,] 0.9900498 1.0000000 0.9900498
## [3,] 0.9607894 0.9900498 1.0000000
```

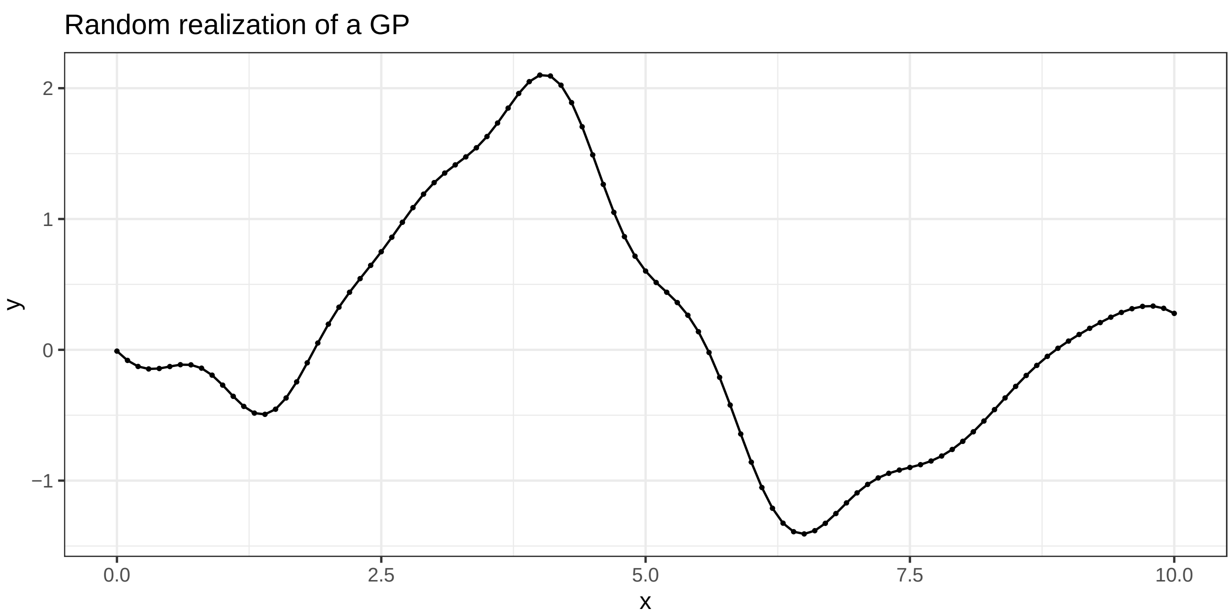
```
x[c(1,10, 50,100)]
```

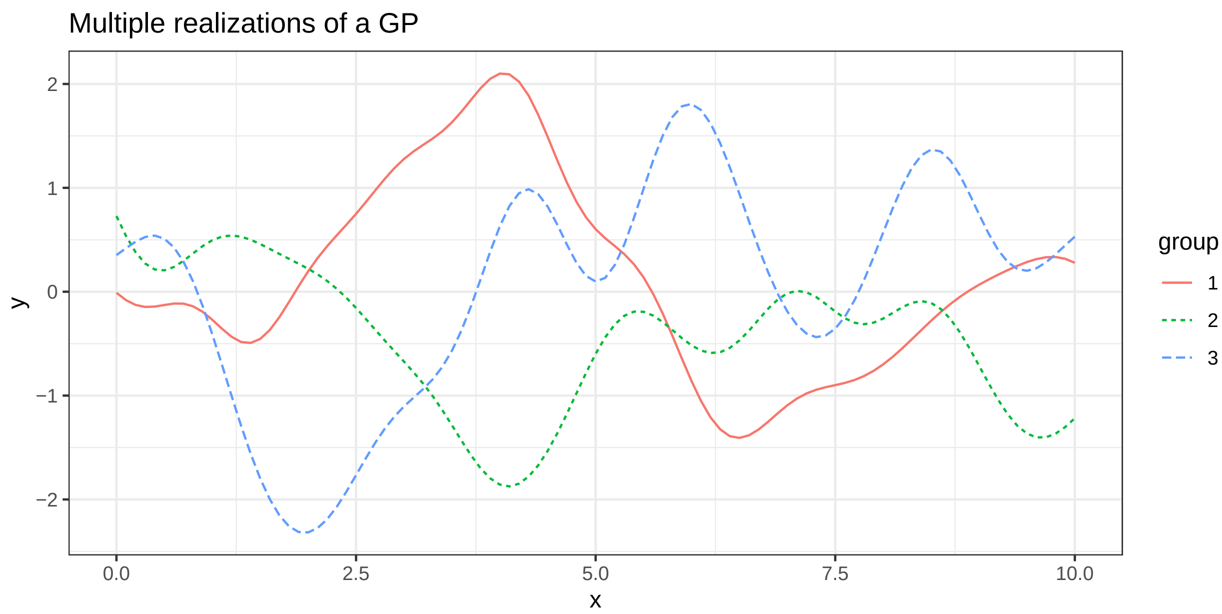
```
## [1] 0.0 0.9 4.9 9.9
```

```
H[1,c(10, 50,100)]
```

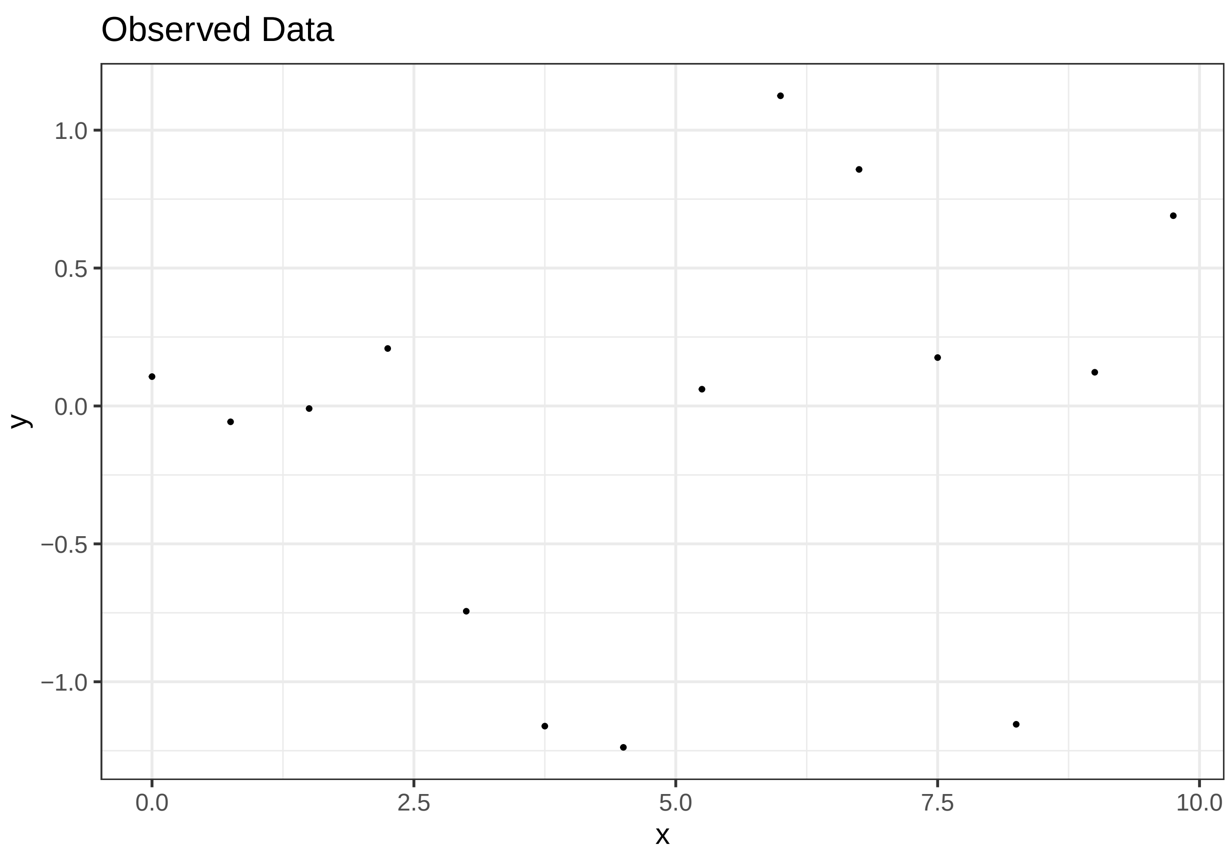
```
## [1] 4.448581e-01 3.737571e-11 2.721434e-43
```

```
y <- rmnorm(1, rep(0,n),H)
```





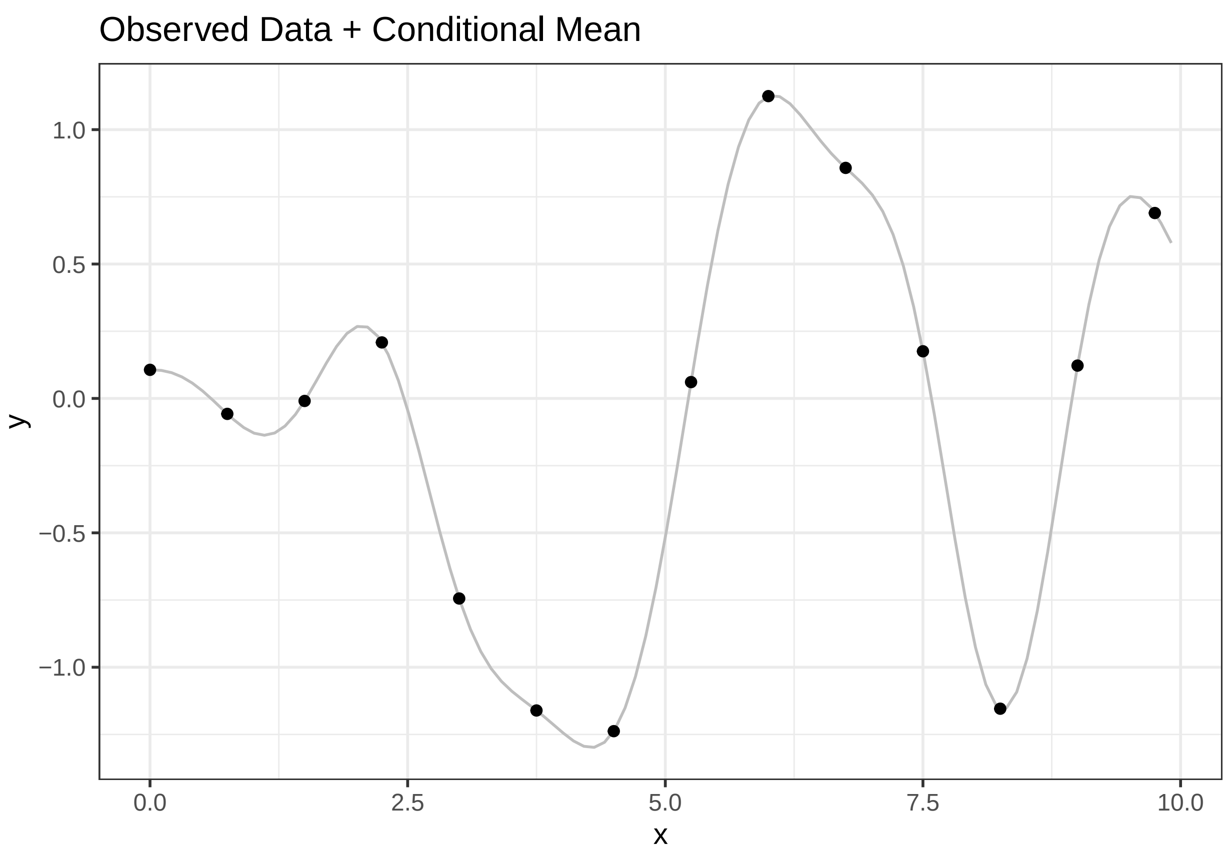
Connecting a GP to conditional normal Now consider a discrete set of points, say $\underline{y_2}$, how can we estimate the response for the remainder of the values in the interval [0,1].



We can connect the dots (with uncertainty) using:

$$\underline{y_1}|\underline{y_2} \sim N\left(X_1\beta + \Sigma_{12}\Sigma_{22}^{-1}\left(\underline{y_2} - X_2\beta\right), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$$

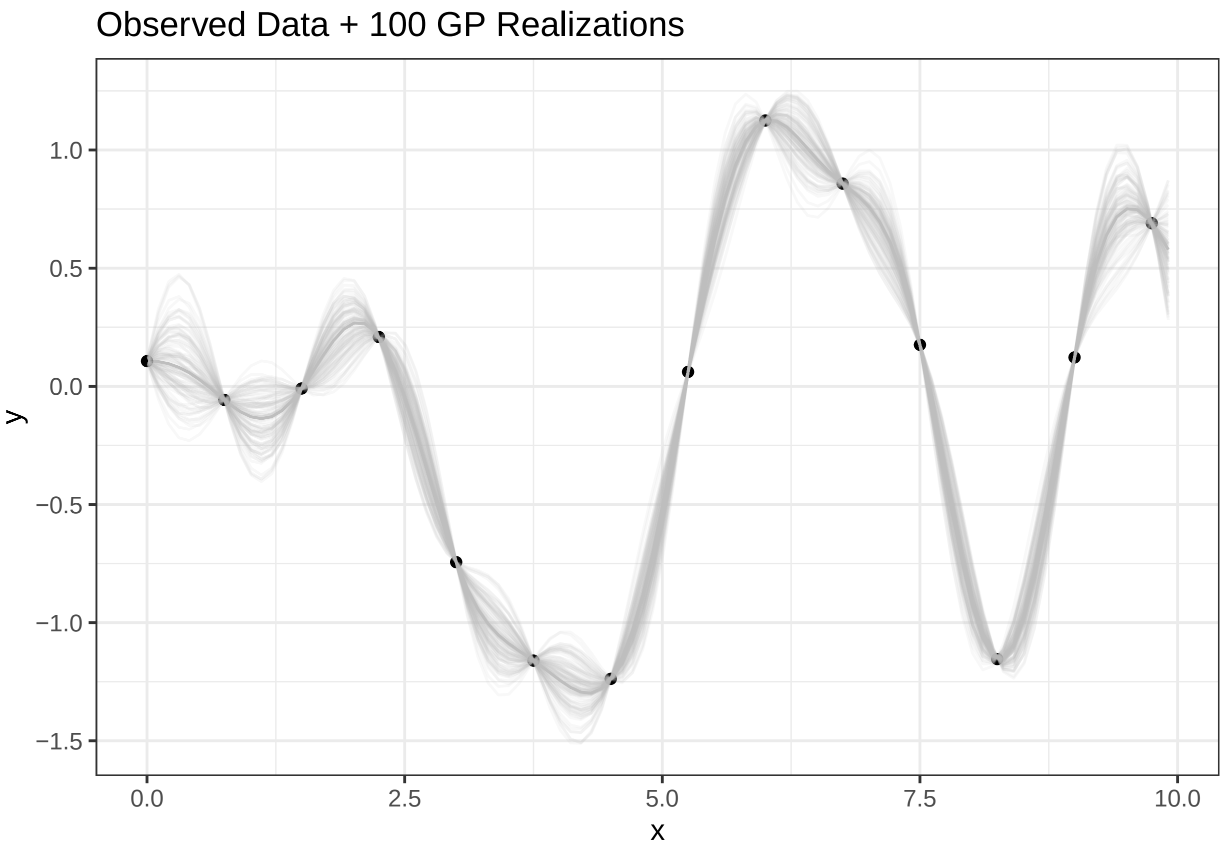
```
x1 <- seq(0.01, 10, .1)
n <- length(x1)
d1 <- plgp::distance(x1)
H11 <- exp(-d1) + diag(eps, n)
d12 <- plgp::distance(x1,x2)
H12 <- exp(-d12)
mu_1given2 <- H12 %*% solve(H22) %*% matrix(y2, nrow = length(y2), ncol = 1)
Sigma_1given2 <- H11 - H12 %*% solve(H22) %*% t(H12)
```



```
num_sims <- 100
y1_sims <- rmnorm(num_sims, mu_1given2, Sigma_1given2)

long_sims <- y1_sims %>% melt() %>% bind_cols(tibble(x = rep(x1, each = num_sims)))

data_and_mean +
  geom_line(aes(y = value, x = x, group = Var1), inherit.aes = F,
            data = long_sims, alpha = .1, color = 'gray') +
  ggtitle('Observed Data + 100 GP Realizations')
```



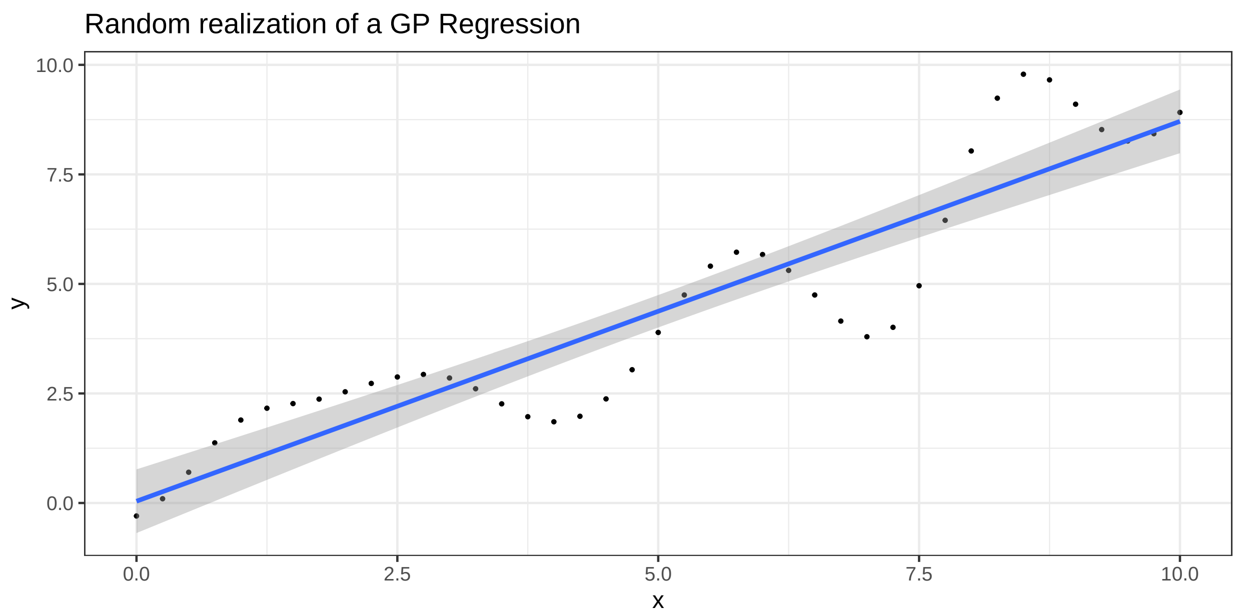
You can also calculate and plot quantile-based intervals.

```
quantiles <- apply(y1_sims,1, quantile, probs = c(.025,.975))[1:2,1:5]
```

GP Regression

Now rather than specifying a zero-mean GP, let the mean be $X\underline{\beta}$.

```
x <- seq(0, 10, by = .25)
beta <- 1
n <- length(x)
d <- plgp::distance(x)
eps <- sqrt(.Machine$double.eps)
H <- exp(-d) + diag(eps, n)
y <- rmnorm(1, x * beta ,H)
```



More details on fitting GP regression models with STAN: https://mc-stan.org/docs/2_19/stan-users-guide/gaussian-processes-chapter.html

GP in 2D (or spatial Kriging)

Q: How does this differ in two (or more) dimensions? *Actually there is very little difference (beyond the challenge of data visualization). The correlation function requires the Euclidean distance in 2-d rather than 1-d.*