```
> X <- seq(0, 1, length = 50)
> XX <- seq(0, 1, length = 99)
> Z <- 1 + 2 * X + rnorm(length(X), sd = 0.25)
```

Using `tgp` on this data with a Bayesian hierarchical linear model goes as follows:

```
> lin.blm <- blm(X = X, XX = XX, Z = Z)

tree[alpha,beta,nmin]=[0,0,10]
n=50, d=1, nn=99
BTE=(1000,4000,3), R=1, linburn=0
preds: data
linear prior: flat
s2[a0,g0]=[5,10]
s2 lambda[a0,g0]=[0.2,10]
corr prior: separable power
nug[a,b][0,1]=[1,1],[1,1]
nug prior fixed
gamlin=[-1,0.2,0.7]
d[a,b][0]=[1,20],[10,20]
d prior fixed

burn in:
r=1000 corr=[0] : n = 50

Obtaining samples (nn=99 predictive locations):
r=1000 corr=[0] : mh=1 n = 50
r=2000 corr=[0] : mh=1 n = 50
r=3000 corr=[0] : mh=1 n = 50

finished repetition 1 of 1
removed 0 leaves from the tree
```

The first group of text printed to `stdout` is a summary of inputs to the `C` code, and the prior parameterization. Then, MCMC progress indicators are printed every 1,000 rounds. The linear model is indicated by `cor=[0]`. In terminal versions, e.g. `Unix`, the progress indicators can give a sense of when the code will finish. GUI versions of `R`—`Windows` or `MacOS X`—usually buffer `stdout`, rendering this feature essentially useless as a real-time indicator of progress.

The generic `plot` method can be used to visualize the fitted posterior predictive surface (with option `layout = 'surf'`) in terms of means and credible intervals. Figure 3 shows how to do it, and what you get. The default option `layout = 'both'` shows both a predictive surface and error (or uncertainty) plot, side by side. The error plot can be obtained alone via `layout = 'as'`. Examples of these layouts appear later.

```
> plot(lin.blm, main = "Linear Model,", layout = "surf")
> abline(1, 2, lty = 3, col = "blue")
```

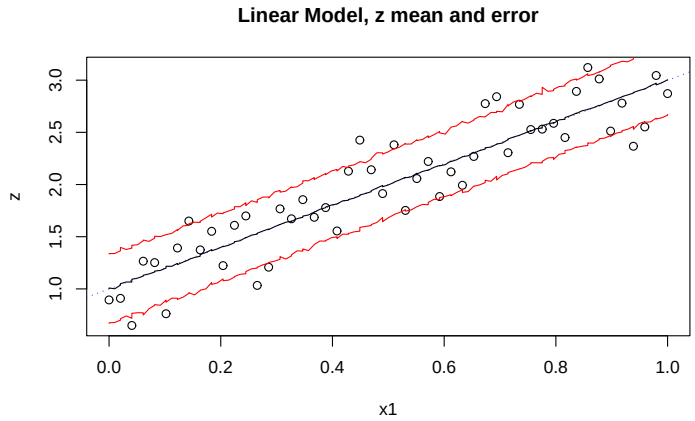

Figure 3: Posterior predictive distribution using `blm` on synthetic linear data: mean and 90% credible interval. The actual generating lines are shown as blue-dotted.

If, say, you were unsure about the dubious "linearness" of this data, you might try a GP LLM (using `btgpllm`) and let a more flexible model speak as to the linearity of the process.

```
> lin.gpllm <- bgpllm(X = X, XX = XX, Z = Z)

tree[alpha,beta,nmin]=[0,0,10]
n=50, d=1, nn=99
BTE=(1000,4000,2), R=1, linburn=0
preds: data
linear prior: flat
s2[a0,g0]=[5,10]
s2 lambda[a0,g0]=[0.2,10]
corr prior: separable power
nug[a,b][0,1]=[1,1],[1,1]
nug prior fixed
gamlin=[10,0.2,0.7]
d[a,b][0]=[1,20],[10,20]
d prior fixed

burn in:
r=1000 corr=[0] : n = 50

Obtaining samples (nn=99 predictive locations):
r=1000 corr=[0] : mh=1 n = 50
r=2000 corr=[0] : mh=1 n = 50
```