```
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
```

```
> plot(lin.gpllm, main = "GP LLM,", layout = "surf")
> abline(1, 2, lty = 4, col = "blue")
```

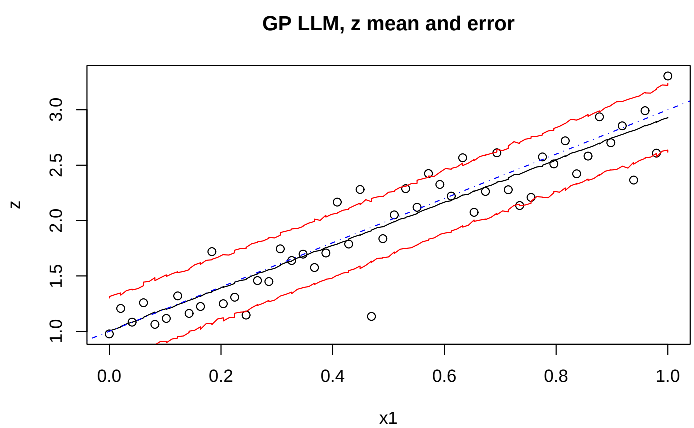


Figure 4: Posterior predictive distribution using `bgpllm` on synthetic linear data: mean and 90% credible interval. The actual generating lines are shown as blue-dotted.

Whenever the progress indicators show `corr[0]` the process is under the LLM in that round, and the GP otherwise. A plot of the resulting surface is shown in Figure 4 for comparison. Since the data is linear, the resulting predictive surfaces should look strikingly similar to one another. On occasion, the GP LLM may find some bendy-ness in the surface. This happens rarely with samples as large as $N = 50$, but is quite a bit more common for $N < 20$.

3.2 1-d Synthetic Sine Data

Consider 1-dimensional simulated data which is partly a mixture of sines and cosines, and partly linear.

$$z(x) = \begin{cases} \sin\left(\frac{\pi x}{5}\right) + \frac{1}{5}\cos\left(\frac{4\pi x}{5}\right) & x < 10 \\ x/10 - 1 & \text{otherwise} \end{cases} \tag{14}$$

The `R` code below obtains $N = 100$ evenly spaced samples from this data in the domain $[0, 20]$, with noise added to keep things interesting. Some evenly spaced predictive locations `XX` are also created.

```
> X <- seq(0, 20, length = 100)
> XX <- seq(0, 20, length = 99)
> Z <- (sin(pi * X/5) + 0.2 * cos(4 * pi * X/5)) *
+     (X <= 9.6)
> lin <- X > 9.6
> Z[lin] <- -1 + X[lin]/10
> Z <- Z + rnorm(length(Z), sd = 0.1)
```

By design, the data is clearly nonstationary. Perhaps not knowing this, good first model choice for this data might be a GP.

```
> sin.bgp <- bgp(X = X, Z = Z, XX = XX)
```

```
> plot(sin.bgp, main = "GP,", layout = "surf")
```

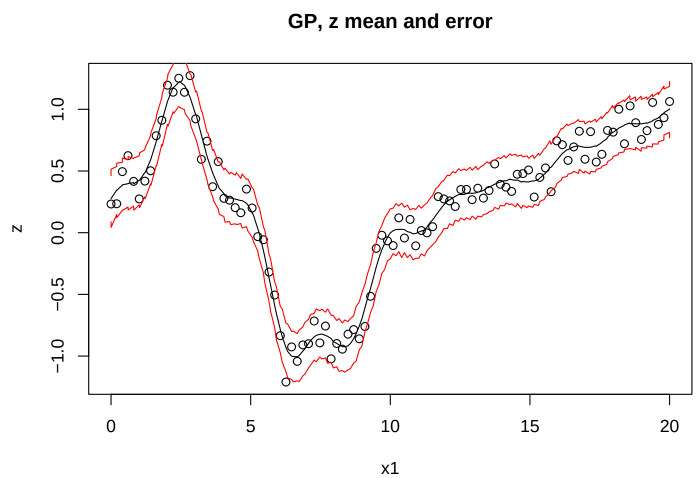


Figure 5: Posterior predictive distribution using `bgp` on synthetic sinusoidal data: mean and 90% credible interval

Progress indicators have been suppressed. Figure 5 shows the resulting posterior predictive surface under the GP. Notice how the (stationary) GP gets the wiggliness of the sinusoidal region, but fails to capture the smoothness of the linear region. This is becuase the data comes from a process that is nonstationary.

So one might consider a Bayesian CART model instead.

```
> sin.btlm <- btlm(X = X, Z = Z, XX = XX)
```

```
n=100, d=1, nn=99
BTE=(2000,7000,2), R=1, linburn=0
preds: data
tree[alpha,beta,nmin]=[0.25,2,10]
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:
**GROW** @depth 0: [0,0.494949], n=(50,50)
**GROW** @depth 1: [0,0.20202], n=(21,29)
**GROW** @depth 1: [0,0.10101], n=(11,10)
**GROW** @depth 3: [0,0.353535], n=(11,11)
r=1000 corr=[0] [0] [0] [0] [0] : n = 10 14 10 13 53
**PRUNE** @depth 3: [0,0.222222]
r=2000 corr=[0] [0] [0] [0] : n = 10 20 17 53

Obtaining samples (nn=99 predictive locations):
r=1000 corr=[0] [0] [0] [0] : mh=4 n = 10 21 16 53
r=2000 corr=[0] [0] [0] [0] : mh=4 n = 15 15 17 53
r=3000 corr=[0] [0] [0] [0] : mh=4 n = 15 15 17 53
r=4000 corr=[0] [0] [0] [0] : mh=4 n = 15 15 17 53
r=5000 corr=[0] [0] [0] [0] : mh=4 n = 16 14 17 53
Grow: 0.01072%, Prune: 0.002899%, Change: 0.1496%, Swap: 0.7925%

finished repetition 1 of 1
removed 4 leaves from the tree
```

MCMC progress indicators printed to `stdout` indicate successful *grow* and *prune* operations as they happen, and region sizes n every 1,000 rounds.

Figure 6 shows the resulting posterior predictive surface (*top*) and trees (*bottom*). The MAP partition ($\hat{\mathcal{T}}$) is also drawn onto the surface plot (*top*) in the form of vertical lines. The CART model captures the smoothness of the linear region just fine, but comes up short in the sinusoidal region—doing the best it can with piecewise linear models.

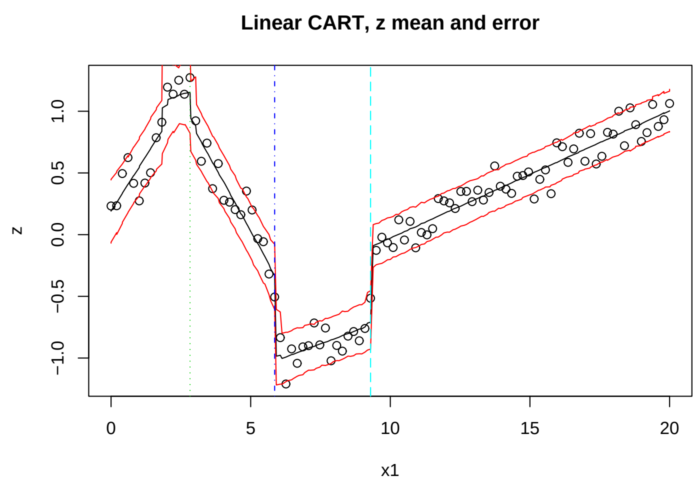
The ideal model for this data is the Bayesian treed GP becuase it can be both smooth and wiggly.

```
> sin.btgp <- btgp(X = X, Z = Z, XX = XX)
```

Progress indicators have been suppressed. Figure 7 shows the resulting posterior predictive surface (*top*) and trees (*bottom*).

Finally, speedups can be obtained if the GP is allowed to jump to the LLM [10], since half of the response surface is *very* smooth, or linear. This is not shown here since the results are very similar to those above, replacing `btgp`

```
> plot(sin.btlm, main = "Linear CART,", layout = "surf")
```




```
> tgp.trees(sin.btlm)
```

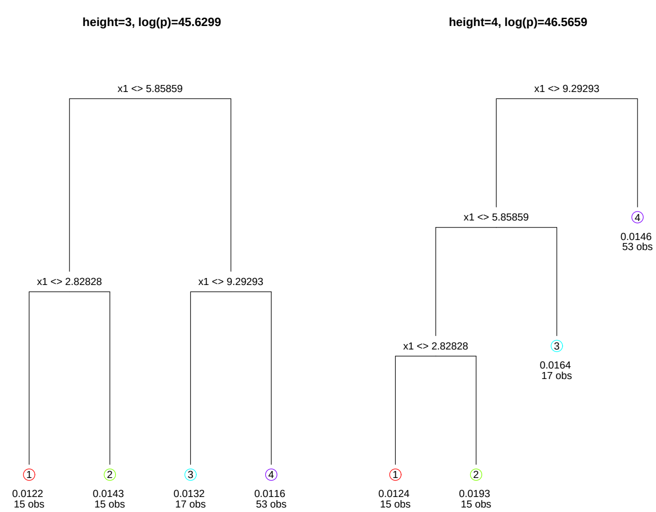
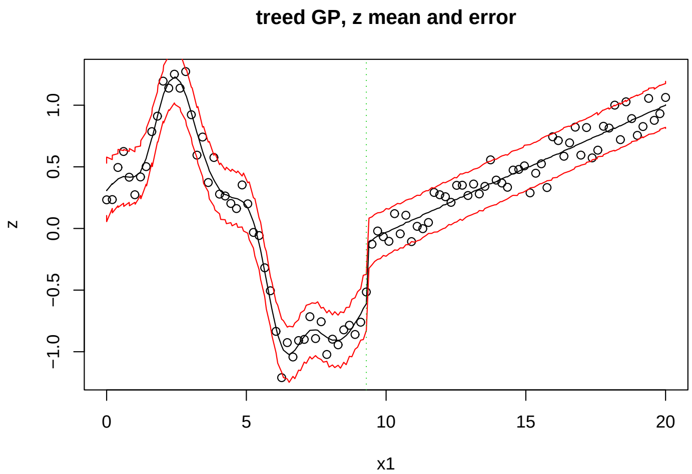


Figure 6: *Top:* Posterior predictive distribution using `btlm` on synthetic sinusoidal data: mean and 90% credible interval, and MAP partition ($\hat{\mathcal{T}}$); *Bottom* MAP trees for each height encountered in the Markov chain showing $\hat{\sigma}^2$ and the number of observation n, at each leaf.

with `btgpllm`. The example in the next subsection offers a comparison for 2-d data.




```
> tgp.trees(sin.btgp)
```

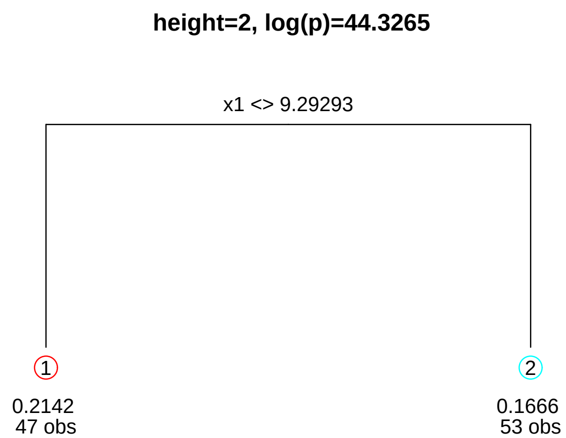


Figure 7: *Top:* Posterior predictive distribution using **btgp** on synthetic sinusoidal data: mean and 90% credible interval, and MAP partition ($\hat{\mathcal{T}}$); *Bottom* MAP trees for each height encountered in the Markov chain.

3.3 Synthetic 2-d Exponential Data

The next example involves a two-dimensional input space in $[-2, 6] \times [-2, 6]$. The true response is given by

$$z(\mathbf{x}) = x_1 \exp(-x_1^2 - x_2^2). \tag{15}$$