

# synthetic datasets

Warning: package 'bartMachine' was built under R version 4.3.3

Warning: package 'randomForest' was built under R version 4.3.3

Warning: package 'missForest' was built under R version 4.3.3

Warning: package 'dbarts' was built under R version 4.3.3

Warning: package 'BART' was built under R version 4.3.3

Warning: package 'bench' was built under R version 4.3.3

## create dataset

```
linear_dgp_fun <- function(n_train, n_test, p, beta, noise_sd) {  
  n <- n_train + n_test  
  X <- matrix(rnorm(n * p), nrow = n, ncol = p)  
  y <- X %*% beta + rnorm(n, sd = noise_sd)  
  data_list <- list(  
    X_train = X[1:n_train, , drop = FALSE],  
    y_train = y[1:n_train],  
    X_test = X[(n_train + 1):n, , drop = FALSE],  
    y_test = y[(n_train + 1):n]  
  )  
  return(data_list)  
}  
linear_dgp <- create_dgp(  
  .dgp_fun = linear_dgp_fun, .name = "Linear DGP",  
  # additional named parameters to pass to .dgp_fun()  
  n_train = 350, n_test = 120, p = 4, beta = c(1,2,1.5,3), noise_sd = 1  
)
```

## build BART model

```
BART_fun <- function(X_train, y_train, X_test, y_test, num_trees,alpha,beta) {  
  train_df <- dplyr::bind_cols(data.frame(X_train), y = y_train)  
  t <- bench::mark(fit <- pbart(x.train = X_train,  
                                y.train = y_train,  
                                x.test = X_test,  
                                ntree = num_trees,  
                                base = alpha,  
                                power = beta))  
  
  time <- mean(t$time[[1]])  
  predictions <- colMeans(fit$yhat.test)  
  mse_score <- mean((y_test - predictions)^2)  
  
  return(list(time = time, mse = mse_score))  
}
```

```
BART <- create_method(  
  .method_fun = BART_fun, .name = "BART",  
  # additional named parameters to pass to .method_fun()  
  num_trees=20,alpha=0.95,beta=2  
)  
  
# Create experiment  
experiment <- create_experiment(name = "Test Experiment") %>%  
  add_dgp(linear_dgp) %>%  
  
  add_method(BART) %>%  
  
  # vary across noise parameter in linear dgp  
  add_vary_across(  
    .dgp = "Linear DGP",  
    noise_sd = c(0.1, 0.5, 1, 2)  
  )  
  
results <- run_experiment(experiment, n_reps = 4, save = TRUE)
```

Fitting Test Experiment...

Warning in qnorm(mean(y.train)): NaNs produced

Warning in qnorm(mean(y.train)): NaNs produced

```
Warning in qnorm(mean(y.train)): NaNs produced
```

```
Warning: Some expressions had a GC in every iteration; so filtering is disabled.
```

```
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
Warning in qnorm(mean(y.train)): NaNs produced
```

```
Saving fit results...
```

```
Fit results saved | time taken: 0.025581 seconds
```

```
4 reps completed (totals: 4/4) | time taken: 0.415921 minutes
```

```
=====
```

```
No evaluators to evaluate. Skipping evaluation.
```

```
=====
```

```
No visualizers to visualize. Skipping visualization.
```

```
=====
```

```
# Render automated documentation and view results
```

```
#render_docs(experiment)
```

```
results$fit_results
```

```
# A tibble: 16 x 6
```

	.rep	.dgp_name	.method_name	noise_sd	time	mse
	<chr>	<chr>	<chr>	<dbl>	<list>	<dbl>
1	1	Linear DGP	BART	0.1	<bench_tm [1]>	NaN
2	1	Linear DGP	BART	0.5	<bench_tm [1]>	14.4
3	1	Linear DGP	BART	1	<bench_tm [1]>	NaN

4	1	Linear	DGP	BART	2	<bench_tm [1]>	24.4
5	2	Linear	DGP	BART	0.1	<bench_tm [1]>	18.1
6	2	Linear	DGP	BART	0.5	<bench_tm [1]>	NaN
7	2	Linear	DGP	BART	1	<bench_tm [1]>	17.0
8	2	Linear	DGP	BART	2	<bench_tm [1]>	NaN
9	3	Linear	DGP	BART	0.1	<bench_tm [1]>	15.1
10	3	Linear	DGP	BART	0.5	<bench_tm [1]>	19.8
11	3	Linear	DGP	BART	1	<bench_tm [1]>	NaN
12	3	Linear	DGP	BART	2	<bench_tm [1]>	NaN
13	4	Linear	DGP	BART	0.1	<bench_tm [1]>	18.7
14	4	Linear	DGP	BART	0.5	<bench_tm [1]>	16.5
15	4	Linear	DGP	BART	1	<bench_tm [1]>	15.7
16	4	Linear	DGP	BART	2	<bench_tm [1]>	21.6