Climate and Institutions

Daniel K Baissa

3/5/2022

Data Setup

For this project I will use two datasets one is a shapefile that contains country data bio 12 bio six soil data and globe cover and the other is the grand data set that we have from the banks data as well as polity.

```
library(readr)
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.1.2
```

```
## Linking to GEOS 3.9.1, GDAL 3.2.1, PROJ 7.2.1; sf_use_s2() is TRUE
```

```
raw_df <- df<- read_csv("Data/cleaned_grand_data.csv")
```

```
## New names:
## * `` -> ...1
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 14563 Columns: 4374
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (30): country, ccid, code, country_name, histname, v2lpname, v2slpnam...
## dbl  (4340): ...1, year, area1, area2, area3, computer1, computer2, computer...
## lgl     (2): v3elupvtlg, v3elupvtsm
## date    (1): historical_date
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
World_data <- read_sf("Shapefiles/country_bio12_bio6_soil_globcover.shp")
```

For the purposes of this project I do not need the shapefiles geometry so I will convert it to a dataframe and then I will merge it with the grand dataset.

```
World_data2 <- as.data.frame(World_data)
```

Now I will create a variable that captures the percentage of soil types and terrain useage by country.

```
d <- merge(df, World_data2, by.x = "country", by.y = "COUNTRY")
```

Now let's select data greater than 2005

```
d <- d %>%
  filter(year > 2005) %>%
  # select(c(economics2, energy1, contains("Soil_HIS"), contains("Globcove"), contains("bio")))
  select(c(economics2, energy1, contains("bio"))) %>%
  mutate(abs_temp = abs(bio6_media)) %>%
  select(c(abs_temp, economics2, energy1, bio12_vari ))
  # select(c(abs_temp, economics2, energy1, bio12_medi)) %>%
  # mutate(log_abs_temp = log(abs_temp+1)) %>%
  # mutate(log_rainfall = log(bio12_medi)) %>%
  # select(c(log_abs_temp, economics2, energy1, log_rainfall))
```

```
 Economics1 is devoted to national income per capita, economics2 to gross domestic product (at factor c
      Gross national product at market prices is the market value of the product, before
      deduction of provisions for the consumption of fixed capital, attributable to the
      factors of production supplied by normal residents of the given country. It is
      identically equal to the sum of consumption expenditure and gross domestic capital
      formation, private and public, and the net exports of goods and services plus the net
      factor incomes received from abroad.
      Gross domestic product at factor cost is the value at factor cost of the product,
      before deduction of provisions for the consumption of fixed capital, attributable to
      factor services rendered to resident producers of the given country. It differs from
      the gross domestic product at market prices by the exclusion of the excess of
      indirect taxes over subsidies.
      National income is the sum of the incomes accruing to factors of production
      supplied by normal residents of the given country before deduction of direct taxes.
      (UN Yearbook of National Accounts Statistics, 1969, v. 1, p. xi.)
```

Now let's remove all of the NAs for economics 3

BART Model

Now we can set up and run the bartMachine.

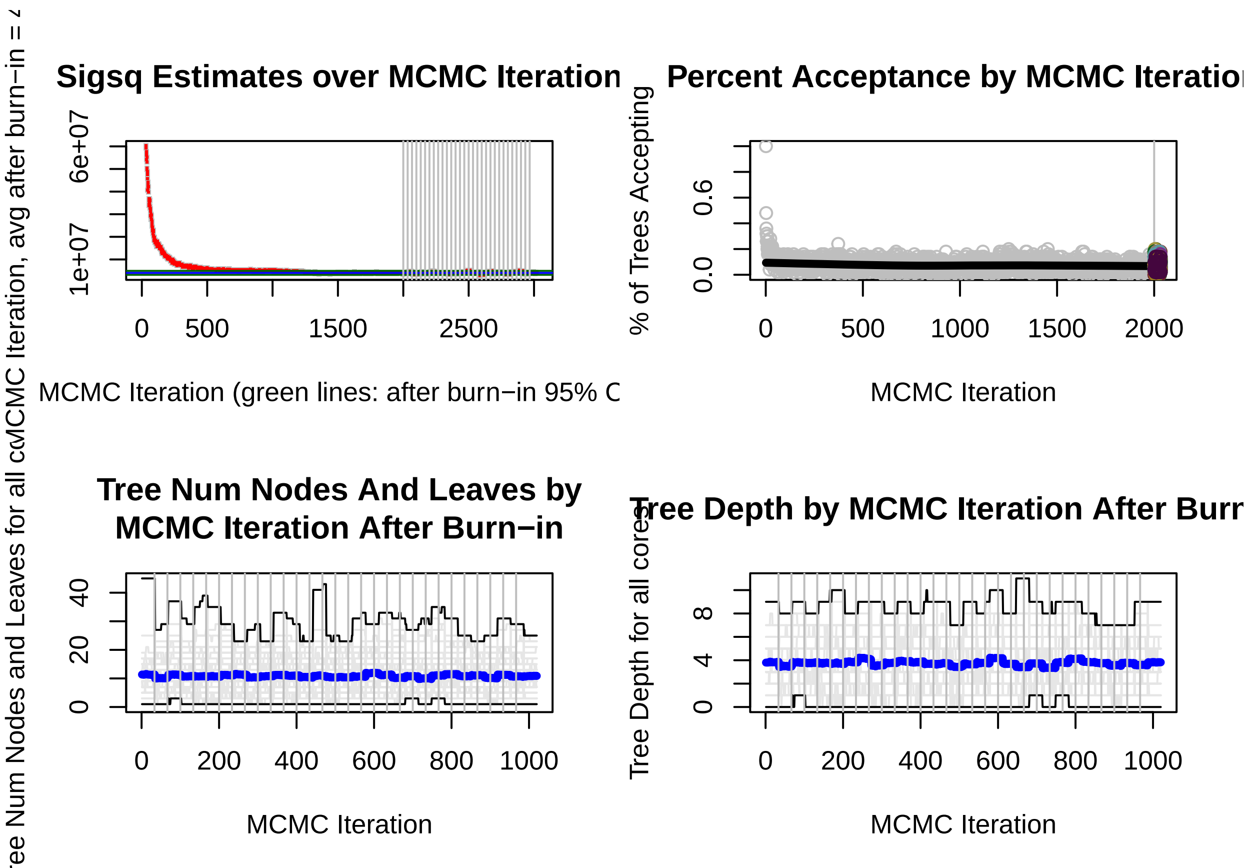
```
bart_machine = bartMachine(X = X, y = y, num_burn_in = 2000)
```

```
## bartMachine initializing with 50 trees...
## bartMachine vars checked...
## bartMachine java init...
## bartMachine factors created...
## bartMachine before preprocess...
## bartMachine after preprocess... 4 total features...
## bartMachine sigsq estimated...
## bartMachine training data finalized...
## Now building bartMachine for regression...
## evaluating in sample data...done
```

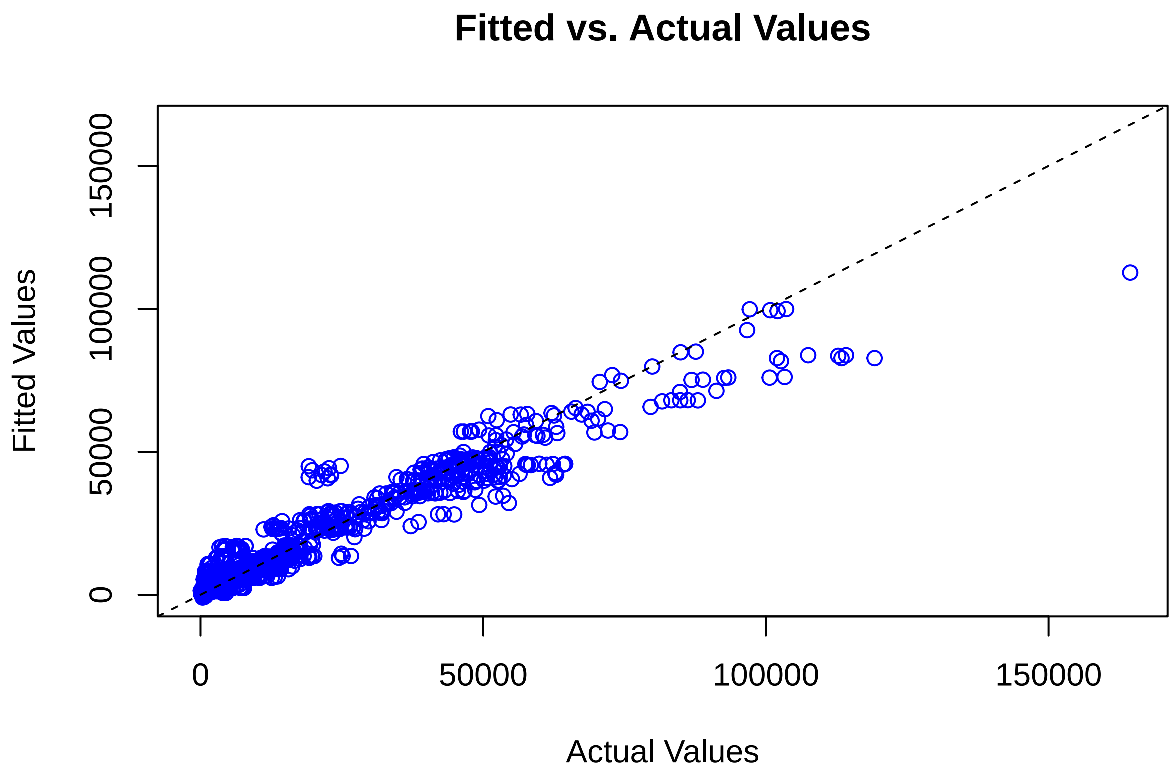
```r
bart_machine
```

```
## bartMachine v1.2.6 for regression
##
## training data n = 1697 and p = 3
## built in 2.81 mins on 30 cores, 50 trees, 2000 burn-in and 1000 post. samples
##
## sigsq est for y beforehand: 334602045.066
## avg sigsq estimate after burn-in: 4058820.23635
##
## in-sample statistics:
##  L1 = 4175695.5
##  L2 = 41031355641.33
##  rmse = 4917.19
##  Pseudo-Rsq = 0.9377
## p-val for shapiro-wilk test of normality of residuals: 0
## p-val for zero-mean noise: 0.30673
```

```r
plot_convergence_diagnostics(bart_machine)
```



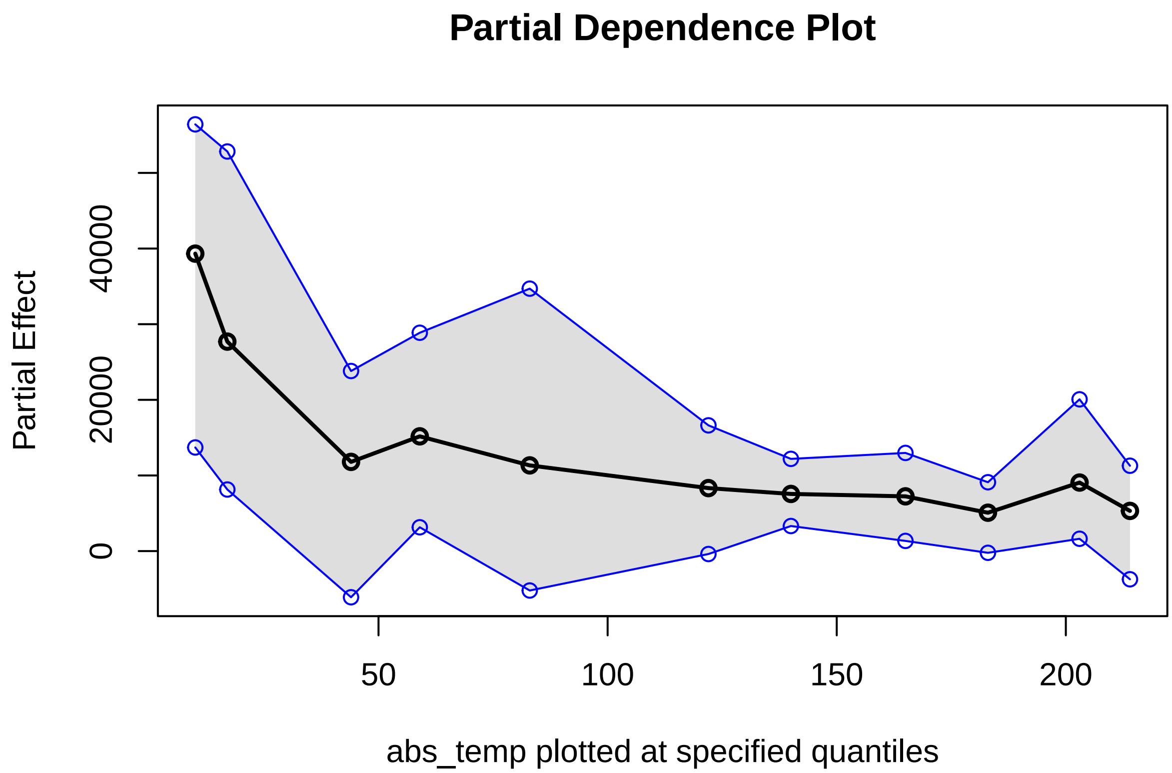

```r
plot_y_vs_yhat(bart_machine)
```



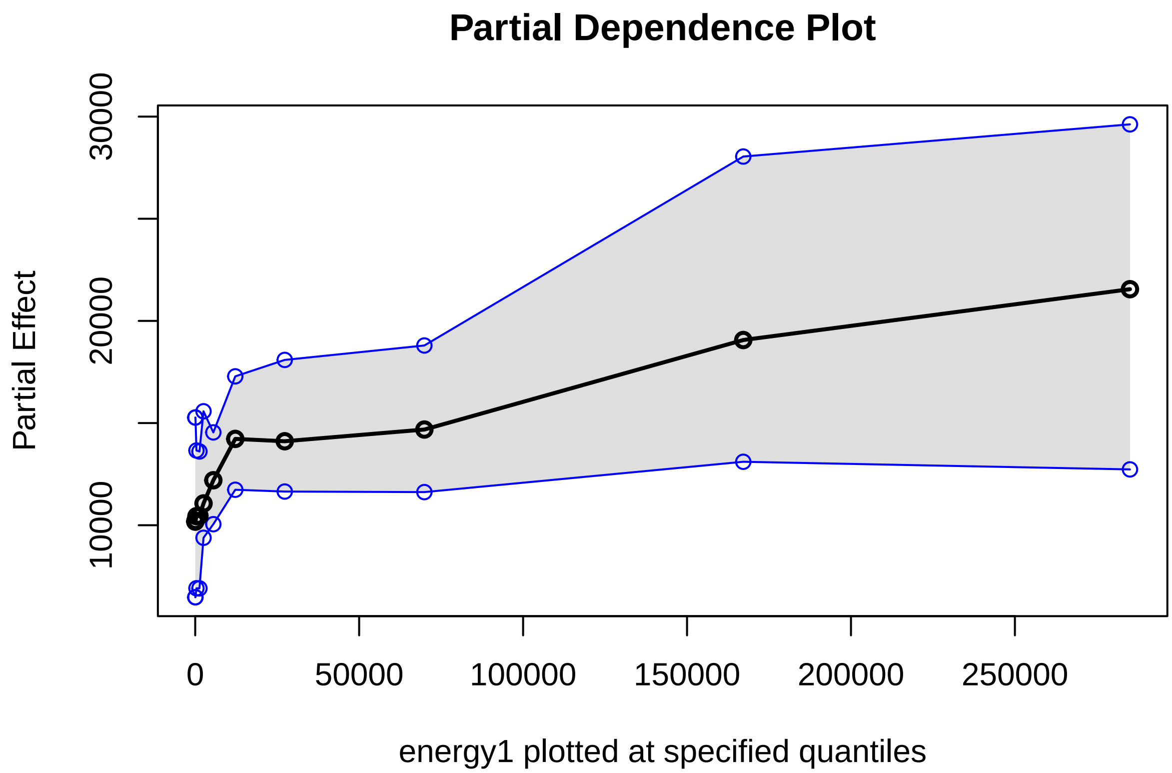
Partial Dependance Plots

```
try(

  for (i in 1:length(bart_machine[["training_data_features"]])) {

      pd_plot(bart_machine,  bart_machine[["training_data_features"]][i])

  }
)
```

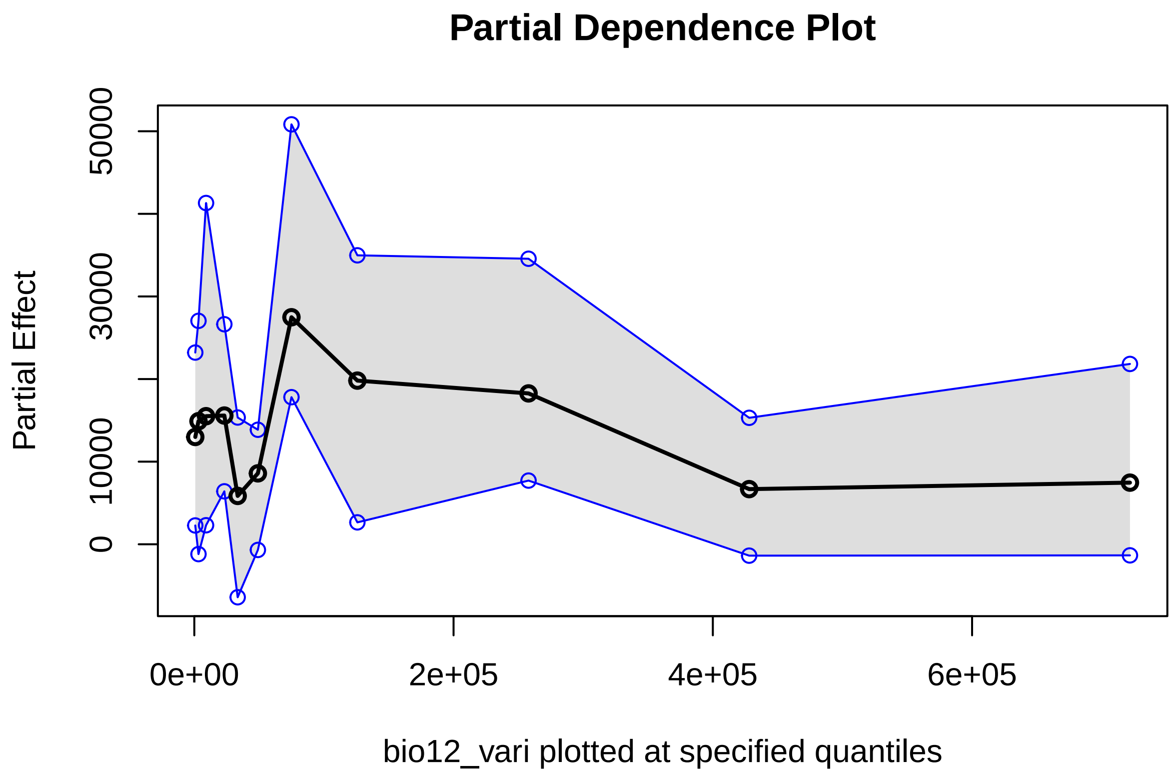
```
## ...........
```


Partial Dependence Plot
Partial Effect
0
20000
40000
50
100
150
200
abs_temp plotted at specified quantiles

```
## ...........
```


Partial Dependence Plot
Partial Effect
30000
20000
10000
0
50000
100000
150000
200000
250000
energy1 plotted at specified quantiles

```
## ...........
```


Partial Dependence Plot
Partial Effect
0
10000
30000
50000
0e+00
2e+05
4e+05
6e+05
bio12_vari plotted at specified quantiles