Simulations and Comparisons

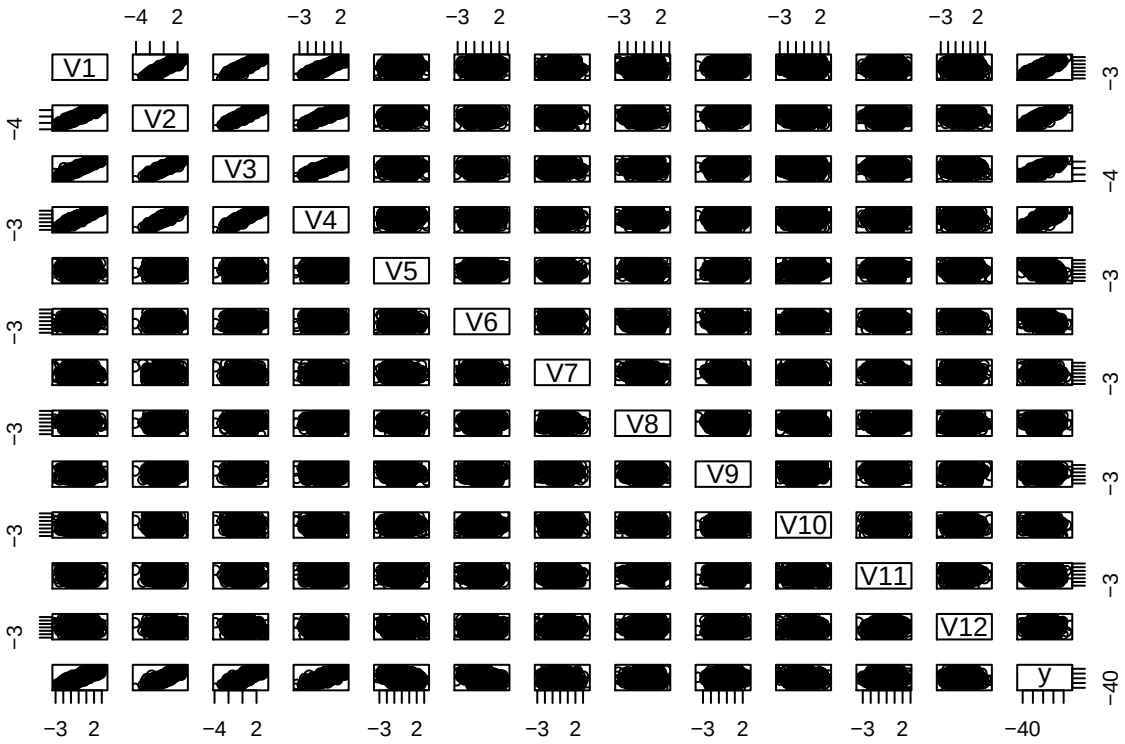
Simulated Data

- From Strobl et al:

To aid in comparisons between the methods, one of the simulated datasets considered in this paper will be generated from the same method as used in (Strobl et al, 2008???). Under this method, the 13 x 1000 data set, D_1, has 12 predictors, $V_1, .., V_{12}$, where $V_j \sim N(0, 1)$. The first four are, however, block correlated to each other with $\rho = .9$. They are related to Y by the linear equation:

$$Y = 5 \cdot V_1 + 5 \cdot V_2 + 2 \cdot V_3 + 0 \cdot V_4 + -5 \cdot V_5 + -5 \cdot V_6 + 0 \cdot V_7 + 0 \cdot ..... + E, E \sim N(0, \frac{1}{2})$$

Note that the coefficents for $V_7, ..., V_{12}$ are all zero.



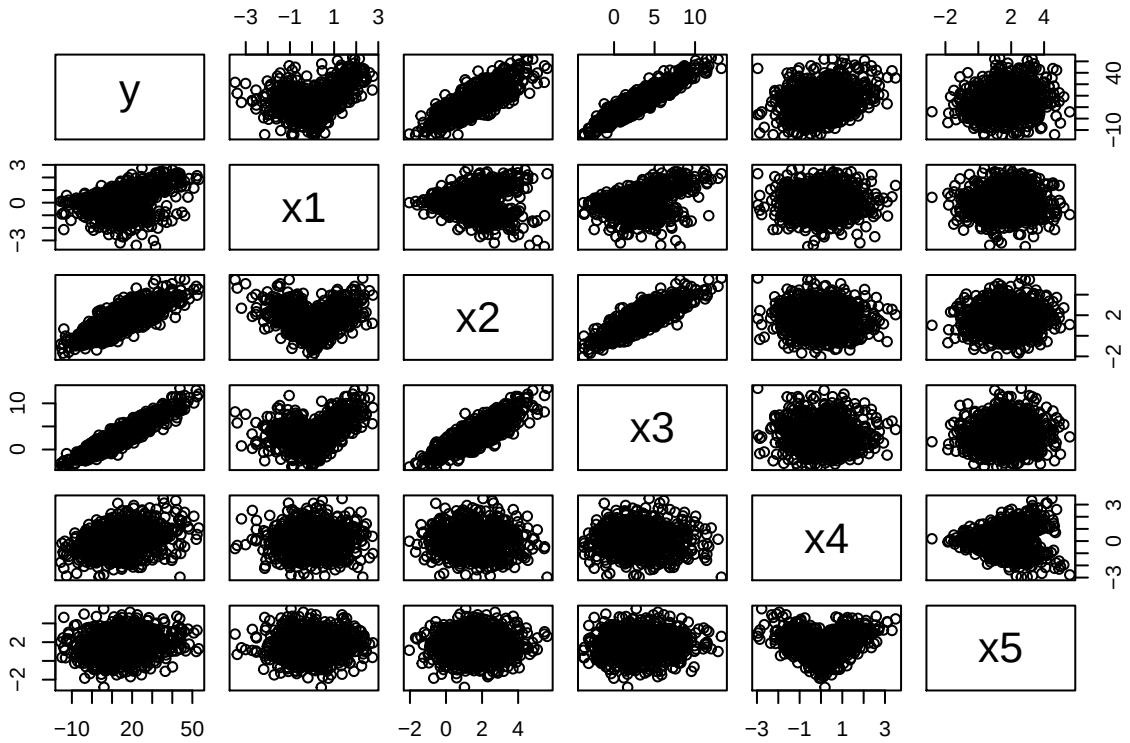
Let's move on to a more difficult situation. The dataset $D2$ contains five predictors, $X_1, ...X_5$, that have an interesting structure- several of the predictors are correlated, but are not one -to -one. This violates an important assumtion of the linear model and means that these variables have low correlation. Note that this only makes sense in higher dimensions where we are estimating the value of X_j given $X_1, ..., X_n$.

```
x1 <- rnorm(1000)

x2 <- 2*sqrt(abs(x1)) + rnorm(1000)

x3 <- x1 + 2*x2 + rnorm(1000)

x4 <- rnorm(1000)

x5 <- 2*sqrt(abs(x4)) + rnorm(1000)

y <- x1 + 2*x2 + 3 * x3 + 4*x4 + 1*x5 + rnorm(1000)
```

```
d2 <- data.frame(y,x1,x2,x3,x4,x5)

cor(d2)
```

```
##             y         x1          x2          x3          x4         x5
## y  1.0000000 0.37370066  0.81852208  0.92926643  0.30615522 0.15357237
## x1 0.3737007 1.00000000  0.01253512  0.37828457  0.02553116 0.01196494
## x2 0.8185221 0.01253512  1.00000000  0.85944303 -0.04770287 0.04253338
## x3 0.9292664 0.37828457  0.85944303  1.00000000 -0.03116991 0.04579883
## x4 0.3061552 0.02553116 -0.04770287 -0.03116991  1.00000000 0.01984835
## x5 0.1535724 0.01196494  0.04253338  0.04579883  0.01984835 1.00000000
```

```
plot(d2)
```



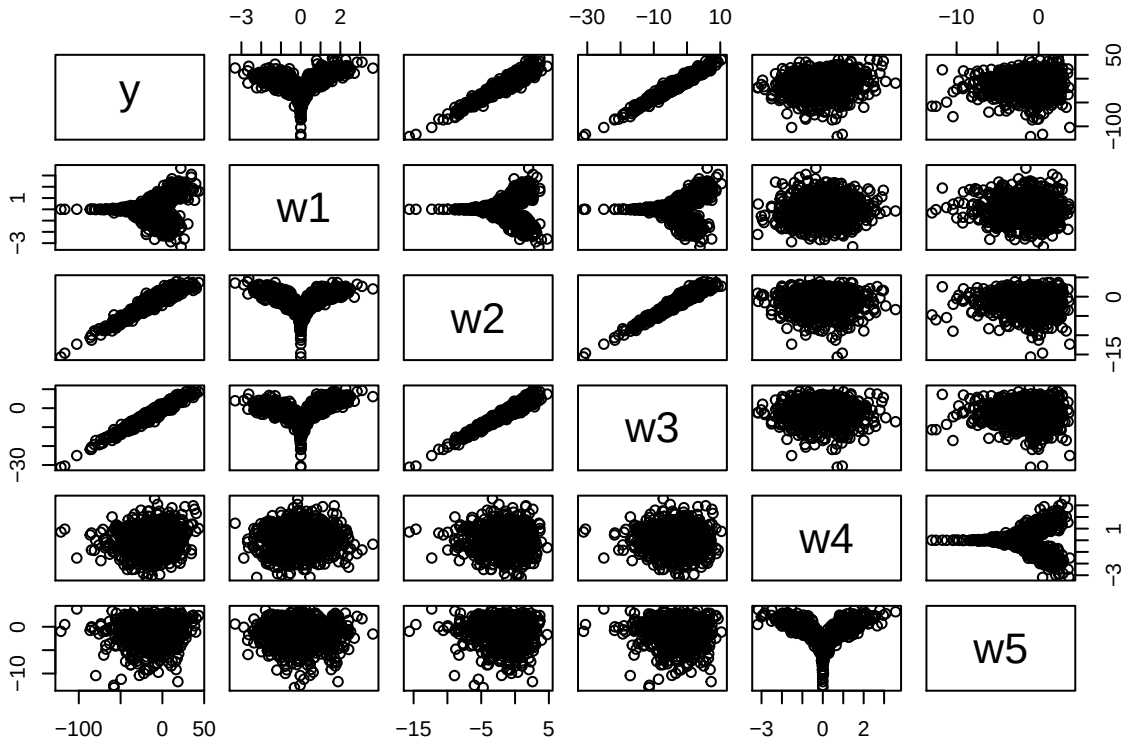
The tricker relationship between the variables in $D2$ was because they were generated using the square root of the absolute value of the other variable. Here, in $D3$ the process is repeated but with the log of the absolute value.

```
w1 <- rnorm(1000)

w2 <- 2*log(abs(w1)) + rnorm(1000)

w3 <- w1 + 2*w2 + rnorm(1000)

w4 <- rnorm(1000)

w5 <- 2*log(abs(w4)) + rnorm(1000)

y <- w1 + 2*w2 + 3 * w3 + 4*w4 + 1*w5 + rnorm(1000)

d3 <- data.frame(y,w1,w2,w3,w4,w5)
```

```
cor(d3)
```

```
##             y          w1          w2          w3          w4          w5
## y  1.00000000  0.15728407  0.94764704  0.97416878  0.17279746  0.05578763
## w1 0.15728407  1.00000000 -0.02714818  0.15752508  0.03050113 -0.05109218
## w2 0.94764704 -0.02714818  1.00000000  0.96531545 -0.01975103 -0.04414540
## w3 0.97416878  0.15752508  0.96531545  1.00000000 -0.01042494 -0.05419898
## w4 0.17279746  0.03050113 -0.01975103 -0.01042494  1.00000000 -0.03325170
## w5 0.05578763 -0.05109218 -0.04414540 -0.05419898 -0.03325170  1.00000000
```

```
plot(d3)
```



Models and Comparisons

Trees

CART

In 1984, Breiman et al introduces a revolutionary new algorithm for trees. **Need to acquire** *Classification and Regression Trees* **to make sure the method discused in MASS is the same that Breiman uses/is used in** `randomForest`

Tree Algorithm CART?

Begin by considering the entire feature space $X_1, ..., X_n$. Then:

1. Consider every possible pair of partitions of this feature space, P_1, P_2, so that if $X_1 = x_1, X_2 = x_2, ..., X_n = x_n$ where $x_1, ..., x_n \in P_1$ then our prediction is the mean value of y given $x_1, .., x_n \in P_1$.
2. Choose the partitions that minimize RSS

3. For each new partition, repeat steps 1 and 2 until some stopping condition is reached.

Binary Recursive Partitioning

Conditional Inference Trees

CI trees 1. For case weights w test the global null hypothesis of independence between any of the m covariates and the response. Stop if this hypothesis cannot be rejected. Otherwise select the j_{th} covariate Xj with strongest association to Y.

2. Choose a set $A \subset X_j$ in order to split X_j into two disjoint sets A and $X_j \ \ A$. The case weights w_{left} and w_{right} determine the two subgroups with $w_{left,i} = w_i I(X_{j \cdot i} \in A)$ and $w_{right,i} = w_i I(X_{ji} \in A)$ for all $i = 1, ..., n$ ($I(\hat{u})$ denotes the indicator function).

3. Recursively repeat steps 1 and 2 with modified case weights w_left and w_right, respectively.

from https://eeecon.uibk.ac.at/~zeileis/papers/Hothorn+Hornik+Zeileis-2006.pdf

After step 1 is completed, any goodness of fit method can be used to generate the split and choose the set A. Note that in this method the splitting is done separately from the variable selection.

Bagged Forests

Random Forests