# LASSO - $n$ Feature Selection Intuition

I am struggling to understand the intuition behind why LASSO can only select at most $n$ features when $n \ll p$, where $n$ is the number of samples, and $p$ is the number of features.

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

Generate a mock dataset.

```
data <- data.frame(x1 = c(1),
                   x2 = c(2),
                   y = c(5))
```

Generate a grid search for coefficient values.

```
range <- 5
b1 <- seq(-range, range, length.out = 50)
b2 <- seq(-range, range, length.out = 50)
model <- expand.grid(b1 = b1, b2 = b2)
```

Helper function to compute loss:

```
compute_loss <- function(model, data, lambda) {
  y_hat <- model["b1"] * data$x1 + model["b2"] * data$x2
  rss <- mean((data$y - y_hat)^2)
  l1 <- lambda * sum(abs(model))

  data.frame(rss = rss,
             lasso = rss + l1,
             b1 = model["b1"],
             b2 = model["b2"],
             y_hat = y_hat)
}
```

Now compute the loss across our grid, with e.g., a $\lambda = 1$.

```
lambda <- 1
costs <- apply(model, 1, compute_loss, data, lambda) %>%
  dplyr::bind_rows() %>%
  tibble::remove_rownames()
```

We can see that the optimal solution for lambdas in this simulated grid search with $\lambda = 1$ does not contain only $n$ features.

```
costs[which.min(costs$lasso),]
```

```
            rss    lasso        b1       b2    y_hat
1826 0.04164931 2.490629 0.1020408 2.346939 4.795918
```

And a visualization of the LASSO grid search where $\lambda = 1$ for reference: