Can we ever fit models with gene expression data?

- We've primarily dealt with design matrices where #rows > #columns

- There are 25,000 human genes. We would need >25,000 samples!

- We can fit models where #rows < #columns (or #rows << #columns), but we need to be careful.

Example: simulated data

```
X = rand(15,50);
b = [10 20 30 40 50]';

y = X(:,1:5)*b +
        0.1*randn(15,1);

fitlm(X,y,'intercept',false)
```

Example: simulated data

```
X = rand(15,50);
b = [10 20 30 40 50]';

y = X(:,1:5)*b +
        0.1*randn(15,1);

fitlm(X,y,'intercept',false)
```

```
Warning: Regression design matrix
is rank deficient to within machine
precision.
ans =

Linear regression model:
 y ~ x1 + x2 + x3 + x4 + x5 + x6
+ x7 + x8 + x9 + x10 + x11 + x12
+ x13 + x14 + x15 + x16 + x17 + x18
+ x19 + x20 + x21 + x22 + x23 + x24
+ x25 + x26 + x27 + x28 + x29 + x30
+ x31 + x32 + x33 + x34 + x35 + x36
+ x37 + x38 + x39 + x40 + x41 + x42
+ x43 + x44 + x45 + x46 + x47 + x48
+ x49 + x50
```

Estimated Coefficients:

	Estimate	SE	tStat	pValue
x1	0	0	NaN	NaN
x2	0	0	NaN	NaN
x3	0	0	NaN	NaN
x4	0	0	NaN	NaN
x5	0	0	NaN	NaN
x6	0	0	NaN	NaN
x7	-27.911	0	-Inf	NaN
x8	0	0	NaN	NaN
x9	28.158	0	Inf	NaN
x10	0	0	NaN	NaN
x11	0	0	NaN	NaN
x12	0	0	NaN	NaN
x13	-16.228	0	-Inf	NaN
x14	0	0	NaN	NaN
x15	0	0	NaN	NaN
x16	0	0	NaN	NaN
x17	0	0	NaN	NaN
x18	0	0	NaN	NaN
x19	16.751	0	Inf	NaN
x20	0	0	NaN	NaN
x21	0	0	NaN	NaN
x22	0	0	NaN	NaN
x23	0	0	NaN	NaN
x24	0	0	NaN	NaN
x25	0	0	NaN	NaN
x26	0	0	NaN	NaN
x27	0	0	NaN	NaN
x28	0	0	NaN	NaN
x29	0	0	NaN	NaN
x30	0	0	NaN	NaN
x31	0	0	NaN	NaN
x32	-15.226	0	-Inf	NaN
x33	12.458	0	Inf	NaN
x34	0	0	NaN	NaN
x35	0	0	NaN	NaN
x36	0	0	NaN	NaN
x37	23.653	0	Inf	NaN
x38	35.082	0	Inf	NaN
x39	-61.538	0	-Inf	NaN
x40	35.885	0	Inf	NaN
x41	0	0	NaN	NaN
x42	39.197	0	Inf	NaN
x43	0.9938	0	Inf	NaN
x44	0	0	NaN	NaN
x45	15.388	0	Inf	NaN
x46	42.296	0	Inf	NaN
x47	0	0	NaN	NaN
x48	0	0	NaN	NaN
x49	0	0	NaN	NaN
x50	-5.9355	0	-Inf	NaN

LASSO (Least Absolute Shrinkage & Selection Operator)

- Standard least squares for regression:

$$\min_{\beta} \sum_{i=1}^{m} (y_i - \beta_0 - \beta_1 x_1 - \cdots - \beta_n x_n)^2$$

- Limiting the total “fitting” that can be done:

$$\min_{\beta} \sum_{i=1}^{m} (y_i - \beta_0 - \beta_1 x_1 - \cdots - \beta_n x_n)^2 \quad \text{s.t.} \quad \sum_{i=1}^{n} |\beta_i| \leq \kappa$$

- Equivalently, we can penalize (tax) use of coefficients:

$$\min_{\beta} \left[\sum_{i=1}^{m} (y_i - \beta_0 - \beta_1 x_1 - \cdots - \beta_n x_n)^2 + \lambda \sum_{i=1}^{n} |\beta_i| \right]$$

Example: simulated data

```
X = rand(15,50);
b = [10 20 30 40 50]';

y = X(:,1:5)*b +
        0.1*randn(15,1);

fitlm(X,y,'intercept',false)
```

```
Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
x1	0	0	NaN	NaN
x2	0	0	NaN	NaN
x3	0	0	NaN	NaN
x4	0	0	NaN	NaN
x5	0	0	NaN	NaN
x6	0	0	NaN	NaN
x7	-27.911	0	-Inf	NaN
x8	0	0	NaN	NaN
x9	28.158	0	Inf	NaN
x10	0	0	NaN	NaN
x11	0	0	NaN	NaN
x12	0	0	NaN	NaN
x13	-16.228	0	-Inf	NaN

<snip>

```
B = lasso(X,y);
B(:,1)
ans =
    9.6699
   19.4583
   30.5237
   36.8148
   47.1173
         0
         0
         0
         0
         0
      <snip>
```

Example: simulated data

```
X = rand(15,50);
b = [10 20 30 40 50]';

y = X(:,1:5)*b +
        0.1*randn(15,1);

fitlm(X,y,'intercept',false)
```

```
Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
x1	0	0	NaN	NaN
x2	0	0	NaN	NaN
x3	0	0	NaN	NaN
x4	0	0	NaN	NaN
x5	0	0	NaN	NaN
x6	0	0	NaN	NaN
x7	-27.911	0	-Inf	NaN
x8	0	0	NaN	NaN
x9	28.158	0	Inf	NaN
x10	0	0	NaN	NaN
x11	0	0	NaN	NaN
x12	0	0	NaN	NaN
x13	-16.228	0	-Inf	NaN

<snip>

```
B = lasso(X,y);
B(:,1)
ans =
    9.6699
   19.4583
   30.5237
   36.8148
   47.1173
         0
         0
         0
         0
         0
    <snip>
```

```
B(:,39)
ans =
         0
         0
   23.0954
         0
    2.0033
         0
         0
         0
         0
         0
    <snip>
```