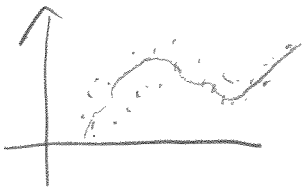
Model Selection

Tools so far:
- t-test on individual coefficients βs
- F-test comparing 2 models
- Which models? → Backward search

Ultimate validation tool = PREDICTION

Does the model generalize to future data?

Usually it is safer to use as simple models as possible for prediction. Why? Complex models are more difficult to estimate (especially if you have a small sample size).

Flexible models	Simple models
- Many parameters ⇓ - adapts to data	- few parameters ⇓ - rigid
	
- little or no bias (since flexible)	- possible bias (since rigid)
- high estimation variance (small changes to data → large impact)	- smaller variance (since rigid, can't change much)

} bias = average across many data sets, how much model deviates from true relationship

To summarize:

Bias – Variance trade-off

$E(\hat{y}) - E(y)$

- $E(\hat{y})$: Average fitted value across many data sets
- $E(y)$: True model value

If the fitted model is incorrect, can't attain true value

Flexible models – bias ↓

Simple models – bias ↑

$V(\hat{y})$

↑ a function of $\hat{\beta}$ since $\hat{y} = x\hat{\beta}$

The more complex model is, the more uncertainty in $\hat{\beta}$'s → and therefore $\hat{y}$'s

Flexible models – variance ↑

Simple models – variance ↓

We want to control both bias <u>and</u> variance.

Just controlling bias is not enough. Yes, on <u>average</u> (across many data sets) we do well, but if variance is high we can be far from this average in one particular data set <u>and</u> we don't know when.

Combining bias & variance ⇒ MSE = BIAS^2 + VARIANCE

So, we should choose a model that minimizes

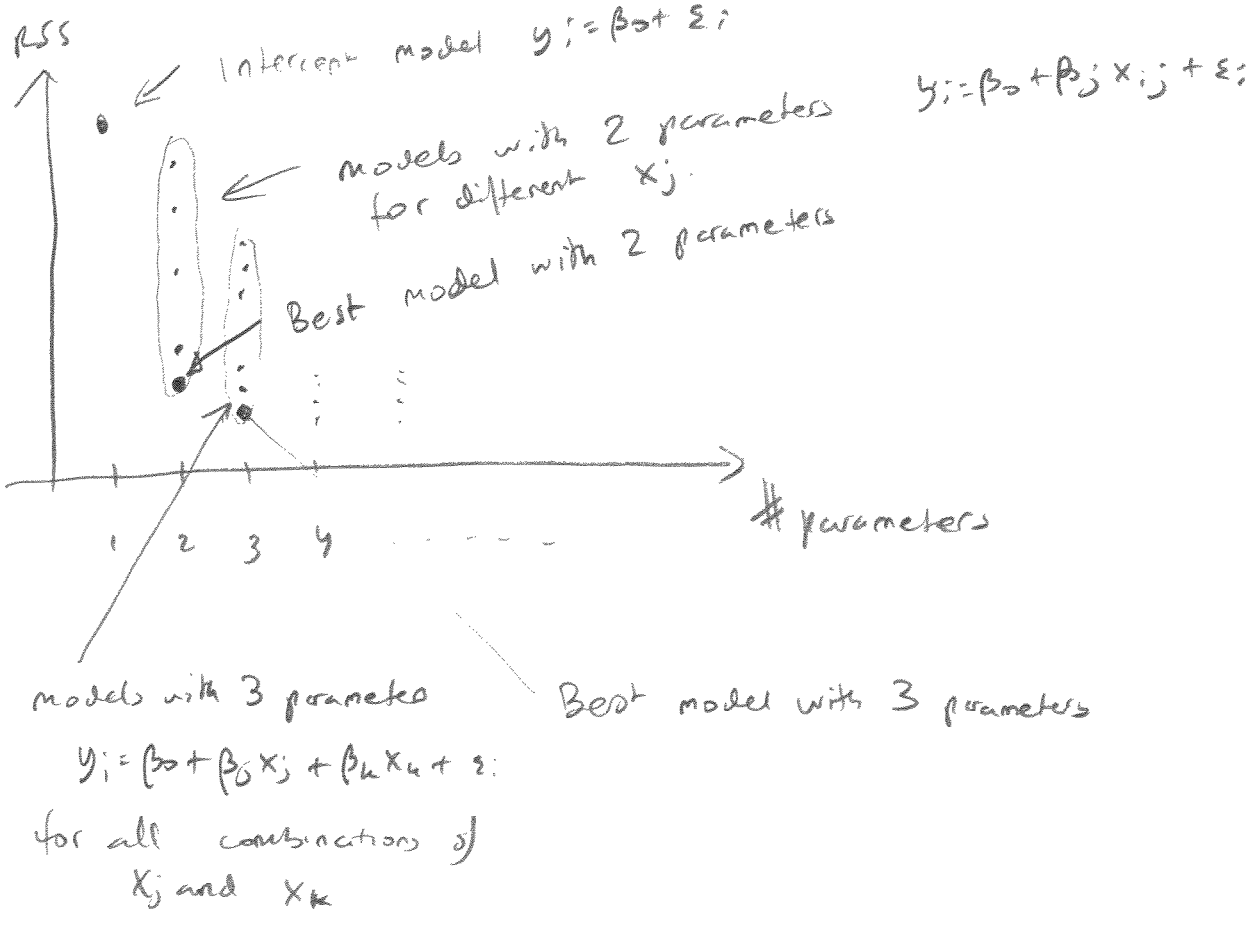
$MSE = BIAS^2 + VARIANCE$.

How? To compute the bias we need to know the true model !!!

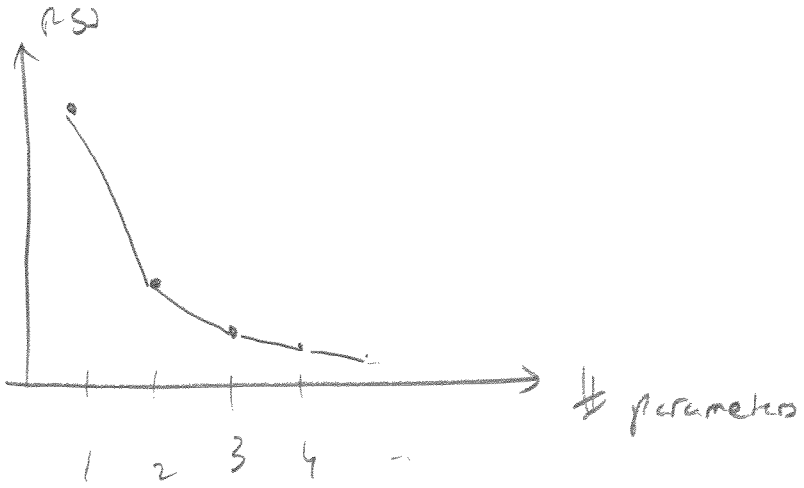
Can we just use the residual sum of squares (RSS)?

NO! $RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ always decreases the more complex the model is (more parameters to help match the model to the data).

In fact, if you use n parameters you get a perfect fit, $RSS = 0$!



RSS-curve



RSS-curve – curve that connects the best model of each size (i.e. best one-variable model
best two-variable model
etc)

So RSS is minimized for the largest model.

RSS is not a good substitute for the prediction MSE.

A hypothetical situation

What if we had Training Data and Test Data

Fit models here ⟶ Check if these fitted models work for prediction on test data

Setup Training Data $(\underset{\sim}{x}_i, y_i)_{i=1}^n$, $\underset{\sim}{x}_i = (x_{i1}, x_{i2}, \ldots, x_{i,p-1})$

Test Data $(\underset{\sim}{x}_i, y_i^{new})_{i=1}^n$, $\underset{\sim}{x}_i = (x_{i1}, x_{i2}, \ldots, x_{i,p-1})$

That is, assume we get to observe new y-values at same x-locations

True model $\beta^* = (\beta_0^*, \beta_1^*, \beta_2^*, \ldots, \beta_{p-1}^*)$

some of these are 0

$$\begin{cases} \text{Training data}: & y_i = x_i \beta^* + \varepsilon_i, \quad \varepsilon_i \text{ drawn from } N(0, \sigma^2) \\ \text{Test data}: & y_i^{new} = x_i \beta^* + \varepsilon^{new}, \quad \varepsilon_i^{new} \sim N(0, \sigma^2) \end{cases}$$

Definition

$$\begin{cases} RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 & \hat{y}_i = x_i \hat{\beta} \leftarrow \text{LS estimate from training data} \\ MSE_{train} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 & \text{(training mean-squared-error)} \end{cases}$$

$$MSE_{test} = \frac{1}{n} \sum_{i=1}^{n} (y_i^{new} - \hat{y}_i)^2 \quad \text{(test or prediction MSE)}$$

$\hat{y}_i = x_i \hat{\beta}$ ← LS estimate from training data

If this is small, the model generalizes to test data.

So no refitting of models using test data.

Strategy

① Enumerate all models of interest (maybe all possible models)

$M_1, M_2, \ldots, M_M$

M_j (model j) is a particular combination of variables

(Example $M_j = \{X_1, X_5, X_{10}\}$ – a 3-variable model involving variable 1, 5 and 10. (4 parameters))

② On the training data, fit all M models to the data using least squares

$M_1, M_2, \ldots, M_M$

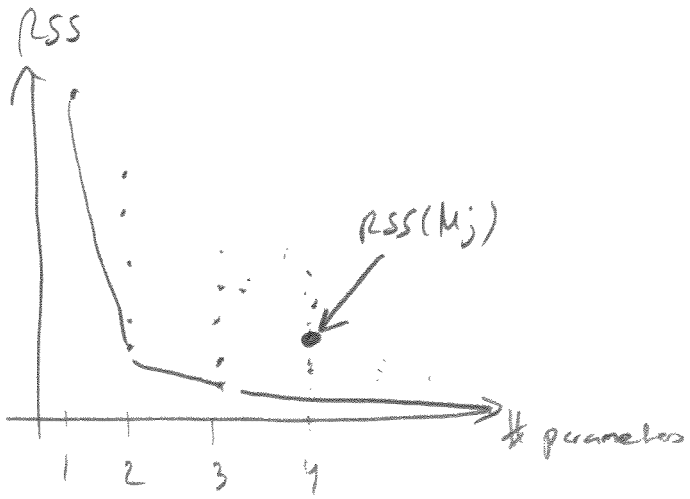
coefficient estimates ⇓ $\hat{\beta}(M_1), \hat{\beta}(M_2), \ldots, \hat{\beta}(M_M)$

⇓

RSS $RSS(M_1), RSS(M_2), \ldots, RSS(M_M)$

Example

$$\hat{\beta}(M_j) = \begin{pmatrix} \hat{\beta}_1(M_j) \\ 0 \\ 0 \\ 0 \\ \hat{\beta}_5(M_j) \\ 0 \\ 0 \\ 0 \\ 0 \\ \hat{\beta}_{10}(M_j) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$



(3) On the test data, evaluate the prediction MSE for all models

$M_1, M_2 \quad \ldots \quad M_M$

$\hat{\beta}(M_1), \hat{\beta}(M_2) \quad \ldots \quad \hat{\beta}(M_M)$ ⟵ from training data

$\Downarrow$

$pMSE(M_1), pMSE(M_2) \quad \ldots \quad pMSE(M_M)$

$$pMSE(M_j) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i^{new} - \hat{y}_i(M_j) \right)^2$$

$\hat{y}_i(M_j)$: fitted values for model M_j

$$= \frac{1}{n} \sum_{i=1}^{n} \left(y_i^{new} - \underset{\sim}{x}_i \hat{\beta}(M_j) \right)^2$$

$\underset{\sim}{x}_i \hat{\beta}(M_j)$: coefficient estimates from training data

(4)