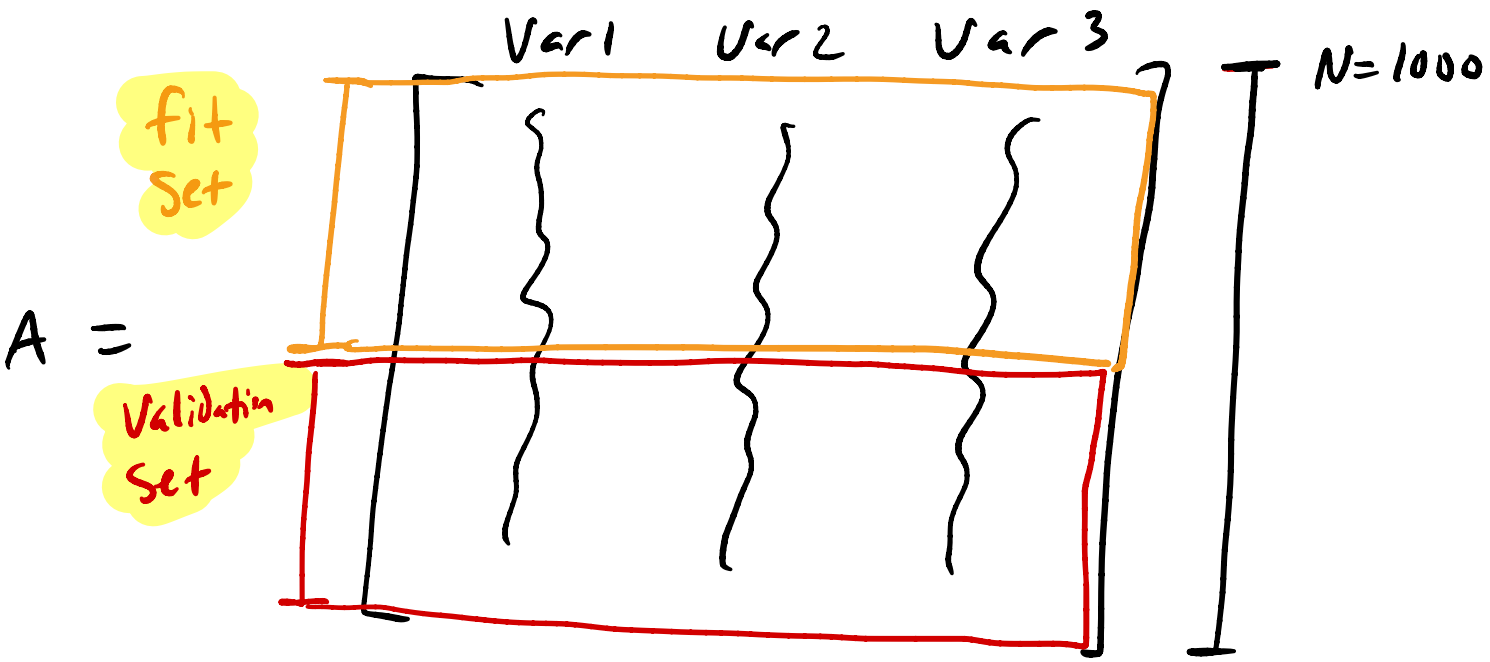
Fit and Validation Sets

To check whether we are overfitting when doing polynomial regression [and in general], it is common to subdivide your data into 2 parts:



Example: If $N = 1000$, you could make $N = 500$ part of the fit set and the other $N = 500$ part of the validation set.

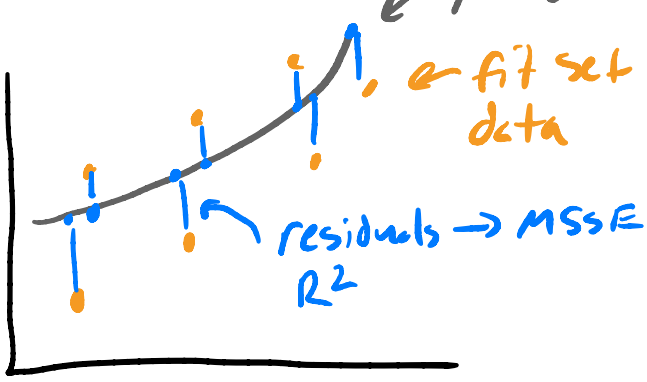
☆ Samples placed either in <u>one</u> or the other, <u>not both</u>.

☆ Usually better to randomly assign samples to either set.

[Above the 1st 50% get added to fit set, the second half to validation set. In project 3, polynomial regression data already randomly shuffled so you can simply take that strategy]

Regression fit and validation set workflow:

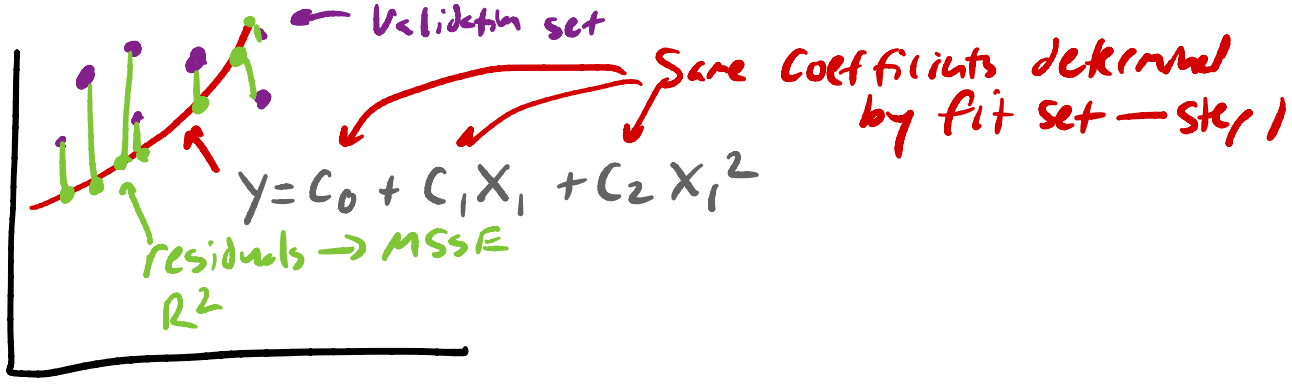
1) Fit regression model to fit set data, get and save coefficients $\vec{c}$.



2) Compute residuals between:
- predicted values of fit set on regression curve.
- Actual fit set values.

⇒ Compute quality of fit metrics on fit set (e.g. MSSE, R^2 above)

3) In in a new plot, plot the Validation set data and the regression model using the fit set coefficients — do not redo regression on Validation set!



4) Compute residuals between:

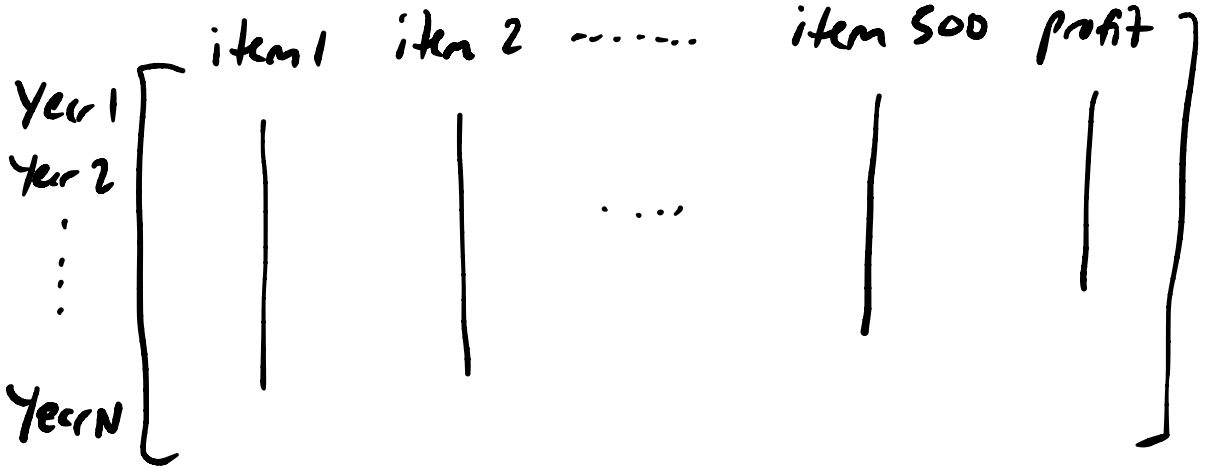
- predicted Validation set values on regression curve
- Validation set samples

⇒ Determine R^2, MSSE, etc.

Stepwise linear regression:

problem: You have tons of variables in a huge dataset and you do not even know which you should include in a regression — you don't know what vars might be important.

Example: predict supermarket sales from 500 products — which combination are most strongly associated with annual profit?



Stepwise linear regression is a greedy algorithm.

1) Start regression with intercept only: $A = [\vec{1}]$

2) Add one of every one of the 500 vars to regression: $A = [\vec{1}, \underbrace{\vec{x}_i}_{\text{regression has 1 var in it}}]$

⇒ Compute R^2 each time.

⇒ Keep track of variable that yields highest R^2

⇒ Officially add the var that gets us the highest R^2 to the running model — say it is $\vec{x}_{30}$.

Model now: $A = [\vec{1}, \vec{x}_{30}]$

3) With updated model, try adding one of each of the remaining 499 variables ($\vec{x}_{30}$ cannot be added twice), compute R^2,

Officially add the var that gets us the highest R^2, ...

repeat...

4) Algorithm terminates when all vars added to regression model.

Algorithm almost certainly will overfit your data — but plotting the progressive R^2 values helps show you which vars might be important:

Look at plot & you decide threshold before curve plateaus

- Does not actually use R^2, but a variant called <u>adjusted R^2</u> that penalizes model complexity (# vars in model ← large = likely to overfit)

$$R^2_{adj} = 1 - \left[\frac{(1-R^2)(N-1)}{N-K-1} \right]$$

K: # ind vars currently added to regression

Note: R^2_{adj} can be < 0 if fit very bad.

If $\max\left(R^2_{adj}\right) < 0$ for your stepwise algorithm → abort / return prematurely

e.g. after adding 490 vars instead of 500.