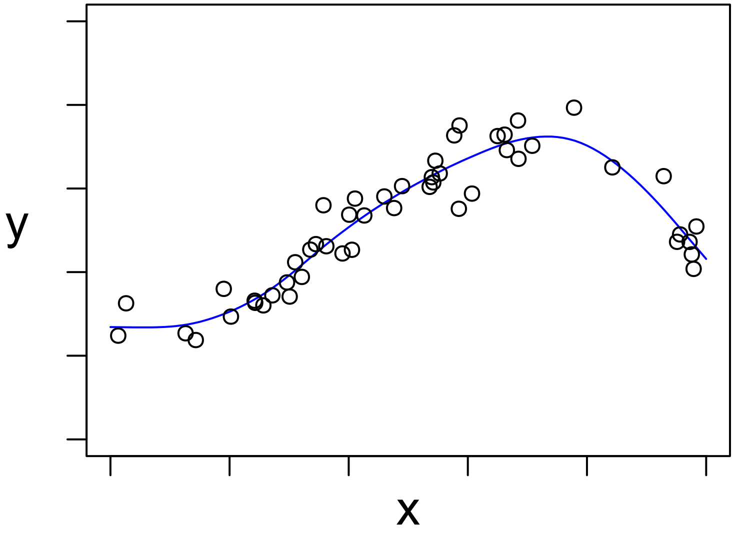
Today

- Continue ML workflow with ants
- Cross-Validation (CV)
  - inference algorithm
  - algorithm from scratch
  - pseudocode to R code

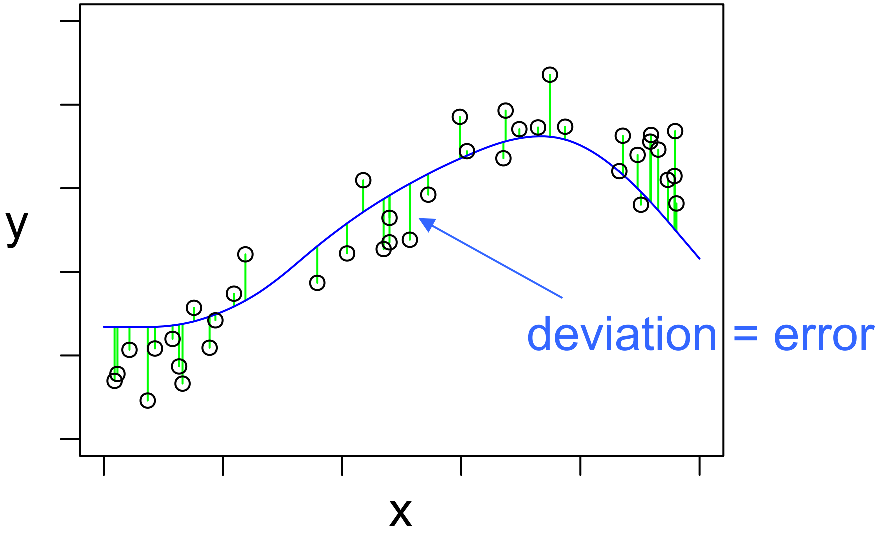
Inference algorithm

Basic algorithm: out-of-sample validation

1. Train model on training dataset



2. Test model on validation dataset



deviation = error

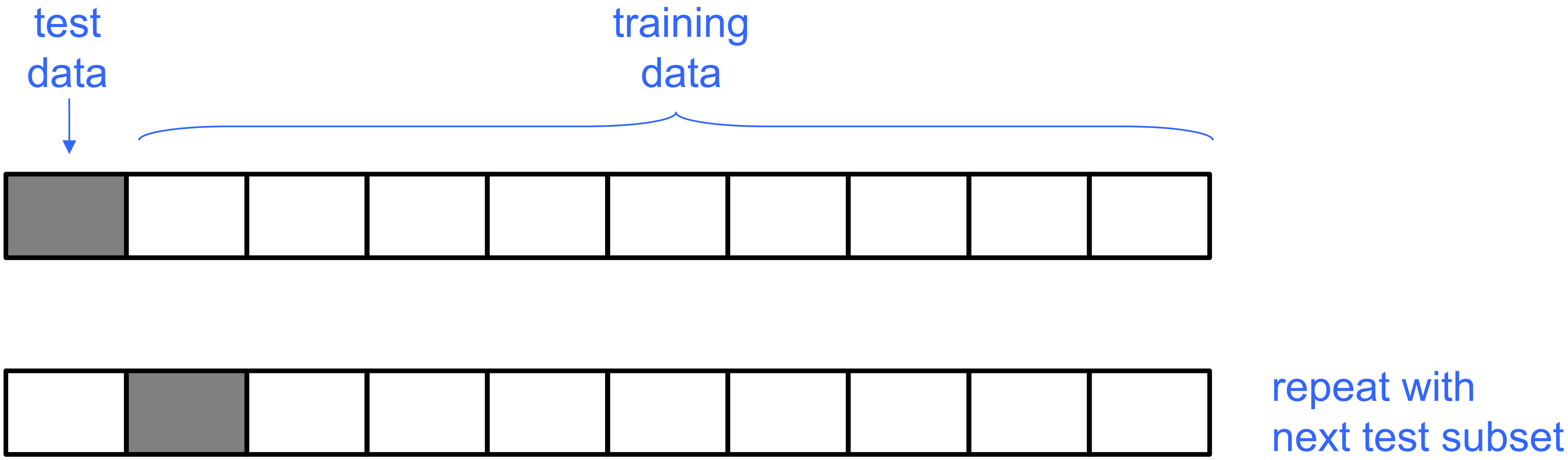
e.g. mean squared error (MSE) or
root mean square error (RMSE)

Cross validation (CV)

- Some approaches:
- Different datasets for train and test
- Holdout portion of a dataset (e.g. 10%)
  - aka train-test split
  - often used for huge datasets
- Both the above can suffer from bias because we have only one test set
- k-fold CV: replicate test sets

k-fold cross validation (CV)

Divide dataset into k parts (preferably randomly)



test data

training data

repeat with next test subset

... repeat with each test subset

k-fold CV inference algorithm

Algorithm

```
divide dataset into k parts i = 1...k
for each i
        test dataset = part i
        training dataset = remaining data
        find f using training dataset
        use f to predict for test dataset
        e_i = prediction error
CV_error = mean(e)
```

Typical values for k: 5, 10, n

Tuning parameters

- Order of polynomial
- Different values of tuning parameters give different models
- Use CV inference algorithm to choose model with best predictive performance

Code

- ants_cv_polynomial.R
- ants_cv_polynomial.py