# Cross Validation: Takeaways

## Syntax

- Implementing k-fold cross validation in trainControl():
  ```

  # Define k to be some number

  k <- 5

  # Use it in trainControl()

  five_fold_control <- trainControl(method = "cv", number = k) ```

- Examining the results of the k-fold cross-validation in the resample object:
  ```

  # Assuming a trained knn model is in `knn_model`

  knn_model$resample

  | RMSE | Rsquared | MAE | Resample |
  |---|---|---|---|

  1 101.40860 0.3324885 55.92774 Fold1 2 115.89932 0.3612587 55.83190 Fold2 3 93.64825 0.3184112 51.88761 Fold3 4 100.92235 0.3991235 50.78818 Fold4 5 122.82986 0.4312819 50.84859 Fold5
  ```

## Concepts

- K-fold cross-validation includes:
  - Splitting the full data set into `k` equal length partitions:
    - Selecting `k-1` partitions as the training set.
    - Selecting the remaining partition as the test set.
  - Training the model on the training set.
  - Using the trained model to predict outcomes on the test fold.
  - Computing the test fold's error metric.
  - Repeating all of the above steps `k-1` times, until each partition has been used as the test set for an iteration.
  - Calculating the mean of the `k` error metrics.
  ```

- Parameters for cross-validation are set in the `trainControl()` function, using the arguments:
  - `method = "cv"` indicates that we want to use cross-validation
  - `number= 5` : indicates that we want to use five folds.
- You can look at the results of the k-fold cross-validation in the `resample` object contained in your trained k-nearest neighbors model
- Bias describes how far an estimator is from the target value it's trying to estimate. Variance describes how varied your estimates. In an ideal world, we want both low bias and low variance when creating machine learning models, but often have to strike a compromise between the two.

## Resources

- Accepted values for scoring criteria
- Bias-variance Trade-off