L06: Validating prediction models

“Clu, we don't have much time left to find that file. This is top priority.”

Kevin Flynn, Tron

Things we cover in this session

- Validating predictions using leave-one-out validations

Things you need for this session

- W06-1 Leave-one-out validation

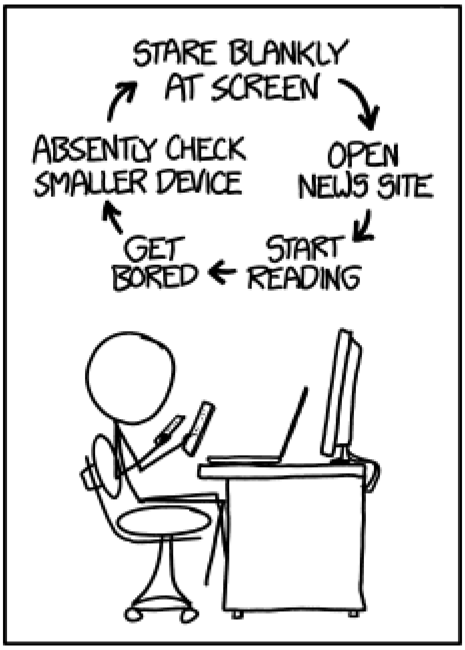
Things to take home from this session

At the end of this session you should be able to

- handle simple for loops
- compute leave-one-out validations
- interpret the validation results

For loop

If you have to apply certain code segments to multiple objects but can not vectorized it (i.e. can not apply it to e.g. an entire data frame directly as all examples in this course so far), you can use repeating structures. Such structures area called loop since they execute code over and over again until the loop comes to a defined end.



While more than one types of loops exist, the for loop is the mother of all loops. Regarding this course, it can always be the loop of your choice, especially if this is your first contact with programming.

A for loop looks like

```
for(<iteration variable> in <control vector>){
    <do something>
}
```

The code within the () brackets is the control statement and controls the number of repetitions and the code within the {} brackets is the content of the loop. The number of repetitions is defined by the elements of the control vector in the control statement. When the loop is executed for the first time, the iteration variable in the control statement is set to the first element of the vector. If it is executed for the second time, the variable is set to the second element of the vector and so on. Hence, the variable can be used within the content of the loop as some kind of iterator.

Have a look at C-06-1 For loops now for more information on them.

Leave-one-out validation

In a perfect world, one would have plenty of data to build (i.e. fit) a prediction model and plenty of other data to validate it. In real world applications, ecological field surveys are generally so time consuming, that the data set is quite limited. Therefore, the same data set has to be used for both model development and model validation.

The problem in using the same data set for model development and validation is that some models tend to fit the training data set quite well (e.g. kriging) so if you just use the same data set for validation, your estimation of the prediction accuracy for unknown values might likely be much to large.

A rather simple solution for this problem is to leave one or more pairs of observational variables out of the model development and then predict the model for the independent data values and compare them to the corresponding dependent values in terms of the difference between the observed and predicted value. Since you leave only one or a few variables out of the model development, you still have a large data set.

Since the above solution would only give an estimate for very few observations, the procedure can be repeated by leaving another variable (or a few) out of the model development and perform the same validation procedure again. This can be repeated until a large number of validation runs is reached (e.g. 200) or until you have left out every value pair once. By averaging over all absolute values of the individual prediction errors , you get a quite good idea about the performance of your model.

Since you know already about for loops, the implementation is not a huge problem. For example, if you have 50 data pairs, you will repeat the validation 50 times.

A pseudo-code structure for a leave-one-out validation would be:

```
predicted_values <- c()
observed_values <- c()
for(i in seq(<number of observations>){
  training <- <your_data>[-i]
  linear_model <- <your fitted model using training data>
  predicted_values[i] <- predict(linear_model, <your_data>$independent[i])
  observed_values[i] <- <your_data>$dependent[i]
}
prediction_error <- abs(predicted_values - observed_values)
mean(prediction_error)
```

If you implement the leave-one-out validation like this and do not compute the error inside the loop, you can get an R^2 for your prediction by fitting a linear model to your predicted/observed value pairs.

Time for practice

W06-1 Leave-one-out validation