



Data slicing

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

SPAM Example: Data splitting

```
library(caret); library(kernlab); data(spam)
inTrain <- createDataPartition(y=spam$type,
                               p=0.75, list=FALSE)
training <- spam[inTrain,]
testing <- spam[-inTrain,]
dim(training)
```

use 75% for training set
25% for test set

```
[1] 3451  58
```

SPAM Example: K-fold

cross validation

```
set.seed(32323)
folds <- createFolds(y=spam$type,k=10, split spam into k=10 folds, based on outcome = type
                    list=TRUE,returnTrain=TRUE)
                    each subset return the training set (and not the test set)
                    is a list
sapply(folds,length)
```

Fold01	Fold02	Fold03	Fold04	Fold05	Fold06	Fold07	Fold08	Fold09	Fold10
4141	4140	4141	4142	4140	4142	4141	4141	4140	4141

```
folds[[1]][1:10]
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

SPAM Example: Return test

```
set.seed(32323)
folds <- createFolds(y=spam$type,k=10,
                     list=TRUE,returnTrain=FALSE)
# now return the test set, not the training set
sapply(folds,length)
```

Fold01	Fold02	Fold03	Fold04	Fold05	Fold06	Fold07	Fold08	Fold09	Fold10
460	461	460	459	461	459	460	460	461	460

```
folds[[1]][1:10]
```

```
[1] 24 27 32 40 41 43 55 58 63 68
```

SPAM Example: Resampling

```
set.seed(32323)
folds <- createResample(y=spam$type, times=10,
                        list=TRUE)
sapply(folds, length)
```

```
Resample01 Resample02 Resample03 Resample04 Resample05 Resample06 Resample07 Resample08 Resample09
      4601      4601      4601      4601      4601      4601      4601      4601      4601
Resample10
      4601
```

```
folds[[1]][1:10]
```

```
[1] 1 2 3 3 3 5 5 7 8 12
```

manchmal kommt derselbe Datensatz mehrmals vor,
weil wir Ziehen mit Zuruecklegen benutzen.

SPAM Example: Time Slices

```
set.seed(32323)
tme <- 1:1000
folds <- createTimeSlices(y=tme,initialWindow=20,
                          horizon=10)
names(folds)
```

20 Datensätze werden für das Training verwendet,
die 10 folgenden für die Prediction.

```
[1] "train" "test"
```

```
folds$train[[1]]
```

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

```
folds$test[[1]]
```

```
[1] 21 22 23 24 25 26 27 28 29 30
```

Further information

- Caret tutorials:
 - http://www.edii.uclm.es/~useR-2013/Tutorials/kuhn/user_caret_2up.pdf
 - <http://cran.r-project.org/web/packages/caret/vignettes/caret.pdf>
- A paper introducing the caret package
 - <http://www.jstatsoft.org/v28/i05/paper>