Resampling

Manual Splitting – what could possibly go wrong?

```
library(mlr)
```

```
## Loading required package: ParamHelpers
```

```
task = makeClassifTask(data = iris, target = "Species")
learner = makeLearner("classif.randomForest")
```

The Good

```
model = train(learner, task, subset = c(1:30, 51))
model
```

```
## Model for learner.id=classif.randomForest; learner.class=classif.randomForest
## Trained on: task.id = iris; obs = 31; features = 4
## Hyperparameters:
```

```
predictions = predict(model, task = task, subset = 31:50)
predictions
```

```
## Prediction: 20 observations
## predict.type: response
## threshold:
## time: 0.00
##    id  truth response
## 31 31 setosa   setosa
## 32 32 setosa   setosa
## 33 33 setosa   setosa
## 34 34 setosa   setosa
## 35 35 setosa   setosa
## 36 36 setosa   setosa
## ... (20 rows, 3 cols)
```

```
performance(predictions, measures = acc)
```

```
## acc
##   1
```

```
calculateConfusionMatrix(predictions)
```

```
##             predicted
## true          setosa versicolor virginica -err.-
##   setosa          20          0         0      0
##   versicolor       0          0         0      0
##   virginica        0          0         0      0
##   -err.-           0          0         0      0
```

The Bad

```
model = train(learner, task, subset = 1:100)
model
```

```
## Model for learner.id=classif.randomForest; learner.class=classif.randomForest
## Trained on: task.id = iris; obs = 100; features = 4
## Hyperparameters:
```

```
predictions = predict(model, task = task, subset = 101:150)
predictions
```

```
## Prediction: 50 observations
## predict.type: response
## threshold:
## time: 0.29
##      id     truth   response
## 101 101 virginica versicolor
## 102 102 virginica versicolor
## 103 103 virginica versicolor
## 104 104 virginica versicolor
## 105 105 virginica versicolor
## 106 106 virginica versicolor
## ... (50 rows, 3 cols)
```

```
performance(predictions, measures = acc)
```

```
## acc
##   0
```

```
calculateConfusionMatrix(predictions)
```

```
##              predicted
## true         setosa versicolor virginica -err.-
##   setosa          0          0         0      0
##   versicolor      0          0         0      0
##   virginica       0         50         0     50
##   -err.-          0         50         0     50
```

The Ugly

```
model = train(learner, task, subset = c(1:45, 51:95, 101:110))
model
```

```
## Model for learner.id=classif.randomForest; learner.class=classif.randomForest
## Trained on: task.id = iris; obs = 100; features = 4
## Hyperparameters:
```

```
predictions = predict(model, task = task, subset = c(46:50, 96:100, 111:150))
predictions
```

```
## Prediction: 50 observations
## predict.type: response
## threshold:
## time: 0.00
##    id      truth   response
## 46 46     setosa     setosa
## 47 47     setosa     setosa
## 48 48     setosa     setosa
## 49 49     setosa     setosa
## 50 50     setosa     setosa
## 96 96 versicolor versicolor
## ... (50 rows, 3 cols)
```

```
performance(predictions, measures = acc)
```

```
##  acc
## 0.84
```

```
calculateConfusionMatrix(predictions)
```

```
##             predicted
## true          setosa versicolor virginica -err.-
##   setosa          5          0         0      0
##   versicolor      0          5         0      0
##   virginica       0          8        32      8
##   -err.-          0          8         0      8
```

Automatic Splitting

Holdout

```
rdesc = makeResampleDesc(method = "Holdout", split = 2/3)
result = resample(learner, task, rdesc, measures = acc)
```

```
## [Resample] holdout iter 1: acc.test.mean=0.98
## [Resample] Aggr. Result: acc.test.mean=0.98
```

```
predictions = getRRPredictions(result)
performance(predictions, measures = acc)
```

```
##  acc
## 0.98
```

```
calculateConfusionMatrix(predictions)
```

```
##             predicted
## true          setosa versicolor virginica -err.-
##   setosa         17          0         0      0
##   versicolor      0         20         1      1
##   virginica       0          0        12      0
##   -err.-          0          0         1      1
```

Using the `holdout` function:

```
result = holdout(learner, task, measures = acc, split = 2/3)
```

```
## [Resample] holdout iter 1: acc.test.mean=0.92
## [Resample] Aggr. Result: acc.test.mean=0.92
```

```
predictions = getRRPredictions(result)
performance(predictions, measures = acc)
```

```
##  acc
## 0.92
```

```r
calculateConfusionMatrix(predictions)
```

```
##             predicted
## true         setosa versicolor virginica -err.-
##   setosa         13          0         0      0
##   versicolor      0         18         1      1
##   virginica       0          3        15      3
##   -err.-          0          3         1      4
```

Stratification

```r
result = holdout(learner, task, measures = acc, split = 2/3, stratify = TRUE)
```

```
## [Resample] holdout iter 1: acc.test.mean=0.961
## [Resample] Aggr. Result: acc.test.mean=0.961
```

```r
predictions = getRRPredictions(result)
performance(predictions, measures = acc)
```

```
##       acc
## 0.9607843
```

```r
calculateConfusionMatrix(predictions)
```

```
##             predicted
## true         setosa versicolor virginica -err.-
##   setosa         17          0         0      0
##   versicolor      0         17         0      0
##   virginica       0          2        15      2
##   -err.-          0          2         0      2
```

Subsample

```r
rdesc = makeResampleDesc(method = "Subsample", iters = 10, split = 2/3, predict = "both")
# or use the "subsample" function
result = resample(learner, task, rdesc, measures = acc)
```

```
## [Resample] subsampling iter 1: acc.test.mean=0.94
## [Resample] subsampling iter 2: acc.test.mean=0.94
## [Resample] subsampling iter 3: acc.test.mean=   1
## [Resample] subsampling iter 4: acc.test.mean=0.98
## [Resample] subsampling iter 5: acc.test.mean=   1
## [Resample] subsampling iter 6: acc.test.mean=0.94
## [Resample] subsampling iter 7: acc.test.mean=0.98
## [Resample] subsampling iter 8: acc.test.mean=0.94
## [Resample] subsampling iter 9: acc.test.mean=0.96
## [Resample] subsampling iter 10: acc.test.mean= 0.9
## [Resample] Aggr. Result: acc.test.mean=0.958
```

Details for each iteration

```r
getRRPredictions(result)
```

```
## Resampled Prediction for:
## Resample description: subsampling with 10 iterations and 0.67 split rate.
## Predict: both
## Stratification: FALSE
## predict.type: response
## threshold: 
## time (mean): 0.00
##    id      truth   response iter  set
## 1  79 versicolor versicolor    1 test
## 2  37     setosa     setosa    1 test
## 3 142  virginica  virginica    1 test
## 4  33     setosa     setosa    1 test
## 5  70 versicolor versicolor    1 test
## 6  76 versicolor versicolor    1 test
## ... (1500 rows, 5 cols)
```

```r
predictionList = getRRPredictionList(result)
sapply(predictionList$test, performance, measures = acc)
```

```
##  1.acc  2.acc  3.acc  4.acc  5.acc  6.acc  7.acc  8.acc  9.acc 10.acc 
##   0.94   0.94   1.00   0.98   1.00   0.94   0.98   0.94   0.96   0.90
```

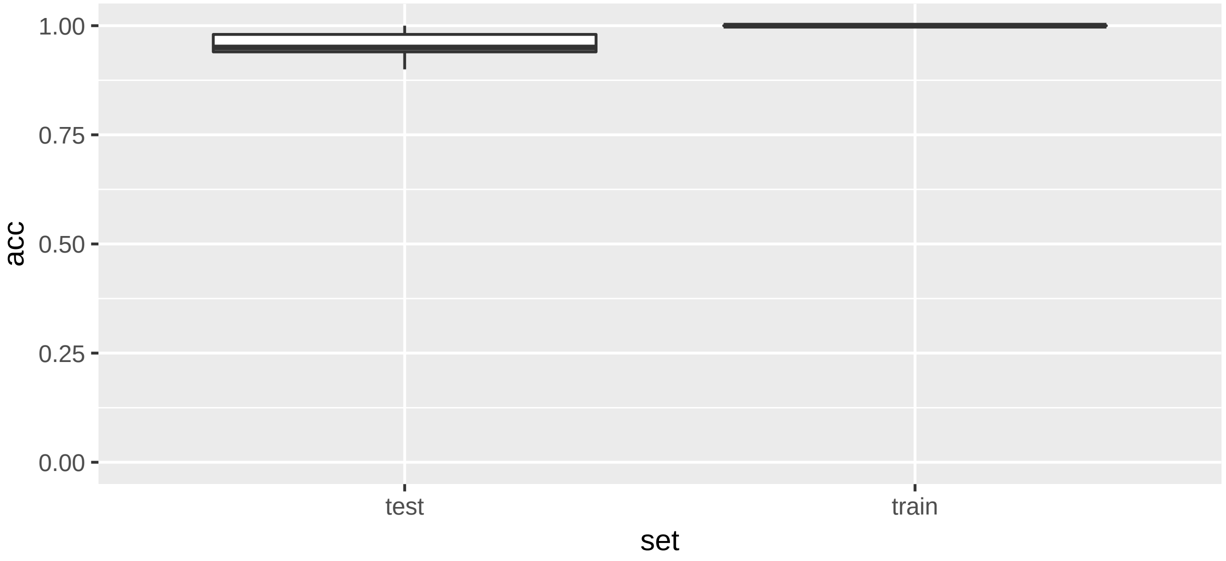
```r
# this is also directly available in the resample result
result$measures.test
```

```
##    iter  acc
## 1     1 0.94
## 2     2 0.94
## 3     3 1.00
## 4     4 0.98
## 5     5 1.00
## 6     6 0.94
## 7     7 0.98
## 8     8 0.94
## 9     9 0.96
## 10   10 0.90
```

```r
lapply(predictionList$test, calculateConfusionMatrix)
```

```
## $`1`
##               predicted
## true           setosa versicolor virginica -err.-
##   setosa           16          0         0      0
##   versicolor        0         18         1      1
##   virginica         0          2        13      2
##   -err.-            0          2         1      3
## 
## $`2`
##               predicted
## true           setosa versicolor virginica -err.-
##   setosa           19          0         0      0
##   versicolor        0         16         2      2
##   virginica       0          1        12      1
##   -err.-          0          1         2      3
##
## $`3`
##             predicted
## true         setosa versicolor virginica -err.-
##   setosa         19          0         0      0
##   versicolor      0         19         0      0
##   virginica       0          0        12      0
##   -err.-          0          0         0      0
##
## $`4`
##             predicted
## true         setosa versicolor virginica -err.-
##   setosa         17          0         0      0
##   versicolor      0         11         1      1
##   virginica       0          0        21      0
##   -err.-          0          0         1      1
##
## $`5`
##             predicted
## true         setosa versicolor virginica -err.-
##   setosa         18          0         0      0
##   versicolor      0         18         0      0
##   virginica       0          0        14      0
##   -err.-          0          0         0      0
##
## $`6`
##             predicted
## true         setosa versicolor virginica -err.-
##   setosa         15          0         0      0
##   versicolor      0         17         0      0
##   virginica       0          3        15      3
##   -err.-          0          3         0      3
##
## $`7`
##             predicted
## true         setosa versicolor virginica -err.-
##   setosa         15          0         0      0
##   versicolor      0         21         1      1
##   virginica       0          0        13      0
##   -err.-          0          0         1      1
##
## $`8`
##             predicted
## true         setosa versicolor virginica -err.-
##   setosa         17          0         0      0
##   versicolor      0         14         2      2
##   virginica       0          1        16      1
##   -err.-          0          1         2      3
##
## $`9`
##             predicted
## true         setosa versicolor virginica -err.-
##   setosa         15          0          0      0
##   versicolor      0         18          1      1
##   virginica       0          1         15      1
##   -err.-          0          1          1      2
##
## $`10`
##               predicted
## true           setosa versicolor virginica -err.-
##   setosa         16          0          0      0
##   versicolor      0         15          1      1
##   virginica       0          4         14      4
##   -err.-          0          4          1      5
```

```
library(ggplot2)
ggplot(data.frame(acc = c(result$measures.train$acc, result$measures.test$acc),
                  set = rep(c("train", "test"), each = nrow(result$measures.train))),
    aes(set, acc)) + geom_boxplot() + ylim(0, 1)
```

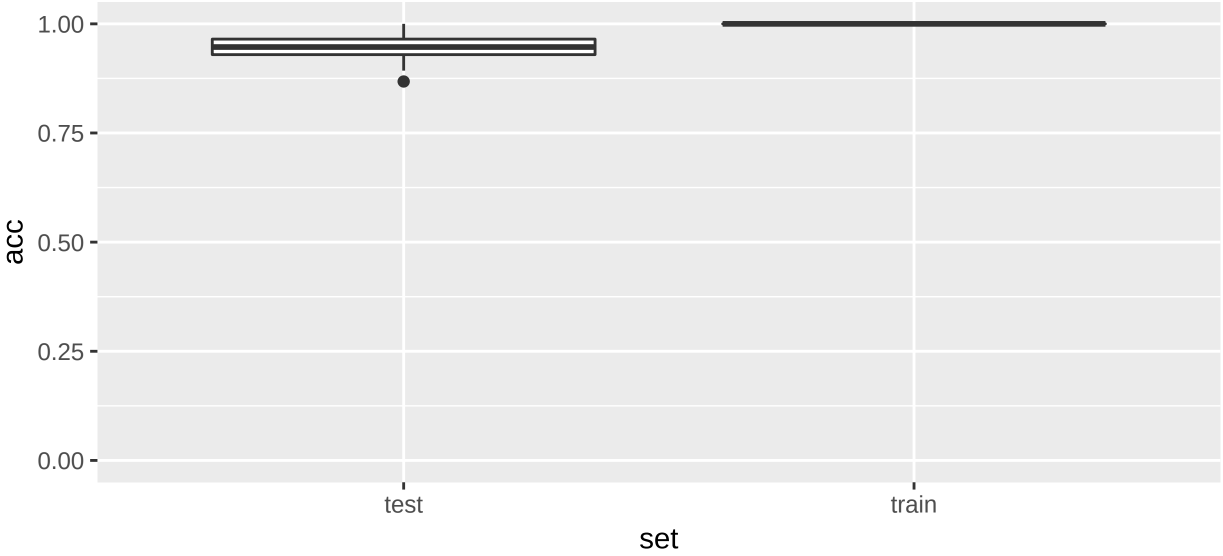


Bootstrap

```
rdesc = makeResampleDesc(method = "Bootstrap", predict = "both")
# or use the "bootstrapOOB" function
result = resample(learner, task, rdesc, measures = acc)
```

```
## [Resample] OOB bootstrapping iter 1: acc.test.mean=0.946
## [Resample] OOB bootstrapping iter 2: acc.test.mean=0.979
## [Resample] OOB bootstrapping iter 3: acc.test.mean=   1
## [Resample] OOB bootstrapping iter 4: acc.test.mean=0.929
## [Resample] OOB bootstrapping iter 5: acc.test.mean=0.966
## [Resample] OOB bootstrapping iter 6: acc.test.mean=0.929
## [Resample] OOB bootstrapping iter 7: acc.test.mean=0.926
## [Resample] OOB bootstrapping iter 8: acc.test.mean=0.868
## [Resample] OOB bootstrapping iter 9: acc.test.mean=0.961
## [Resample] OOB bootstrapping iter 10: acc.test.mean=0.964
## [Resample] OOB bootstrapping iter 11: acc.test.mean=0.925
## [Resample] OOB bootstrapping iter 12: acc.test.mean=0.951
## [Resample] OOB bootstrapping iter 13: acc.test.mean=0.923
## [Resample] OOB bootstrapping iter 14: acc.test.mean=0.978
## [Resample] OOB bootstrapping iter 15: acc.test.mean=0.966
## [Resample] OOB bootstrapping iter 16: acc.test.mean=0.946
## [Resample] OOB bootstrapping iter 17: acc.test.mean=0.946
## [Resample] OOB bootstrapping iter 18: acc.test.mean=0.947
## [Resample] OOB bootstrapping iter 19: acc.test.mean=0.982
## [Resample] OOB bootstrapping iter 20: acc.test.mean=0.98
## [Resample] OOB bootstrapping iter 21: acc.test.mean=   1
## [Resample] OOB bootstrapping iter 22: acc.test.mean=0.962
## [Resample] OOB bootstrapping iter 23: acc.test.mean=0.944
## [Resample] OOB bootstrapping iter 24: acc.test.mean=0.932
## [Resample] OOB bootstrapping iter 25: acc.test.mean=0.893
## [Resample] OOB bootstrapping iter 26: acc.test.mean=0.919
## [Resample] OOB bootstrapping iter 27: acc.test.mean=0.944
## [Resample] OOB bootstrapping iter 28: acc.test.mean=0.947
## [Resample] OOB bootstrapping iter 29: acc.test.mean=0.943
## [Resample] OOB bootstrapping iter 30: acc.test.mean=0.964
## [Resample] Aggr. Result: acc.test.mean=0.949
```

```r
ggplot(data.frame(acc = c(result$measures.train$acc, result$measures.test$acc),
                  set = rep(c("train", "test"), each = nrow(result$measures.train))),
    aes(set, acc)) + geom_boxplot() + ylim(0, 1)
```

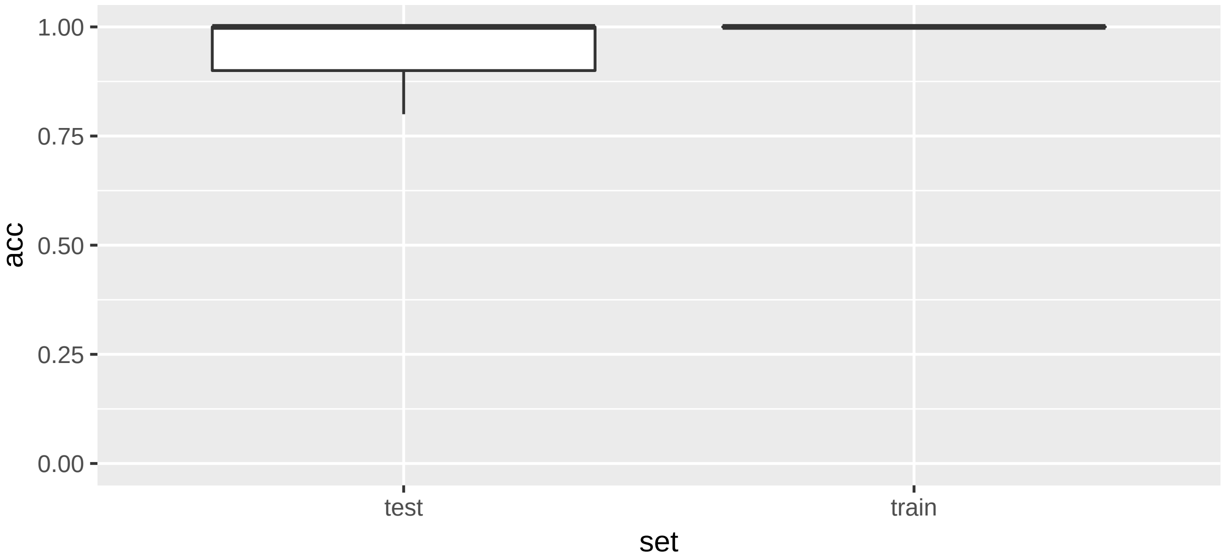


Cross-Validation

```r
rdesc = makeResampleDesc(method = "CV", predict = "both")
# or use the "crossval" function
result = resample(learner, task, rdesc, measures = acc)
```

```
## [Resample] cross-validation iter 1: acc.test.mean=   1
## [Resample] cross-validation iter 2: acc.test.mean=   1
## [Resample] cross-validation iter 3: acc.test.mean=0.867
## [Resample] cross-validation iter 4: acc.test.mean= 0.8
## [Resample] cross-validation iter 5: acc.test.mean=   1
## [Resample] cross-validation iter 6: acc.test.mean=0.867
## [Resample] cross-validation iter 7: acc.test.mean=   1
## [Resample] cross-validation iter 8: acc.test.mean=   1
## [Resample] cross-validation iter 9: acc.test.mean=   1
## [Resample] cross-validation iter 10: acc.test.mean=   1
## [Resample] Aggr. Result: acc.test.mean=0.953
```

```r
ggplot(data.frame(acc = c(result$measures.train$acc, result$measures.test$acc),
                  set = rep(c("train", "test"), each = nrow(result$measures.train))),
    aes(set, acc)) + geom_boxplot() + ylim(0, 1)
```

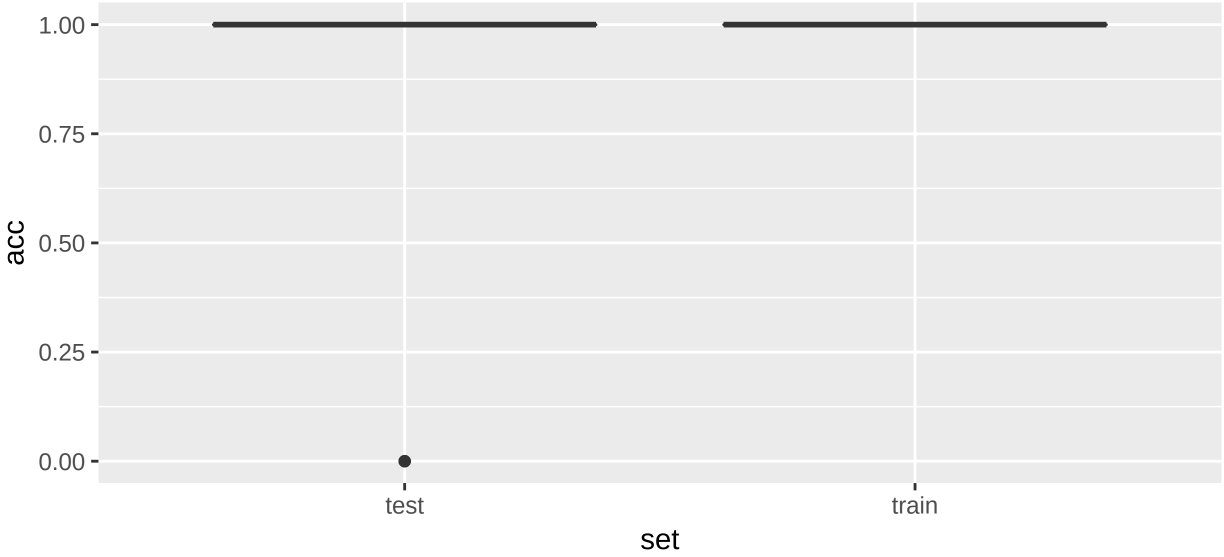


Leave-One-Out Cross-Valdation

```r
# or set number of folds to n
rdesc = makeResampleDesc(method = "LOO", predict = "both")
result = resample(learner, task, rdesc, measures = acc)
```

```
## [Resample] LOO iter 1: acc.test.mean=   1
## [Resample] LOO iter 2: acc.test.mean=   1
## [Resample] LOO iter 3: acc.test.mean=   1
## [Resample] LOO iter 4: acc.test.mean=   1
## [Resample] LOO iter 5: acc.test.mean=   1
## [Resample] LOO iter 6: acc.test.mean=   1
## [Resample] LOO iter 7: acc.test.mean=   1
## [Resample] LOO iter 8: acc.test.mean=   1
## [Resample] LOO iter 9: acc.test.mean=   1
## [Resample] LOO iter 10: acc.test.mean=   1
## [Resample] LOO iter 11: acc.test.mean=   1
## [Resample] LOO iter 12: acc.test.mean=   1
## [Resample] LOO iter 13: acc.test.mean=   1
## [Resample] LOO iter 14: acc.test.mean=   1
## [Resample] LOO iter 15: acc.test.mean=   1
## [Resample] LOO iter 16: acc.test.mean=   1
## [Resample] LOO iter 17: acc.test.mean=   1
## [Resample] LOO iter 18: acc.test.mean=   1
## [Resample] LOO iter 19: acc.test.mean=   1
## [Resample] LOO iter 20: acc.test.mean=   1
## [Resample] LOO iter 21: acc.test.mean=   1
## [Resample] LOO iter 22: acc.test.mean=   1
## [Resample] LOO iter 23: acc.test.mean=   1
## [Resample] LOO iter 24: acc.test.mean=   1
## [Resample] LOO iter 25: acc.test.mean=   1
## [Resample] LOO iter 26: acc.test.mean=   1
## [Resample] LOO iter 27: acc.test.mean=   1
## [Resample] LOO iter 28: acc.test.mean=   1
## [Resample] LOO iter 29: acc.test.mean=   1
## [Resample] LOO iter 30: acc.test.mean=   1
## [Resample] LOO iter 31: acc.test.mean=   1
## [Resample] LOO iter 32: acc.test.mean=   1
## [Resample] LOO iter 33: acc.test.mean=   1
## [Resample] LOO iter 34: acc.test.mean=   1
## [Resample] LOO iter 35: acc.test.mean=   1
## [Resample] LOO iter 36: acc.test.mean=   1
## [Resample] LOO iter 37: acc.test.mean=   1
## [Resample] LOO iter 38: acc.test.mean=   1
## [Resample] LOO iter 39: acc.test.mean=   1
## [Resample] LOO iter 40: acc.test.mean=   1
## [Resample] LOO iter 41: acc.test.mean=   1
## [Resample] LOO iter 42: acc.test.mean=   1
## [Resample] LOO iter 43: acc.test.mean=   1
## [Resample] LOO iter 44: acc.test.mean=   1
## [Resample] LOO iter 45: acc.test.mean=   1
## [Resample] LOO iter 46: acc.test.mean=   1
## [Resample] LOO iter 47: acc.test.mean=   1
## [Resample] LOO iter 48: acc.test.mean=   1
## [Resample] LOO iter 49: acc.test.mean=   1
## [Resample] LOO iter 50: acc.test.mean=   1
## [Resample] LOO iter 51: acc.test.mean=   1
## [Resample] LOO iter 52: acc.test.mean=   1
## [Resample] LOO iter 53: acc.test.mean=   1
## [Resample] LOO iter 54: acc.test.mean=   1
## [Resample] LOO iter 55: acc.test.mean=   1
## [Resample] LOO iter 56: acc.test.mean=   1
## [Resample] LOO iter 57: acc.test.mean=   1
## [Resample] LOO iter 58: acc.test.mean=   1
## [Resample] LOO iter 59: acc.test.mean=   1
## [Resample] LOO iter 60: acc.test.mean=   1
## [Resample] LOO iter 61: acc.test.mean=   1
## [Resample] LOO iter 62: acc.test.mean=   1
## [Resample] LOO iter 63: acc.test.mean=   1
## [Resample] LOO iter 64: acc.test.mean=   1
## [Resample] LOO iter 65: acc.test.mean=   1
## [Resample] LOO iter 66: acc.test.mean=   1
## [Resample] LOO iter 67: acc.test.mean=   1
## [Resample] LOO iter 68: acc.test.mean=   1
## [Resample] LOO iter 69: acc.test.mean=   1
## [Resample] LOO iter 70: acc.test.mean=   1
## [Resample] LOO iter 71: acc.test.mean=   0
## [Resample] LOO iter 72: acc.test.mean=   1
## [Resample] LOO iter 73: acc.test.mean=   1
## [Resample] LOO iter 74: acc.test.mean=   1
## [Resample] LOO iter 75: acc.test.mean=   1
## [Resample] LOO iter 76: acc.test.mean=   1
## [Resample] LOO iter 77: acc.test.mean=   1
## [Resample] LOO iter 78: acc.test.mean=   0
## [Resample] LOO iter 79: acc.test.mean=   1
## [Resample] LOO iter 80: acc.test.mean=   1
## [Resample] LOO iter 81: acc.test.mean=   1
## [Resample] LOO iter 82: acc.test.mean=   1
## [Resample] LOO iter 83: acc.test.mean=   1
## [Resample] LOO iter 84: acc.test.mean=   0
## [Resample] LOO iter 85: acc.test.mean=   1
## [Resample] LOO iter 86: acc.test.mean=   1
## [Resample] LOO iter 87: acc.test.mean=   1
## [Resample] LOO iter 88: acc.test.mean=   1
## [Resample] LOO iter 89: acc.test.mean=   1
## [Resample] LOO iter 90: acc.test.mean=   1
## [Resample] LOO iter 91: acc.test.mean=   1
## [Resample] LOO iter 92: acc.test.mean=   1
## [Resample] LOO iter 93: acc.test.mean=   1
## [Resample] LOO iter 94: acc.test.mean=   1
## [Resample] LOO iter 95: acc.test.mean=   1
## [Resample] LOO iter 96: acc.test.mean=   1
## [Resample] LOO iter 97: acc.test.mean=   1
## [Resample] LOO iter 98: acc.test.mean=   1
## [Resample] LOO iter 99: acc.test.mean=   1
## [Resample] LOO iter 100: acc.test.mean=   1
## [Resample] LOO iter 101: acc.test.mean=   1
## [Resample] LOO iter 102: acc.test.mean=   1
## [Resample] LOO iter 103: acc.test.mean=   1
## [Resample] LOO iter 104: acc.test.mean=   1
## [Resample] LOO iter 105: acc.test.mean=   1
## [Resample] LOO iter 106: acc.test.mean=   1
## [Resample] LOO iter 107: acc.test.mean=   0
## [Resample] LOO iter 108: acc.test.mean=   1
## [Resample] LOO iter 109: acc.test.mean=   1
## [Resample] LOO iter 110: acc.test.mean=   1
## [Resample] LOO iter 111: acc.test.mean=   1
## [Resample] LOO iter 112: acc.test.mean=   1
## [Resample] LOO iter 113: acc.test.mean=   1
## [Resample] LOO iter 114: acc.test.mean=   1
## [Resample] LOO iter 115: acc.test.mean=   1
## [Resample] LOO iter 116: acc.test.mean=   1
## [Resample] LOO iter 117: acc.test.mean=   1
## [Resample] LOO iter 118: acc.test.mean=   1
## [Resample] LOO iter 119: acc.test.mean=   1
## [Resample] LOO iter 120: acc.test.mean=   0
## [Resample] LOO iter 121: acc.test.mean=   1
## [Resample] LOO iter 122: acc.test.mean=   1
## [Resample] LOO iter 123: acc.test.mean=   1
## [Resample] LOO iter 124: acc.test.mean=   1
## [Resample] LOO iter 125: acc.test.mean=   1
## [Resample] LOO iter 126: acc.test.mean=   1
## [Resample] LOO iter 127: acc.test.mean=   1
## [Resample] LOO iter 128: acc.test.mean=   1
## [Resample] LOO iter 129: acc.test.mean=   1
## [Resample] LOO iter 130: acc.test.mean=   1
## [Resample] LOO iter 131: acc.test.mean=   1
## [Resample] LOO iter 132: acc.test.mean=   1
## [Resample] LOO iter 133: acc.test.mean=   1
## [Resample] LOO iter 134: acc.test.mean=   0
## [Resample] LOO iter 135: acc.test.mean=   1
## [Resample] LOO iter 136: acc.test.mean=   1
## [Resample] LOO iter 137: acc.test.mean=   1
## [Resample] LOO iter 138: acc.test.mean=   1
## [Resample] LOO iter 139: acc.test.mean=   1
## [Resample] LOO iter 140: acc.test.mean=   1
## [Resample] LOO iter 141: acc.test.mean=   1
## [Resample] LOO iter 142: acc.test.mean=   1
## [Resample] LOO iter 143: acc.test.mean=   1
## [Resample] LOO iter 144: acc.test.mean=   1
## [Resample] LOO iter 145: acc.test.mean=   1
## [Resample] LOO iter 146: acc.test.mean=   1
## [Resample] LOO iter 147: acc.test.mean=   1
## [Resample] LOO iter 148: acc.test.mean=   1
## [Resample] LOO iter 149: acc.test.mean=   1
## [Resample] LOO iter 150: acc.test.mean=   1
## [Resample] Aggr. Result: acc.test.mean=0.96
```

```r
ggplot(data.frame(acc = c(result$measures.train$acc, result$measures.test$acc),
                  set = rep(c("train", "test"), each = nrow(result$measures.train))),
    aes(set, acc)) + geom_boxplot() + ylim(0, 1)
```



Blocking

```r
task = makeClassifTask(data = iris, target = "Species", blocking = iris$Species)
rdesc = makeResampleDesc(method = "CV", iters = 3)
result = resample(learner, task, rdesc, measures = acc)
```

```
## [Resample] cross-validation iter 1: acc.test.mean=   0
## [Resample] cross-validation iter 2: acc.test.mean=   0
## [Resample] cross-validation iter 3: acc.test.mean=   0
## [Resample] Aggr. Result: acc.test.mean=   0
```