Input Processing

Data Inspection

```
data = iris
data$Sepal.Length[sample(1:150, 20)] = NA
data$Sepal.Width[sample(1:150, 20)] = NA
data$Petal.Length[sample(1:150, 20)] = NA
data$Petal.Width[sample(1:150, 20)] = NA
data$foo = 1
data$bar = 2
head(data)
```

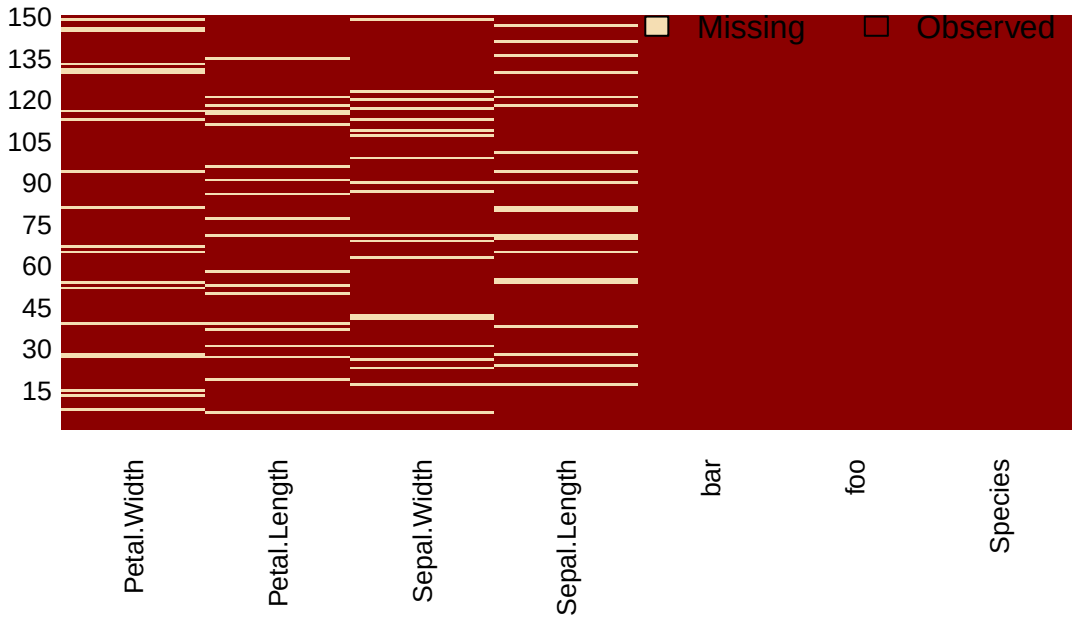
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species foo bar
## 1          5.1         3.5          1.4         0.2  setosa   1   2
## 2          4.9          NA          1.4          NA  setosa   1   2
## 3          4.7         3.2          1.3         0.2  setosa   1   2
## 4           NA         3.1          1.5         0.2  setosa   1   2
## 5          5.0         3.6          1.4          NA  setosa   1   2
## 6          5.4         3.9          1.7          NA  setosa   1   2
```

```
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.4, built: 2015-12-05)
## ## Copyright (C) 2005-2017 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
missmap(data)

library(GGally)
```




```
ggpairs(data)
```

```
## Warning: Removed 20 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 37 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 37 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 34 rows containing missing values
```

```
## Warning: Removed 20 rows containing non-finite values (stat_boxplot).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 20 rows containing missing values
```

```
## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 20 rows containing missing values
```

```
## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```

```
## Warning: Removed 20 rows containing non-finite values (stat_density).
```

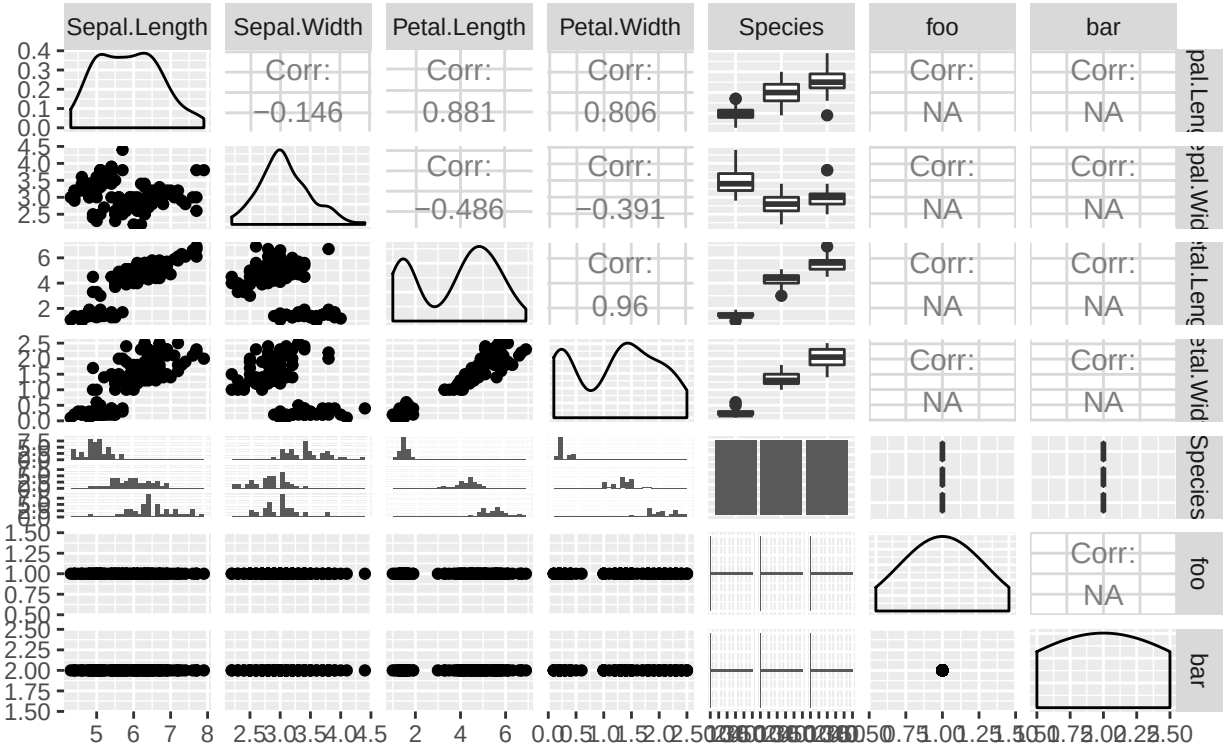
```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 37 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 38 rows containing missing values
```

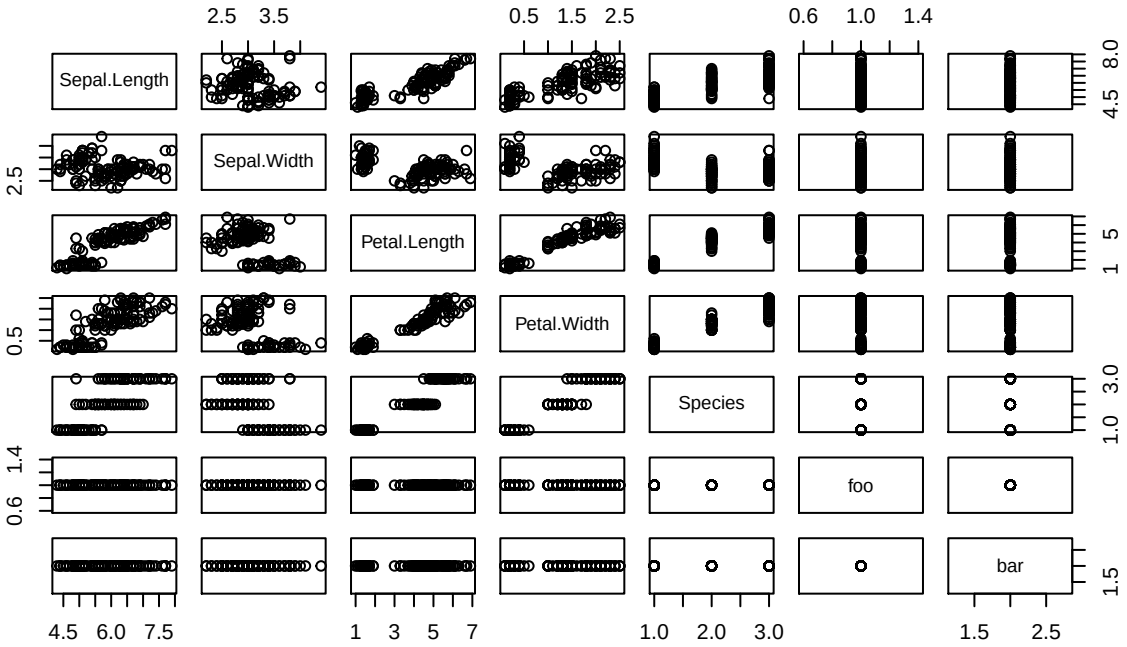
```
## Warning: Removed 20 rows containing non-finite values (stat_boxplot).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 20 rows containing missing values

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 20 rows containing missing values

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning: Removed 37 rows containing missing values (geom_point).

## Warning: Removed 37 rows containing missing values (geom_point).

## Warning: Removed 20 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 37 rows containing missing values

## Warning: Removed 20 rows containing non-finite values (stat_boxplot).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 20 rows containing missing values

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 20 rows containing missing values

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning: Removed 34 rows containing missing values (geom_point).

## Warning: Removed 38 rows containing missing values (geom_point).

## Warning: Removed 37 rows containing missing values (geom_point).

## Warning: Removed 20 rows containing non-finite values (stat_density).

## Warning: Removed 20 rows containing non-finite values (stat_boxplot).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 20 rows containing missing values

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 20 rows containing missing values

## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 20 rows containing non-finite values (stat_bin).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 20 rows containing non-finite values (stat_bin).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 20 rows containing non-finite values (stat_bin).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 20 rows containing non-finite values (stat_bin).
## Warning: Removed 20 rows containing missing values (geom_point).
## Warning: Removed 20 rows containing missing values (geom_point).
## Warning: Removed 20 rows containing missing values (geom_point).
## Warning: Removed 20 rows containing missing values (geom_point).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning in cor(x, y, method = method, use = use): the standard deviation is
## zero
## Warning in cor(x, y, method = method, use = use): Removed 20 rows
## containing missing values (geom_point).
## Warning in cor(x, y, method = method, use = use): Removed 20 rows
## containing missing values (geom_point).
## Warning in cor(x, y, method = method, use = use): Removed 20 rows
## containing missing values (geom_point).
## Warning in cor(x, y, method = method, use = use): Removed 20 rows
## containing missing values (geom_point).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
plot(data)
```



Imputing missing values

Learner with missing values

```
library(mlr)
```

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

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

resample(learner, task, rdesc)
```

```
## Error in checkLearnerBeforeTrain(task, learner, weights): Task 'data' has missing values in 'Sepal.L
```

Imputing

```
learner = makeImputeWrapper("classif.randomForest",
    classes = list(numeric = imputeMedian()))

resample(learner, task, rdesc)
```

```
## [Resample] cross-validation iter 1: mmce.test.mean=0.0667
## [Resample] cross-validation iter 2: mmce.test.mean=0.0667
## [Resample] cross-validation iter 3: mmce.test.mean=0.0667
## [Resample] cross-validation iter 4: mmce.test.mean=0.133
## [Resample] cross-validation iter 5: mmce.test.mean=   0
```

```
## [Resample] cross-validation iter 6: mmce.test.mean=0.0667
## [Resample] cross-validation iter 7: mmce.test.mean=   0
## [Resample] cross-validation iter 8: mmce.test.mean=   0
## [Resample] cross-validation iter 9: mmce.test.mean=0.133
## [Resample] cross-validation iter 10: mmce.test.mean=0.0667
## [Resample] Aggr. Result: mmce.test.mean=0.06
```

```
## Resample Result
## Task: data
## Learner: classif.randomForest.imputed
## Aggr perf: mmce.test.mean=0.06
## Runtime: 1.09004
```

Imputing with dummy columns

```
learner = makeImputeWrapper("classif.randomForest",
    classes = list(numeric = imputeMedian()),
    dummy.classes = c("numeric"))

resample(learner, task, rdesc)
```

```
## [Resample] cross-validation iter 1: mmce.test.mean=   0
## [Resample] cross-validation iter 2: mmce.test.mean=0.0667
## [Resample] cross-validation iter 3: mmce.test.mean=0.0667
## [Resample] cross-validation iter 4: mmce.test.mean=   0
## [Resample] cross-validation iter 5: mmce.test.mean=0.0667
## [Resample] cross-validation iter 6: mmce.test.mean=0.0667
## [Resample] cross-validation iter 7: mmce.test.mean=0.133
## [Resample] cross-validation iter 8: mmce.test.mean=0.0667
## [Resample] cross-validation iter 9: mmce.test.mean=0.267
## [Resample] cross-validation iter 10: mmce.test.mean=   0
## [Resample] Aggr. Result: mmce.test.mean=0.0733
```

```
## Resample Result
## Task: data
## Learner: classif.randomForest.imputed
## Aggr perf: mmce.test.mean=0.0733
## Runtime: 0.726308
```

Removing constant features

```
head(getTaskData(task))
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species foo bar
## 1          5.1         3.5          1.4         0.2  setosa   1   2
## 2          4.9          NA          1.4          NA  setosa   1   2
## 3          4.7         3.2          1.3         0.2  setosa   1   2
## 4           NA         3.1          1.5         0.2  setosa   1   2
## 5          5.0         3.6          1.4          NA  setosa   1   2
## 6          5.4         3.9          1.7          NA  setosa   1   2
```

```
task = removeConstantFeatures(task)
```

```
## Removing 2 columns: foo,bar
```

```
head(getTaskData(task))
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9          NA          1.4          NA  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4           NA         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4          NA  setosa
## 6          5.4         3.9          1.7          NA  setosa
```

Unbalanced classes

```
task = makeClassifTask(data = iris[51:102,], target = "Species")
```

```
## Warning in makeClassifTask(data = iris[51:102, ], target = "Species"):
## Target column 'Species' contains empty factor levels
```

```
table(getTaskTargets(task))
```

```
##
## versicolor  virginica
##         50          2
```

```
learner = makeLearner("classif.rpart")
res = resample(learner, task, rdesc)
```

```
## [Resample] cross-validation iter 1: mmce.test.mean=0.167
## [Resample] cross-validation iter 2: mmce.test.mean=   0
## [Resample] cross-validation iter 3: mmce.test.mean=   0
## [Resample] cross-validation iter 4: mmce.test.mean=   0
## [Resample] cross-validation iter 5: mmce.test.mean=   0
## [Resample] cross-validation iter 6: mmce.test.mean=   0
## [Resample] cross-validation iter 7: mmce.test.mean=   0
## [Resample] cross-validation iter 8: mmce.test.mean=   0
## [Resample] cross-validation iter 9: mmce.test.mean=   0
## [Resample] cross-validation iter 10: mmce.test.mean=0.167
## [Resample] Aggr. Result: mmce.test.mean=0.0333
```

```
table(getPredictionResponse(getRRPredictions(res)))
```

```
##
## versicolor  virginica
##         52          0
```

Undersampling

```
learner.undersample = makeUndersampleWrapper(learner, usw.rate = .03)
res = resample(learner.undersample, task, rdesc)
```

```
## [Resample] cross-validation iter 1: mmce.test.mean=   1
## [Resample] cross-validation iter 2: mmce.test.mean=   1
## [Resample] cross-validation iter 3: mmce.test.mean=   1
## [Resample] cross-validation iter 4: mmce.test.mean=0.167
## [Resample] cross-validation iter 5: mmce.test.mean=   1
## [Resample] cross-validation iter 6: mmce.test.mean=   1
## [Resample] cross-validation iter 7: mmce.test.mean=0.167
## [Resample] cross-validation iter 8: mmce.test.mean=   1
## [Resample] cross-validation iter 9: mmce.test.mean=   1
## [Resample] cross-validation iter 10: mmce.test.mean=   1
## [Resample] Aggr. Result: mmce.test.mean=0.833
```

```
table(getPredictionResponse(getRRPredictions(res)))
```

```
##
## versicolor  virginica
##         12         40
```

Oversampling

```
learner.oversample = makeOversampleWrapper(learner, osw.rate = 10)
res = resample(learner.oversample, task, rdesc)
```

```
## [Resample] cross-validation iter 1: mmce.test.mean=   0
## [Resample] cross-validation iter 2: mmce.test.mean= 0.2
## [Resample] cross-validation iter 3: mmce.test.mean=   0
## [Resample] cross-validation iter 4: mmce.test.mean=   0
## [Resample] cross-validation iter 5: mmce.test.mean=   0
## [Resample] cross-validation iter 6: mmce.test.mean=   0
## [Resample] cross-validation iter 7: mmce.test.mean=0.167
## [Resample] cross-validation iter 8: mmce.test.mean=   0
## [Resample] cross-validation iter 9: mmce.test.mean=0.167
## [Resample] cross-validation iter 10: mmce.test.mean= 0.2
## [Resample] Aggr. Result: mmce.test.mean=0.0733
```

```
table(getPredictionResponse(getRRPredictions(res)))
```

```
##
## versicolor  virginica
##         50          2
```