22 Data Sets

There are a few data sets included in `caret`. The first four are computational chemistry problems where the object is to relate the molecular structure of compounds (via molecular descriptors) to some property of interest (Clark and Pickett (2000)). Similar data sets can be found in the `QSARdata` R pacakge.

Other R packages with data are:

- `mlbench`,
- `SMCRM` and
- `AppliedPredictiveModeling`.

22.1 Blood-Brain Barrier Data

Mente and Lombardo (2005) developed models to predict the log of the ratio of the concentration of a compound in the brain and the concentration in blood. For each compound, they computed three sets of molecular descriptors: MOE 2D, rule-of-five and Charge Polar Surface Area (CPSA). In all, 134 descriptors were calculated. Included in this package are 208 non-proprietary literature compounds. The vector `logBBB` contains the log concentration ratio and the data fame `bbbDescr` contains the descriptor values.

22.2 COX-2 Activity Data

From Sutherland, O'Brien, and Weaver (2003): A set of 467 cyclooxygenase-2 (COX-2) inhibitors has been assembled from the published work of a single research group, with in vitro activities against human recombinant enzyme expressed as IC50 values ranging from 1 nM to >100 uM (53 compounds have indeterminate IC50 values).

A set of 255 descriptors (MOE2D and QikProp) were generated. To classify the data, we used a cutoff of 2^{2.5} to determine activity.

Using `data(cox2)` exposes three R objects: `cox2Descr` is a data frame with the descriptor data, `cox2IC50` is a numeric vector of IC50 assay values and `cox2Class` is a factor vector with the activity results.

22.3 DHFR Inhibition

Sutherland and Weaver (2004) discuss QSAR models for dihydrofolate reductase (DHFR) inhibition. This data set contains values for 325 compounds. For each compound, 228 molecular descriptors have been calculated. Additionally, each samples is designated as “active” or “inactive”.

The data frame `dhfr` contains a column called `Y` with the outcome classification. The remainder of the columns are molecular descriptor values.

22.4 Tecator NIR Data

These data can be found in the datasets section of StatLib. The data consist of 100 near infrared absorbance spectra used to predict the moisture, fat and protein values of chopped meat.

From StatLib:

> These data are recorded on a Tecator Infratec Food and Feed Analyzer working in the wavelength range 850 - 1050 nm by the Near Infrared Transmission (NIT) principle. Each sample contains finely chopped pure meat with different moisture, fat and protein contents. If results from these data are used in a publication we want you to mention the instrument and company name (Tecator) in the publication. In addition, please send a preprint of your article to: Karin Thente, Tecator AB, Box 70, S-263 21 Hoganas, Sweden.

One reference for these data is Borggaard and Thodberg (1992).

Using `data(tecator)` loads a 215 x 100 matrix of absorbance spectra and a 215 x 3 matrix of outcomes.

22.5 Fatty Acid Composition Data

Brodnjak-Voncina et al. (2005) describe a set of data where seven fatty acid compositions were used to classify commercial oils as either pumpkin (labeled `A`), sunflower (`B`), peanut (`C`), olive (`D`), soybean (`E`), rapeseed (`F`) and corn (`G`). There were 96 data points contained in their Table 1 with known results. The breakdown of the classes is given in below:

```
data(oil)
dim(fattyAcids)
```

```
## [1] 96  7
```

```
table(oilType)
```

```
## oilType
##  A  B  C  D  E  F  G
## 37 26  3  7 11 10  2
```

As a note, the paper states on page 32 that there are 37 unknown samples while the table on pages 33 and 34 shows that there are 34 unknowns.

22.6 German Credit Data

Data from Dr. Hans Hofmann of the University of Hamburg and stored at the UC Irvine Machine Learning Repository.

These data have two classes for the credit worthiness: good or bad. There are predictors related to attributes, such as: checking account status, duration, credit history, purpose of the loan, amount of the loan, savings accounts or bonds, employment duration, Installment rate in percentage of disposable income, personal information, other debtors/guarantors, residence duration, property, age, other installment plans, housing, number of existing credits, job information, Number of people being liable to provide maintenance for, telephone, and foreign worker status.

Many of these predictors are discrete and have been expanded into several 0/1 indicator variables

```
library(caret)
data(GermanCredit)
## Show the first 10 columns
str(GermanCredit[, 1:10])
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ status             : Factor w/ 4 levels "... < 100 DM",..: 1 2 4 1 1 4 4 2 4 2 ...
##  $ duration           : num  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history     : Factor w/ 5 levels "no credits taken/all credits paid back dul
##  $ purpose            : Factor w/ 10 levels "car (new)","car (used)",..: 5 5 8 4 1 8 4
##  $ amount             : num  1169 5951 2096 7882 4870 ...
##  $ savings            : Factor w/ 5 levels "... < 100 DM",..: 5 1 1 1 1 5 3 1 4 1 ...
##  $ employment_duration: Ord.factor w/ 5 levels "unemployed"<"... < 1 year"<..: 5 3 4 4
##  $ installment_rate   : num  4 2 2 2 3 2 3 2 2 4 ...
##  $ personal_status_sex: Factor w/ 5 levels "male : divorced/separated",..: 3 2 3 3 3 3
##  $ other_debtors      : Factor w/ 3 levels "none","co-applicant",..: 1 1 1 3 1 1 1 1 1
```

22.7 Kelly Blue Book

Resale data for 2005 model year GM cars Kuiper (2008) collected data on Kelly Blue Book resale data for 804 GM cars (2005 model year).

`cars` is data frame of the suggested retail price (column `Price`) and various characteristics of each car (columns `Mileage`, `Cylinder`, `Doors`, `Cruise`, `Sound`, `Leather`, `Buick`, `Cadillac`, `Chevy`, `Pontiac`, `Saab`, `Saturn`, `convertible`, `coupe`, `hatchback`, `sedan` and `wagon`)

```
data(cars)
str(cars)
```

```
## 'data.frame':    804 obs. of  18 variables:
##  $ Price      : num  22661 21725 29143 30732 33359 ...
##  $ Mileage    : int  20105 13457 31655 22479 17590 23635 17381 27558 25049 17319 ...
##  $ Cylinder   : int  6 6 4 4 4 4 4 4 4 4 ...
##  $ Doors      : int  4 2 2 2 2 2 2 2 2 4 ...
##  $ Cruise     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Sound      : int  0 1 1 0 1 0 1 0 0 0 ...
##  $ Leather    : int  0 0 1 0 1 0 1 1 0 1 ...
##  $ Buick      : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Cadillac   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chevy      : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Pontiac    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Saab       : int  0 0 1 1 1 1 1 1 1 1 ...
##  $ Saturn     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ convertible: int  0 0 1 1 1 1 1 1 1 0 ...
##  $ coupe      : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ hatchback  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sedan      : int  1 0 0 0 0 0 0 0 0 1 ...
##  $ wagon      : int  0 0 0 0 0 0 0 0 0 0 ...
```

22.8 Cell Body Segmentation Data

Hill, LaPan, Li and Haney (2007) develop models to predict which cells in a high content screen were well segmented. The data consists of 119 imaging measurements on 2019. The original analysis used 1009 for training and 1010 as a test set (see the column called `Case`).

The outcome class is contained in a factor variable called `Class` with levels `PS` for poorly segmented and `WS` for well segmented.

```
data(segmentationData)
str(segmentationData[,1:10])
```

```
## 'data.frame':    2019 obs. of  10 variables:
##  $ Cell                  : int  207827637 207932307 207932463 207932470 207932455 2078
##  $ Case                  : Factor w/ 2 levels "Test","Train": 1 2 2 2 1 1 1 1 1 1 ...
##  $ Class                 : Factor w/ 2 levels "PS","WS": 1 1 2 1 1 2 2 1 2 2 ...
##  $ AngleCh1              : num  143.25 133.75 106.65 69.15 2.89 ...
##  $ AreaCh1               : int  185 819 431 298 285 172 177 251 495 384 ...
##  $ AvgIntenCh1           : num  15.7 31.9 28 19.5 24.3 ...
##  $ AvgIntenCh2           : num  4.95 206.88 116.32 102.29 112.42 ...
##  $ AvgIntenCh3           : num  9.55 69.92 63.94 28.22 20.47 ...
##  $ AvgIntenCh4           : num  2.21 164.15 106.7 31.03 40.58 ...
##  $ ConvexHullAreaRatioCh1: num  1.12 1.26 1.05 1.2 1.11 ...
```

22.9 Sacramento House Price Data

This data frame contains house and sale price data for 932 homes in Sacramento CA. The original data were obtained from the website for the SpatialKey software. From their website: "The Sacramento real estate transactions file is a list of 985 real estate transactions in the Sacramento area reported over a five-day period, as reported by the Sacramento Bee." Google was used to fill in missing/incorrect data.

```
data(Sacramento)
str(Sacramento)
```

```
## 'data.frame':    932 obs. of  9 variables:
##  $ city     : Factor w/ 37 levels "ANTELOPE","AUBURN",..: 34 34 34 34 34 34 34 34 29 3
##  $ zip      : Factor w/ 68 levels "z95603","z95608",..: 64 52 44 44 53 65 66 49 24 25
##  $ beds     : int  2 3 2 2 2 3 3 3 2 3 ...
##  $ baths    : num  1 1 1 1 1 1 2 1 2 2 ...
##  $ sqft     : int  836 1167 796 852 797 1122 1104 1177 941 1146 ...
##  $ type     : Factor w/ 3 levels "Condo","Multi_Family",..: 3 3 3 3 3 1 3 3 1 3 ...
##  $ price    : int  59222 68212 68880 69307 81900 89921 90895 91002 94905 98937 ...
##  $ latitude : num  38.6 38.5 38.6 38.6 38.5 ...
##  $ longitude: num  -121 -121 -121 -121 -121 ...
```

22.10 Animal Scat Data

Reid (2105) collected data on animal feses in coastal California. The data consist of DNA verified species designations as well as fields related to the time and place of the collection and the scat itself. The data frame `scat_orig` contains while scat contains data on the three main species.

```
data(scat)
str(scat)
```

```
## 'data.frame':    110 obs. of  19 variables:
##  $ Species  : Factor w/ 3 levels "bobcat","coyote",..: 2 2 1 2 2 2 1 1 1 1 ...
##  $ Month    : Factor w/ 9 levels "April","August",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Year     : int  2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...
##  $ Site     : Factor w/ 2 levels "ANNU","YOLA": 2 2 2 2 2 2 1 1 1 1 ...
##  $ Location : Factor w/ 3 levels "edge","middle",..: 1 1 2 2 1 1 3 3 3 2 ...
##  $ Age      : int  5 3 3 5 5 5 1 3 5 5 ...
##  $ Number   : int  2 2 2 2 4 3 5 7 2 1 ...
##  $ Length   : num  9.5 14 9 8.5 8 9 6 5.5 11 20.5 ...
##  $ Diameter : num  25.7 25.4 18.8 18.1 20.7 21.2 15.7 21.9 17.5 18 ...
##  $ Taper    : num  41.9 37.1 16.5 24.7 20.1 28.5 8.2 19.3 29.1 21.4 ...
##  $ TI       : num  1.63 1.46 0.88 1.36 0.97 1.34 0.52 0.88 1.66 1.19 ...
##  $ Mass     : num  15.9 17.6 8.4 7.4 25.4 ...
##  $ d13C     : num  -26.9 -29.6 -28.7 -20.1 -23.2 ...
##  $ d15N     : num  6.94 9.87 8.52 5.79 7.01 8.28 4.2 3.89 7.34 6.06 ...
##  $ CN       : num  8.5 11.3 8.1 11.5 10.6 9 5.4 5.6 5.8 7.7 ...
##  $ ropey    : int  0 0 1 1 0 1 1 0 0 1 ...
##  $ segmented: int  0 0 1 0 1 0 1 1 1 1 ...
##  $ flat     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ scrape   : int  0 0 1 0 0 0 1 0 0 0 ...
```