

### Libraries to Import

library(readr)	library(dplyr)
library(ggplot2)	library(broom)
library(caret)	library(rpart)
library(splines)	library(party)
library(leaps)	library(glmnet)
library(MASS)	library(class)

### Data Conversion

as.array(x)	as.character(x)
as.data.frame(x)	as.factor(x)
as.logical(x)	as.numeric(x)

### K-Fold

**folds <- crossv\_kfold(data, k = 5)** [k is the number of folds]

### Regularization - Ridge & Lasso

**lambdas\_to\_try <- 10seq(-3, 5, length.out = 100)**<sup>^</sup>

**ridge\_cv <- cv.glmnet(X, y, alpha = 0, lambda = lambdas\_to\_try, standardize = TRUE, nfolds = 10)** [Setting alpha = 0 implements ridge regression]

**lasso <- glmnet(data\_x, data\_y, alpha = 1.0)**

### Random Forest

**randomForest(formula, data)** [formula is a formula describing the predictor and response variables. data is the name of the data set used]

### Basic Codes

**read\_csv("path/nhanes.csv")** [Read nhanes.csv in the path/ folder (readr)]

**View(df)** [View tabular data frame df in a graphical viewer]

**mean, median, range** [Descriptive stats. Remember na.rm=TRUE if desired]

**filter(df, ...)** [Filters data frame according to condition ... (dplyr)]

### Basic Codes (cont)

**factor(x, levels=c("wt", "mutant"))**  
[Create factor specifying level order]

**relevel(x, ref="wildtype")** [Re-level a factor variable]

**t.test(y~grp, data=df)** [T-test mean y across grp in data df]

**lmfit <- lm(y~x1+x2, data=df)** [Fit linear model y against two x's]

**anova(lmfit)** [Print ANOVA table on object returned from lm()]

**summary(lmfit)** [Get summary information about a model fit with lm()]

**TukeyHSD(aov(lmfit))** [ANOVA Post-hoc pairwise contrasts]

**wilcox.test(y~grp, data=df)** [Wilcoxon rank sum / Mann-Whitney U test]

**xt <- xtabs(~x1+x2, data=df)** [Cross-tabulate a contingency table]

**addmargins(xt)** [Adds summary margin to a contingency table xt]

**addmargins(xt)** [Adds summary margin to a contingency table xt]

**chisq.test(xt)** [Chi-square test on a contingency table xt]

**fisher.test(xt)** [Fisher's exact test on a contingency table xt]

**mosaicplot(xt)** [Mosaic plot for a contingency table xt]

**power.t.test(n, power, sd, delta)** [T-test power calculations]

**power.prop.test(n, power, p1, p2)** [Proportions test power calculations]

**tidy() augment() glance()** [Model tidying functions in the broom package]

### Data Information

is.na(x)	is.null(x)
is.nan(x)	is.array(x)
is.data.frame(x)	is.numeric(x)
is.complex(x)	is.character(x)
head(x)	tail(x)
summary(x)	str(x)
length(x)	dim(x)
dimnames(x)	attr(x,which)
nrow(x)	ncol(x)
NROW(x)	NCOL(x)
class(x)	unclass(x)

### Data Splitting

**createDataPartition(y,p=0.8)** [createDataPartition splits a vector 'y' with 80 percent data in one part and 20 percent in other part partition(y,p=0.8)]

**trainControl( summaryFunction=<Rfunction>,classProbs=<logical>)** [It is used for controlling training parameters like resampling, number of folds, iteration etc.]

**densityplot.rfe(x,data,...)** [Lattice functions for plotting resampling results of recursive feature selection]

**featureplot(x,y,plot...)** [A shortcut to produce lattice plots]

