```
> library(tidyverse)
── Attaching packages ─────────────────────────────────────── tidyverse 1.2.1 ──
✔ ggplot2 3.2.1     ✔ purrr   0.3.2
✔ tibble  2.1.3     ✔ dplyr   0.8.3
✔ tidyr   0.8.3     ✔ stringr 1.4.0
✔ readr   1.3.1     ✔ forcats 0.4.0
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
> library(caret)
Loading required package: lattice

Attaching package: 'caret'

The following object is masked from 'package:purrr':

    lift

> set.seed(1996, sample.kind="Rounding")
Warning message:
In set.seed(1996, sample.kind = "Rounding") :
  non-uniform 'Rounding' sampler used
> n <- 1000
> p <- 10000
> x <- matrix(rnorm(n*p), n, p)
> colnames(x) <- paste("x", 1:ncol(x), sep = "_")
> y <- rbinom(n, 1, 0.5) %>% factor()
>
> x_subset <- x[ ,sample(p, 100)]
>
> class(x)
[1] "matrix"
> nrow(x)
[1] 1000
> ncol(x)
[1] 10000
> class(y)
[1] "factor"
> length(y)
[1] 1000
> y[1:10]
 [1] 1 1 0 0 1 0 0 1 1 0
Levels: 0 1
> class(x_subset)
[1] "matrix"
> nrow(x_subset)
[1] 1000
> ncol(x_subset)
[1] 100
>
>
> fit <- train(x_subset, y, method = 'glm')
> fit
Generalized Linear Model

1000 samples
 100 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 1000, 1000, 1000, 1000, 1000, 1000, ...
Resampling results:

  Accuracy   Kappa
```

```
  0.4987228  -0.002558887

> class(fit)
[1] "train"
> names(fit)
 [1] "method"       "modelInfo"    "modelType"    "results"      "pred"
 [6] "bestTune"     "call"         "dots"         "metric"       "control"
[11] "finalModel"   "preProcess"   "trainingData" "resample"     "resampledCM"
[16] "perfNames"    "maximize"     "yLimits"      "times"        "levels"
> fit$results
  parameter  Accuracy        Kappa AccuracySD    KappaSD
1      none 0.4987228 -0.002558887 0.02112972 0.04240727
>
>
> library(genefilter)

Attaching package: 'genefilter'

The following object is masked from 'package:readr':

    spec

> tt <- colttests(x, y)
> class(tt)
[1] "data.frame"
> dim(tt)
[1] 10000     3
> head(tt)
     statistic          dm    p.value
x_1 -0.8875044 -0.05406068 0.37502121
x_2  0.7902597  0.05051311 0.42956387
x_3 -0.7428500 -0.04933976 0.45774730
x_4 -0.2910172 -0.01848353 0.77109863
x_5 -0.1859136 -0.01185577 0.85255029
x_6  1.6852477  0.10731283 0.09225341
>
>
> ind <- which(tt$p.value <= 0.01)
> length(ind)
[1] 108
>
>
> x_subset <- x[,ind]
> class(x_subset)
[1] "matrix"
> nrow(x_subset)
[1] 1000
> ncol(x_subset)
[1] 108
> fit <- train(x_subset, y, method = 'glm')
> fit
Generalized Linear Model

1000 samples
 108 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 1000, 1000, 1000, 1000, 1000, 1000, ...
Resampling results:

  Accuracy   Kappa
  0.7564915  0.5120586
```

```
> fit$results
  parameter  Accuracy     Kappa AccuracySD    KappaSD
1      none 0.7564915 0.5120586 0.02277298 0.04528652
>
>
> k = seq(101, 301, 25)
> fit <- train(x_subset, y, method = "knn", tuneGrid = data.frame(k = k))
> class(fit)
[1] "train"
> fit
k-Nearest Neighbors

1000 samples
 108 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 1000, 1000, 1000, 1000, 1000, 1000, ...
Resampling results across tuning parameters:

  k    Accuracy   Kappa
  101  0.7332153  0.4632081
  126  0.7316208  0.4600619
  151  0.7299220  0.4565029
  176  0.7336308  0.4638693
  201  0.7322994  0.4607487
  226  0.7318058  0.4597253
  251  0.7303479  0.4567133
  276  0.7321866  0.4605668
  301  0.7323656  0.4608626

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 176.
> ggplot(fit)
>
>
> library(dslabs)
> data(tissue_gene_expression)
> class(tissue_gene_expression)
[1] "list"
> names(tissue_gene_expression)
[1] "x" "y"
> dim(tissue_gene_expression$x)
[1] 189 500
> dim(as.matrix(tissue_gene_expression$y))
[1] 189   1
> k <- seq(1,7,2)
> fit <- train(tissue_gene_expression$x, tissue_gene_expression$y, method = "knn", tuneGrid = data.frame(
k = k))
> fit
k-Nearest Neighbors

189 samples
500 predictors
  7 classes: 'cerebellum', 'colon', 'endometrium', 'hippocampus', 'kidney', 'liver', 'placenta'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 189, 189, 189, 189, 189, 189, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  1  0.9906966  0.9887216
  3  0.9784504  0.9738911
```

```
  5  0.9741887  0.9687972
  7  0.9687547  0.9622020

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 1.
> fit$results
  k  Accuracy     Kappa AccuracySD    KappaSD
1 1 0.9906031 0.9884838 0.01398458 0.01716581
2 3 0.9794253 0.9749698 0.02034206 0.02469029
3 5 0.9801134 0.9757701 0.02451657 0.02988792
4 7 0.9771149 0.9721463 0.02477876 0.03022948
> ggplot(fit)
>
```