```r
library(MASS)
library(ISLR)
head(Smarket)
```

```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5 Volume  Today Direction
## 1 2001  0.381 -0.192 -2.624 -1.055  5.010 1.1913  0.959        Up
## 2 2001  0.959  0.381 -0.192 -2.624 -1.055 1.2965  1.032        Up
## 3 2001  1.032  0.959  0.381 -0.192 -2.624 1.4112 -0.623      Down
## 4 2001 -0.623  1.032  0.959  0.381 -0.192 1.2760  0.614        Up
## 5 2001  0.614 -0.623  1.032  0.959  0.381 1.2057  0.213        Up
## 6 2001  0.213  0.614 -0.623  1.032  0.959 1.3491  1.392        Up
```

Logistic Regression

```r
glm_fit <- glm(Direction ~ Lag1 + Lag2,
    data = Smarket,
    family = binomial
)
glm_fit
```

```
##
## Call:  glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = Smarket)
##
## Coefficients:
## (Intercept)         Lag1         Lag2
##     0.07425     -0.07151     -0.04450
##
## Degrees of Freedom: 1249 Total (i.e. Null);  1247 Residual
## Null Deviance:       1731
## Residual Deviance: 1728  AIC: 1734
```

```r
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------

## v ggplot2 3.1.0       v purrr   0.2.5
## v tibble  2.0.1       v dplyr   0.8.0.1
## v tidyr   0.8.1       v stringr 1.4.0
## v readr   1.1.1       v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```r
library(modelr)
new_data <- read_csv("Lag1, Lag2
                      0.5, 0.3
                      0.4, 0.3")
new_data %>% add_predictions(glm_fit) %>% mutate(prob = exp(pred)/ (1 + exp(pred)))
```

```
## # A tibble: 2 x 4
##    Lag1  Lag2   pred  prob
##   <dbl> <dbl>  <dbl> <dbl>
## 1   0.5   0.3 0.0251 0.506
## 2   0.4   0.3 0.0323 0.508
```

ROC curve

```
Smarket2 <- Smarket %>%
    add_predictions(glm_fit) %>%
    mutate(prob = exp(pred)/ (1 + exp(pred)), EstDir = ifelse(prob > 0.5, "Up", "Down"))
Smarket2 %>% count(Direction, EstDir) %>% spread(Direction, n)
```

```
## # A tibble: 2 x 3
##   EstDir  Down    Up
##   <chr>  <int> <int>
## 1 Down     114   102
## 2 Up       488   546
```

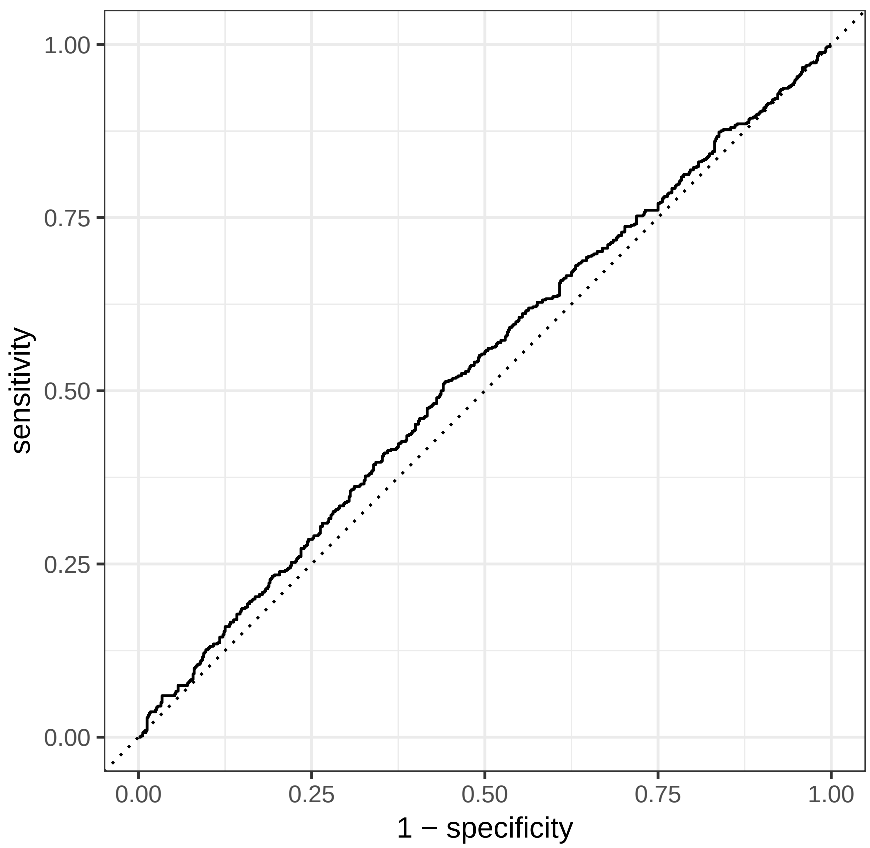
```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------------------------------------------------

## v broom     0.5.0     v recipes   0.1.4
## v dials     0.0.2     v rsample   0.0.4
## v infer     0.4.0     v yardstick 0.0.2
## v parsnip   0.0.1

## -- Conflicts -----------------------------------------------------------------------------------------
## x broom::bootstrap() masks modelr::bootstrap()
## x scales::discard()  masks purrr::discard()
## x dplyr::filter()    masks stats::filter()
## x recipes::fixed()   masks stringr::fixed()
## x dplyr::lag()       masks stats::lag()
## x yardstick::mae()   masks modelr::mae()
## x yardstick::mape()  masks modelr::mape()
## x yardstick::rmse()  masks modelr::rmse()
## x dplyr::select()    masks MASS::select()
## x yardstick::spec()  masks readr::spec()
## x recipes::step()    masks stats::step()
## x yardstick::tidy()  masks rsample::tidy(), recipes::tidy(), broom::tidy()
```

```
autoplot(roc_curve(Smarket2, Direction, prob))
```




```
roc_auc(Smarket2, Direction, prob)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.531
```

Multinomial logistic

variable name	type	about the variable
id	scale	student id
female	nominal	(0/1)
race	nominal	ethnicity (1=hispanic 2=asian 3=african-amer 4=white)
ses	ordinal	socio economic status (1=low 2=middle 3=high)
schtyp	nominal	type of school (1=public 2=private)
prog	nominal	type of program (1=general 2=academic 3=vocational)
read	scale	standardized reading score
write	scale	standardized writing score
math	scale	standardized math score
science	scale	standardized science score
socst	scale	standardized social studies score
hon	nominal	honors english (0/1)

```r
ml <- read_csv("hsb2.csv")
```

```
## Parsed with column specification:
## cols(
##   id = col_integer(),
##   female = col_integer(),
##   race = col_integer(),
##   ses = col_integer(),
##   schtyp = col_integer(),
##   prog = col_integer(),
##   read = col_integer(),
##   write = col_integer(),
##   math = col_integer(),
##   science = col_integer(),
##   socst = col_integer()
## )
```

```r
hsb2 <- ml %>% mutate(
    prog = recode_factor(prog, `1` = "general", `2` = "academic", `3` = "vocational"),
    ses = recode_factor(ses, `1` = "low", `2` = "middle", `3` = "high"))
```

```r
hsb2 %>% count(prog, ses)  %>% spread(prog, n)
```

```
## # A tibble: 3 x 4
##   ses    general academic vocational
##   <fct>    <int>    <int>      <int>
## 1 low         16       19         12
## 2 middle      20       44         31
## 3 high         9       42          7
```

```r
hsb2 %>% group_by(prog) %>% summarize(mwrite = mean(write))
```

```
## # A tibble: 3 x 2
##   prog       mwrite
##   <fct>       <dbl>
## 1 general      51.3
## 2 academic     56.3
## 3 vocational   46.8
```

```r
library(nnet)
multi_fit <- multinom(prog ~ ses + write, data = hsb2)
```

```
## # weights:  15 (8 variable)
## initial  value 219.722458
## iter  10 value 179.985215
## final  value 179.981726
## converged
```

```r
new_data <- tibble(ses = "middle", write = 56)
predict(multi_fit, new_data, type = "probs")
```

```
##   general   academic vocational
##  0.2174957  0.5485545  0.2339499
```

```r
# tidyverse
new_data %>% add_predictions(multi_fit)
```

```
## # A tibble: 1 x 3
##   ses    write pred
##   <chr>  <dbl> <fct>
## 1 middle    56 academic
```

```r
predict(multi_fit, new_data)
```

```
## [1] academic
## Levels: general academic vocational
```

```r
# a few month later
# new_data %>% add_predictions(multi_fit, type = "probs")
```

LDA

```r
lda_fit <- lda(Direction ~ Lag1 + Lag2, data = Smarket)
```

```r
Smarket3 <- Smarket %>% mutate(EstDir = predict(lda_fit, newdata = Smarket)$class)
Smarket3 %>% count(Direction, EstDir) %>% spread(Direction, n)
```

```
## # A tibble: 2 x 3
##   EstDir  Down    Up
##   <fct>  <int> <int>
## 1 Down     114   102
## 2 Up       488   546
```

```r
new_data <- read_csv("Lag1, Lag2
                      0.5, 0.3
                      0.4, 0.3")

predict(lda_fit, new_data)
```

```
## $class
## [1] Up Up
## Levels: Down Up
##
## $posterior
##         Down        Up
```

```
## 1 0.4936720 0.5063280
## 2 0.4918909 0.5081091
##
## $x
##          LD1
## 1 -0.5148696
## 2 -0.4391935
```

```
# modelr magic doesn't work now
# new_data %>% add_predictions(lda_fit)
```

An exmple with more than one class

```
head(iris)
```

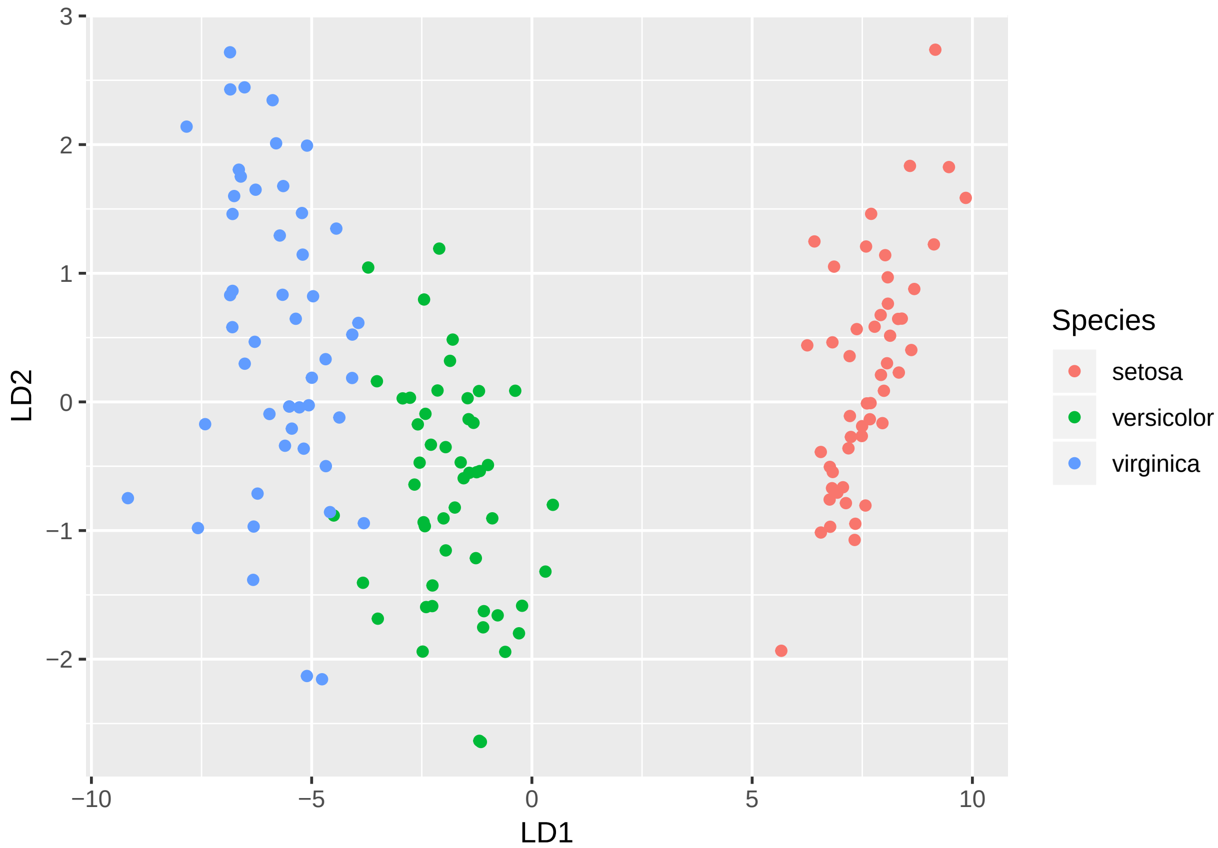
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
iris_fit <- lda(Species ~ .,  data = iris)
```

```
new_data <-  tibble(Sepal.Length = 5.906, Sepal.Width = 2.77, Petal.Length = 3, Petal.Width = 0.246)
predict(iris_fit, new_data)$posterior
```

```
##      setosa   versicolor    virginica
## 1 0.9996068 0.0003931945 7.917338e-19
```

```
iris2 <- iris %>% bind_cols(as_tibble(predict(iris_fit, newdata = iris)$x))
ggplot(iris2) + geom_point(aes(LD1, LD2, color = Species))
```



Reduced rank LDA

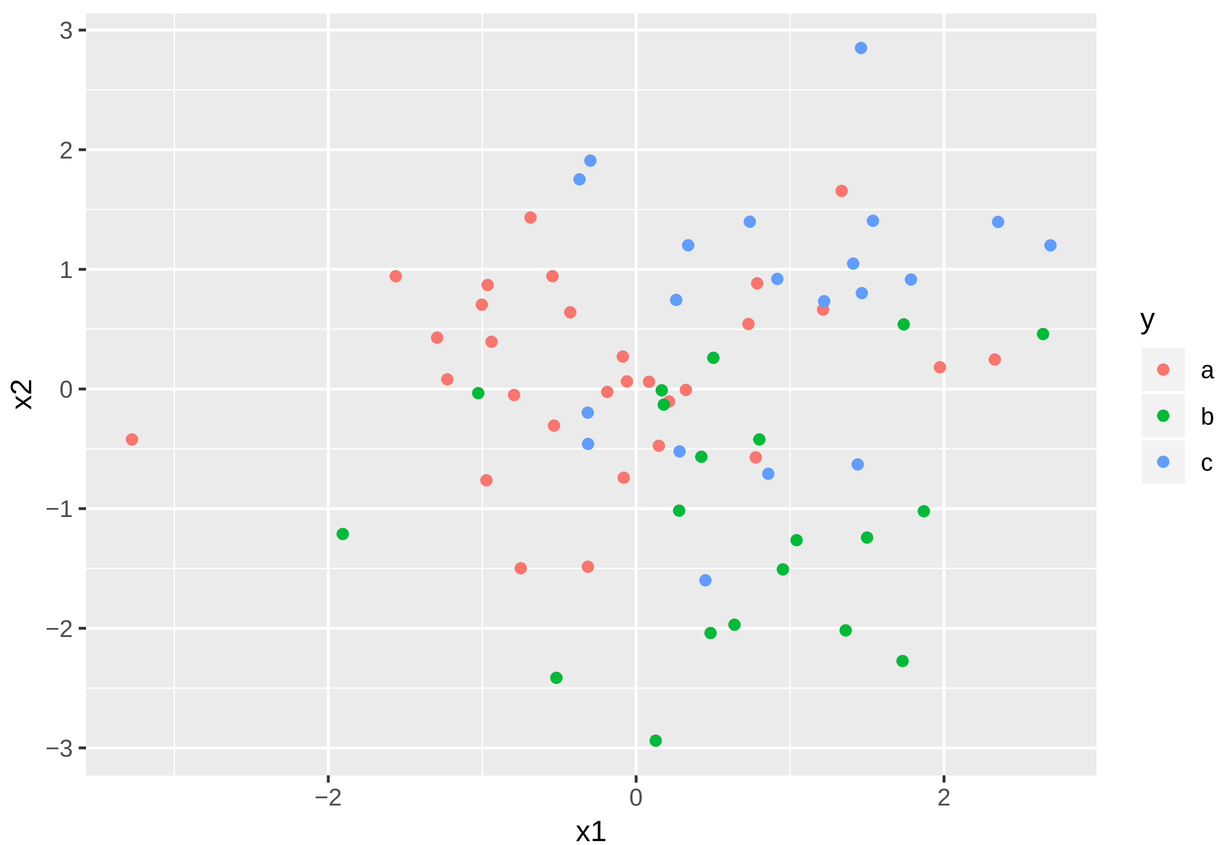
```r
# only use LD1
predict(iris_fit, new_data, dimen = 1)
```

```
## $class
## [1] setosa
## Levels: setosa versicolor virginica
##
## $posterior
##      setosa   versicolor    virginica
## 1 0.9999577 4.230918e-05 1.925451e-18
##
## $x
##        LD1
## 1 3.958893
```

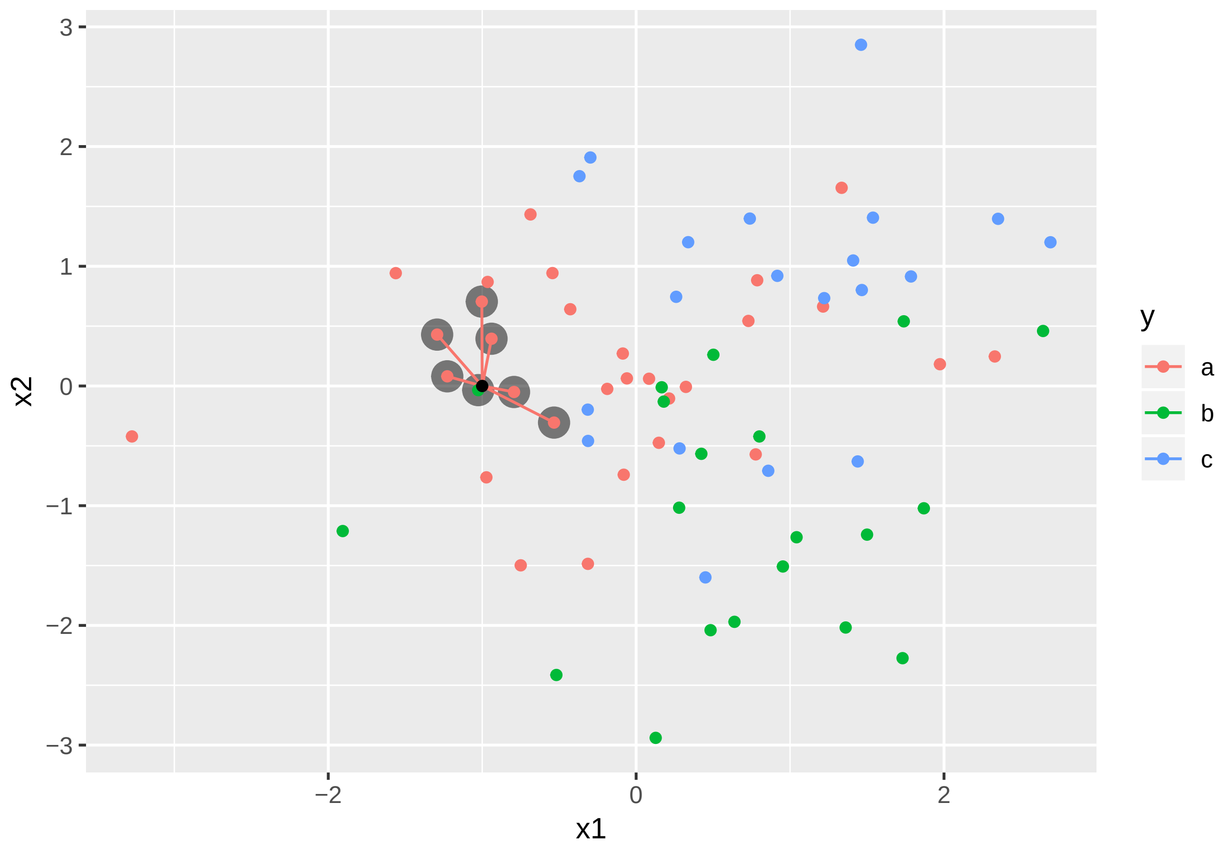
Explaining kNN

```
library(mvtnorm)
g1 <- rmvnorm(30, mean=c(-0.5,0))
g2 <- rmvnorm(20, mean=c(1,-1))
g3 <- rmvnorm(20, mean=c(1,1))
x <- rbind(g1, g2, g3)
colnames(x) <- c("x1", "x2")
x <- as_tibble(x)
y <- rep(c("a", "b", "c"), c(30, 20, 20))
knn_example <- bind_cols(x, y = y)
```

```
ggplot(knn_example) + geom_point(aes(x1, x2, color = y))
```




```
k = 7
point = c(-1, 0)
knn_neighor <- knn_example %>%
  mutate(dist = sqrt((x1 - point[1])^2 + (x2 - point[2])^2)) %>%
  filter(row_number(dist) <= k)
ggplot(knn_example) +
  geom_point(data = knn_neighor, aes(x1, x2), alpha = 0.5, size = 5) +
  geom_segment(data = knn_neighor, aes(x = x1, y = x2, xend = point[1], yend = point[2], color = y)) +
  geom_point(aes(x1, x2, color = y)) +
  annotate("point", x = point[1], y = point[2])
```




```
library(class)
new_data <- read_csv("x1, x2
                      0.2, 1
                      0.6, -1")
new_data %>% mutate(est_class = knn(knn_example %>% select(x1, x2), new_data, knn_example %>% pull(y)))
```

```
## # A tibble: 2 x 3
##      x1    x2 est_class
##   <dbl> <int> <fct>
## 1   0.2     1 c
## 2   0.6    -1 b
```