Logistic regression in R

Data

Wisconsin Breast Cancer Database covers 683 observations of 10 variables in relation to examining tumors in the breast.

- Nine clinical variables with a score between 0 and 10.
- The binary variable `Class` with levels `benign/malignant`.

We will use 4 of the predictors, where 2 have been discretized.

```
library(tidyverse)
```

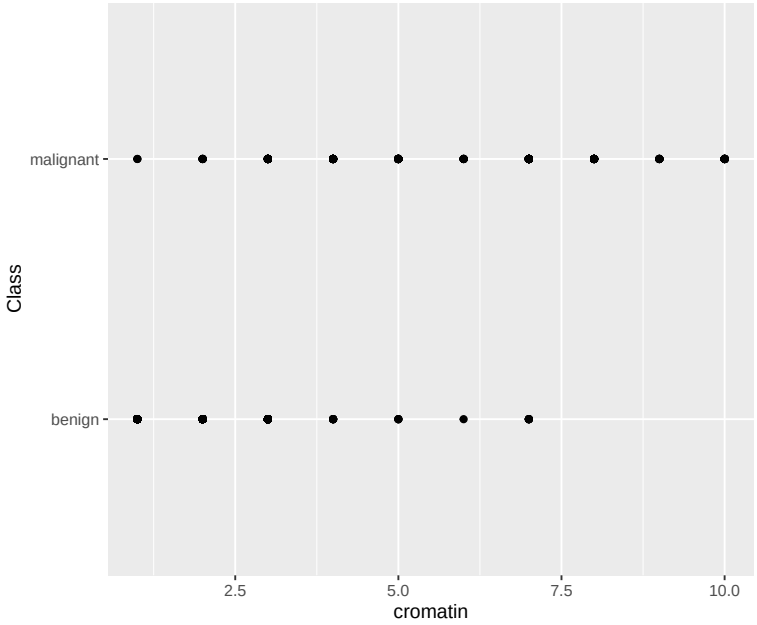
```
## -- Attaching core tidyverse packages ------------------------------- tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
BC <- read_delim("https://asta.math.aau.dk/datasets?file=BC0.dat",
                 col_types = cols(Class = col_factor()))
BC |> print(n = 6)
```

```
## # A tibble: 683 x 6
##   nuclei cromatin Size.low Size.medium Shape.low Class
##    <dbl>    <dbl> <lgl>    <lgl>       <lgl>     <fct>
## 1      1        3 TRUE     FALSE       TRUE      benign
## 2     10        3 FALSE    TRUE        FALSE     benign
## 3      2        3 TRUE     FALSE       TRUE      benign
## 4      4        3 FALSE    FALSE       FALSE     benign
## 5      1        3 TRUE     FALSE       TRUE      benign
## 6     10        9 FALSE    FALSE       FALSE     malignant
## # i 677 more rows
```

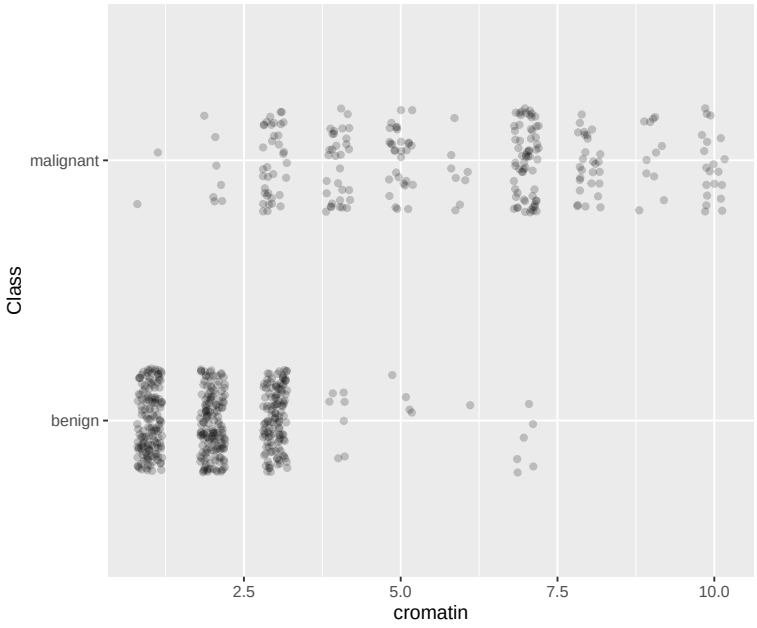
Simple scatter plot

```
BC |>
  ggplot(aes(x = cromatin, y = Class)) +
  geom_point()
```



Jittered scatter plot

```
BC |>
  ggplot(aes(x = cromatin, y = Class)) +
  geom_jitter(width = .2, height = .2, alpha = .2)
```



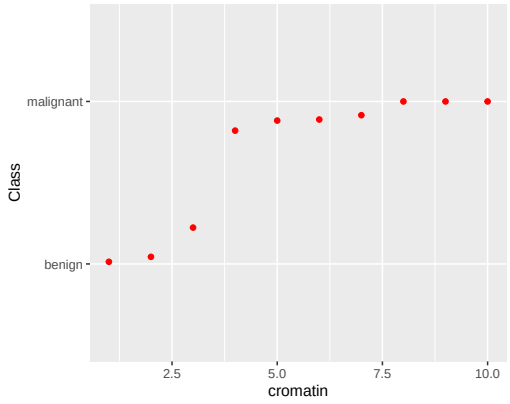
Empirical probabilities

```
BCtable <- BC |>
  group_by(cromatin, Class) |>
  summarise(n = n()) |>
  mutate(prop = n / sum(n))
```

```
## `summarise()` has grouped output by 'cromatin'. You can override using the `.groups`
## argument.
```

```
props <- BCtable |>
  filter(Class == "malignant")

prop_plot <- BC |> ggplot(aes(x = cromatin, y = Class)) +
  geom_point(alpha = 0) +
  geom_point(aes(y = 1+prop), data = props, col = "red")
prop_plot
```



Estimated simple logistic regression and confidence intervals

```
library(broom)
model <- glm(Class ~ cromatin, data = BC, family = "binomial")
info <- tidy(model) |>
  mutate(low_ci = estimate-1.96*std.error, high_ci = estimate+1.96*std.error) |>
  mutate(exp_low_ci = exp(low_ci), exp_high_ci = exp(high_ci))
info
```

```
## # A tibble: 2 x 9
##   term        estimate std.error statistic  p.value low_ci high_ci exp_low_ci exp_high_ci
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>  <dbl>   <dbl>      <dbl>       <dbl>
## 1 (Intercept)    -5.28     0.392     -13.5 2.20e-41  -6.05   -4.51    0.00236      0.0110
## 2 cromatin        1.37     0.117      11.6 2.62e-31   1.14    1.60    3.11         4.93
```

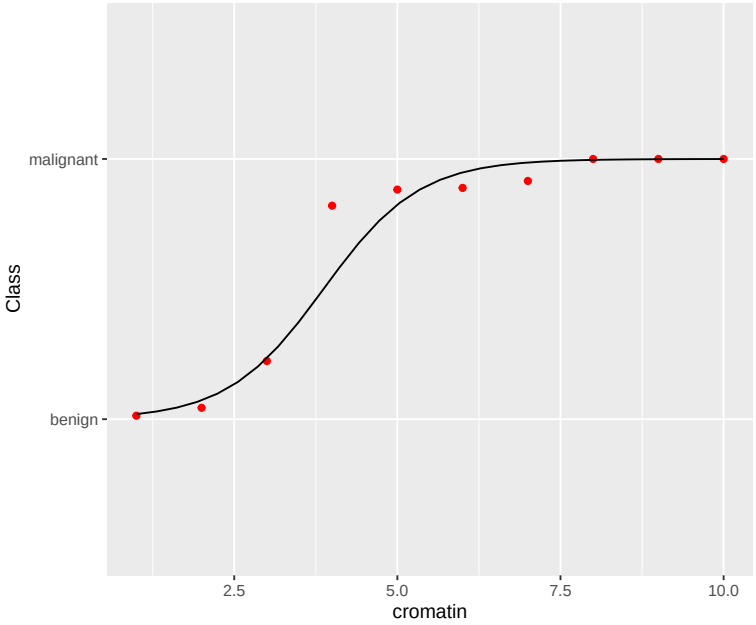
Plot of model predictions against actual data

```
library(modelr)
```

```
##
## Attaching package: 'modelr'
```

```
## The following object is masked from 'package:broom':
##
##     bootstrap
```

```
pred_data <- BC |>
  data_grid(cromatin = seq_range(cromatin, n = 30)) |>
  add_predictions(model, type = "response")
prop_plot +
  geom_line(aes(x = cromatin, y = pred + 1), data = pred_data)
```



Lethal dose 50%

The value of the covariate corresponding to predicted probability of 0.5:

$$\beta_0 + \beta_1 x_1 = \text{logit}(0.5) = \log(\frac{0.5}{1-0.5}) = 0$$

So LD50 is:

$$x_1 = -\beta_0/\beta_1$$

The estimated LD50 is given by plugging in the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$. The *delta method* is a general technique to find the approximate std. error of an expression involving the parameter estimates. With the package `car` we find it like this:

```
car::deltaMethod(model, "-Intercept/cromatin")
```

```
##                      Estimate      SE   2.5 % 97.5 %
## -Intercept/cromatin  3.86717 0.11382 3.64408 4.0903
```

```
car::deltaMethod(model, "-b0/b1", parameterNames = c("b0", "b1"))
```

```
##        Estimate      SE   2.5 % 97.5 %
## -b0/b1  3.86717 0.11382 3.64408 4.0903
```

Alternatively you can use `MASS::dose.p()` for LD50:

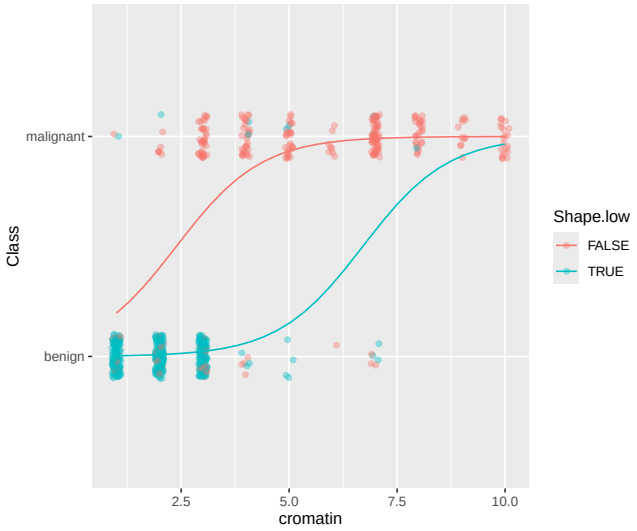
```
MASS::dose.p(model)
```

```
##              Dose        SE
## p = 0.5: 3.867168 0.1138234
```

Multiple logistic regression

Additive model where the effect of `cromatin` is the same for every level of `Shape.low`:

```
model1 <- glm(Class ~ cromatin + Shape.low, data = BC, family = binomial)
pred_data <- BC |>
  data_grid(Shape.low, cromatin = seq_range(cromatin, n = 30)) |>
  add_predictions(model1, type = "response")
ggplot(BC, aes(x = cromatin, color = Shape.low, y = Class)) +
  geom_jitter(width=.1, height=.1, alpha = 0.4) +
  geom_line(aes(x = cromatin, y = pred + 1, color = Shape.low), data = pred_data)
```



Multiple logistic regression

Interaction model where the effect of `cromatin` is allowed to depend on the level of `Shape.low`:

```
model2 <- glm(Class ~ cromatin * Shape.low, data = BC, family = binomial)
tidy(model2)
```

```
## # A tibble: 4 x 5
##   term                   estimate std.error statistic  p.value
##   <chr>                     <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)              -2.38      0.525    -4.54  5.52e- 6
## 2 cromatin                  1.00      0.157     6.40  1.57e-10
## 3 Shape.lowTRUE            -4.49      1.02     -4.40  1.08e- 5
## 4 cromatin:Shape.lowTRUE    0.0307    0.255     0.120 9.04e- 1
```

Multiple logistic regression

Test for combined effect of all predictors vs. no predictors:

```
noEffects <- glm(Class ~ 1, data = BC, family = binomial)
mainEffects <- glm(Class ~ ., data = BC, family = binomial)
anova(noEffects, mainEffects, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Class ~ 1
## Model 2: Class ~ nuclei + cromatin + Size.low + Size.medium + Shape.low
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       682     884.35
## 2       677     135.06  5   749.29 < 2.2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Recall that `~ .` means "all remaining variables in data".

Interactions

```
intEffects <- glm(Class ~ .^2, data = BC, family = binomial)
final <- step(intEffects, k = log(nrow(BC)), trace = 0)
tidy(final)
```

```
## # A tibble: 7 x 5
##   term                estimate std.error statistic    p.value
##   <chr>                  <dbl>     <dbl>     <dbl>      <dbl>
## 1 (Intercept)           0.0337    0.903     0.0373 0.970
## 2 nuclei                0.302     0.0837    3.60   0.000314
## 3 cromatin              0.446     0.144     3.09   0.00198
## 4 Size.lowTRUE         -5.42      1.14     -4.77   0.00000182
## 5 Size.mediumTRUE      -2.29      0.690    -3.33   0.000874
## 6 Shape.lowTRUE        -2.25      0.649    -3.47   0.000525
## 7 nuclei:Size.lowTRUE   0.569     0.236     2.41   0.0157
```

Predicted probabilities

```
BCpred <- BC |>
  add_predictions(final, type = "response") |>
  mutate(pred_50 = ifelse(pred>.5, "malignant", "benign")) |>
  mutate(pred_10 = ifelse(pred>.1, "malignant", "benign"))
BCpred
```

```
## # A tibble: 683 x 9
##    nuclei cromatin Size.low Size.medium Shape.low Class        pred pred_50   pred_10
##     <dbl>    <dbl> <lgl>    <lgl>       <lgl>     <fct>       <dbl> <chr>     <chr>
##  1      1        3 TRUE     FALSE       TRUE      benign    0.00437 benign    benign
##  2     10        3 FALSE    TRUE        FALSE     benign    0.890   malignant malignant
##  3      2        3 TRUE     FALSE       TRUE      benign    0.0104  benign    benign
##  4      4        3 FALSE    FALSE       FALSE     benign    0.929   malignant malignant
##  5      1        3 TRUE     FALSE       TRUE      benign    0.00437 benign    benign
##  6     10        9 FALSE    FALSE       FALSE     malignant 0.999   malignant malignant
##  7     10        3 TRUE     FALSE       TRUE      benign    0.917   malignant malignant
##  8      1        3 TRUE     FALSE       TRUE      benign    0.00437 benign    benign
##  9      1        1 TRUE     FALSE       TRUE      benign    0.00180 benign    benign
## 10      1        2 TRUE     FALSE       TRUE      benign    0.00280 benign    benign
## # i 673 more rows
```

```
BCpred |> count(Class, pred_50)
```

```
## # A tibble: 4 x 3
##   Class     pred_50       n
##   <fct>     <chr>     <int>
## 1 benign    benign      432
## 2 benign    malignant    12
## 3 malignant benign       11
## 4 malignant malignant   228
```

```
BCpred |> count(Class, pred_10)
```

```
## # A tibble: 4 x 3
##   Class     pred_10       n
##   <fct>     <chr>     <int>
## 1 benign    benign      418
## 2 benign    malignant    26
## 3 malignant benign        2
## 4 malignant malignant   237
```

Confusion table format:

```
BCpred |> count(Class, pred_50) |> pivot_wider(names_from = pred_50, values_from = n)
```

```
## # A tibble: 2 x 3
##   Class     benign malignant
##   <fct>      <int>     <int>
## 1 benign       432        12
## 2 malignant     11       228
```

```
BCpred |> count(Class, pred_10) |> pivot_wider(names_from = pred_10, values_from = n)
```

```
## # A tibble: 2 x 3
##   Class     benign malignant
##   <fct>      <int>     <int>
## 1 benign       418        26
## 2 malignant      2       237
```

LD50 for multiple logistic regression

The "usual" LD50 formula for cromatin score corresponds to assuming all other predictors are zero:

```
car::deltaMethod(final, "-Intercept/cromatin")
```

```
##                       Estimate       SE    2.5 % 97.5 %
## -Intercept/cromatin -0.07562  2.04105 -4.07601 3.9248
```

In our case that would be a case with `nuclei` score of 0 and both shape and size score `"high"` (reference group which is coded as 0). If instead we are interested in a case with `nuclei` score 1 (median value) with shape and size score `"low"`:

```
car::deltaMethod(final, "-(b0+b3+b5+(b1+b6)*1)/b2", parameterNames = paste0("b", 0:6))
```

```
##                                     Estimate      SE   2.5 % 97.5 %
## -(b0 + b3 + b5 + (b1 + b6) * 1)/b2  15.1829  4.3660  6.6256  23.74
```

Estimation from aggregated data

```
BCtable
```

```
## # A tibble: 17 x 4
## # Groups:   cromatin [10]
##    cromatin Class         n   prop
##       <dbl> <fct>     <int>  <dbl>
##  1        1 benign      148 0.987
##  2        1 malignant     2 0.0133
##  3        2 benign      153 0.956
##  4        2 malignant     7 0.0438
```

```
##  5        3 benign      125 0.776
##  6        3 malignant    36 0.224
##  7        4 benign        7 0.179
##  8        4 malignant    32 0.821
##  9        5 benign        4 0.118
## 10        5 malignant    30 0.882
## 11        6 benign        1 0.111
## 12        6 malignant     8 0.889
## 13        7 benign        6 0.0845
## 14        7 malignant    65 0.915
## 15        8 malignant    28 1
## 16        9 malignant    11 1
## 17       10 malignant    20 1
```

```
model <- glm(Class ~ cromatin, weights = n, data = BCtable, family = "binomial")
tidy(model)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    -5.28     0.392     -13.5 2.22e-41
## 2 cromatin        1.37     0.117      11.6 2.65e-31
```

```
BCtable2 <- BCtable |>
  pivot_wider(id_cols = cromatin, names_from = Class, values_from = n, values_fill = 0)
pander::pander(BCtable2)
```

cromatin	benign	malignant
1	148	2
2	153	7
3	125	36
4	7	32
5	4	30
6	1	8
7	6	65
8	0	28
9	0	11
10	0	20

```
model <- glm(cbind(malignant, benign) ~ cromatin, data = BCtable2, family = "binomial")
tidy(model)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    -5.28     0.392     -13.5 2.22e-41
## 2 cromatin        1.37     0.117      11.6 2.64e-31
```

From aggregated data to individual cases

```
tidy_BC <- BCtable |>
  select(-prop) |>
  uncount(n)
tidy_BC
```

```
## # A tibble: 683 x 2
## # Groups:   cromatin [10]
##    cromatin Class
##       <dbl> <fct>
##  1        1 benign
##  2        1 benign
##  3        1 benign
##  4        1 benign
##  5        1 benign
##  6        1 benign
##  7        1 benign
##  8        1 benign
##  9        1 benign
## 10        1 benign
## # i 673 more rows
```

```
tidy_BC2 <- BCtable2 |>
  pivot_longer(c(benign, malignant),
                names_to = "type",
                values_to = "n") |>
  uncount(n)
tidy_BC2
```

```
## # A tibble: 683 x 2
## # Groups:   cromatin [10]
##    cromatin type
##       <dbl> <chr>
##  1        1 benign
##  2        1 benign
##  3        1 benign
##  4        1 benign
##  5        1 benign
##  6        1 benign
##  7        1 benign
##  8        1 benign
##  9        1 benign
## 10        1 benign
## # i 673 more rows
```

```
tidy_BC3 <- doBy::binomial_to_bernoulli_data(BCtable2, y = malignant, size = benign)
```