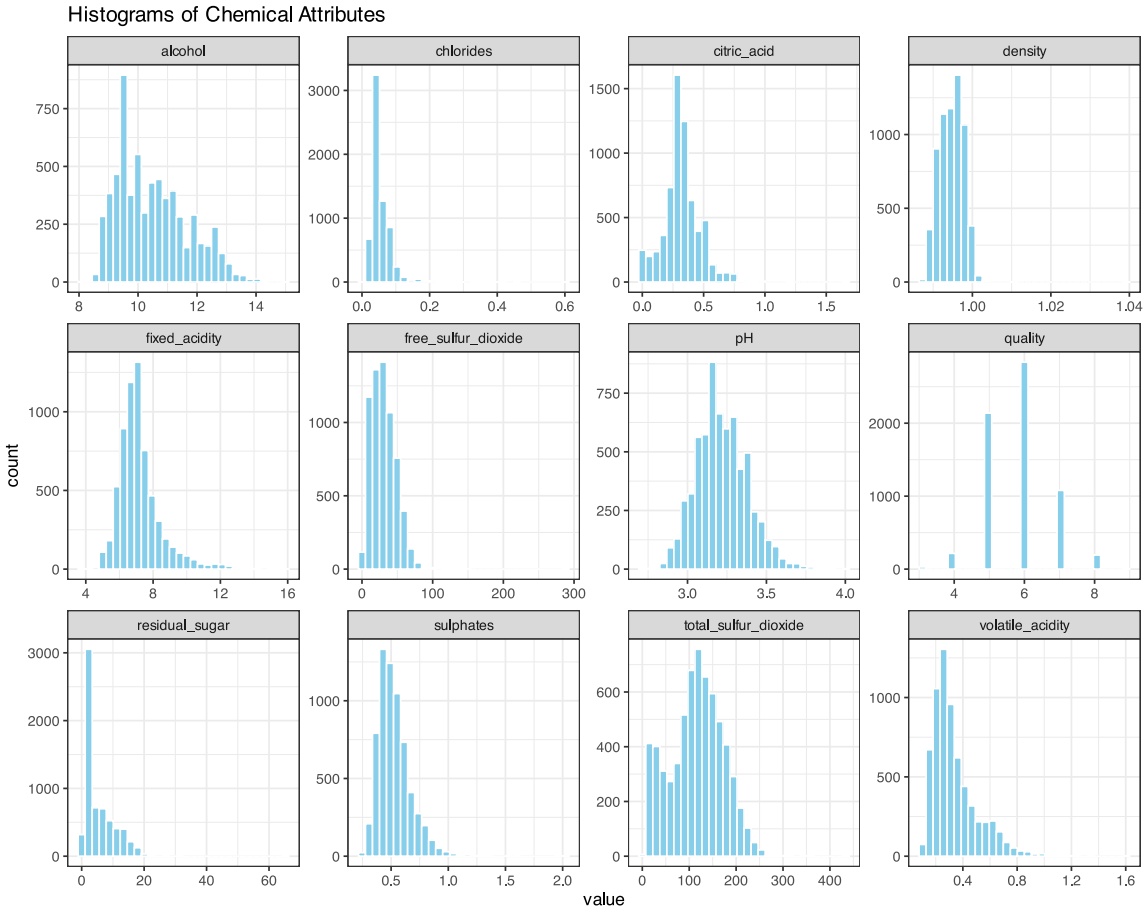
Data Processing

```
red <- read.csv("data/winequality-red.csv", sep = ";") |> mutate(type = "0")
white <- read.csv("data/winequality-white.csv", sep = ";") |> mutate(type =
"1")

wine <- bind_rows(red, white)
colnames(wine) <- str_replace_all(colnames(wine), pattern = "\\.", "_")
wine|> write_csv("data/wine.csv")
```

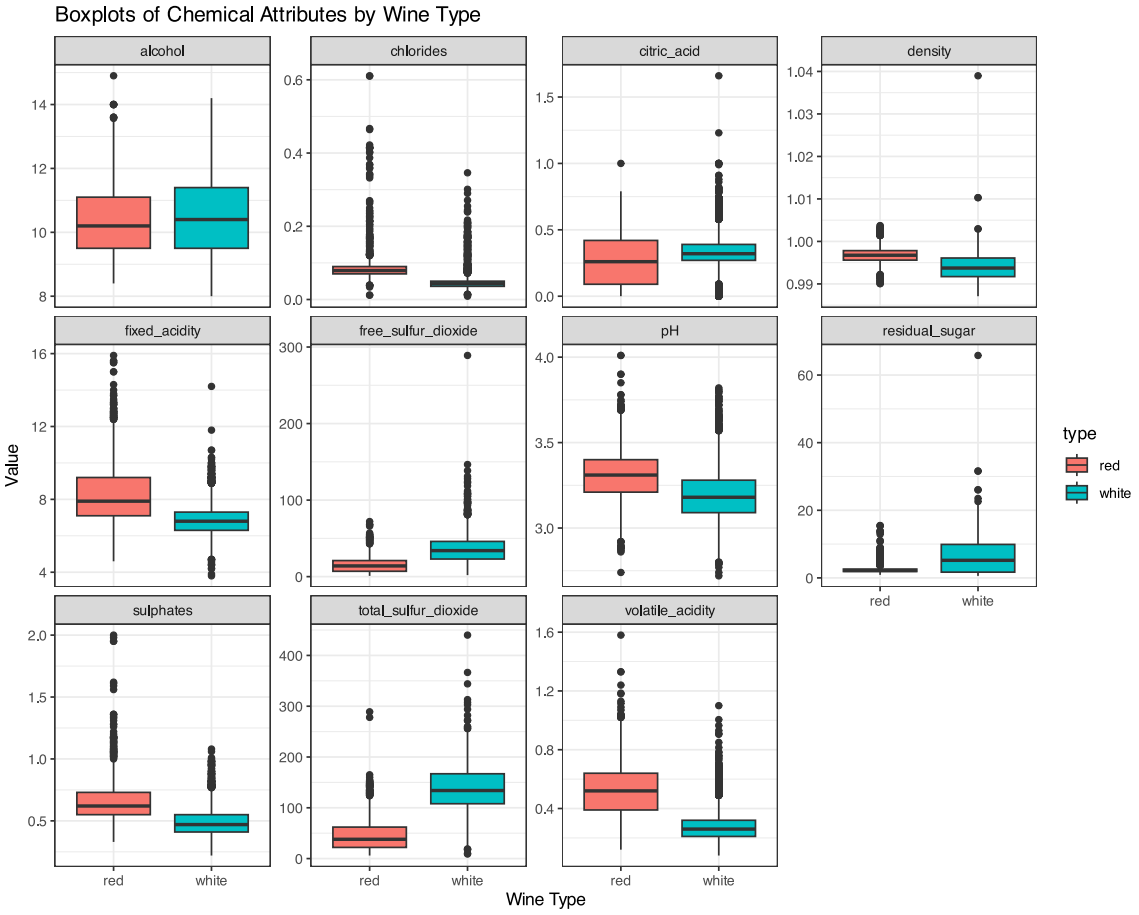
EDA

```
# preparation
wine <- read_csv("data/wine.csv")
wine <- wine |>
  mutate(
    type = factor(
      type,
      levels = c(0, 1),
      labels = c("red", "white")
    )
  )
```

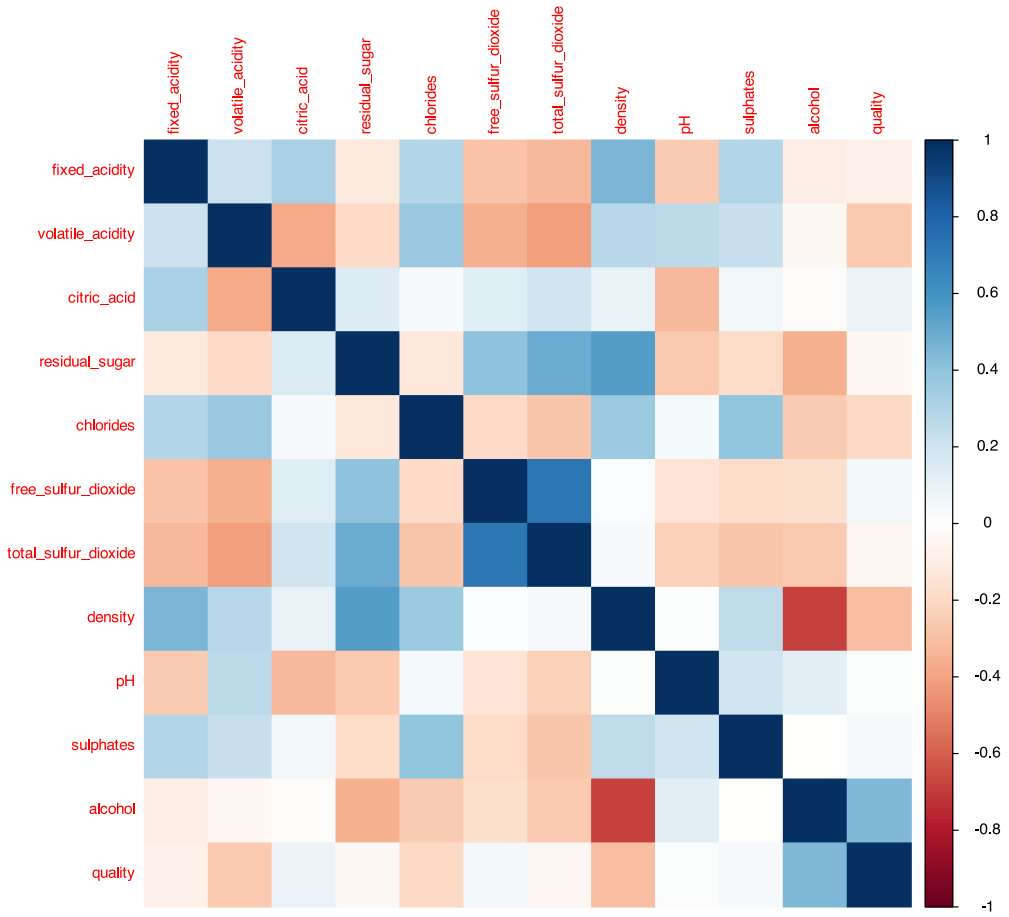
```
# only numerical variables, excluding types
wine_numeric <- wine[, sapply(wine, is.numeric)]
wine_long <- pivot_longer(wine_numeric,
                          cols = everything(),
                          names_to = "variable",
                          values_to = "value")

ggplot(wine_long, aes(x = value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "white") +
  facet_wrap(~ variable, scales = "free") +
  theme_bw() +
  labs(title = "Histograms of Chemical Attributes")
```



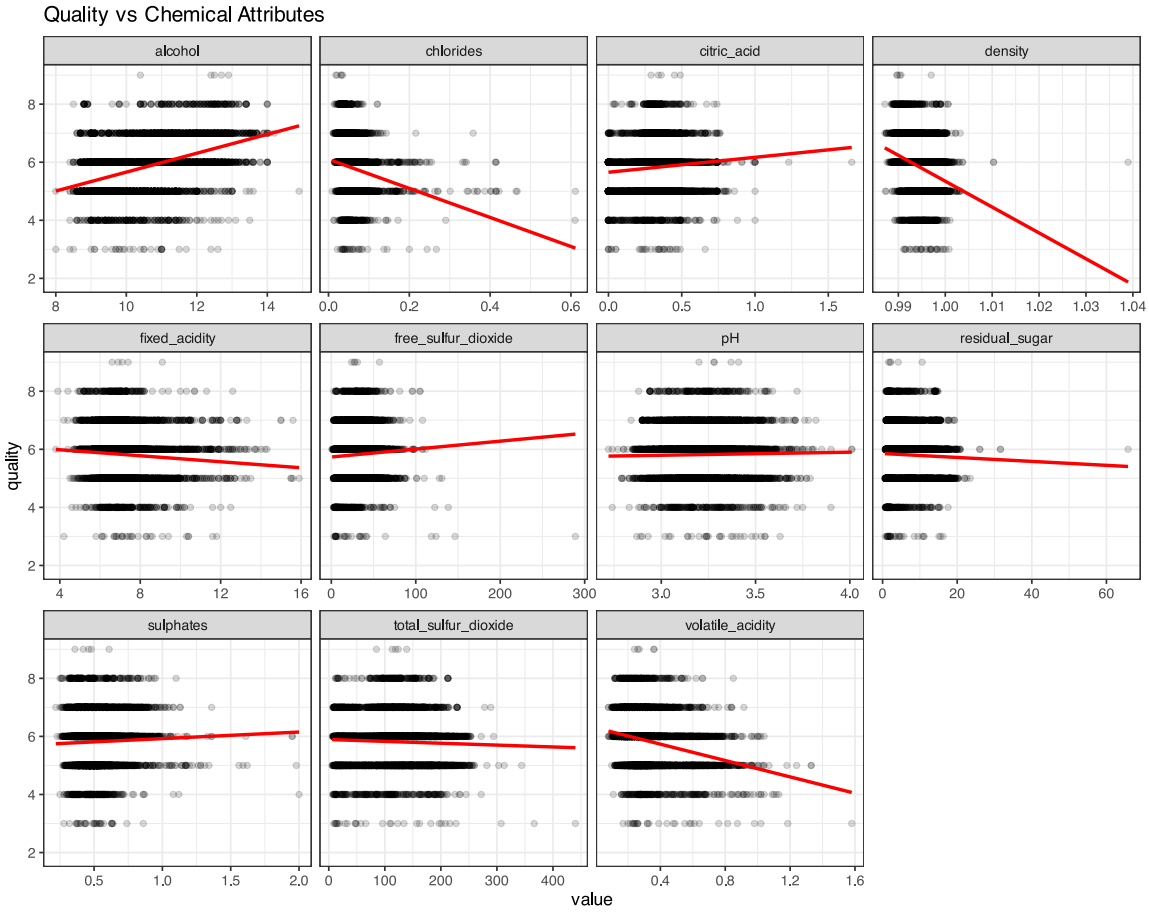

```
# Boxplot: chemical variable by type
wine_long2 <- pivot_longer(wine,
                           cols = -c(type, quality),
                           names_to = "variable",
                           values_to = "value")

ggplot(wine_long2, aes(x = type, y = value, fill = type)) +
  geom_boxplot() +
  facet_wrap(~ variable, scales = "free_y") +
  theme_bw() +
  labs(title = "Boxplots of Chemical Attributes by Wine Type",
       x = "Wine Type", y = "Value")
```




```r
# correlation heatmap
corr_matrix <- cor(wine_numeric)   # numeric variables only
corrplot(corr_matrix, method = "color", tl.cex = 0.8)
```

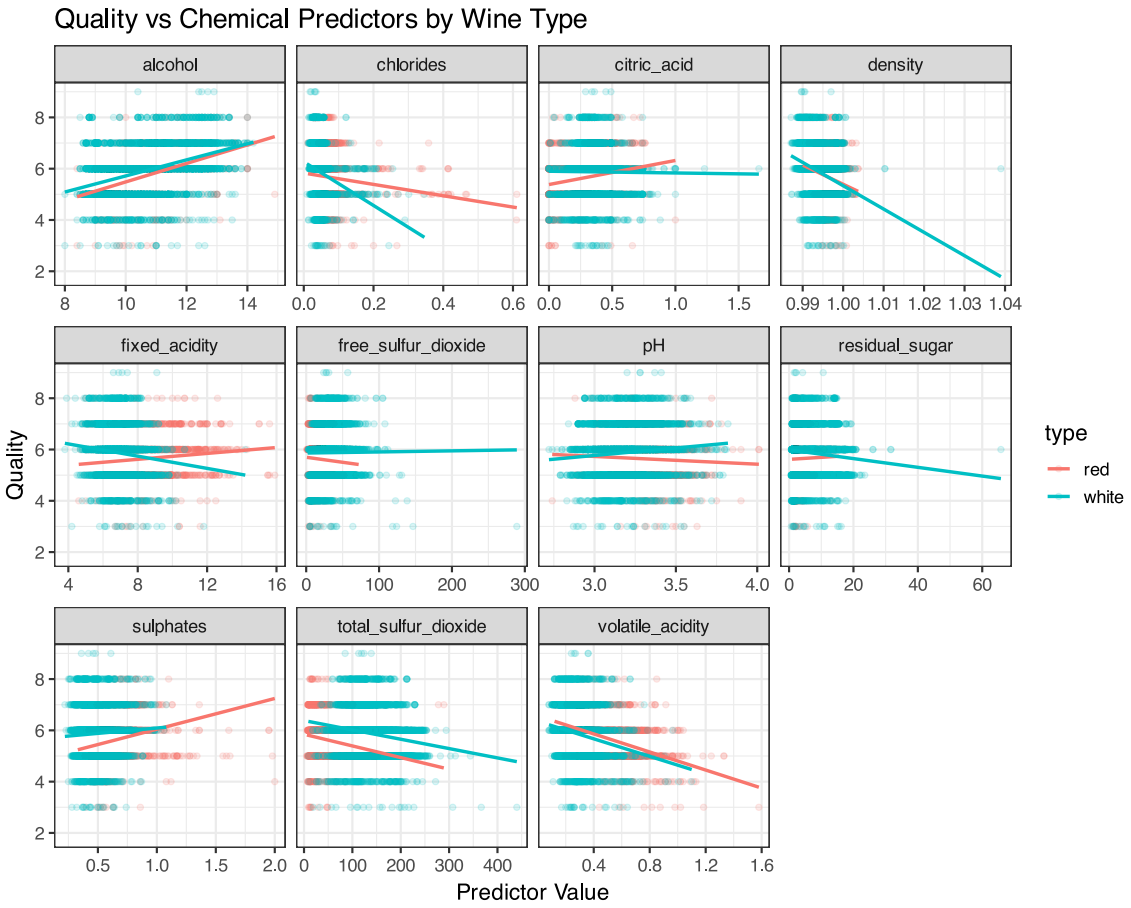



```r
# scatterplot
ggplot(wine_long2, aes(x = value, y = quality)) +
  geom_point(alpha = 0.15) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  facet_wrap(~ variable, scales = "free_x") +
  theme_bw() +
  labs(title = "Quality vs Chemical Attributes")
```

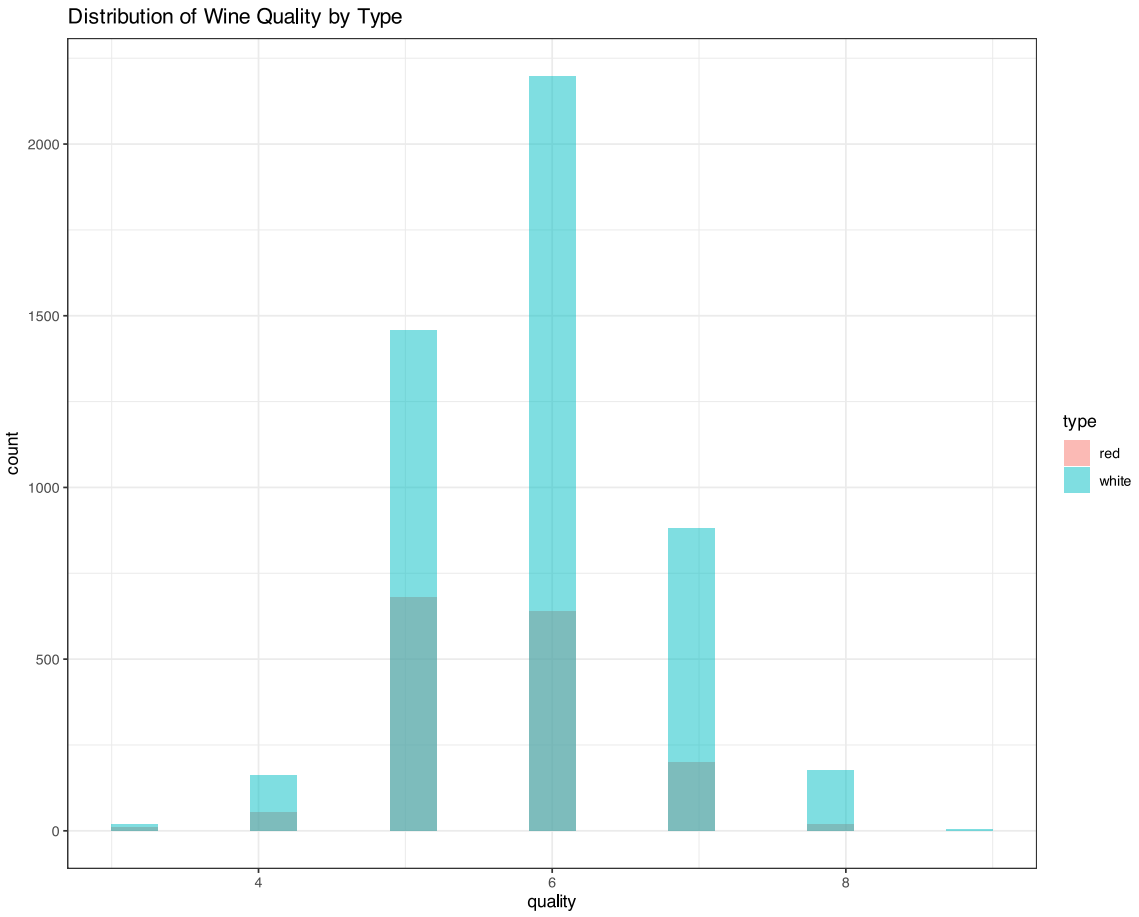



```r
# Quality VS Type
predictors <- wine %>%
  select(-quality, -type) %>%
  select(where(is.numeric)) %>%
  colnames()

wine_long <- wine %>%
  pivot_longer(
    cols = all_of(predictors),
    names_to = "variable",
    values_to = "value"
  )

ggplot(wine_long, aes(x = value, y = quality, color = type)) +
  geom_point(alpha = 0.15) +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~ variable, scales = "free_x") +
  theme_bw(base_size = 14) +
  labs(
```

```
    title = "Quality vs Chemical Predictors by Wine Type",
    x = "Predictor Value",
    y = "Quality"
  )
```




```
ggplot(wine, aes(x = quality, fill = type)) +
  geom_histogram(position = "identity", alpha = 0.5, bins = 20) +
  theme_bw() +
  labs(title = "Distribution of Wine Quality by Type")
```



Model Candidates

```r
# Transformations
wine_tf <- data.frame(
  # response
  quality = wine$quality,
  type = wine$type,

  # transformed predictors
  residual_sugar_t        = log(wine$residual_sugar + 0.1),
  total_sulfur_dioxide_t  = log(wine$total_sulfur_dioxide + 1),
  free_sulfur_dioxide_t   = log(wine$free_sulfur_dioxide + 1),
  chlorides_t             = log(wine$chlorides),
  volatile_acidity_t      = log(wine$volatile_acidity),
  sulphates_t             = log(wine$sulphates),
  citric_acid_t           = log(wine$citric_acid + 0.1),

  # raw predictors kept as-is
  alcohol          = wine$alcohol,
```

```
  fixed_acidity    = wine$fixed_acidity,
  pH               = wine$pH,
  density          = wine$density
)
```

A

baseline / main effects / no interactions

```
mA <- lm(
  quality ~ alcohol + residual_sugar_t + volatile_acidity_t +
    chlorides_t + sulphates_t + citric_acid_t +
    total_sulfur_dioxide_t + fixed_acidity + pH + type,
  data = wine_tf
)
```

B

strong interactions: - volatile_acidity × type - chlorides × type - residual_sugar × type

```
mB <- lm(
  quality ~ alcohol +
    residual_sugar_t * type +
    volatile_acidity_t * type +
    chlorides_t * type +
    total_sulfur_dioxide_t +
    fixed_acidity + pH,
  data = wine_tf
)
```

C

strong + moderate interactions: - total_sulfur_dioxide × type - residual_sugar × type - volatile_acidity × type - chlorides × type

```
mC <- lm(
  quality ~ alcohol +
    residual_sugar_t * type +
    volatile_acidity_t * type +
    chlorides_t * type +
    total_sulfur_dioxide_t * type +
    fixed_acidity + pH,
  data = wine_tf
)
```

D

Strong + moderate + weak

```r
mD <- lm(
  quality ~ alcohol +
    residual_sugar_t * type +
    volatile_acidity_t * type +
    chlorides_t * type +
    total_sulfur_dioxide_t * type +
    citric_acid_t * type +
    sulphates_t * type +
    fixed_acidity + pH,
  data = wine_tf
)
```

E

Replace SO_2 variable to test collinearity sensitivity

```r
mE <- lm(
  quality ~ alcohol +
    residual_sugar_t * type +
    volatile_acidity_t * type +
    chlorides_t * type +
    free_sulfur_dioxide_t +    # 替代 total_SO₂
    fixed_acidity + pH,
  data = wine_tf
)
```

F

Minimal interpretable interaction model

```r
mF <- lm(
  quality ~ alcohol +
    volatile_acidity_t * type +
    residual_sugar_t * type +
    fixed_acidity + pH,
  data = wine_tf
)
```

G

Additive nonlinear expansion model

```r
mG <- lm(
  quality ~ alcohol + I(alcohol^2) +
    volatile_acidity_t + I(volatile_acidity_t^2) +
    residual_sugar_t + I(residual_sugar_t^2) +
    total_sulfur_dioxide_t +
    type,
```

```
  data = wine_tf
)
```

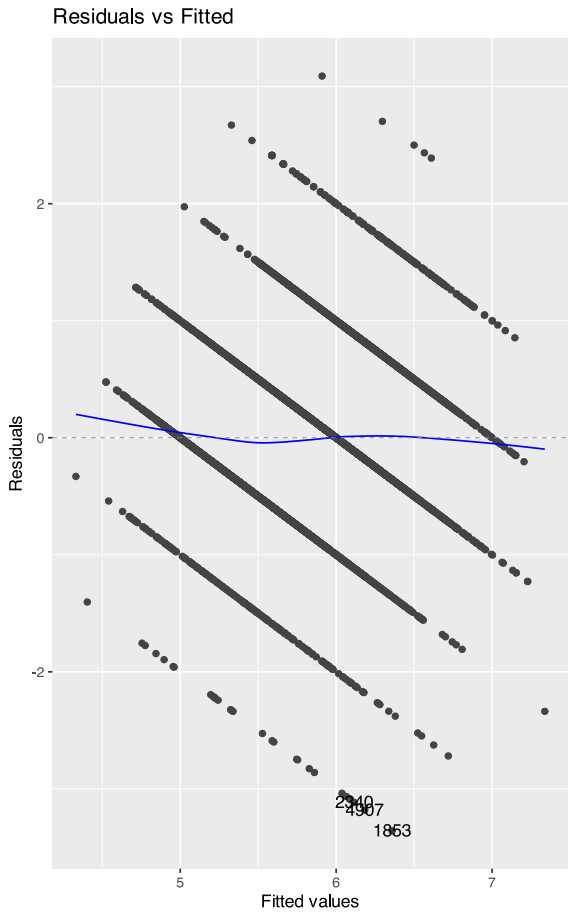
Model Comparison

```
models <- list(mA = mA, mB = mB, mC = mC, mD = mD, mE = mE, mF = mF, mG = mG)
model_summaries <- map_df(
  models,
  ~ glance(.x),
  .id = "model"
)
model_summaries
```

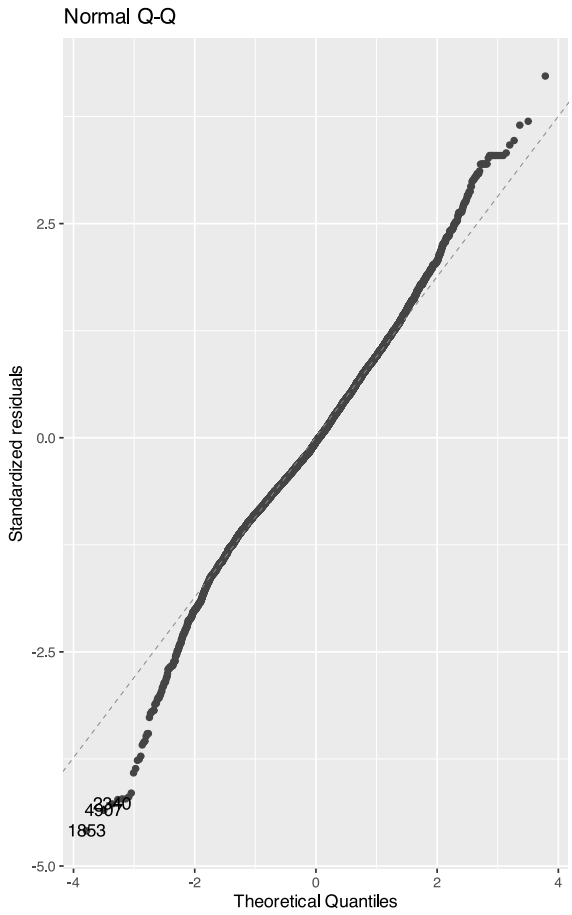
```
# A tibble: 7 × 13
  model r.squared adj.r.squared sigma statistic p.value    df logLik    AIC
  <chr>     <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl>  <dbl>
1 mA        0.292         0.291 0.735      267.       0    10 -7217. 14459.
2 mB        0.286         0.285 0.738      236.       0    11 -7243. 14512.
3 mC        0.289         0.288 0.737      220.       0    12 -7230. 14489.
4 mD        0.298         0.296 0.733      172.       0    16 -7189. 14413.
5 mE        0.296         0.295 0.733      248.       0    11 -7198. 14422.
6 mF        0.284         0.283 0.739      322.       0     8 -7251. 14522.
7 mG        0.286         0.285 0.738      325.       0     8 -7243. 14505.
# ℹ 4 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>, nobs <int>
```

Diagnotics

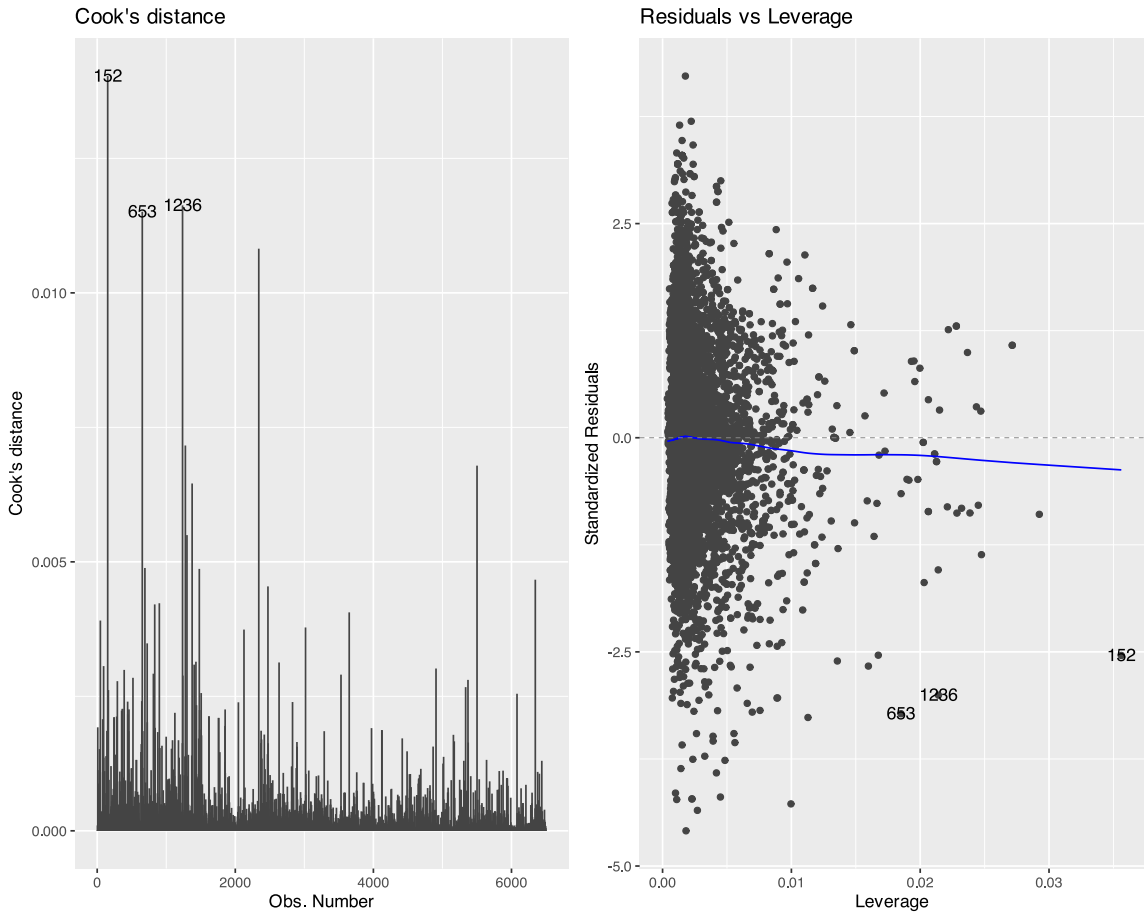
```
autoplot(mD, which = 1)
```



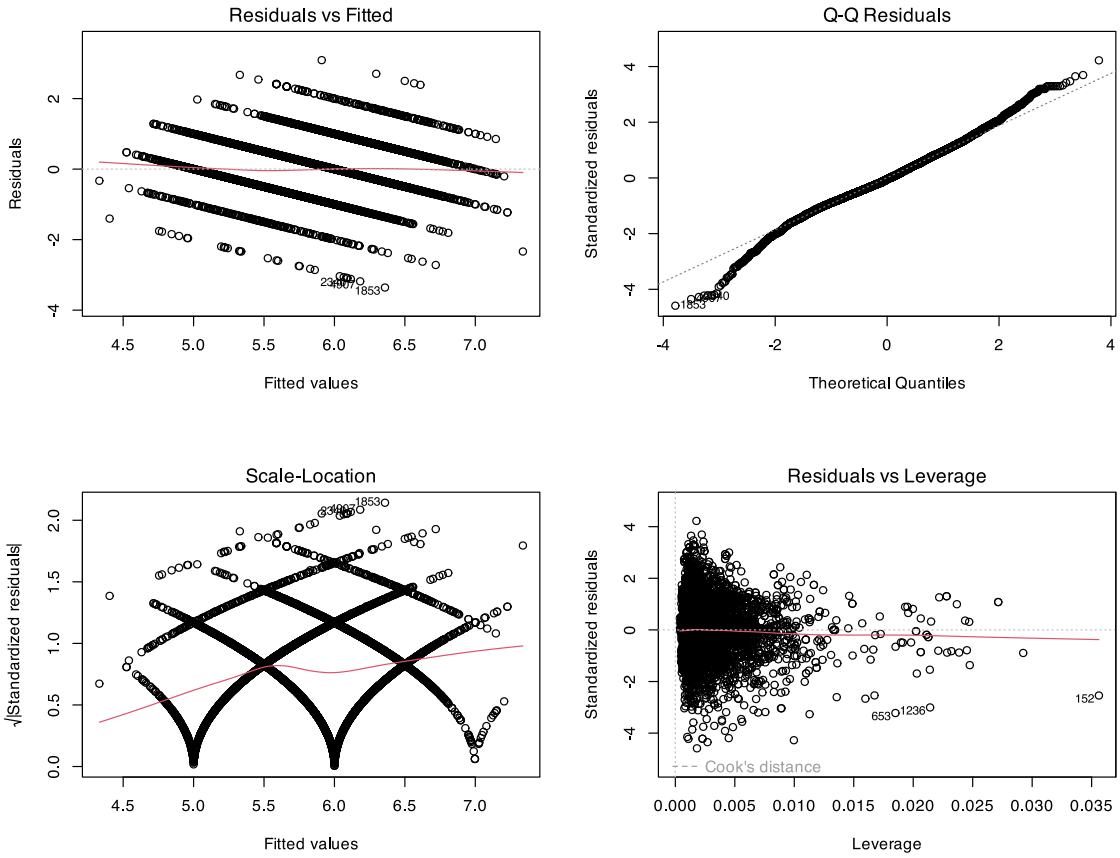

```
autoplot(mD, which = 2)
```



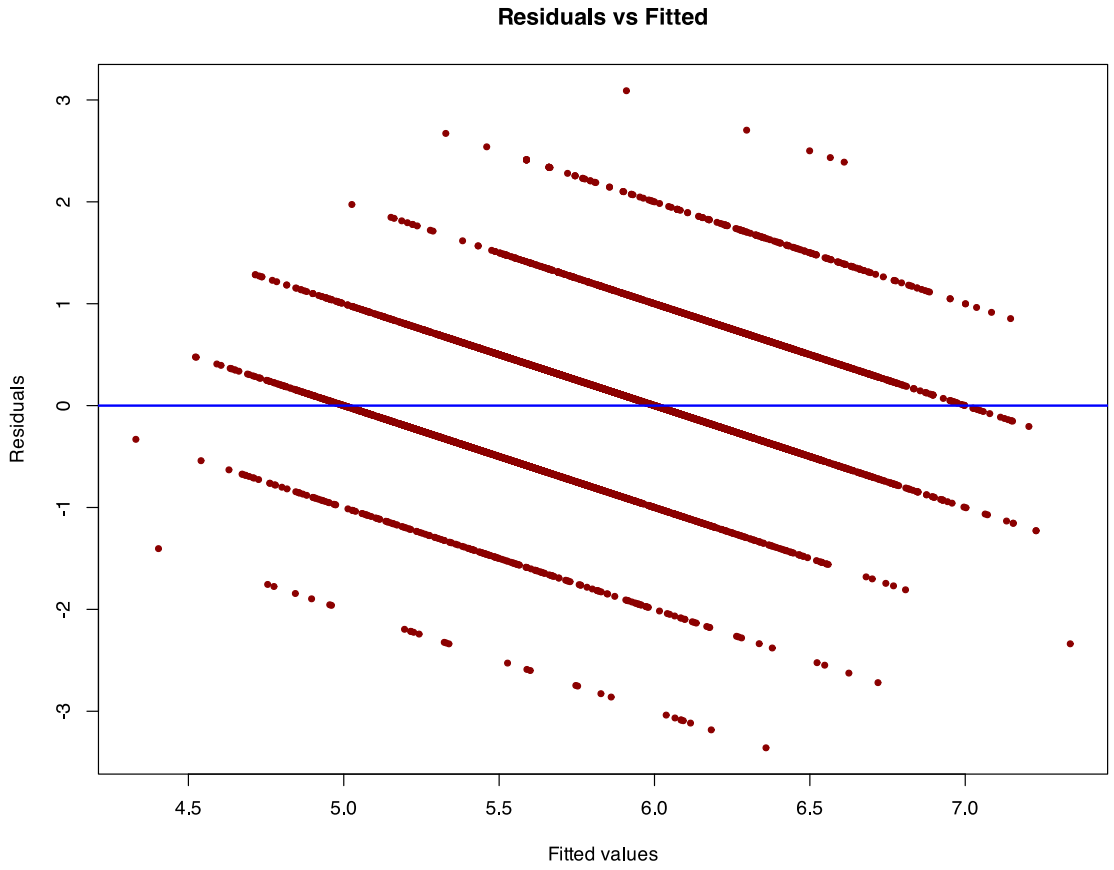

```
autoplot(mD, which = 4:5)
```



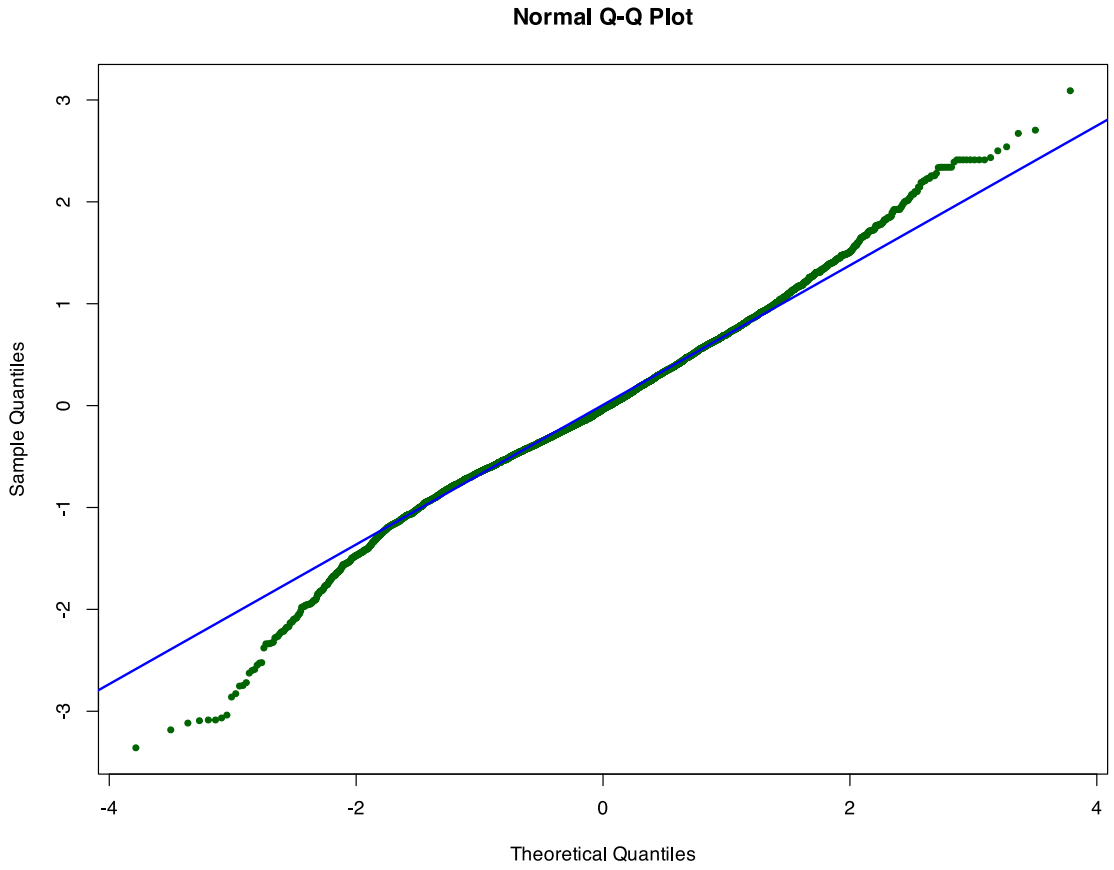

```
# Standard diagnostic panel
par(mfrow = c(2, 2))
plot(mD)
```



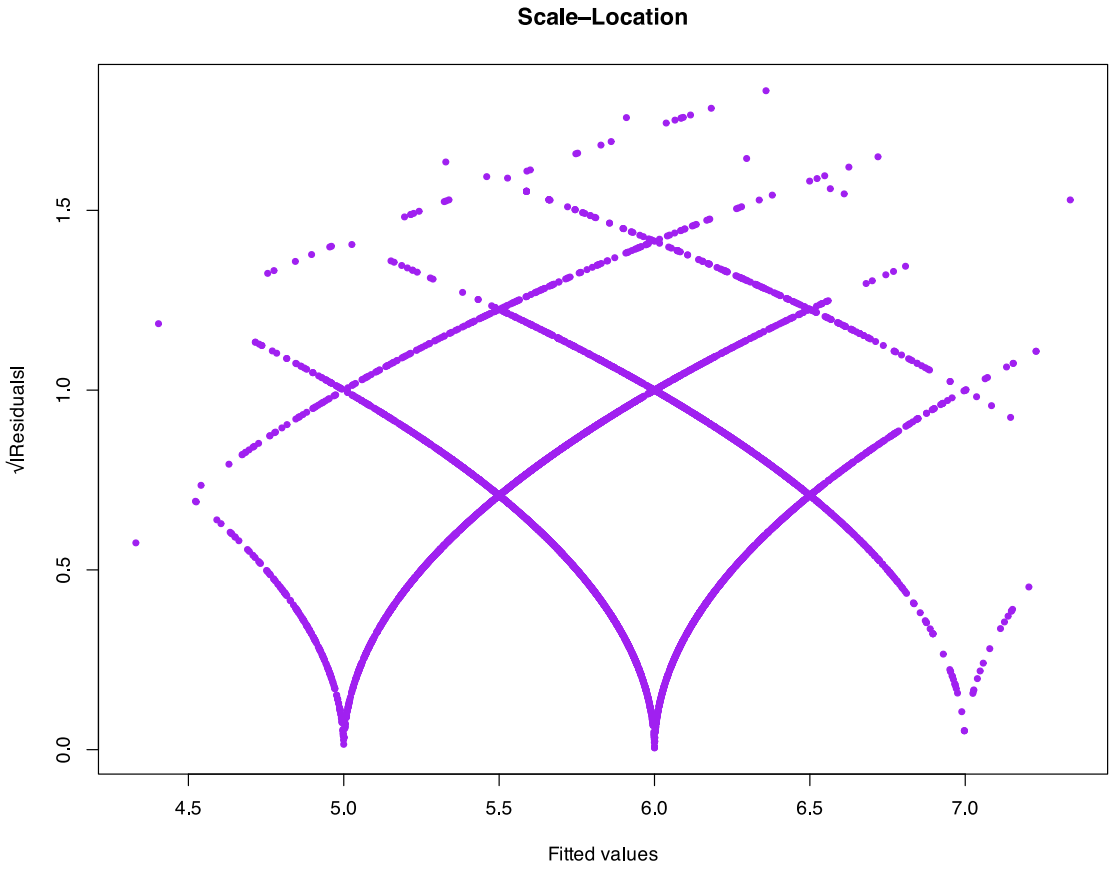

```r
par(mfrow = c(1, 1))

# Residuals vs Fitted
plot(
  fitted(mD), resid(mD),
  pch = 20, col = "darkred",
  xlab = "Fitted values",
  ylab = "Residuals",
  main = "Residuals vs Fitted"
)
abline(h = 0, col = "blue", lwd = 2)
```



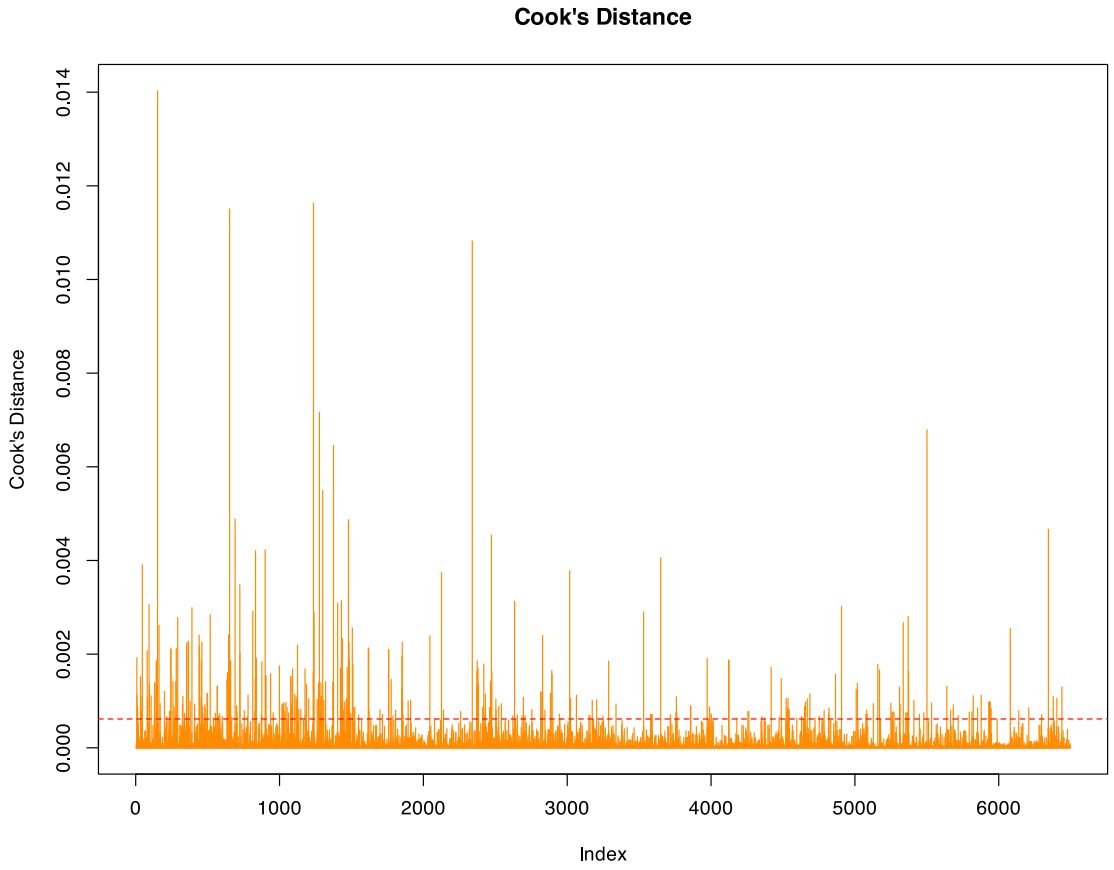

```
# QQ Plot
qqnorm(resid(mD), pch = 20, col = "darkgreen",
       main = "Normal Q-Q Plot")
qqline(resid(mD), col = "blue", lwd = 2)
```



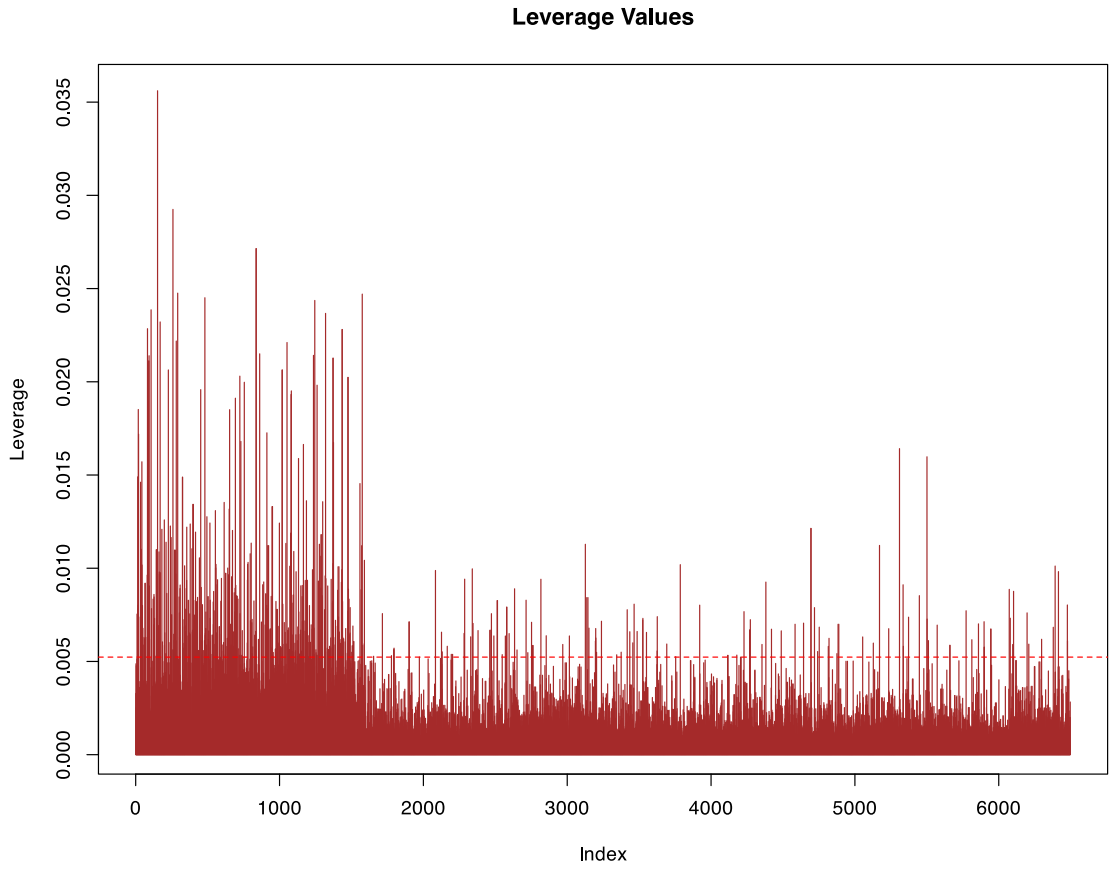

```r
# Scale–Location plot
plot(
  fitted(mD), sqrt(abs(resid(mD))),
  pch = 20, col = "purple",
  xlab = "Fitted values",
  ylab = "√|Residuals|",
  main = "Scale–Location"
)
```



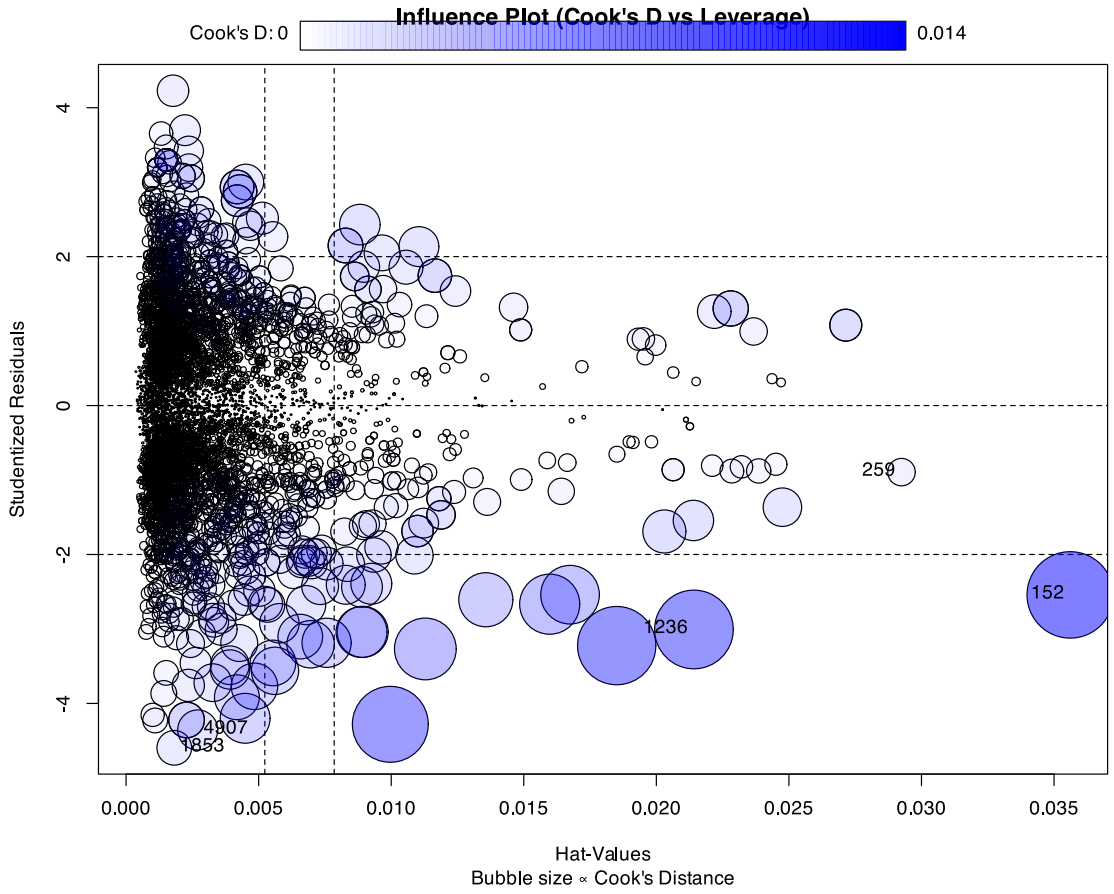

```r
# Cook's Distance
cook <- cooks.distance(mD)
n <- length(cook)
cook_cut <- 4 / n

plot(
  cook, type = "h",
  col = "darkorange",
  main = "Cook's Distance",
  ylab = "Cook's Distance"
)
abline(h = cook_cut, col = "red", lty = 2)
```




```r
# Leverage (hat values)
lev <- hatvalues(mD)
p <- length(coef(mD)) - 1
lev_cut <- 2 * (p + 1) / n

plot(
  lev, type = "h",
  col = "brown",
  main = "Leverage Values",
  ylab = "Leverage"
)
abline(h = lev_cut, col = "red", lty = 2)
```




```r
# Influence plot (optional but strong)
influencePlot(
  mD,
  main = "Influence Plot (Cook's D vs Leverage)",
  sub = "Bubble size ∝ Cook's Distance"
)
```




```
       StudRes         Hat       CookD
152  -2.5425742 0.035610662 0.014030072
259  -0.8940066 0.029246607 0.001416487
1236 -3.0069424 0.021417549 0.011626126
1853 -4.5961419 0.001818063 0.002256267
4907 -4.3563302 0.002702876 0.003017109
```