22-Leveraging-Multiple-Models.R

nandi

2024-12-06

```
options(
  digits = 2,
  scipen = 999,
  warn = -1
)
rm(
  list = ls()
)
library(magrittr)

M_covtype <- read.csv(
  file = "D:\\training.csv"
)
M_covtype$class <- as.numeric(
  factor(M_covtype$class, levels = unique(M_covtype$class))
)
M_covtype$Feature_Sum <- rowSums(
  x = M_covtype[, c("b1", "b2", "b3", "b4", "b5")]
)
M_covtype$Feature_Avg <- rowMeans(
  x = M_covtype[, c("b6", "b7", "b8", "b9")]
)
preProcess_YeoJohnson <- caret::preProcess(
  x = M_covtype %>% dplyr::select(b1, b2, b3, b4, b5),
  method = "YeoJohnson"
)
M_covtype <- predict(
  object = preProcess_YeoJohnson,
  newdata = M_covtype
)
M_covtype <- M_covtype %>%
  dplyr::select(-dplyr::matches("pred_minus_obs_S|pred_minus_obs_H"))
M_covtype$b1_squared <- M_covtype$b1^2
list_glm <- list()
for (j in unique(M_covtype$class)) {
  list_glm[[as.character(j)]] <- glm(
    formula = target ~ .,
    family = "binomial",
    data = M_covtype %>%
      dplyr::mutate(
        target = as.numeric(class == j)
      ) %>%
      dplyr::select(-class)
  )
}
list_forward <- lapply(
  X = list_glm,
  FUN = MASS::stepAIC,
  direction = "forward",
  trace = 0
)
list_predict <- lapply(
  X = list_forward,
  FUN = function(model) predict(
    object = model,
    newdata = M_covtype %>% dplyr::select(-class),
    type = "response"
  )
)
M_predict <- dplyr::bind_cols(
  list_predict
)
if (nrow(M_predict) > 0) {
  M_covtype <- M_covtype %>%
    dplyr::mutate(
      predict = apply(
        X = M_predict,
        MARGIN = 1,
        FUN = which.max
      ),
      predict = factor(
```

```
        x = predict,
        levels = unique(class)
      )
    )
}

head(M_predict)
```

```
## # A tibble: 6 × 4
##        `1`          `2`         `3`       `4`
##      <dbl>        <dbl>       <dbl>     <dbl>
## 1 1.00e+ 0 0.000131    0.00302    2.22e-16
## 2 2.22e-16 0.996       0.00636    2.22e-16
## 3 2.22e-16 0.000300    0.996      2.22e-16
## 4 2.22e-16 0.000247    0.990      2.22e-16
## 5 1   e+ 0 0.000000357 0.00000555 2.22e-16
## 6 2.22e-16 1.00        0.000246   1.54e- 9
```

```
# This model's performance evaluation
# Let's look at the training statistics
M_covtype %>%
  dplyr::select(class, predict) %>%
  table()
```

```
##      predict
## class  1  2  3  4
##     1 54  0  0  0
##     2  0 47  1  0
##     3  0  3 56  0
##     4  0  0  0 37
```

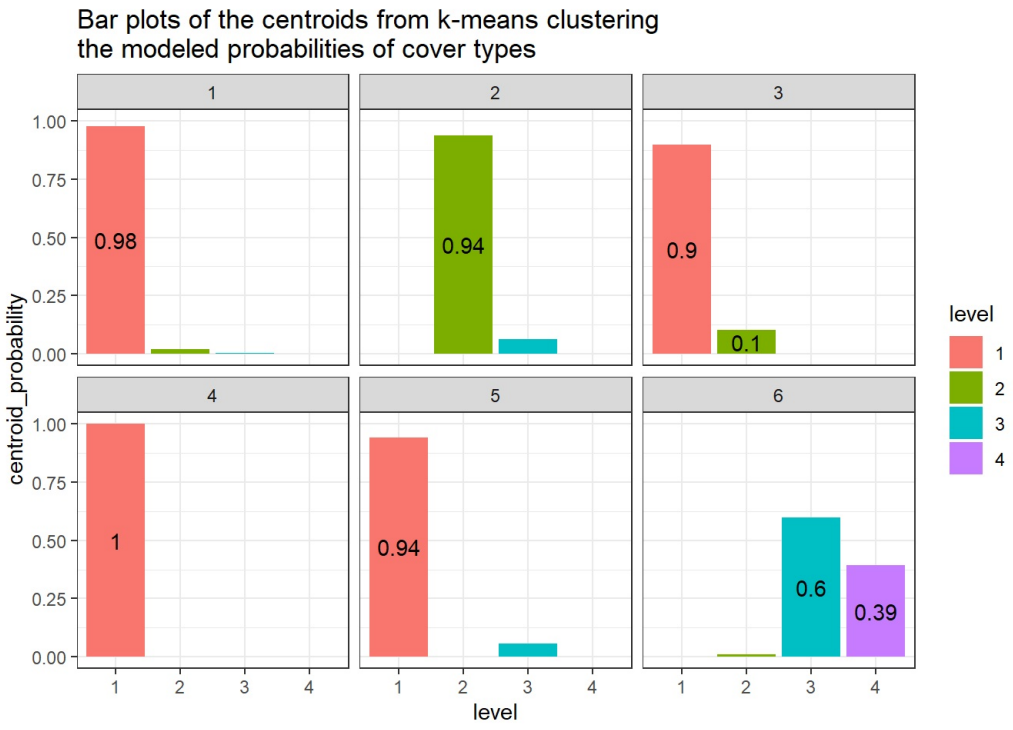
```
caret::confusionMatrix(
  M_covtype %>%
    dplyr::select(class,predict) %>%
    table()
)
```

```
## Confusion Matrix and Statistics
##
##      predict
## class  1  2  3  4
##     1 54  0  0  0
##     2  0 47  1  0
##     3  0  3 56  0
##     4  0  0  0 37
##
## Overall Statistics
##
##                Accuracy : 0.98
##                  95% CI : (0.949, 0.994)
##     No Information Rate : 0.288
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                   Kappa : 0.973
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4
## Sensitivity             1.000    0.940    0.982    1.000
## Specificity             1.000    0.993    0.979    1.000
## Pos Pred Value          1.000    0.979    0.949    1.000
## Neg Pred Value          1.000    0.980    0.993    1.000
## Prevalence              0.273    0.253    0.288    0.187
## Detection Rate          0.273    0.237    0.283    0.187
## Detection Prevalence    0.273    0.242    0.298    0.187
## Balanced Accuracy       1.000    0.967    0.981    1.000
```

```
M_predict <- M_predict / rowSums(M_predict)
M_centers <- rbind(
  diag(ncol(M_predict)),
  rep(1 / ncol(M_predict), ncol(M_predict)),
  prop.table(table(M_covtype$class))
) %>%
  as.data.frame()
M_centers <- M_centers[!duplicated(M_centers), ]
set.seed(123)
kmeans_predict <- kmeans(
  x = M_predict,
  centers = nrow(M_centers),
)
prop.table(table(kmeans_predict$cluster))
```

```
##
##      1      2      3      4      5      6
## 0.0202 0.2525 0.0051 0.2374 0.0101 0.4747
```

```
library(ggplot2)
kmeans_predict$centers %>%
  as.data.frame() %>%
  dplyr::mutate(
    cluster = 1:nrow(kmeans_predict$centers)
  ) %>%
  tidyr::gather(
    key = "level",
    value = "centroid_probability",
    -cluster
  ) %>%
  ggplot() +
  aes(
    x = level,
    y = centroid_probability,
    fill = level,
    label = ifelse(centroid_probability > 0.1,round(centroid_probability,2),"")
  ) +
  geom_col() +
  geom_text(position = position_stack(0.5)) +
  facet_wrap(~cluster) +
  theme_bw() +
  labs(
    title = "Bar plots of the centroids from k-means clustering\nthe modeled probabilities of cover types"
  )
```



```
M_house_prices <- readr::read_csv(
  file = "D:\\housing.csv",
  name_repair = janitor::make_clean_names
) %>%
  dplyr::filter(!is.na(total_bedrooms)) %>%
  dplyr::mutate(
    train_test = dplyr::if_else(
      condition = runif(dplyr::n()) > 2/3,
      true = "Train",
      false = dplyr::if_else(
        condition = runif(dplyr::n()) > 1/2,
        true = "Validation",
        false = "Test"
      )
    )
  ) %>%
  as.data.frame()
```

```
## Rows: 20640 Columns: 10
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ocean_proximity
## dbl (9): longitude, latitude, housing_median_age, total_rooms, total_bedroom...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
v_house_prices <- c(
  "housing_median_age", "total_rooms", "total_bedrooms", "population",
  "households", "median_income", "median_house_value"
)

M_house_prices[, v_house_prices] <- scale(
  x = M_house_prices[, v_house_prices]
)
M_loans <- readr::read_csv(
  file = "D:\\loan_data.csv",
  col_types = "cnnnnnn"
) %>%
  dplyr::mutate(
    train_test = dplyr::if_else(
      condition = runif(dplyr::n()) > 2/3,
      true = "Train",
      false = dplyr::if_else(
        condition = runif(dplyr::n()) > 1/2,
        true = "Validation",
        false = "Test"
      )
    )
  ) %>%
  dplyr::mutate(
    loan_status = factor(loan_status)
  ) %>%
  as.data.frame()

v_loans <- c(
  "person_education_2_moderate", "person_education_3_high",
  "person_home_ownership_3_high", "loan_intent_2_moderate", "loan_intent_3_high"
)

M_loans[, v_loans] <- scale(
  x = M_loans[, v_loans]
)

M <- M_house_prices %>%
  dplyr::select(dplyr::any_of(c(v_house_prices, "train_test", "median_house_value"))) %>%
  dplyr::mutate(
    target = median_house_value - median(median_house_value, na.rm = TRUE),
    sign_target = as.numeric(
      x = sign(target) > 0
    ),
    absolute_value_target = abs(target)
  )

glm_sign <- glm(
  formula = as.formula(
    object = paste0(
      "sign_target ~ ",
      paste(v_house_prices, collapse = " + ")
    )
  ),
  data = M %>%
    dplyr::filter(
      train_test == "Train"
    ),
  family = "binomial"
) %>%
  MASS::stepAIC(
    direction = "forward",
    trace = 0
  )

glm_sign
```

```
##
## Call:  glm(formula = sign_target ~ housing_median_age + total_rooms +
##     total_bedrooms + population + households + median_income +
##     median_house_value, family = "binomial", data = M %>% dplyr::filter(train_test ==
##     "Train"))
##
## Coefficients:
##        (Intercept)  housing_median_age         total_rooms      total_bedrooms
##            1621.05                5.96              -29.81             -149.24
##         population          households       median_income  median_house_value
##             -38.09              203.76              -13.18             6900.08
##
## Degrees of Freedom: 6705 Total (i.e. Null);  6698 Residual
## Null Deviance:       9300
## Residual Deviance: 0.00072   AIC: 16
```

```
lm_absolute_value <- lm(
  formula = as.formula(
    object = paste0(
      "absolute_value_target ~ ",
      paste(v_house_prices, collapse = " + ")
    )
  ),
  data = M %>%
    dplyr::filter(
      train_test == "Train"
    )
) %>%
  MASS::stepAIC(
    direction = "forward",
    trace = 0
  )

lm_absolute_value
```

```
##
## Call:
## lm(formula = absolute_value_target ~ housing_median_age + total_rooms +
##     total_bedrooms + population + households + median_income +
##     median_house_value, data = M %>% dplyr::filter(train_test ==
##     "Train"))
##
## Coefficients:
##        (Intercept)  housing_median_age         total_rooms      total_bedrooms
##             0.7670              0.0210              0.1361              0.0137
##         population          households       median_income  median_house_value
##            -0.0785             -0.1166             -0.0482              0.5129
```

```
M %>%
  dplyr::select(
    train_test, median_house_value, target, sign_target, absolute_value_target
  ) %>%
  dplyr::mutate(
    sign_predict = predict(
      object = glm_sign,
      newdata = M,
      type = "response"
    ),
    absolute_value_predict = predict(
      object = lm_absolute_value,
      newdata = M,
      type = "response"
    ),
    predict = sign(sign_predict - 0.5) * absolute_value_predict,
    sign_residual = sign_predict - sign_target,
    absolute_value_residual = absolute_value_predict - absolute_value_target,
    residual = predict - target
  ) %>%
  dplyr::select(
    train_test, sign_residual, absolute_value_residual, residual
  ) %>%
  tidyr::gather(
    key = "source",
    value = "residual",
    -train_test
  ) %>%
  dplyr::group_by(
    train_test, source
  ) %>%
  dplyr::summarise(
    mse = mean(residual^2, na.rm = TRUE)
  ) %>%
  tidyr::spread(
    key = source,
    value = mse
  )
```

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

```
## # A tibble: 3 × 4
## # Groups:   train_test [3]
##   train_test absolute_value_residual residual sign_residual
##   <chr>                        <dbl>    <dbl>         <dbl>
## 1 Test                         0.221    0.221      1.19e- 3
## 2 Train                        0.224    0.224      4.10e-12
## 3 Validation                   0.224    0.224      1.17e- 3
```

```
kmeans_house_prices <- stats::kmeans(
  x = M_house_prices %>%
    dplyr::filter(train_test == "Train") %>%
    dplyr::select(dplyr::any_of(v_house_prices)) %>%
    as.data.frame(),
  centers = 2
)

M_house_prices <- M_house_prices %>%
  dplyr::mutate(
    partition = apply(
      X = as.matrix(M_house_prices %>%
                      dplyr::select(dplyr::any_of(v_house_prices))),
      MARGIN = 1,
      FUN = function(row) {
        distances <- apply(kmeans_house_prices$centers, 1, function(center) sum((row - center)^2))
        which.min(distances)
      }
    )
  )

kmeans_house_prices
```

```
## K-means clustering with 2 clusters of sizes 730, 5976
##
```

```
## Cluster means:
##   housing_median_age total_rooms total_bedrooms population households
## 1             -0.910        1.95           2.02       1.84       2.00
## 2              0.095       -0.24          -0.25      -0.22      -0.25
##   median_income median_house_value
## 1       0.16404             0.0930
## 2      -0.00062             0.0014
##
## Clustering vector:
##    [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2
##   [38] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
##   [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [149] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
##  [186] 1 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
##  [260] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 1 2 1 2 2 1 1 1 2 2
##  [297] 2 2 2 2 2 2 2 2 2 1 1 1 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2
##  [334] 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [371] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
##  [408] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [445] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
##  [482] 2 2 2 2 2 2 2 2 1 2 1 1 1 2 2 2 2 2 2 2 1 2 1 1 2 2 1 2 2 2 2 1 2 2 2
##  [519] 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 1 2 2
##  [556] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [593] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
##  [630] 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
##  [667] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2
##  [704] 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [741] 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [778] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [815] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
##  [852] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [889] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2
##  [926] 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
##  [963] 2 2 2 2 2 2 2 2 2 1 1 1 2 2 1 1 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 1 2 2 2
## [1000] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [1037] 2 2 2 2 2 2 1 2 2 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1074] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1111] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 1 2 2 1 2 1 2 2
## [1148] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 1 2 2 2 1 2 1 2 1 2 2
## [1185] 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1222] 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2
## [1259] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
## [1296] 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 2
## [1333] 2 2 1 1 2 2 2 2 2 2 2 1 1 2 2 1 2 2 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2
## [1370] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [1407] 2 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
## [1444] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1481] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 2
## [1518] 1 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 1
## [1555] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1592] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1629] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1666] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1703] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2
## [1740] 2 2 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2
## [1777] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2
## [1814] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1
## [1851] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 1 2 1 2 2
## [1888] 2 2 2 2 2 2 1 2 1 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1925] 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 1 2 2 2
## [1962] 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2 1 2 2 2 1 2 2 2 2 2
## [1999] 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2
## [2036] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2
## [2073] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2110] 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [2147] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2184] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 1 2
## [2221] 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2258] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2295] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2
## [2332] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2369] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2
## [2406] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2443] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2
## [2480] 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
## [2517] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1
## [2554] 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2
```

```
## [2591] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
## [2628] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2665] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [2702] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 1
## [2739] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
## [2776] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
## [2813] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2
## [2850] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
## [2887] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1
## [2924] 2 1 1 2 2 2 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 2 2 1 2 2 2 2 2 2 1
## [2961] 2 2 2 2 2 1 2 2 2 2 2 2 1 1 2 1 2 1 2 2 2 2 1 2 1 2 1 2 1 1 2 2 2 2 2 2 2 2 2
## [2998] 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 1 1 1 2 2 2 2 2 1 2 2 1 2 2 1
## [3035] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3072] 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
## [3109] 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 1 1 2
## [3146] 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2
## [3183] 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1
## [3220] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3257] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2
## [3294] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3331] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1
## [3368] 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2
## [3405] 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2
## [3442] 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [3479] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3516] 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 1 2 2 2
## [3553] 1 1 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 1 1 2 2 2 2 2 2
## [3590] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
## [3627] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
## [3664] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2
## [3701] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2
## [3738] 1 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2
## [3775] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3812] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 1 2
## [3849] 1 1 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [3886] 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 1
## [3923] 1 2 1 2 2 1 1 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 1 1 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2
## [3960] 1 1 2 1 1 1 1 2 1 1 1 1 2 1 1 2 2 2 1 1 1 2 1 2 2 2 2 2 1 2 1 2 1 1 1 1 2
## [3997] 2 2 2 1 1 1 2 1 2 2 2 1 2 2 1 2 1 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
## [4034] 1 1 2 2 2 2 1 2 1 1 1 2 1 1 1 1 1 2 2 1 1 1 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2
## [4071] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4108] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 1 2 2
## [4145] 2 2 2 2 1 2 2 1 1 1 1 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2
## [4182] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1 2 2 2
## [4219] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4256] 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 1 2 2 2 2 2 2 2 1
## [4293] 2 2 1 2 1 2 2 2 2 2 1 1 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [4330] 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4367] 2 2 2 1 2 2 2 2 2 2 1 2 2 2 1 2 2 1 1 1 1 2 2 1 2 1 2 2 2 2 2 1 1 2 1 2 2
## [4404] 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2
## [4441] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 2
## [4478] 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2
## [4515] 2 2 1 2 2 1 1 1 2 1 2 2 2 1 2 1 1 1 1 1 1 2 2 2 2 2 1 2 2 1 2 2 1 2 2 2 1
## [4552] 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
## [4589] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4626] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4663] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4700] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [4737] 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4774] 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2
## [4811] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [4848] 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 1 1 1 2 2
## [4885] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 1 2 1 2 1 2 2 2 2 2 1 2 2 2 2
## [4922] 1 2 1 1 2 2 1 2 1 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2
## [4959] 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2
## [4996] 2 2 2 2 2 1 2 2 1 2 2 1 2 2 2 1 2 1 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 1
## [5033] 2 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 1 2 2 1 2 2 2 2 2 1 2 1 2 2 2 2 1 2 1 2 2
## [5070] 2 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5107] 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5144] 2 2 2 2 1 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2
## [5181] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2
## [5218] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5255] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5292] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2
## [5329] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [5366] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5403] 2 2 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5440] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2
## [5477] 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 2 2 2 2 2 2
```

```
## [5514] 1 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2
## [5551] 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
## [5588] 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
## [5625] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5662] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2
## [5699] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5736] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2
## [5773] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 2
## [5810] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 1 2 2 2 2
## [5847] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5884] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2
## [5921] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
## [5958] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5995] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6032] 2 2 1 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 1
## [6069] 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [6106] 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
## [6143] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1
## [6180] 2 2 2 2 2 2 2 2 1 2 1 1 2 1 2 1 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2
## [6217] 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [6254] 2 2 2 2 2 2 2 2 1 2 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2
## [6291] 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6328] 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6365] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 1 2 2
## [6402] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6439] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6476] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6513] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6550] 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6587] 2 2 2 2 2 2 2 2 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 2 1 2 2 2 2 2 2
## [6624] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2
## [6661] 2 2 1 1 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
## [6698] 2 2 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1]  7374 23705
##  (between_SS / total_SS =  29.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
list_lm <- list()
for (j in sort(unique(M_house_prices$partition))) {
  list_lm[[j]] <- lm(
    formula = paste(
      "median_house_value ~ ", paste(v_house_prices, collapse = " + ")
    ),
    data = M_house_prices %>%
      dplyr::filter(train_test == "Train", partition == j) %>%
      dplyr::select(dplyr::any_of(c("median_house_value", v_house_prices)))
  ) %>%
    MASS::stepAIC(
      trace = 0
    )
}

list_lm
```

```
## [[1]]
##
## Call:
## lm(formula = median_house_value ~ housing_median_age + total_rooms +
##     total_bedrooms + population + households + median_income,
##     data = M_house_prices %>% dplyr::filter(train_test == "Train",
##         partition == j) %>% dplyr::select(dplyr::any_of(c("median_house_value",
##         v_house_prices))))
##
## Coefficients:
##        (Intercept)  housing_median_age         total_rooms      total_bedrooms
##              0.161               0.319              -0.246               0.228
##         population          households       median_income
##             -0.208               0.243               0.853
##
##
## [[2]]
##
## Call:
## lm(formula = median_house_value ~ housing_median_age + total_rooms +
##     total_bedrooms + population + households + median_income,
##     data = M_house_prices %>% dplyr::filter(train_test == "Train",
##         partition == j) %>% dplyr::select(dplyr::any_of(c("median_house_value",
##         v_house_prices))))
##
## Coefficients:
##        (Intercept)  housing_median_age         total_rooms      total_bedrooms
##             0.0294              0.1857             -0.6976              0.6917
##         population          households       median_income
##            -0.5576              0.6639              0.8148
```

```
M_house_prices %>%
  dplyr::mutate(
    predict_1 = predict(
      object = list_lm[[1]],
      newdata = M_house_prices
    ),
    predict_2 = predict(
      object = list_lm[[2]],
      newdata = M_house_prices
    ),
    predict = dplyr::if_else(
      condition = partition == 1,
      true = predict_1,
      false = predict_2
    ),
    residual = predict - median_house_value
  ) %>%
  dplyr::group_by(train_test, partition) %>%
  dplyr::summarise(
    mse = mean(residual^2, na.rm = TRUE)
  )
```

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

```
## # A tibble: 6 × 3
## # Groups:   train_test [3]
##   train_test partition   mse
##   <chr>          <int> <dbl>
## 1 Test               1 0.353
## 2 Test               2 0.408
## 3 Train              1 0.306
## 4 Train              2 0.436
## 5 Validation         1 0.369
## 6 Validation         2 0.417
```