Homework05

```
# install.packages("glmnet")
# install.packages("janitor")
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 0.2.0 --
```

```
## v broom        0.7.12     v recipes      0.2.0
## v dials        0.1.1      v rsample      0.1.1
## v dplyr        1.0.8      v tibble       3.1.6
## v ggplot2      3.3.6      v tidyr        1.2.0
## v infer        1.0.0      v tune         0.2.0
## v modeldata    0.1.1      v workflows    0.2.6
## v parsnip      0.2.1      v workflowsets 0.2.1
## v purrr        0.3.4      v yardstick    0.0.9
```

```
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v readr   1.4.0     v forcats 0.5.1
## v stringr 1.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x stringr::fixed()    masks recipes::fixed()
## x dplyr::lag()        masks stats::lag()
## x readr::spec()       masks yardstick::spec()
```

```
library(rlang)
```

```
##
## Attaching package: 'rlang'
```

```
## The following objects are masked from 'package:purrr':
##
##     %@%, as_function, flatten, flatten_chr, flatten_dbl, flatten_int,
##     flatten_lgl, flatten_raw, invoke, splice
```

```
library(knitr)
library(discrim)
```

```
##
```

```
## Attaching package: 'discrim'

## The following object is masked from 'package:dials':
##
##     smoothness
```

```
library(klaR)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-1
```

```
library("janitor")
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
tidymodels_prefer()
```

```
pokemon <- read_csv("data/Pokemon.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   `#` = col_double(),
##   Name = col_character(),
##   `Type 1` = col_character(),
##   `Type 2` = col_character(),
##   Total = col_double(),
##   HP = col_double(),
##   Attack = col_double(),
##   Defense = col_double(),
##   `Sp. Atk` = col_double(),
##   `Sp. Def` = col_double(),
##   Speed = col_double(),
##   Generation = col_double(),
##   Legendary = col_logical()
## )
```

```
head(pokemon)
```

```
## # A tibble: 6 x 13
##      `#` Name    `Type 1` `Type 2` Total    HP Attack Defense `Sp. Atk` `Sp. Def`
##   <dbl> <chr>   <chr>    <chr>    <dbl> <dbl>  <dbl>   <dbl>     <dbl>     <dbl>
## 1     1 Bulbas~ Grass    Poison     318    45     49      49        65        65
## 2     2 Ivysaur Grass    Poison     405    60     62      63        80        80
## 3     3 Venusa~ Grass    Poison     525    80     82      83       100       100
## 4     3 Venusa~ Grass    Poison     625    80    100     123       122       120
## 5     4 Charma~ Fire     <NA>       309    39     52      43        60        50
## 6     5 Charme~ Fire     <NA>       405    58     64      58        80        65
## # ... with 3 more variables: Speed <dbl>, Generation <dbl>, Legendary <lgl>
```

Exercise 1

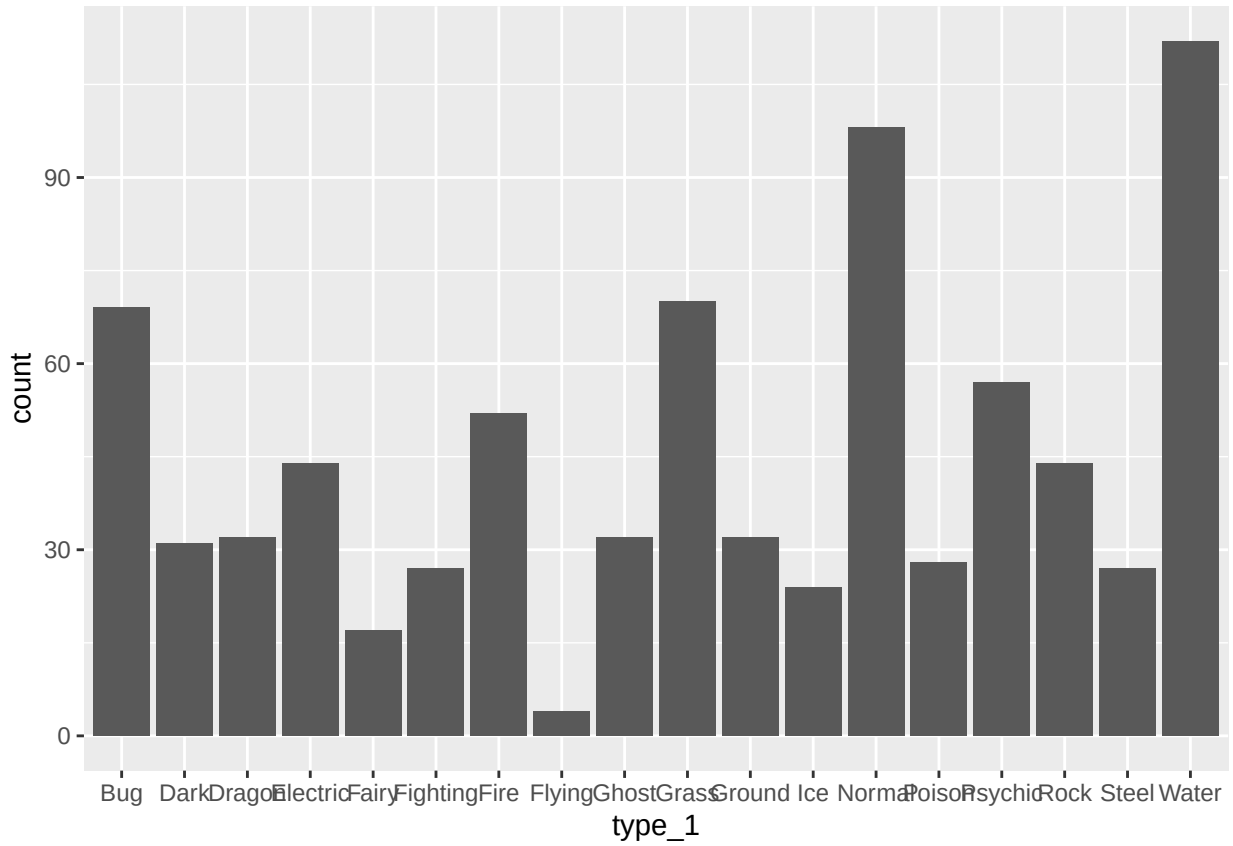
```
pokemon <- pokemon %>%
  clean_names()
head(pokemon)
```

```
## # A tibble: 6 x 13
##   number name       type_1 type_2 total    hp attack defense sp_atk sp_def speed
##    <dbl> <chr>      <chr>  <chr>  <dbl> <dbl>  <dbl>   <dbl>  <dbl>  <dbl> <dbl>
## 1      1 Bulbasaur  Grass  Poison   318    45     49      49     65     65    45
## 2      2 Ivysaur    Grass  Poison   405    60     62      63     80     80    60
## 3      3 Venusaur   Grass  Poison   525    80     82      83    100    100    80
## 4      3 VenusaurM~ Grass  Poison   625    80    100     123    122    120    80
## 5      4 Charmander Fire   <NA>     309    39     52      43     60     50    65
## 6      5 Charmeleon Fire   <NA>     405    58     64      58     80     65    80
## # ... with 2 more variables: generation <dbl>, legendary <lgl>
```

1. What happened to the data? Column names are cleaned. For example, column name has changed from "#" to "number" and all the uppercase letters became lowercase letters. Also, it removes unnecessary spaces and replaces "." to "_".
2. why is clean_names() useful? It cleans dirty data and make it easier to process data.

Exercise 2

```
pokemon %>%
  ggplot(aes(x = type_1)) +
  geom_bar()
```




```
table(pokemon$type_1)
```

```
##
##      Bug     Dark   Dragon Electric    Fairy Fighting     Fire   Flying
##       69       31       32       44       17       27       52        4
##    Ghost    Grass   Ground      Ice   Normal   Poison  Psychic     Rock
##       32       70       32       24       98       28       57       44
##    Steel    Water
##       27      112
```

```
unique(pokemon$type_1)
```

```
##  [1] "Grass"    "Fire"     "Water"    "Bug"      "Normal"   "Poison"
##  [7] "Electric" "Ground"   "Fairy"    "Fighting" "Psychic"  "Rock"
## [13] "Ghost"    "Ice"      "Dragon"   "Dark"     "Steel"    "Flying"
```

```
# filtered_pokemon <- filter(pokemon, type_1 == "Bug" | type_1 == "Fire" | type_1 == "Grass" | type_1 =

df <- pokemon %>%
  filter(type_1 %in% c("Bug", "Fire", "Grass", "Normal", "Water", "Psychic"))
df
```

```
## # A tibble: 458 x 13
##    number name      type_1 type_2 total    hp attack defense sp_atk sp_def speed
##     <dbl> <chr>     <chr>  <chr>  <dbl> <dbl>  <dbl>   <dbl>  <dbl>  <dbl> <dbl>
##  1      1 Bulbasaur Grass  Poison   318    45     49      49     65     65    45
##  2      2 Ivysaur   Grass  Poison   405    60     62      63     80     80    60
##  3      3 Venusaur  Grass  Poison   525    80     82      83    100    100    80
##  4      3 Venusaur~ Grass  Poison   625    80    100     123    122    120    80
```

```
##  5      4 Charmand~ Fire   <NA>     309    39     52      43     60     50    65
##  6      5 Charmele~ Fire   <NA>     405    58     64      58     80     65    80
##  7      6 Charizard Fire   Flying   534    78     84      78    109     85   100
##  8      6 Charizar~ Fire   Dragon   634    78    130     111    130     85   100
##  9      6 Charizar~ Fire   Flying   634    78    104      78    159    115   100
## 10      7 Squirtle  Water  <NA>     314    44     48      65     50     64    43
## # ... with 448 more rows, and 2 more variables: generation <dbl>,
## #   legendary <lgl>
```

```
# Converting "type_1" and "legendary" to factors
df$type_1 <- factor(df$type_1)
df$legendary <- factor(df$legendary)
```

1. How many classes of outcome are there? There are 18 classes.
2. Are there any Pokemon types with very few Pokemon? Which ones? "Flying" has only 4.

Exercise 3

```
set.seed(1014)

df_split <- initial_split(df, prop = 0.70,
                          strata = type_1)
df_train <- training(df_split)
df_test <- testing(df_split)

dim(df_train)
```

```
## [1] 318  13
```

```
dim(df_test)
```

```
## [1] 140  13
```

```
df_folds <- vfold_cv(df_train, v = 5, strata = type_1)
df_folds
```

```
## #  5-fold cross-validation using stratification
## # A tibble: 5 x 2
##   splits           id
##   <list>           <chr>
## 1 <split [252/66]> Fold1
## 2 <split [253/65]> Fold2
## 3 <split [253/65]> Fold3
## 4 <split [256/62]> Fold4
## 5 <split [258/60]> Fold5
```

1. Number of observations for training and test sets? 318 observations for train sets and 140 observations for test sets.
2. Why stratifying the folds be useful? If we do not split the data into different sets the model would be evaluated on the same data it has seen during training. We could avoid overfitting by stratifying sampling.

Exercise 4

```
df_recipe <- recipe(type_1 ~ legendary + generation + sp_atk + attack + speed + defense + hp + sp_def,
                    data = df_train) %>%
  step_dummy(c(legendary, generation)) %>%
  step_normalize(all_predictors())
```

Exercise 5

```
tune_engine <- multinom_reg(penalty = tune(), mixture = tune()) %>%
  set_mode("classification") %>%
  set_engine("glmnet")

pokemon_wf <- workflow() %>%
  add_model(tune_engine) %>%
  add_recipe(df_recipe)

penalty_grid <- grid_regular(penalty(range = c(-5,5)), mixture(range = c(0,1)), levels = 10)
```

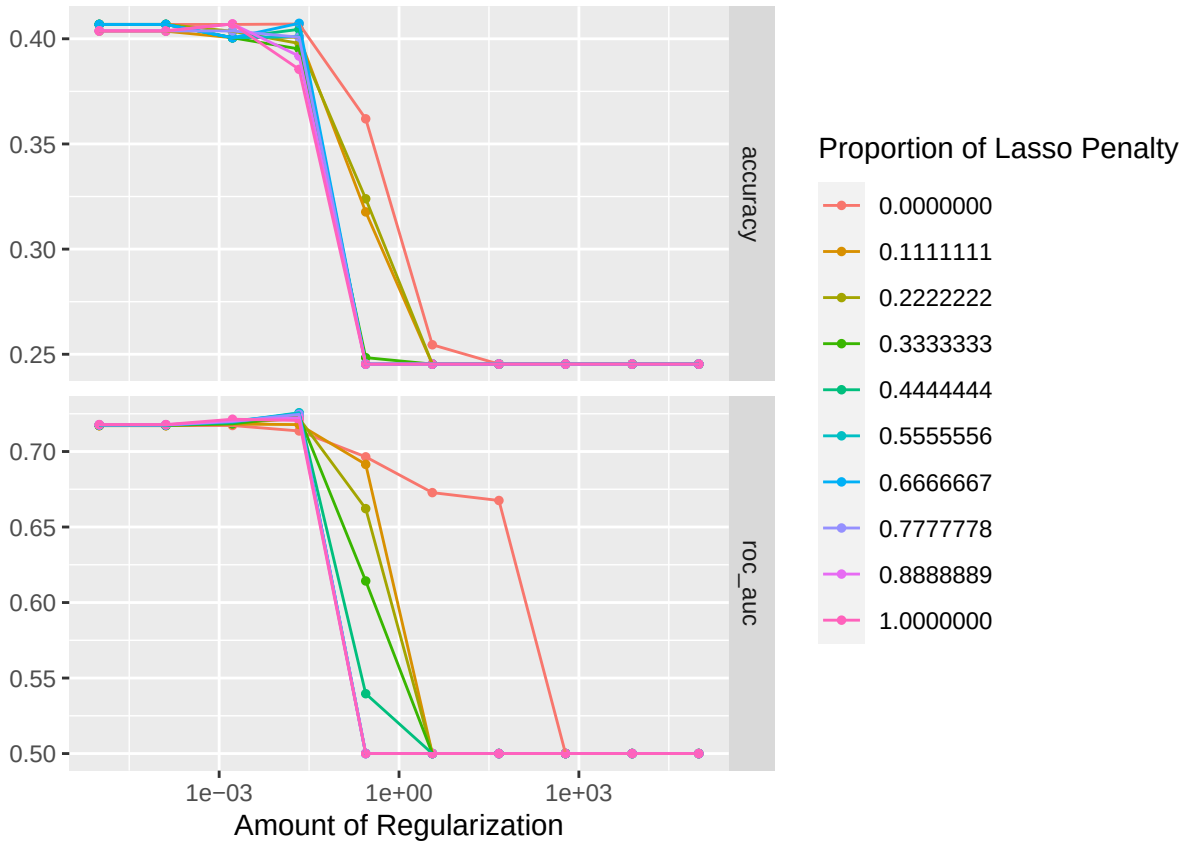
1. How many total models will you be fitting when you fit these models to your folded data? 500 models

Exercise 6

```
tune_res <- tune_grid(
  pokemon_wf,
  resamples = df_folds,
  grid = penalty_grid
)
```

```
## ! Fold1: preprocessor 1/1: The following variables are not factor vectors and wil...
## ! Fold2: preprocessor 1/1: The following variables are not factor vectors and wil...
## ! Fold3: preprocessor 1/1: The following variables are not factor vectors and wil...
## ! Fold4: preprocessor 1/1: The following variables are not factor vectors and wil...
## ! Fold5: preprocessor 1/1: The following variables are not factor vectors and wil...
```

```
autoplot(tune_res)
```



Exercise 7

```
best_penalty <- select_best(tune_res, metrix = "roc_auc")
```

```
## Warning: No value of `metric` was given; metric 'roc_auc' will be used.
```

```
final <- finalize_workflow(pokemon_wf, best_penalty)

final_fit <- fit(final, data = df_train)
```

```
## Warning: The following variables are not factor vectors and will be ignored:
## `generation`
```

```
predict(final_fit, new_data = df_test, type = "class")
```

```
## # A tibble: 140 x 1
##    .pred_class
##    <fct>
##  1 Water
##  2 Water
##  3 Water
##  4 Water
##  5 Normal
##  6 Normal
##  7 Water
##  8 Bug
##  9 Water
```

```
## 10 Water
## # ... with 130 more rows
```

```
test_acc <- augment(final_fit, new_data = df_test) %>%
  accuracy(truth = type_1, estimate = .pred_class)

test_acc
```

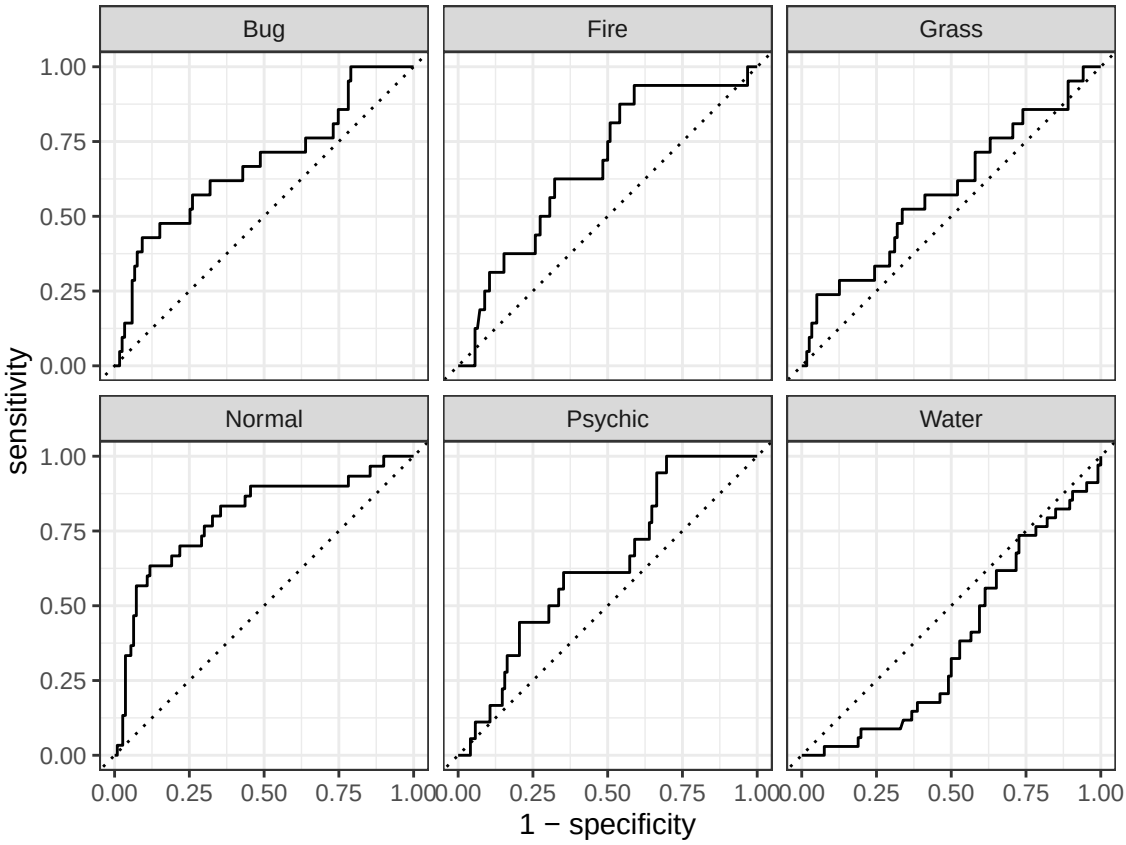
```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.321
```

Exercise 8

```
augment(final_fit, new_data = df_test) %>%
  roc_auc(type_1, .pred_Bug, .pred_Fire, .pred_Grass, .pred_Normal, .pred_Water, .pred_Psychic)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc hand_till      0.616
```

```
augment(final_fit, new_data = df_test) %>%
  roc_curve(type_1, .pred_Bug, .pred_Fire, .pred_Grass, .pred_Normal, .pred_Water, .pred_Psychic) %>%
  autoplot()
```



Exercise 9

```
data <- data.frame(sample(x = c(1, 0), prob = c(0.421, 0.579), size = 801, replace = TRUE))
data
```

```
##     sample.x...c.1..0...prob...c.0.421..0.579...size...801..replace...TRUE.
## 1                                                                           0
## 2                                                                           1
## 3                                                                           0
## 4                                                                           0
## 5                                                                           0
## 6                                                                           0
## 7                                                                           0
## 8                                                                           0
## 9                                                                           1
## 10                                                                          0
## 11                                                                          0
## 12                                                                          0
## 13                                                                          0
## 14                                                                          0
## 15                                                                          1
## 16                                                                          0
## 17                                                                          0
## 18                                                                          1
## 19                                                                          1
## 20                                                                          1
## 21                                                                          0
## 22                                                                          0
## 23                                                                          0
## 24                                                                          1
## 25                                                                          0
## 26                                                                          0
## 27                                                                          0
## 28                                                                          1
## 29                                                                          0
## 30                                                                          1
## 31                                                                          0
## 32                                                                          0
## 33                                                                          0
## 34                                                                          1
## 35                                                                          1
## 36                                                                          1
## 37                                                                          0
## 38                                                                          1
## 39                                                                          0
## 40                                                                          1
## 41                                                                          1
## 42                                                                          1
## 43                                                                          1
## 44                                                                          0
## 45                                                                          0
## 46                                                                          1
## 47                                                                          0
## 48                                                                          0
```

```
## 49                                                                    0
## 50                                                                    0
## 51                                                                    0
## 52                                                                    0
## 53                                                                    1
## 54                                                                    0
## 55                                                                    1
## 56                                                                    1
## 57                                                                    0
## 58                                                                    0
## 59                                                                    0
## 60                                                                    0
## 61                                                                    1
## 62                                                                    1
## 63                                                                    0
## 64                                                                    0
## 65                                                                    1
## 66                                                                    0
## 67                                                                    0
## 68                                                                    0
## 69                                                                    0
## 70                                                                    0
## 71                                                                    0
## 72                                                                    0
## 73                                                                    1
## 74                                                                    0
## 75                                                                    0
## 76                                                                    0
## 77                                                                    1
## 78                                                                    0
## 79                                                                    0
## 80                                                                    0
## 81                                                                    0
## 82                                                                    0
## 83                                                                    1
## 84                                                                    0
## 85                                                                    0
## 86                                                                    0
## 87                                                                    1
## 88                                                                    0
## 89                                                                    1
## 90                                                                    1
## 91                                                                    0
## 92                                                                    1
## 93                                                                    1
## 94                                                                    1
## 95                                                                    0
## 96                                                                    1
## 97                                                                    0
## 98                                                                    0
## 99                                                                    1
## 100                                                                   0
## 101                                                                   1
## 102                                                                   0
```

```
## 103 1
## 104 1
## 105 0
## 106 1
## 107 1
## 108 0
## 109 0
## 110 1
## 111 1
## 112 0
## 113 0
## 114 1
## 115 0
## 116 1
## 117 1
## 118 0
## 119 1
## 120 0
## 121 1
## 122 0
## 123 0
## 124 0
## 125 0
## 126 0
## 127 0
## 128 0
## 129 0
## 130 0
## 131 1
## 132 0
## 133 1
## 134 0
## 135 0
## 136 0
## 137 1
## 138 0
## 139 1
## 140 0
## 141 1
## 142 1
## 143 0
## 144 0
## 145 0
## 146 0
## 147 0
## 148 0
## 149 1
## 150 0
## 151 0
## 152 1
## 153 0
## 154 0
## 155 0
## 156 1
```

```
## 157 1
## 158 0
## 159 0
## 160 0
## 161 0
## 162 1
## 163 0
## 164 1
## 165 1
## 166 0
## 167 1
## 168 0
## 169 0
## 170 0
## 171 0
## 172 0
## 173 0
## 174 0
## 175 0
## 176 1
## 177 0
## 178 1
## 179 0
## 180 0
## 181 0
## 182 0
## 183 0
## 184 0
## 185 0
## 186 0
## 187 0
## 188 0
## 189 0
## 190 1
## 191 1
## 192 1
## 193 1
## 194 1
## 195 0
## 196 0
## 197 1
## 198 0
## 199 0
## 200 0
## 201 0
## 202 0
## 203 0
## 204 1
## 205 0
## 206 1
## 207 0
## 208 0
## 209 0
## 210 1
```

```
## 211 1
## 212 1
## 213 0
## 214 1
## 215 0
## 216 0
## 217 1
## 218 1
## 219 0
## 220 0
## 221 1
## 222 1
## 223 1
## 224 0
## 225 1
## 226 1
## 227 0
## 228 0
## 229 1
## 230 1
## 231 0
## 232 1
## 233 1
## 234 0
## 235 0
## 236 0
## 237 1
## 238 1
## 239 0
## 240 0
## 241 1
## 242 0
## 243 0
## 244 1
## 245 0
## 246 1
## 247 0
## 248 0
## 249 1
## 250 0
## 251 1
## 252 0
## 253 0
## 254 1
## 255 0
## 256 0
## 257 1
## 258 0
## 259 1
## 260 0
## 261 0
## 262 0
## 263 0
## 264 0
```

```
## 265 0
## 266 1
## 267 0
## 268 0
## 269 0
## 270 1
## 271 1
## 272 0
## 273 0
## 274 1
## 275 0
## 276 1
## 277 1
## 278 1
## 279 1
## 280 0
## 281 1
## 282 0
## 283 0
## 284 0
## 285 0
## 286 0
## 287 0
## 288 0
## 289 1
## 290 1
## 291 0
## 292 0
## 293 1
## 294 1
## 295 0
## 296 1
## 297 0
## 298 1
## 299 0
## 300 1
## 301 1
## 302 0
## 303 1
## 304 1
## 305 0
## 306 1
## 307 0
## 308 0
## 309 0
## 310 0
## 311 0
## 312 1
## 313 1
## 314 0
## 315 0
## 316 1
## 317 0
## 318 1
```

```
## 319 0
## 320 1
## 321 1
## 322 1
## 323 0
## 324 1
## 325 1
## 326 0
## 327 1
## 328 0
## 329 1
## 330 0
## 331 1
## 332 0
## 333 1
## 334 1
## 335 0
## 336 0
## 337 0
## 338 1
## 339 1
## 340 0
## 341 0
## 342 0
## 343 1
## 344 0
## 345 1
## 346 0
## 347 0
## 348 0
## 349 1
## 350 1
## 351 1
## 352 0
## 353 0
## 354 0
## 355 0
## 356 1
## 357 0
## 358 0
## 359 0
## 360 1
## 361 0
## 362 1
## 363 0
## 364 0
## 365 0
## 366 1
## 367 0
## 368 0
## 369 0
## 370 0
## 371 0
## 372 1
```

```
## 373 1
## 374 0
## 375 0
## 376 0
## 377 1
## 378 1
## 379 0
## 380 0
## 381 0
## 382 0
## 383 0
## 384 0
## 385 1
## 386 0
## 387 1
## 388 0
## 389 1
## 390 0
## 391 1
## 392 0
## 393 0
## 394 0
## 395 1
## 396 0
## 397 0
## 398 1
## 399 1
## 400 0
## 401 0
## 402 0
## 403 0
## 404 0
## 405 1
## 406 1
## 407 0
## 408 0
## 409 1
## 410 0
## 411 1
## 412 1
## 413 1
## 414 0
## 415 0
## 416 1
## 417 1
## 418 0
## 419 0
## 420 1
## 421 1
## 422 1
## 423 0
## 424 0
## 425 0
## 426 1
```

```
## 427 0
## 428 0
## 429 0
## 430 0
## 431 1
## 432 0
## 433 0
## 434 1
## 435 0
## 436 0
## 437 0
## 438 1
## 439 0
## 440 0
## 441 0
## 442 0
## 443 0
## 444 1
## 445 1
## 446 0
## 447 0
## 448 0
## 449 1
## 450 0
## 451 0
## 452 0
## 453 0
## 454 1
## 455 0
## 456 0
## 457 0
## 458 1
## 459 1
## 460 1
## 461 0
## 462 0
## 463 0
## 464 1
## 465 1
## 466 1
## 467 0
## 468 1
## 469 1
## 470 0
## 471 0
## 472 1
## 473 0
## 474 0
## 475 1
## 476 0
## 477 1
## 478 0
## 479 1
## 480 0
```

```
## 481 1
## 482 0
## 483 0
## 484 1
## 485 1
## 486 0
## 487 0
## 488 0
## 489 0
## 490 0
## 491 0
## 492 1
## 493 1
## 494 0
## 495 1
## 496 1
## 497 0
## 498 0
## 499 1
## 500 0
## 501 0
## 502 1
## 503 0
## 504 1
## 505 0
## 506 0
## 507 1
## 508 0
## 509 0
## 510 0
## 511 1
## 512 0
## 513 0
## 514 0
## 515 1
## 516 0
## 517 0
## 518 1
## 519 0
## 520 0
## 521 0
## 522 0
## 523 0
## 524 0
## 525 0
## 526 1
## 527 0
## 528 0
## 529 0
## 530 0
## 531 1
## 532 0
## 533 0
## 534 0
```

```
## 535 1
## 536 0
## 537 0
## 538 0
## 539 0
## 540 1
## 541 1
## 542 1
## 543 0
## 544 1
## 545 0
## 546 0
## 547 0
## 548 0
## 549 0
## 550 0
## 551 0
## 552 0
## 553 1
## 554 0
## 555 0
## 556 0
## 557 0
## 558 1
## 559 1
## 560 0
## 561 1
## 562 0
## 563 0
## 564 1
## 565 0
## 566 0
## 567 1
## 568 1
## 569 1
## 570 0
## 571 0
## 572 0
## 573 1
## 574 0
## 575 1
## 576 1
## 577 1
## 578 0
## 579 1
## 580 0
## 581 1
## 582 1
## 583 1
## 584 1
## 585 1
## 586 0
## 587 1
## 588 0
```

```
## 589 0
## 590 0
## 591 1
## 592 0
## 593 1
## 594 1
## 595 0
## 596 1
## 597 0
## 598 0
## 599 1
## 600 1
## 601 0
## 602 0
## 603 1
## 604 1
## 605 0
## 606 1
## 607 1
## 608 1
## 609 0
## 610 0
## 611 0
## 612 1
## 613 1
## 614 1
## 615 0
## 616 0
## 617 0
## 618 0
## 619 0
## 620 0
## 621 1
## 622 1
## 623 0
## 624 0
## 625 0
## 626 1
## 627 1
## 628 0
## 629 0
## 630 1
## 631 1
## 632 1
## 633 0
## 634 0
## 635 0
## 636 0
## 637 1
## 638 0
## 639 0
## 640 1
## 641 0
## 642 0
```

```
## 643 0
## 644 0
## 645 0
## 646 0
## 647 0
## 648 0
## 649 0
## 650 1
## 651 1
## 652 1
## 653 1
## 654 1
## 655 1
## 656 1
## 657 1
## 658 1
## 659 1
## 660 0
## 661 1
## 662 1
## 663 0
## 664 1
## 665 1
## 666 1
## 667 0
## 668 1
## 669 1
## 670 0
## 671 0
## 672 1
## 673 1
## 674 1
## 675 1
## 676 0
## 677 0
## 678 1
## 679 0
## 680 0
## 681 0
## 682 0
## 683 1
## 684 0
## 685 1
## 686 0
## 687 0
## 688 1
## 689 0
## 690 1
## 691 1
## 692 0
## 693 0
## 694 1
## 695 1
## 696 0
```

```
## 697 1
## 698 0
## 699 0
## 700 0
## 701 0
## 702 0
## 703 1
## 704 0
## 705 0
## 706 1
## 707 1
## 708 0
## 709 0
## 710 1
## 711 1
## 712 0
## 713 1
## 714 0
## 715 1
## 716 1
## 717 0
## 718 0
## 719 1
## 720 0
## 721 0
## 722 0
## 723 1
## 724 1
## 725 0
## 726 1
## 727 1
## 728 0
## 729 0
## 730 0
## 731 1
## 732 1
## 733 1
## 734 1
## 735 1
## 736 0
## 737 0
## 738 0
## 739 1
## 740 1
## 741 1
## 742 0
## 743 0
## 744 0
## 745 0
## 746 0
## 747 0
## 748 0
## 749 1
## 750 0
```

```
## 751                                                                  0
## 752                                                                  0
## 753                                                                  0
## 754                                                                  0
## 755                                                                  0
## 756                                                                  0
## 757                                                                  1
## 758                                                                  1
## 759                                                                  0
## 760                                                                  1
## 761                                                                  1
## 762                                                                  0
## 763                                                                  1
## 764                                                                  1
## 765                                                                  1
## 766                                                                  1
## 767                                                                  0
## 768                                                                  0
## 769                                                                  0
## 770                                                                  1
## 771                                                                  0
## 772                                                                  1
## 773                                                                  1
## 774                                                                  1
## 775                                                                  0
## 776                                                                  0
## 777                                                                  0
## 778                                                                  1
## 779                                                                  1
## 780                                                                  1
## 781                                                                  1
## 782                                                                  0
## 783                                                                  0
## 784                                                                  0
## 785                                                                  0
## 786                                                                  1
## 787                                                                  1
## 788                                                                  1
## 789                                                                  0
## 790                                                                  1
## 791                                                                  1
## 792                                                                  1
## 793                                                                  0
## 794                                                                  0
## 795                                                                  1
## 796                                                                  1
## 797                                                                  0
## 798                                                                  0
## 799                                                                  1
## 800                                                                  0
## 801                                                                  0
```

```
fg_boot <- bootstraps(data, times = 1e3, apparent = TRUE)
```