데이터마트

데이터마트란 데이터의 한 부분으로서 특정 사용자가 관심을 갖는 데이터들을 담은 비교적 작은 규모의 데이터 웨어하우스이다. 데이터 웨어하우스와 데이터 마트의 구분은 사용자의 기능 및 제공 범위를 기준으로 한다. 데이터마트

## 1. R reshape2의 melt(), cast() 함수

설문조사 데이처를 받았을 때 혹은 분석 과정 중의 데이터셋이 분석가가 하고자 하는 통계 분석에 필요한 데이터 구조로 딱 맞아 떨어지지 않는 경우가 굉장히 많다. 이때 필요한 것이 데이터를 분석 목적, 기법에 맞게 자유자재로 변형하여 재구조화(reshape)하는 일이다. 실제로 reshape 함수는 실무 업무에 많이 사용된다.

# install reshape package  
#install.packages("reshape")  
#install.packages("reshape2")  
library(reshape); library(reshape2)

## Warning: package 'reshape' was built under R version 4.0.3

## Warning: package 'reshape2' was built under R version 4.0.3

##   
## Attaching package: 'reshape2'

## The following objects are masked from 'package:reshape':  
##   
## colsplit, melt, recast

data(airquality) #let's use airquality data  
??airquality

## starting httpd help server ...

## done

head(airquality)

## Ozone Solar.R Wind Temp Month Day  
## 1 41 190 7.4 67 5 1  
## 2 36 118 8.0 72 5 2  
## 3 12 149 12.6 74 5 3  
## 4 18 313 11.5 62 5 4  
## 5 NA NA 14.3 56 5 5  
## 6 28 NA 14.9 66 5 6

names(airquality) <- tolower(names(airquality))

melt()함수의 이용: melt()를 이용하여 여러 변수를 하나의 명목형 변수로 reshape하게되면 R의 장점인 시각화를 하기가 쉬워진다.

T <- melt(airquality, id = c("month", "day"), na.rm = TRUE)  
T

## month day variable value  
## 1 5 1 ozone 41.0  
## 2 5 2 ozone 36.0  
## 3 5 3 ozone 12.0  
## 4 5 4 ozone 18.0  
## 6 5 6 ozone 28.0  
## 7 5 7 ozone 23.0  
## 8 5 8 ozone 19.0  
## 9 5 9 ozone 8.0  
## 11 5 11 ozone 7.0  
## 12 5 12 ozone 16.0  
## 13 5 13 ozone 11.0  
## 14 5 14 ozone 14.0  
## 15 5 15 ozone 18.0  
## 16 5 16 ozone 14.0  
## 17 5 17 ozone 34.0  
## 18 5 18 ozone 6.0  
## 19 5 19 ozone 30.0  
## 20 5 20 ozone 11.0  
## 21 5 21 ozone 1.0  
## 22 5 22 ozone 11.0  
## 23 5 23 ozone 4.0  
## 24 5 24 ozone 32.0  
## 28 5 28 ozone 23.0  
## 29 5 29 ozone 45.0  
## 30 5 30 ozone 115.0  
## 31 5 31 ozone 37.0  
## 38 6 7 ozone 29.0  
## 40 6 9 ozone 71.0  
## 41 6 10 ozone 39.0  
## 44 6 13 ozone 23.0  
## 47 6 16 ozone 21.0  
## 48 6 17 ozone 37.0  
## 49 6 18 ozone 20.0  
## 50 6 19 ozone 12.0  
## 51 6 20 ozone 13.0  
## 62 7 1 ozone 135.0  
## 63 7 2 ozone 49.0  
## 64 7 3 ozone 32.0  
## 66 7 5 ozone 64.0  
## 67 7 6 ozone 40.0  
## 68 7 7 ozone 77.0  
## 69 7 8 ozone 97.0  
## 70 7 9 ozone 97.0  
## 71 7 10 ozone 85.0  
## 73 7 12 ozone 10.0  
## 74 7 13 ozone 27.0  
## 76 7 15 ozone 7.0  
## 77 7 16 ozone 48.0  
## 78 7 17 ozone 35.0  
## 79 7 18 ozone 61.0  
## 80 7 19 ozone 79.0  
## 81 7 20 ozone 63.0  
## 82 7 21 ozone 16.0  
## 85 7 24 ozone 80.0  
## 86 7 25 ozone 108.0  
## 87 7 26 ozone 20.0  
## 88 7 27 ozone 52.0  
## 89 7 28 ozone 82.0  
## 90 7 29 ozone 50.0  
## 91 7 30 ozone 64.0  
## 92 7 31 ozone 59.0  
## 93 8 1 ozone 39.0  
## 94 8 2 ozone 9.0  
## 95 8 3 ozone 16.0  
## 96 8 4 ozone 78.0  
## 97 8 5 ozone 35.0  
## 98 8 6 ozone 66.0  
## 99 8 7 ozone 122.0  
## 100 8 8 ozone 89.0  
## 101 8 9 ozone 110.0  
## 104 8 12 ozone 44.0  
## 105 8 13 ozone 28.0  
## 106 8 14 ozone 65.0  
## 108 8 16 ozone 22.0  
## 109 8 17 ozone 59.0  
## 110 8 18 ozone 23.0  
## 111 8 19 ozone 31.0  
## 112 8 20 ozone 44.0  
## 113 8 21 ozone 21.0  
## 114 8 22 ozone 9.0  
## 116 8 24 ozone 45.0  
## 117 8 25 ozone 168.0  
## 118 8 26 ozone 73.0  
## 120 8 28 ozone 76.0  
## 121 8 29 ozone 118.0  
## 122 8 30 ozone 84.0  
## 123 8 31 ozone 85.0  
## 124 9 1 ozone 96.0  
## 125 9 2 ozone 78.0  
## 126 9 3 ozone 73.0  
## 127 9 4 ozone 91.0  
## 128 9 5 ozone 47.0  
## 129 9 6 ozone 32.0  
## 130 9 7 ozone 20.0  
## 131 9 8 ozone 23.0  
## 132 9 9 ozone 21.0  
## 133 9 10 ozone 24.0  
## 134 9 11 ozone 44.0  
## 135 9 12 ozone 21.0  
## 136 9 13 ozone 28.0  
## 137 9 14 ozone 9.0  
## 138 9 15 ozone 13.0  
## 139 9 16 ozone 46.0  
## 140 9 17 ozone 18.0  
## 141 9 18 ozone 13.0  
## 142 9 19 ozone 24.0  
## 143 9 20 ozone 16.0  
## 144 9 21 ozone 13.0  
## 145 9 22 ozone 23.0  
## 146 9 23 ozone 36.0  
## 147 9 24 ozone 7.0  
## 148 9 25 ozone 14.0  
## 149 9 26 ozone 30.0  
## 151 9 28 ozone 14.0  
## 152 9 29 ozone 18.0  
## 153 9 30 ozone 20.0  
## 154 5 1 solar.r 190.0  
## 155 5 2 solar.r 118.0  
## 156 5 3 solar.r 149.0  
## 157 5 4 solar.r 313.0  
## 160 5 7 solar.r 299.0  
## 161 5 8 solar.r 99.0  
## 162 5 9 solar.r 19.0  
## 163 5 10 solar.r 194.0  
## 165 5 12 solar.r 256.0  
## 166 5 13 solar.r 290.0  
## 167 5 14 solar.r 274.0  
## 168 5 15 solar.r 65.0  
## 169 5 16 solar.r 334.0  
## 170 5 17 solar.r 307.0  
## 171 5 18 solar.r 78.0  
## 172 5 19 solar.r 322.0  
## 173 5 20 solar.r 44.0  
## 174 5 21 solar.r 8.0  
## 175 5 22 solar.r 320.0  
## 176 5 23 solar.r 25.0  
## 177 5 24 solar.r 92.0  
## 178 5 25 solar.r 66.0  
## 179 5 26 solar.r 266.0  
## 181 5 28 solar.r 13.0  
## 182 5 29 solar.r 252.0  
## 183 5 30 solar.r 223.0  
## 184 5 31 solar.r 279.0  
## 185 6 1 solar.r 286.0  
## 186 6 2 solar.r 287.0  
## 187 6 3 solar.r 242.0  
## 188 6 4 solar.r 186.0  
## 189 6 5 solar.r 220.0  
## 190 6 6 solar.r 264.0  
## 191 6 7 solar.r 127.0  
## 192 6 8 solar.r 273.0  
## 193 6 9 solar.r 291.0  
## 194 6 10 solar.r 323.0  
## 195 6 11 solar.r 259.0  
## 196 6 12 solar.r 250.0  
## 197 6 13 solar.r 148.0  
## 198 6 14 solar.r 332.0  
## 199 6 15 solar.r 322.0  
## 200 6 16 solar.r 191.0  
## 201 6 17 solar.r 284.0  
## 202 6 18 solar.r 37.0  
## 203 6 19 solar.r 120.0  
## 204 6 20 solar.r 137.0  
## 205 6 21 solar.r 150.0  
## 206 6 22 solar.r 59.0  
## 207 6 23 solar.r 91.0  
## 208 6 24 solar.r 250.0  
## 209 6 25 solar.r 135.0  
## 210 6 26 solar.r 127.0  
## 211 6 27 solar.r 47.0  
## 212 6 28 solar.r 98.0  
## 213 6 29 solar.r 31.0  
## 214 6 30 solar.r 138.0  
## 215 7 1 solar.r 269.0  
## 216 7 2 solar.r 248.0  
## 217 7 3 solar.r 236.0  
## 218 7 4 solar.r 101.0  
## 219 7 5 solar.r 175.0  
## 220 7 6 solar.r 314.0  
## 221 7 7 solar.r 276.0  
## 222 7 8 solar.r 267.0  
## 223 7 9 solar.r 272.0  
## 224 7 10 solar.r 175.0  
## 225 7 11 solar.r 139.0  
## 226 7 12 solar.r 264.0  
## 227 7 13 solar.r 175.0  
## 228 7 14 solar.r 291.0  
## 229 7 15 solar.r 48.0  
## 230 7 16 solar.r 260.0  
## 231 7 17 solar.r 274.0  
## 232 7 18 solar.r 285.0  
## 233 7 19 solar.r 187.0  
## 234 7 20 solar.r 220.0  
## 235 7 21 solar.r 7.0  
## 236 7 22 solar.r 258.0  
## 237 7 23 solar.r 295.0  
## 238 7 24 solar.r 294.0  
## 239 7 25 solar.r 223.0  
## 240 7 26 solar.r 81.0  
## 241 7 27 solar.r 82.0  
## 242 7 28 solar.r 213.0  
## 243 7 29 solar.r 275.0  
## 244 7 30 solar.r 253.0  
## 245 7 31 solar.r 254.0  
## 246 8 1 solar.r 83.0  
## 247 8 2 solar.r 24.0  
## 248 8 3 solar.r 77.0  
## 252 8 7 solar.r 255.0  
## 253 8 8 solar.r 229.0  
## 254 8 9 solar.r 207.0  
## 255 8 10 solar.r 222.0  
## 256 8 11 solar.r 137.0  
## 257 8 12 solar.r 192.0  
## 258 8 13 solar.r 273.0  
## 259 8 14 solar.r 157.0  
## 260 8 15 solar.r 64.0  
## 261 8 16 solar.r 71.0  
## 262 8 17 solar.r 51.0  
## 263 8 18 solar.r 115.0  
## 264 8 19 solar.r 244.0  
## 265 8 20 solar.r 190.0  
## 266 8 21 solar.r 259.0  
## 267 8 22 solar.r 36.0  
## 268 8 23 solar.r 255.0  
## 269 8 24 solar.r 212.0  
## 270 8 25 solar.r 238.0  
## 271 8 26 solar.r 215.0  
## 272 8 27 solar.r 153.0  
## 273 8 28 solar.r 203.0  
## 274 8 29 solar.r 225.0  
## 275 8 30 solar.r 237.0  
## 276 8 31 solar.r 188.0  
## 277 9 1 solar.r 167.0  
## 278 9 2 solar.r 197.0  
## 279 9 3 solar.r 183.0  
## 280 9 4 solar.r 189.0  
## 281 9 5 solar.r 95.0  
## 282 9 6 solar.r 92.0  
## 283 9 7 solar.r 252.0  
## 284 9 8 solar.r 220.0  
## 285 9 9 solar.r 230.0  
## 286 9 10 solar.r 259.0  
## 287 9 11 solar.r 236.0  
## 288 9 12 solar.r 259.0  
## 289 9 13 solar.r 238.0  
## 290 9 14 solar.r 24.0  
## 291 9 15 solar.r 112.0  
## 292 9 16 solar.r 237.0  
## 293 9 17 solar.r 224.0  
## 294 9 18 solar.r 27.0  
## 295 9 19 solar.r 238.0  
## 296 9 20 solar.r 201.0  
## 297 9 21 solar.r 238.0  
## 298 9 22 solar.r 14.0  
## 299 9 23 solar.r 139.0  
## 300 9 24 solar.r 49.0  
## 301 9 25 solar.r 20.0  
## 302 9 26 solar.r 193.0  
## 303 9 27 solar.r 145.0  
## 304 9 28 solar.r 191.0  
## 305 9 29 solar.r 131.0  
## 306 9 30 solar.r 223.0  
## 307 5 1 wind 7.4  
## 308 5 2 wind 8.0  
## 309 5 3 wind 12.6  
## 310 5 4 wind 11.5  
## 311 5 5 wind 14.3  
## 312 5 6 wind 14.9  
## 313 5 7 wind 8.6  
## 314 5 8 wind 13.8  
## 315 5 9 wind 20.1  
## 316 5 10 wind 8.6  
## 317 5 11 wind 6.9  
## 318 5 12 wind 9.7  
## 319 5 13 wind 9.2  
## 320 5 14 wind 10.9  
## 321 5 15 wind 13.2  
## 322 5 16 wind 11.5  
## 323 5 17 wind 12.0  
## 324 5 18 wind 18.4  
## 325 5 19 wind 11.5  
## 326 5 20 wind 9.7  
## 327 5 21 wind 9.7  
## 328 5 22 wind 16.6  
## 329 5 23 wind 9.7  
## 330 5 24 wind 12.0  
## 331 5 25 wind 16.6  
## 332 5 26 wind 14.9  
## 333 5 27 wind 8.0  
## 334 5 28 wind 12.0  
## 335 5 29 wind 14.9  
## 336 5 30 wind 5.7  
## 337 5 31 wind 7.4  
## 338 6 1 wind 8.6  
## 339 6 2 wind 9.7  
## 340 6 3 wind 16.1  
## 341 6 4 wind 9.2  
## 342 6 5 wind 8.6  
## 343 6 6 wind 14.3  
## 344 6 7 wind 9.7  
## 345 6 8 wind 6.9  
## 346 6 9 wind 13.8  
## 347 6 10 wind 11.5  
## 348 6 11 wind 10.9  
## 349 6 12 wind 9.2  
## 350 6 13 wind 8.0  
## 351 6 14 wind 13.8  
## 352 6 15 wind 11.5  
## 353 6 16 wind 14.9  
## 354 6 17 wind 20.7  
## 355 6 18 wind 9.2  
## 356 6 19 wind 11.5  
## 357 6 20 wind 10.3  
## 358 6 21 wind 6.3  
## 359 6 22 wind 1.7  
## 360 6 23 wind 4.6  
## 361 6 24 wind 6.3  
## 362 6 25 wind 8.0  
## 363 6 26 wind 8.0  
## 364 6 27 wind 10.3  
## 365 6 28 wind 11.5  
## 366 6 29 wind 14.9  
## 367 6 30 wind 8.0  
## 368 7 1 wind 4.1  
## 369 7 2 wind 9.2  
## 370 7 3 wind 9.2  
## 371 7 4 wind 10.9  
## 372 7 5 wind 4.6  
## 373 7 6 wind 10.9  
## 374 7 7 wind 5.1  
## 375 7 8 wind 6.3  
## 376 7 9 wind 5.7  
## 377 7 10 wind 7.4  
## 378 7 11 wind 8.6  
## 379 7 12 wind 14.3  
## 380 7 13 wind 14.9  
## 381 7 14 wind 14.9  
## 382 7 15 wind 14.3  
## 383 7 16 wind 6.9  
## 384 7 17 wind 10.3  
## 385 7 18 wind 6.3  
## 386 7 19 wind 5.1  
## 387 7 20 wind 11.5  
## 388 7 21 wind 6.9  
## 389 7 22 wind 9.7  
## 390 7 23 wind 11.5  
## 391 7 24 wind 8.6  
## 392 7 25 wind 8.0  
## 393 7 26 wind 8.6  
## 394 7 27 wind 12.0  
## 395 7 28 wind 7.4  
## 396 7 29 wind 7.4  
## 397 7 30 wind 7.4  
## 398 7 31 wind 9.2  
## 399 8 1 wind 6.9  
## 400 8 2 wind 13.8  
## 401 8 3 wind 7.4  
## 402 8 4 wind 6.9  
## 403 8 5 wind 7.4  
## 404 8 6 wind 4.6  
## 405 8 7 wind 4.0  
## 406 8 8 wind 10.3  
## 407 8 9 wind 8.0  
## 408 8 10 wind 8.6  
## 409 8 11 wind 11.5  
## 410 8 12 wind 11.5  
## 411 8 13 wind 11.5  
## 412 8 14 wind 9.7  
## 413 8 15 wind 11.5  
## 414 8 16 wind 10.3  
## 415 8 17 wind 6.3  
## 416 8 18 wind 7.4  
## 417 8 19 wind 10.9  
## 418 8 20 wind 10.3  
## 419 8 21 wind 15.5  
## 420 8 22 wind 14.3  
## 421 8 23 wind 12.6  
## 422 8 24 wind 9.7  
## 423 8 25 wind 3.4  
## 424 8 26 wind 8.0  
## 425 8 27 wind 5.7  
## 426 8 28 wind 9.7  
## 427 8 29 wind 2.3  
## 428 8 30 wind 6.3  
## 429 8 31 wind 6.3  
## 430 9 1 wind 6.9  
## 431 9 2 wind 5.1  
## 432 9 3 wind 2.8  
## 433 9 4 wind 4.6  
## 434 9 5 wind 7.4  
## 435 9 6 wind 15.5  
## 436 9 7 wind 10.9  
## 437 9 8 wind 10.3  
## 438 9 9 wind 10.9  
## 439 9 10 wind 9.7  
## 440 9 11 wind 14.9  
## 441 9 12 wind 15.5  
## 442 9 13 wind 6.3  
## 443 9 14 wind 10.9  
## 444 9 15 wind 11.5  
## 445 9 16 wind 6.9  
## 446 9 17 wind 13.8  
## 447 9 18 wind 10.3  
## 448 9 19 wind 10.3  
## 449 9 20 wind 8.0  
## 450 9 21 wind 12.6  
## 451 9 22 wind 9.2  
## 452 9 23 wind 10.3  
## 453 9 24 wind 10.3  
## 454 9 25 wind 16.6  
## 455 9 26 wind 6.9  
## 456 9 27 wind 13.2  
## 457 9 28 wind 14.3  
## 458 9 29 wind 8.0  
## 459 9 30 wind 11.5  
## 460 5 1 temp 67.0  
## 461 5 2 temp 72.0  
## 462 5 3 temp 74.0  
## 463 5 4 temp 62.0  
## 464 5 5 temp 56.0  
## 465 5 6 temp 66.0  
## 466 5 7 temp 65.0  
## 467 5 8 temp 59.0  
## 468 5 9 temp 61.0  
## 469 5 10 temp 69.0  
## 470 5 11 temp 74.0  
## 471 5 12 temp 69.0  
## 472 5 13 temp 66.0  
## 473 5 14 temp 68.0  
## 474 5 15 temp 58.0  
## 475 5 16 temp 64.0  
## 476 5 17 temp 66.0  
## 477 5 18 temp 57.0  
## 478 5 19 temp 68.0  
## 479 5 20 temp 62.0  
## 480 5 21 temp 59.0  
## 481 5 22 temp 73.0  
## 482 5 23 temp 61.0  
## 483 5 24 temp 61.0  
## 484 5 25 temp 57.0  
## 485 5 26 temp 58.0  
## 486 5 27 temp 57.0  
## 487 5 28 temp 67.0  
## 488 5 29 temp 81.0  
## 489 5 30 temp 79.0  
## 490 5 31 temp 76.0  
## 491 6 1 temp 78.0  
## 492 6 2 temp 74.0  
## 493 6 3 temp 67.0  
## 494 6 4 temp 84.0  
## 495 6 5 temp 85.0  
## 496 6 6 temp 79.0  
## 497 6 7 temp 82.0  
## 498 6 8 temp 87.0  
## 499 6 9 temp 90.0  
## 500 6 10 temp 87.0  
## 501 6 11 temp 93.0  
## 502 6 12 temp 92.0  
## 503 6 13 temp 82.0  
## 504 6 14 temp 80.0  
## 505 6 15 temp 79.0  
## 506 6 16 temp 77.0  
## 507 6 17 temp 72.0  
## 508 6 18 temp 65.0  
## 509 6 19 temp 73.0  
## 510 6 20 temp 76.0  
## 511 6 21 temp 77.0  
## 512 6 22 temp 76.0  
## 513 6 23 temp 76.0  
## 514 6 24 temp 76.0  
## 515 6 25 temp 75.0  
## 516 6 26 temp 78.0  
## 517 6 27 temp 73.0  
## 518 6 28 temp 80.0  
## 519 6 29 temp 77.0  
## 520 6 30 temp 83.0  
## 521 7 1 temp 84.0  
## 522 7 2 temp 85.0  
## 523 7 3 temp 81.0  
## 524 7 4 temp 84.0  
## 525 7 5 temp 83.0  
## 526 7 6 temp 83.0  
## 527 7 7 temp 88.0  
## 528 7 8 temp 92.0  
## 529 7 9 temp 92.0  
## 530 7 10 temp 89.0  
## 531 7 11 temp 82.0  
## 532 7 12 temp 73.0  
## 533 7 13 temp 81.0  
## 534 7 14 temp 91.0  
## 535 7 15 temp 80.0  
## 536 7 16 temp 81.0  
## 537 7 17 temp 82.0  
## 538 7 18 temp 84.0  
## 539 7 19 temp 87.0  
## 540 7 20 temp 85.0  
## 541 7 21 temp 74.0  
## 542 7 22 temp 81.0  
## 543 7 23 temp 82.0  
## 544 7 24 temp 86.0  
## 545 7 25 temp 85.0  
## 546 7 26 temp 82.0  
## 547 7 27 temp 86.0  
## 548 7 28 temp 88.0  
## 549 7 29 temp 86.0  
## 550 7 30 temp 83.0  
## 551 7 31 temp 81.0  
## 552 8 1 temp 81.0  
## 553 8 2 temp 81.0  
## 554 8 3 temp 82.0  
## 555 8 4 temp 86.0  
## 556 8 5 temp 85.0  
## 557 8 6 temp 87.0  
## 558 8 7 temp 89.0  
## 559 8 8 temp 90.0  
## 560 8 9 temp 90.0  
## 561 8 10 temp 92.0  
## 562 8 11 temp 86.0  
## 563 8 12 temp 86.0  
## 564 8 13 temp 82.0  
## 565 8 14 temp 80.0  
## 566 8 15 temp 79.0  
## 567 8 16 temp 77.0  
## 568 8 17 temp 79.0  
## 569 8 18 temp 76.0  
## 570 8 19 temp 78.0  
## 571 8 20 temp 78.0  
## 572 8 21 temp 77.0  
## 573 8 22 temp 72.0  
## 574 8 23 temp 75.0  
## 575 8 24 temp 79.0  
## 576 8 25 temp 81.0  
## 577 8 26 temp 86.0  
## 578 8 27 temp 88.0  
## 579 8 28 temp 97.0  
## 580 8 29 temp 94.0  
## 581 8 30 temp 96.0  
## 582 8 31 temp 94.0  
## 583 9 1 temp 91.0  
## 584 9 2 temp 92.0  
## 585 9 3 temp 93.0  
## 586 9 4 temp 93.0  
## 587 9 5 temp 87.0  
## 588 9 6 temp 84.0  
## 589 9 7 temp 80.0  
## 590 9 8 temp 78.0  
## 591 9 9 temp 75.0  
## 592 9 10 temp 73.0  
## 593 9 11 temp 81.0  
## 594 9 12 temp 76.0  
## 595 9 13 temp 77.0  
## 596 9 14 temp 71.0  
## 597 9 15 temp 71.0  
## 598 9 16 temp 78.0  
## 599 9 17 temp 67.0  
## 600 9 18 temp 76.0  
## 601 9 19 temp 68.0  
## 602 9 20 temp 82.0  
## 603 9 21 temp 64.0  
## 604 9 22 temp 71.0  
## 605 9 23 temp 81.0  
## 606 9 24 temp 69.0  
## 607 9 25 temp 63.0  
## 608 9 26 temp 70.0  
## 609 9 27 temp 77.0  
## 610 9 28 temp 75.0  
## 611 9 29 temp 76.0  
## 612 9 30 temp 68.0

cast() 함수의 이용

cast(T, day~month~variable)

## , , variable = ozone  
##   
## month  
## day 5 6 7 8 9  
## 1 41 NA 135 39 96  
## 2 36 NA 49 9 78  
## 3 12 NA 32 16 73  
## 4 18 NA NA 78 91  
## 5 NA NA 64 35 47  
## 6 28 NA 40 66 32  
## 7 23 29 77 122 20  
## 8 19 NA 97 89 23  
## 9 8 71 97 110 21  
## 10 NA 39 85 NA 24  
## 11 7 NA NA NA 44  
## 12 16 NA 10 44 21  
## 13 11 23 27 28 28  
## 14 14 NA NA 65 9  
## 15 18 NA 7 NA 13  
## 16 14 21 48 22 46  
## 17 34 37 35 59 18  
## 18 6 20 61 23 13  
## 19 30 12 79 31 24  
## 20 11 13 63 44 16  
## 21 1 NA 16 21 13  
## 22 11 NA NA 9 23  
## 23 4 NA NA NA 36  
## 24 32 NA 80 45 7  
## 25 NA NA 108 168 14  
## 26 NA NA 20 73 30  
## 27 NA NA 52 NA NA  
## 28 23 NA 82 76 14  
## 29 45 NA 50 118 18  
## 30 115 NA 64 84 20  
## 31 37 NA 59 85 NA  
##   
## , , variable = solar.r  
##   
## month  
## day 5 6 7 8 9  
## 1 190 286 269 83 167  
## 2 118 287 248 24 197  
## 3 149 242 236 77 183  
## 4 313 186 101 NA 189  
## 5 NA 220 175 NA 95  
## 6 NA 264 314 NA 92  
## 7 299 127 276 255 252  
## 8 99 273 267 229 220  
## 9 19 291 272 207 230  
## 10 194 323 175 222 259  
## 11 NA 259 139 137 236  
## 12 256 250 264 192 259  
## 13 290 148 175 273 238  
## 14 274 332 291 157 24  
## 15 65 322 48 64 112  
## 16 334 191 260 71 237  
## 17 307 284 274 51 224  
## 18 78 37 285 115 27  
## 19 322 120 187 244 238  
## 20 44 137 220 190 201  
## 21 8 150 7 259 238  
## 22 320 59 258 36 14  
## 23 25 91 295 255 139  
## 24 92 250 294 212 49  
## 25 66 135 223 238 20  
## 26 266 127 81 215 193  
## 27 NA 47 82 153 145  
## 28 13 98 213 203 191  
## 29 252 31 275 225 131  
## 30 223 138 253 237 223  
## 31 279 NA 254 188 NA  
##   
## , , variable = wind  
##   
## month  
## day 5 6 7 8 9  
## 1 7.4 8.6 4.1 6.9 6.9  
## 2 8.0 9.7 9.2 13.8 5.1  
## 3 12.6 16.1 9.2 7.4 2.8  
## 4 11.5 9.2 10.9 6.9 4.6  
## 5 14.3 8.6 4.6 7.4 7.4  
## 6 14.9 14.3 10.9 4.6 15.5  
## 7 8.6 9.7 5.1 4.0 10.9  
## 8 13.8 6.9 6.3 10.3 10.3  
## 9 20.1 13.8 5.7 8.0 10.9  
## 10 8.6 11.5 7.4 8.6 9.7  
## 11 6.9 10.9 8.6 11.5 14.9  
## 12 9.7 9.2 14.3 11.5 15.5  
## 13 9.2 8.0 14.9 11.5 6.3  
## 14 10.9 13.8 14.9 9.7 10.9  
## 15 13.2 11.5 14.3 11.5 11.5  
## 16 11.5 14.9 6.9 10.3 6.9  
## 17 12.0 20.7 10.3 6.3 13.8  
## 18 18.4 9.2 6.3 7.4 10.3  
## 19 11.5 11.5 5.1 10.9 10.3  
## 20 9.7 10.3 11.5 10.3 8.0  
## 21 9.7 6.3 6.9 15.5 12.6  
## 22 16.6 1.7 9.7 14.3 9.2  
## 23 9.7 4.6 11.5 12.6 10.3  
## 24 12.0 6.3 8.6 9.7 10.3  
## 25 16.6 8.0 8.0 3.4 16.6  
## 26 14.9 8.0 8.6 8.0 6.9  
## 27 8.0 10.3 12.0 5.7 13.2  
## 28 12.0 11.5 7.4 9.7 14.3  
## 29 14.9 14.9 7.4 2.3 8.0  
## 30 5.7 8.0 7.4 6.3 11.5  
## 31 7.4 NA 9.2 6.3 NA  
##   
## , , variable = temp  
##   
## month  
## day 5 6 7 8 9  
## 1 67 78 84 81 91  
## 2 72 74 85 81 92  
## 3 74 67 81 82 93  
## 4 62 84 84 86 93  
## 5 56 85 83 85 87  
## 6 66 79 83 87 84  
## 7 65 82 88 89 80  
## 8 59 87 92 90 78  
## 9 61 90 92 90 75  
## 10 69 87 89 92 73  
## 11 74 93 82 86 81  
## 12 69 92 73 86 76  
## 13 66 82 81 82 77  
## 14 68 80 91 80 71  
## 15 58 79 80 79 71  
## 16 64 77 81 77 78  
## 17 66 72 82 79 67  
## 18 57 65 84 76 76  
## 19 68 73 87 78 68  
## 20 62 76 85 78 82  
## 21 59 77 74 77 64  
## 22 73 76 81 72 71  
## 23 61 76 82 75 81  
## 24 61 76 86 79 69  
## 25 57 75 85 81 63  
## 26 58 78 82 86 70  
## 27 57 73 86 88 77  
## 28 67 80 88 97 75  
## 29 81 77 86 94 76  
## 30 79 83 83 96 68  
## 31 76 NA 81 94 NA

b <- acast(T, month ~ variable, mean)  
b

## ozone solar.r wind temp  
## 5 23.61538 181.2963 11.622581 65.54839  
## 6 29.44444 190.1667 10.266667 79.10000  
## 7 59.11538 216.4839 8.941935 83.90323  
## 8 59.96154 171.8571 8.793548 83.96774  
## 9 31.44828 167.4333 10.180000 76.90000

## 2. sqldf를 이용한 데이터 분석

R에는 많은 데이터 처리 함수가 있어 데이터를 편리하게 핸들링 할 수 있는 장점이 있다. sqldf 패키지는 SQL 문을 사용할 줄 아는 사용자가 쉽게 데이터에 접근할 수 있게 해준다.

#install.packages("sqldf")  
library(sqldf)

## Warning: package 'sqldf' was built under R version 4.0.3

## Loading required package: gsubfn

## Warning: package 'gsubfn' was built under R version 4.0.3

## Loading required package: proto

## Warning: package 'proto' was built under R version 4.0.3

## Loading required package: RSQLite

## Warning: package 'RSQLite' was built under R version 4.0.3

data(iris)  
sqldf("select\*from iris limit 10")

## 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  
## 7 4.6 3.4 1.4 0.3 setosa  
## 8 5.0 3.4 1.5 0.2 setosa  
## 9 4.4 2.9 1.4 0.2 setosa  
## 10 4.9 3.1 1.5 0.1 setosa

sqldf("select count(\*) from iris where species like 'se%'")

## count(\*)  
## 1 50

## 3. plyr

ply() 함수는 앞의 두 개의 문자를 접두사로 가지는데, 첫 번째 문자는 입력하는 데이터 형태를 나타내고, 두번째 문자는 출력하는 데이터 형태를 나타낸다. 예를들면, ddply는 데이터 프레임을 입력받아서, 어떤 데이터 가공한 후 다시 데이터프레임으로 출력하게 된다.

plyr 패키지는 데이터를 받아서 분리하고(split), 분할된 데이터에 특정 함수를 적용(apply), 결과를 재결합(combine) 처리하는 함수를 제공한다.

apply 함수와 multi-core 사용함수를 이용하면 for loop를 사용하지 않고 매우 간단하게 for loop 기능을 처리할 수 있고, apply 함수에 기반에 데이터와 출력변수를 동시에 배열로 치환하여 처리가 가능하다.

set.seed(1) #난수를 생성할 때마다 같은 값의 난수들을 생성  
d <- data.frame(year = rep(2012:2014, each = 6), count = round(runif(9,0,20)))  
# 2012-2014년 각각 6개를 만들고 count라는 변수에 난수를 생선하는 runif(생성할 난수의 수, 최소값, 최대값) 함수를 사용하여 0에서 20사이의 정수 9개를 저장  
d

## year count  
## 1 2012 5  
## 2 2012 7  
## 3 2012 11  
## 4 2012 18  
## 5 2012 4  
## 6 2012 18  
## 7 2013 19  
## 8 2013 13  
## 9 2013 13  
## 10 2013 5  
## 11 2013 7  
## 12 2013 11  
## 13 2014 18  
## 14 2014 4  
## 15 2014 18  
## 16 2014 19  
## 17 2014 13  
## 18 2014 13

# install plyr package  
#install.packages("plyr")  
library(plyr)

## Warning: package 'plyr' was built under R version 4.0.3

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:reshape':  
##   
## rename, round\_any

?ddply  
ddply(d, "year", function(x){  
 mean.count = mean(x$count)  
 sd.count = sd(x$count)  
 cv = sd.count/mean.count  
 data.frame(cv.count = cv)  
})

## year cv.count  
## 1 2012 0.5985621  
## 2 2013 0.4382254  
## 3 2014 0.3978489

ddply(d, .(year), summarise, mean.count = mean(count))

## year mean.count  
## 1 2012 10.50000  
## 2 2013 11.33333  
## 3 2014 14.16667

ddply(d, .(year), transform, total.count = sum(count))

## year count total.count  
## 1 2012 5 63  
## 2 2012 7 63  
## 3 2012 11 63  
## 4 2012 18 63  
## 5 2012 4 63  
## 6 2012 18 63  
## 7 2013 19 68  
## 8 2013 13 68  
## 9 2013 13 68  
## 10 2013 5 68  
## 11 2013 7 68  
## 12 2013 11 68  
## 13 2014 18 85  
## 14 2014 4 85  
## 15 2014 18 85  
## 16 2014 19 85  
## 17 2014 13 85  
## 18 2014 13 85

## 4. 데이터 테이블

데이터 프레임과 유사하지만 보다 빠른 그룹화grouping, 순서화ordering, 짧은 문장 지원 측면 및 데이터 분석 속도에서 데이터 프레임보다 연산속도가 빠르다.

#install.packages("data.table")  
library(data.table)

## Warning: package 'data.table' was built under R version 4.0.3

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:reshape2':  
##   
## dcast, melt

## The following object is masked from 'package:reshape':  
##   
## melt

DT <- data.table(x =c("b", "b" , "b", "a", "a"), v =rnorm(5))  
DT

## x v  
## 1: b -1.539950042  
## 2: b -0.928567035  
## 3: b -0.294720447  
## 4: a -0.005767173  
## 5: a 2.404653389

data(cars)  
str(cars)

## 'data.frame': 50 obs. of 2 variables:  
## $ speed: num 4 4 7 7 8 9 10 10 10 11 ...  
## $ dist : num 2 10 4 22 16 10 18 26 34 17 ...

데이터 프레임으로 된 데이터를 데이터 테이블 형식으로 불러와 CARS에 저장하자.

CARS <- as.data.table(cars)

str(CARS)

## Classes 'data.table' and 'data.frame': 50 obs. of 2 variables:  
## $ speed: num 4 4 7 7 8 9 10 10 10 11 ...  
## $ dist : num 2 10 4 22 16 10 18 26 34 17 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

데이터 테이블과 데이터 프레임의 차이를 알아보자. 지금까지 생성된 데이터 테이블 형식의 데이터를 살펴보기 위해 tables()기능을 이용해 크기가 어떠한지, key는 있는지 용량은 얼마인지 알아본다.

tables()

## NAME NROW NCOL MB COLS KEY  
## 1: CARS 50 2 0 speed,dist   
## 2: DT 5 2 0 x,v   
## Total: 0MB

table() 함수는 모든 데이터 테이블 객체의 목록을 저장한 데이터 테이블을 반환한다.

sapply(CARS, class)

## speed dist   
## "numeric" "numeric"

key를 이용한 빠른 데이터 접근.

setkey(DT, x)  
DT

## x v  
## 1: a -0.005767173  
## 2: a 2.404653389  
## 3: b -1.539950042  
## 4: b -0.928567035  
## 5: b -0.294720447

tables()

## NAME NROW NCOL MB COLS KEY  
## 1: CARS 50 2 0 speed,dist   
## 2: DT 5 2 0 x,v x  
## Total: 0MB

DT["b"] #"b"가 들어간 모든 데이터를 표시하고 싶은 경우

## x v  
## 1: b -1.5399500  
## 2: b -0.9285670  
## 3: b -0.2947204

DT["b", mult = 'first'] #"b"가 들어간 첫번째 결과

## x v  
## 1: b -1.53995

DT["b", mult = "last"]

## x v  
## 1: b -0.2947204

## 5. 결측치 처리와 이상값 검색

일반적으로 결측/무응답을 가진 자료를 분석할 때는 다음 사항을 고려한다. 다음을 고려하여 효율적인 방법으로 처리하여 데이터의 정보가 손실되거나 왜곡되는 것을 피해야 한다.

첫째, 효율성efficiency 문제

둘째, 자료처리 및 분석의 복잡성 문제

셋째, 관측된 자료와 결측된 자료 간의 차이에서 기인하는 bias

### 5.1 결측값의 대치법 imputation of missiong values

결측/무응답을 가진 자료를 무시하지 않고 분석할 수 있는 통계 방법론의 하나인 대치법imputation은 다음과 같다.

#### 1) 완전히 응답한 개체분석 Complete Analysis

불완전 자료는 모두 무시하고 완전하게 관측된 자료만으로 표준적 통계기법에 의해 분석하는 방법. 분석이 쉽다는 장점. 그러나 부분적으로 관측된 자료를 무시하므로 생기는 효율성 상실. 통계적 추론의 타당성 문제

#### 2) 평균대치법 Mean imputation

관측 또는 실험되어 얻어진 자료의 적절한 평균값으로 결측값을 대치해서 불완전한 자료를 완전한 자료로 만든 후, 완전한 자료를 마치 관측된 자료라 생각하고 분석하는 방법

#### 3) 단순확률 대치법 Single stochastic imputation

평균대치법에서 추정량 표준오차의 과소추정 문제를 보완하고자 고안된 방법으로 Hot-deck2 방법, NearestNeighbour2 방법 등이 있다. 기본적 아이디어는 평균대치법에서 관측된 자료를 토대로 추정된 통계량으로 결측값을 대치할 때 어떤 적절한 확률값을 부여한 후 대치하는 방법

#### 4) 다중대치법 Multiple imputation

단순확률대치법을 여러번 시행하여 분석하는 방법. 결측치를 가진 자료 분석에 사용하기 용이하고, 통계적 추론에 사용된 통계량의 효율성 및 일치성 등의 문제를 부분적으로 보완해준다. 그러나 추정량 표준오차의 과소추정 또는 계산의 난해성의 문제를 여전히 가지고 있다.

### 5.2 데이터 기초 통계

data(iris)  
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

str(iris)

## 'data.frame': 150 obs. of 5 variables:  
## $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...  
## $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...  
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...  
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...  
## $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

summary(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width   
## Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100   
## 1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300   
## Median :5.800 Median :3.000 Median :4.350 Median :1.300   
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199   
## 3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800   
## Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500   
## Species   
## setosa :50   
## versicolor:50   
## virginica :50   
##   
##   
##

cov(iris[,1:4])

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## Sepal.Length 0.6856935 -0.0424340 1.2743154 0.5162707  
## Sepal.Width -0.0424340 0.1899794 -0.3296564 -0.1216394  
## Petal.Length 1.2743154 -0.3296564 3.1162779 1.2956094  
## Petal.Width 0.5162707 -0.1216394 1.2956094 0.5810063

cor(iris[,1:4])

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## Sepal.Length 1.0000000 -0.1175698 0.8717538 0.8179411  
## Sepal.Width -0.1175698 1.0000000 -0.4284401 -0.3661259  
## Petal.Length 0.8717538 -0.4284401 1.0000000 0.9628654  
## Petal.Width 0.8179411 -0.3661259 0.9628654 1.0000000

### 5.3 결측값 처리

R에서 결측값 처리 관련 패키지는 Amelia2, Mice, mistools 등이 있다. 아래는 Amelia 패키지를 사용한 예시. 결측값은 NA (not available), 불가능한 값은 NaN (not a number)으로 처리한다.

y <- c(1,2,3,NA)  
is.na(y)

## [1] FALSE FALSE FALSE TRUE

data(iris)  
iris[iris$Petal.Width == 0.2, "Petal.Width"] <- NA

is.na(iris$Petal.Width)

## [1] TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE  
## [13] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE  
## [25] TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE  
## [37] TRUE FALSE TRUE TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE  
## [49] TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [145] FALSE FALSE FALSE FALSE FALSE FALSE

#결측값을 제거하여 문제를 해결하는 방법  
mean(y)

## [1] NA

mean(y, na.rm = TRUE)

## [1] 2

library(reshape)  
data(french\_fries)  
??french\_fries

french\_fries[!complete.cases(french\_fries),]

## time treatment subject rep potato buttery grassy rancid painty  
## 315 5 3 15 1 NA NA NA NA NA  
## 455 7 2 79 1 7.3 NA 0.0 0.7 0  
## 515 8 1 79 1 10.5 NA 0.0 0.5 0  
## 520 8 2 16 1 4.5 NA 1.4 6.7 0  
## 563 8 2 79 2 5.7 0 1.4 2.3 NA

Amelia package 사용

#install.packages("Amelia")  
library(Amelia)

## Warning: package 'Amelia' was built under R version 4.0.3

## Loading required package: Rcpp

## Warning: package 'Rcpp' was built under R version 4.0.3

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.7.6, built: 2019-11-24)  
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

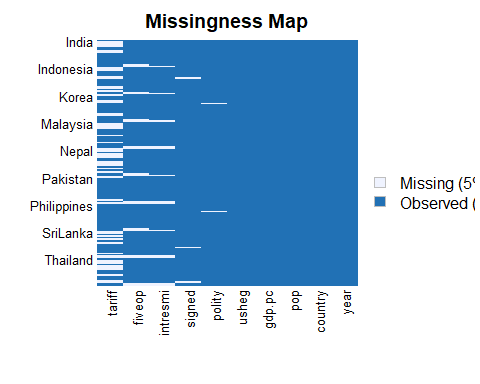
data(freetrade)  
??freetrade  
  
head(freetrade)

## year country tariff polity pop gdp.pc intresmi signed fiveop  
## 1 1981 SriLanka NA 6 14988000 461.0236 1.937347 0 12.4  
## 2 1982 SriLanka NA 5 15189000 473.7634 1.964430 0 12.5  
## 3 1983 SriLanka 41.3 5 15417000 489.2266 1.663936 1 12.3  
## 4 1984 SriLanka NA 5 15599000 508.1739 2.797462 0 12.3  
## 5 1985 SriLanka 31.0 5 15837000 525.5609 2.259116 0 12.3  
## 6 1986 SriLanka NA 5 16117000 538.9237 1.832549 0 12.5  
## usheg  
## 1 0.2593112  
## 2 0.2558008  
## 3 0.2655022  
## 4 0.2988009  
## 5 0.2952431  
## 6 0.2886563

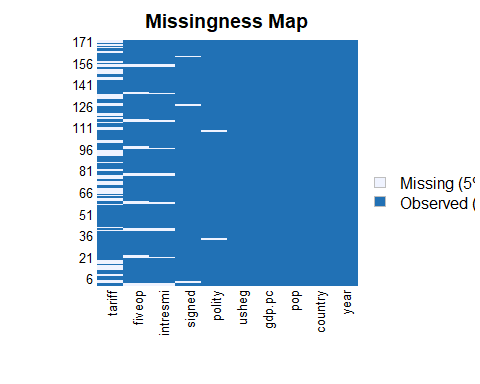
a.out <- amelia(freetrade, m = 5, ts = "year", cs = "country")

## -- Imputation 1 --  
##   
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18  
##   
## -- Imputation 2 --  
##   
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14  
##   
## -- Imputation 3 --  
##   
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
## 21 22 23  
##   
## -- Imputation 4 --  
##   
## 1 2 3 4 5 6 7 8 9 10 11 12  
##   
## -- Imputation 5 --  
##   
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14

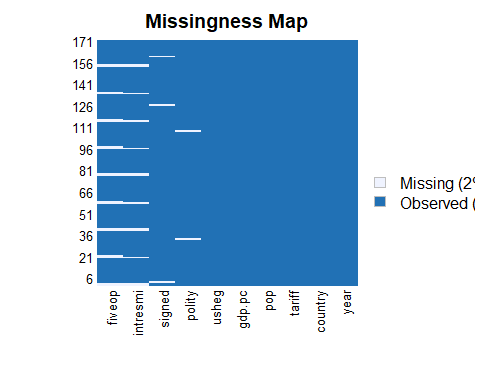
missmap(a.out)



missmap(freetrade)



freetrade$tariff <- a.out$imputations[[5]]$tariff  
missmap(freetrade)



### 5.4 이상값 검색

이상값 outlier 검색은 분석에서 전처리를 어떻게 할 지 결정할 때와 부정사용방지 시스템에서 규칙을 발견하는데 사용할 수 있다.

의도하지 않게 잘못 입력된 경우 (bad data), 의도하지 않게 입력됐으나 분석목적에 부합하지 않아 제거해야하는 경우 (bad data), 의도되지 않은 현상이지만 분석에 포함해야 하는 경우 (이상값), 의도된 이상값 사이의 차이를 인지한다.

# outlier package 사용  
#install.packages("outliers")  
library(outliers)

## Warning: package 'outliers' was built under R version 4.0.3

set.seed(1234)  
y = rnorm(100)  
  
outlier(y, opposite = TRUE) #반대방향으로 가장 차이가 많이 나는 값 출력

## [1] -2.345698

dim(y) <- c(20,5) # 행 20, 열 5 행렬 생성  
  
outlier(y) # 각 열의 평균과 가장 차이가 많은 값을 각 열 별로 출력

## [1] 2.415835 1.102298 1.647817 2.548991 2.121117

outlier(y, opposite = TRUE) # 각 열 별로 반대 방향으로 열 평균과 가장 차이가 많은 값 출력

## [1] -2.345698 -2.180040 -1.806031 -1.390701 -1.372302

boxplot(y)

