D2SC ICMA Notebook

Vincent Espana

2024-10-30

Table of Contents

# Initial Loading

library(tidyverse)

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

# ICMA 9/9/2024

plot(cars)



# ICMA 9/11/2024

This is where we put plain text

# this is where we put r code

ohno\_this\_is\_a\_nightmare <- "hello"  
# dots also work

2+2

## [1] 4

x <- 2 + 2

?mean

## starting httpd help server ... done

mean(c(2, 3, 4))

## [1] 3

myVector <- c(2, 3, 4)

my\_vector <- c(2, 3, 4)

?sd

sd(1:2) ^ 2

## [1] 0.5

# ICMA 9/16/2024

#install.packages() last time  
library(tidyverse)

#some functions cant be used/need more work  
stats::filter()

#does not work  
head(3, billboard)

#whatever output is first is put into the first position  
# 3 will technically be in second position  
billboard %>%  
 head(3)

## # A tibble: 3 × 79  
## artist track date.entered wk1 wk2 wk3 wk4 wk5 wk6 wk7 wk8  
## <chr> <chr> <date> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 Pac Baby… 2000-02-26 87 82 72 77 87 94 99 NA  
## 2 2Ge+her The … 2000-09-02 91 87 92 NA NA NA NA NA  
## 3 3 Doors Do… Kryp… 2000-04-08 81 70 68 67 66 57 54 53  
## # ℹ 68 more variables: wk9 <dbl>, wk10 <dbl>, wk11 <dbl>, wk12 <dbl>,  
## # wk13 <dbl>, wk14 <dbl>, wk15 <dbl>, wk16 <dbl>, wk17 <dbl>, wk18 <dbl>,  
## # wk19 <dbl>, wk20 <dbl>, wk21 <dbl>, wk22 <dbl>, wk23 <dbl>, wk24 <dbl>,  
## # wk25 <dbl>, wk26 <dbl>, wk27 <dbl>, wk28 <dbl>, wk29 <dbl>, wk30 <dbl>,  
## # wk31 <dbl>, wk32 <dbl>, wk33 <dbl>, wk34 <dbl>, wk35 <dbl>, wk36 <dbl>,  
## # wk37 <dbl>, wk38 <dbl>, wk39 <dbl>, wk40 <dbl>, wk41 <dbl>, wk42 <dbl>,  
## # wk43 <dbl>, wk44 <dbl>, wk45 <dbl>, wk46 <dbl>, wk47 <dbl>, wk48 <dbl>, …

my\_numbers <- c(1, 2, 3, 4, 5, 8, 9, 10)

mean(my\_numbers)

## [1] 5.25

my\_numbers %>%  
 mean()

## [1] 5.25

getwd()

## [1] "C:/Users/vince/OneDrive/Documents/GitHub/Data2SciComm/ICMA"

mtcars

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2

write\_csv(mtcars, #object name  
 "mtcars\_fromR.csv") #file name to save

mtcars %>%  
 rownames\_to\_column() %>%  
 write\_csv("mtcars\_fromR\_rownames.csv")

mtcars\_read <- read\_csv("mtcars\_fromR\_rownames.csv")

## Rows: 32 Columns: 12  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): rowname  
## dbl (11): mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb  
##   
## 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.

# ICMA 9/18/2024

?ChickWeight

ChickWeight %>%  
 select(weight)

## weight  
## 1 42  
## 2 51  
## 3 59  
## 4 64  
## 5 76  
## 6 93  
## 7 106  
## 8 125  
## 9 149  
## 10 171  
## 11 199  
## 12 205  
## 13 40  
## 14 49  
## 15 58  
## 16 72  
## 17 84  
## 18 103  
## 19 122  
## 20 138  
## 21 162  
## 22 187  
## 23 209  
## 24 215  
## 25 43  
## 26 39  
## 27 55  
## 28 67  
## 29 84  
## 30 99  
## 31 115  
## 32 138  
## 33 163  
## 34 187  
## 35 198  
## 36 202  
## 37 42  
## 38 49  
## 39 56  
## 40 67  
## 41 74  
## 42 87  
## 43 102  
## 44 108  
## 45 136  
## 46 154  
## 47 160  
## 48 157  
## 49 41  
## 50 42  
## 51 48  
## 52 60  
## 53 79  
## 54 106  
## 55 141  
## 56 164  
## 57 197  
## 58 199  
## 59 220  
## 60 223  
## 61 41  
## 62 49  
## 63 59  
## 64 74  
## 65 97  
## 66 124  
## 67 141  
## 68 148  
## 69 155  
## 70 160  
## 71 160  
## 72 157  
## 73 41  
## 74 49  
## 75 57  
## 76 71  
## 77 89  
## 78 112  
## 79 146  
## 80 174  
## 81 218  
## 82 250  
## 83 288  
## 84 305  
## 85 42  
## 86 50  
## 87 61  
## 88 71  
## 89 84  
## 90 93  
## 91 110  
## 92 116  
## 93 126  
## 94 134  
## 95 125  
## 96 42  
## 97 51  
## 98 59  
## 99 68  
## 100 85  
## 101 96  
## 102 90  
## 103 92  
## 104 93  
## 105 100  
## 106 100  
## 107 98  
## 108 41  
## 109 44  
## 110 52  
## 111 63  
## 112 74  
## 113 81  
## 114 89  
## 115 96  
## 116 101  
## 117 112  
## 118 120  
## 119 124  
## 120 43  
## 121 51  
## 122 63  
## 123 84  
## 124 112  
## 125 139  
## 126 168  
## 127 177  
## 128 182  
## 129 184  
## 130 181  
## 131 175  
## 132 41  
## 133 49  
## 134 56  
## 135 62  
## 136 72  
## 137 88  
## 138 119  
## 139 135  
## 140 162  
## 141 185  
## 142 195  
## 143 205  
## 144 41  
## 145 48  
## 146 53  
## 147 60  
## 148 65  
## 149 67  
## 150 71  
## 151 70  
## 152 71  
## 153 81  
## 154 91  
## 155 96  
## 156 41  
## 157 49  
## 158 62  
## 159 79  
## 160 101  
## 161 128  
## 162 164  
## 163 192  
## 164 227  
## 165 248  
## 166 259  
## 167 266  
## 168 41  
## 169 49  
## 170 56  
## 171 64  
## 172 68  
## 173 68  
## 174 67  
## 175 68  
## 176 41  
## 177 45  
## 178 49  
## 179 51  
## 180 57  
## 181 51  
## 182 54  
## 183 42  
## 184 51  
## 185 61  
## 186 72  
## 187 83  
## 188 89  
## 189 98  
## 190 103  
## 191 113  
## 192 123  
## 193 133  
## 194 142  
## 195 39  
## 196 35  
## 197 43  
## 198 48  
## 199 55  
## 200 62  
## 201 65  
## 202 71  
## 203 82  
## 204 88  
## 205 106  
## 206 120  
## 207 144  
## 208 157  
## 209 41  
## 210 47  
## 211 54  
## 212 58  
## 213 65  
## 214 73  
## 215 77  
## 216 89  
## 217 98  
## 218 107  
## 219 115  
## 220 117  
## 221 40  
## 222 50  
## 223 62  
## 224 86  
## 225 125  
## 226 163  
## 227 217  
## 228 240  
## 229 275  
## 230 307  
## 231 318  
## 232 331  
## 233 41  
## 234 55  
## 235 64  
## 236 77  
## 237 90  
## 238 95  
## 239 108  
## 240 111  
## 241 131  
## 242 148  
## 243 164  
## 244 167  
## 245 43  
## 246 52  
## 247 61  
## 248 73  
## 249 90  
## 250 103  
## 251 127  
## 252 135  
## 253 145  
## 254 163  
## 255 170  
## 256 175  
## 257 42  
## 258 52  
## 259 58  
## 260 74  
## 261 66  
## 262 68  
## 263 70  
## 264 71  
## 265 72  
## 266 72  
## 267 76  
## 268 74  
## 269 40  
## 270 49  
## 271 62  
## 272 78  
## 273 102  
## 274 124  
## 275 146  
## 276 164  
## 277 197  
## 278 231  
## 279 259  
## 280 265  
## 281 42  
## 282 48  
## 283 57  
## 284 74  
## 285 93  
## 286 114  
## 287 136  
## 288 147  
## 289 169  
## 290 205  
## 291 236  
## 292 251  
## 293 39  
## 294 46  
## 295 58  
## 296 73  
## 297 87  
## 298 100  
## 299 115  
## 300 123  
## 301 144  
## 302 163  
## 303 185  
## 304 192  
## 305 39  
## 306 46  
## 307 58  
## 308 73  
## 309 92  
## 310 114  
## 311 145  
## 312 156  
## 313 184  
## 314 207  
## 315 212  
## 316 233  
## 317 39  
## 318 48  
## 319 59  
## 320 74  
## 321 87  
## 322 106  
## 323 134  
## 324 150  
## 325 187  
## 326 230  
## 327 279  
## 328 309  
## 329 42  
## 330 48  
## 331 59  
## 332 72  
## 333 85  
## 334 98  
## 335 115  
## 336 122  
## 337 143  
## 338 151  
## 339 157  
## 340 150  
## 341 42  
## 342 53  
## 343 62  
## 344 73  
## 345 85  
## 346 102  
## 347 123  
## 348 138  
## 349 170  
## 350 204  
## 351 235  
## 352 256  
## 353 41  
## 354 49  
## 355 65  
## 356 82  
## 357 107  
## 358 129  
## 359 159  
## 360 179  
## 361 221  
## 362 263  
## 363 291  
## 364 305  
## 365 39  
## 366 50  
## 367 63  
## 368 77  
## 369 96  
## 370 111  
## 371 137  
## 372 144  
## 373 151  
## 374 146  
## 375 156  
## 376 147  
## 377 41  
## 378 49  
## 379 63  
## 380 85  
## 381 107  
## 382 134  
## 383 164  
## 384 186  
## 385 235  
## 386 294  
## 387 327  
## 388 341  
## 389 41  
## 390 53  
## 391 64  
## 392 87  
## 393 123  
## 394 158  
## 395 201  
## 396 238  
## 397 287  
## 398 332  
## 399 361  
## 400 373  
## 401 39  
## 402 48  
## 403 61  
## 404 76  
## 405 98  
## 406 116  
## 407 145  
## 408 166  
## 409 198  
## 410 227  
## 411 225  
## 412 220  
## 413 41  
## 414 48  
## 415 56  
## 416 68  
## 417 80  
## 418 83  
## 419 103  
## 420 112  
## 421 135  
## 422 157  
## 423 169  
## 424 178  
## 425 41  
## 426 49  
## 427 61  
## 428 74  
## 429 98  
## 430 109  
## 431 128  
## 432 154  
## 433 192  
## 434 232  
## 435 280  
## 436 290  
## 437 42  
## 438 50  
## 439 61  
## 440 78  
## 441 89  
## 442 109  
## 443 130  
## 444 146  
## 445 170  
## 446 214  
## 447 250  
## 448 272  
## 449 41  
## 450 55  
## 451 66  
## 452 79  
## 453 101  
## 454 120  
## 455 154  
## 456 182  
## 457 215  
## 458 262  
## 459 295  
## 460 321  
## 461 42  
## 462 51  
## 463 66  
## 464 85  
## 465 103  
## 466 124  
## 467 155  
## 468 153  
## 469 175  
## 470 184  
## 471 199  
## 472 204  
## 473 42  
## 474 49  
## 475 63  
## 476 84  
## 477 103  
## 478 126  
## 479 160  
## 480 174  
## 481 204  
## 482 234  
## 483 269  
## 484 281  
## 485 42  
## 486 55  
## 487 69  
## 488 96  
## 489 131  
## 490 157  
## 491 184  
## 492 188  
## 493 197  
## 494 198  
## 495 199  
## 496 200  
## 497 42  
## 498 51  
## 499 65  
## 500 86  
## 501 103  
## 502 118  
## 503 127  
## 504 138  
## 505 145  
## 506 146  
## 507 41  
## 508 50  
## 509 61  
## 510 78  
## 511 98  
## 512 117  
## 513 135  
## 514 141  
## 515 147  
## 516 174  
## 517 197  
## 518 196  
## 519 40  
## 520 52  
## 521 62  
## 522 82  
## 523 101  
## 524 120  
## 525 144  
## 526 156  
## 527 173  
## 528 210  
## 529 231  
## 530 238  
## 531 41  
## 532 53  
## 533 66  
## 534 79  
## 535 100  
## 536 123  
## 537 148  
## 538 157  
## 539 168  
## 540 185  
## 541 210  
## 542 205  
## 543 39  
## 544 50  
## 545 62  
## 546 80  
## 547 104  
## 548 125  
## 549 154  
## 550 170  
## 551 222  
## 552 261  
## 553 303  
## 554 322  
## 555 40  
## 556 53  
## 557 64  
## 558 85  
## 559 108  
## 560 128  
## 561 152  
## 562 166  
## 563 184  
## 564 203  
## 565 233  
## 566 237  
## 567 41  
## 568 54  
## 569 67  
## 570 84  
## 571 105  
## 572 122  
## 573 155  
## 574 175  
## 575 205  
## 576 234  
## 577 264  
## 578 264

chick\_clean <- ChickWeight %>%  
 select(chick\_id = Chick,  
 diet = Diet,  
 age\_days = Time,  
 weight\_gm = weight) #will reorder as well as rename  
#will not print because this is only assignment --> will show up in environment

equality operator in R is “==”

chick\_clean %>%  
 arrange(chick\_id)

## chick\_id diet age\_days weight\_gm  
## 1 18 1 0 39  
## 2 18 1 2 35  
## 3 16 1 0 41  
## 4 16 1 2 45  
## 5 16 1 4 49  
## 6 16 1 6 51  
## 7 16 1 8 57  
## 8 16 1 10 51  
## 9 16 1 12 54  
## 10 15 1 0 41  
## 11 15 1 2 49  
## 12 15 1 4 56  
## 13 15 1 6 64  
## 14 15 1 8 68  
## 15 15 1 10 68  
## 16 15 1 12 67  
## 17 15 1 14 68  
## 18 13 1 0 41  
## 19 13 1 2 48  
## 20 13 1 4 53  
## 21 13 1 6 60  
## 22 13 1 8 65  
## 23 13 1 10 67  
## 24 13 1 12 71  
## 25 13 1 14 70  
## 26 13 1 16 71  
## 27 13 1 18 81  
## 28 13 1 20 91  
## 29 13 1 21 96  
## 30 9 1 0 42  
## 31 9 1 2 51  
## 32 9 1 4 59  
## 33 9 1 6 68  
## 34 9 1 8 85  
## 35 9 1 10 96  
## 36 9 1 12 90  
## 37 9 1 14 92  
## 38 9 1 16 93  
## 39 9 1 18 100  
## 40 9 1 20 100  
## 41 9 1 21 98  
## 42 20 1 0 41  
## 43 20 1 2 47  
## 44 20 1 4 54  
## 45 20 1 6 58  
## 46 20 1 8 65  
## 47 20 1 10 73  
## 48 20 1 12 77  
## 49 20 1 14 89  
## 50 20 1 16 98  
## 51 20 1 18 107  
## 52 20 1 20 115  
## 53 20 1 21 117  
## 54 10 1 0 41  
## 55 10 1 2 44  
## 56 10 1 4 52  
## 57 10 1 6 63  
## 58 10 1 8 74  
## 59 10 1 10 81  
## 60 10 1 12 89  
## 61 10 1 14 96  
## 62 10 1 16 101  
## 63 10 1 18 112  
## 64 10 1 20 120  
## 65 10 1 21 124  
## 66 8 1 0 42  
## 67 8 1 2 50  
## 68 8 1 4 61  
## 69 8 1 6 71  
## 70 8 1 8 84  
## 71 8 1 10 93  
## 72 8 1 12 110  
## 73 8 1 14 116  
## 74 8 1 16 126  
## 75 8 1 18 134  
## 76 8 1 20 125  
## 77 17 1 0 42  
## 78 17 1 2 51  
## 79 17 1 4 61  
## 80 17 1 6 72  
## 81 17 1 8 83  
## 82 17 1 10 89  
## 83 17 1 12 98  
## 84 17 1 14 103  
## 85 17 1 16 113  
## 86 17 1 18 123  
## 87 17 1 20 133  
## 88 17 1 21 142  
## 89 19 1 0 43  
## 90 19 1 2 48  
## 91 19 1 4 55  
## 92 19 1 6 62  
## 93 19 1 8 65  
## 94 19 1 10 71  
## 95 19 1 12 82  
## 96 19 1 14 88  
## 97 19 1 16 106  
## 98 19 1 18 120  
## 99 19 1 20 144  
## 100 19 1 21 157  
## 101 4 1 0 42  
## 102 4 1 2 49  
## 103 4 1 4 56  
## 104 4 1 6 67  
## 105 4 1 8 74  
## 106 4 1 10 87  
## 107 4 1 12 102  
## 108 4 1 14 108  
## 109 4 1 16 136  
## 110 4 1 18 154  
## 111 4 1 20 160  
## 112 4 1 21 157  
## 113 6 1 0 41  
## 114 6 1 2 49  
## 115 6 1 4 59  
## 116 6 1 6 74  
## 117 6 1 8 97  
## 118 6 1 10 124  
## 119 6 1 12 141  
## 120 6 1 14 148  
## 121 6 1 16 155  
## 122 6 1 18 160  
## 123 6 1 20 160  
## 124 6 1 21 157  
## 125 11 1 0 43  
## 126 11 1 2 51  
## 127 11 1 4 63  
## 128 11 1 6 84  
## 129 11 1 8 112  
## 130 11 1 10 139  
## 131 11 1 12 168  
## 132 11 1 14 177  
## 133 11 1 16 182  
## 134 11 1 18 184  
## 135 11 1 20 181  
## 136 11 1 21 175  
## 137 3 1 0 43  
## 138 3 1 2 39  
## 139 3 1 4 55  
## 140 3 1 6 67  
## 141 3 1 8 84  
## 142 3 1 10 99  
## 143 3 1 12 115  
## 144 3 1 14 138  
## 145 3 1 16 163  
## 146 3 1 18 187  
## 147 3 1 20 198  
## 148 3 1 21 202  
## 149 1 1 0 42  
## 150 1 1 2 51  
## 151 1 1 4 59  
## 152 1 1 6 64  
## 153 1 1 8 76  
## 154 1 1 10 93  
## 155 1 1 12 106  
## 156 1 1 14 125  
## 157 1 1 16 149  
## 158 1 1 18 171  
## 159 1 1 20 199  
## 160 1 1 21 205  
## 161 12 1 0 41  
## 162 12 1 2 49  
## 163 12 1 4 56  
## 164 12 1 6 62  
## 165 12 1 8 72  
## 166 12 1 10 88  
## 167 12 1 12 119  
## 168 12 1 14 135  
## 169 12 1 16 162  
## 170 12 1 18 185  
## 171 12 1 20 195  
## 172 12 1 21 205  
## 173 2 1 0 40  
## 174 2 1 2 49  
## 175 2 1 4 58  
## 176 2 1 6 72  
## 177 2 1 8 84  
## 178 2 1 10 103  
## 179 2 1 12 122  
## 180 2 1 14 138  
## 181 2 1 16 162  
## 182 2 1 18 187  
## 183 2 1 20 209  
## 184 2 1 21 215  
## 185 5 1 0 41  
## 186 5 1 2 42  
## 187 5 1 4 48  
## 188 5 1 6 60  
## 189 5 1 8 79  
## 190 5 1 10 106  
## 191 5 1 12 141  
## 192 5 1 14 164  
## 193 5 1 16 197  
## 194 5 1 18 199  
## 195 5 1 20 220  
## 196 5 1 21 223  
## 197 14 1 0 41  
## 198 14 1 2 49  
## 199 14 1 4 62  
## 200 14 1 6 79  
## 201 14 1 8 101  
## 202 14 1 10 128  
## 203 14 1 12 164  
## 204 14 1 14 192  
## 205 14 1 16 227  
## 206 14 1 18 248  
## 207 14 1 20 259  
## 208 14 1 21 266  
## 209 7 1 0 41  
## 210 7 1 2 49  
## 211 7 1 4 57  
## 212 7 1 6 71  
## 213 7 1 8 89  
## 214 7 1 10 112  
## 215 7 1 12 146  
## 216 7 1 14 174  
## 217 7 1 16 218  
## 218 7 1 18 250  
## 219 7 1 20 288  
## 220 7 1 21 305  
## 221 24 2 0 42  
## 222 24 2 2 52  
## 223 24 2 4 58  
## 224 24 2 6 74  
## 225 24 2 8 66  
## 226 24 2 10 68  
## 227 24 2 12 70  
## 228 24 2 14 71  
## 229 24 2 16 72  
## 230 24 2 18 72  
## 231 24 2 20 76  
## 232 24 2 21 74  
## 233 30 2 0 42  
## 234 30 2 2 48  
## 235 30 2 4 59  
## 236 30 2 6 72  
## 237 30 2 8 85  
## 238 30 2 10 98  
## 239 30 2 12 115  
## 240 30 2 14 122  
## 241 30 2 16 143  
## 242 30 2 18 151  
## 243 30 2 20 157  
## 244 30 2 21 150  
## 245 22 2 0 41  
## 246 22 2 2 55  
## 247 22 2 4 64  
## 248 22 2 6 77  
## 249 22 2 8 90  
## 250 22 2 10 95  
## 251 22 2 12 108  
## 252 22 2 14 111  
## 253 22 2 16 131  
## 254 22 2 18 148  
## 255 22 2 20 164  
## 256 22 2 21 167  
## 257 23 2 0 43  
## 258 23 2 2 52  
## 259 23 2 4 61  
## 260 23 2 6 73  
## 261 23 2 8 90  
## 262 23 2 10 103  
## 263 23 2 12 127  
## 264 23 2 14 135  
## 265 23 2 16 145  
## 266 23 2 18 163  
## 267 23 2 20 170  
## 268 23 2 21 175  
## 269 27 2 0 39  
## 270 27 2 2 46  
## 271 27 2 4 58  
## 272 27 2 6 73  
## 273 27 2 8 87  
## 274 27 2 10 100  
## 275 27 2 12 115  
## 276 27 2 14 123  
## 277 27 2 16 144  
## 278 27 2 18 163  
## 279 27 2 20 185  
## 280 27 2 21 192  
## 281 28 2 0 39  
## 282 28 2 2 46  
## 283 28 2 4 58  
## 284 28 2 6 73  
## 285 28 2 8 92  
## 286 28 2 10 114  
## 287 28 2 12 145  
## 288 28 2 14 156  
## 289 28 2 16 184  
## 290 28 2 18 207  
## 291 28 2 20 212  
## 292 28 2 21 233  
## 293 26 2 0 42  
## 294 26 2 2 48  
## 295 26 2 4 57  
## 296 26 2 6 74  
## 297 26 2 8 93  
## 298 26 2 10 114  
## 299 26 2 12 136  
## 300 26 2 14 147  
## 301 26 2 16 169  
## 302 26 2 18 205  
## 303 26 2 20 236  
## 304 26 2 21 251  
## 305 25 2 0 40  
## 306 25 2 2 49  
## 307 25 2 4 62  
## 308 25 2 6 78  
## 309 25 2 8 102  
## 310 25 2 10 124  
## 311 25 2 12 146  
## 312 25 2 14 164  
## 313 25 2 16 197  
## 314 25 2 18 231  
## 315 25 2 20 259  
## 316 25 2 21 265  
## 317 29 2 0 39  
## 318 29 2 2 48  
## 319 29 2 4 59  
## 320 29 2 6 74  
## 321 29 2 8 87  
## 322 29 2 10 106  
## 323 29 2 12 134  
## 324 29 2 14 150  
## 325 29 2 16 187  
## 326 29 2 18 230  
## 327 29 2 20 279  
## 328 29 2 21 309  
## 329 21 2 0 40  
## 330 21 2 2 50  
## 331 21 2 4 62  
## 332 21 2 6 86  
## 333 21 2 8 125  
## 334 21 2 10 163  
## 335 21 2 12 217  
## 336 21 2 14 240  
## 337 21 2 16 275  
## 338 21 2 18 307  
## 339 21 2 20 318  
## 340 21 2 21 331  
## 341 33 3 0 39  
## 342 33 3 2 50  
## 343 33 3 4 63  
## 344 33 3 6 77  
## 345 33 3 8 96  
## 346 33 3 10 111  
## 347 33 3 12 137  
## 348 33 3 14 144  
## 349 33 3 16 151  
## 350 33 3 18 146  
## 351 33 3 20 156  
## 352 33 3 21 147  
## 353 37 3 0 41  
## 354 37 3 2 48  
## 355 37 3 4 56  
## 356 37 3 6 68  
## 357 37 3 8 80  
## 358 37 3 10 83  
## 359 37 3 12 103  
## 360 37 3 14 112  
## 361 37 3 16 135  
## 362 37 3 18 157  
## 363 37 3 20 169  
## 364 37 3 21 178  
## 365 36 3 0 39  
## 366 36 3 2 48  
## 367 36 3 4 61  
## 368 36 3 6 76  
## 369 36 3 8 98  
## 370 36 3 10 116  
## 371 36 3 12 145  
## 372 36 3 14 166  
## 373 36 3 16 198  
## 374 36 3 18 227  
## 375 36 3 20 225  
## 376 36 3 21 220  
## 377 31 3 0 42  
## 378 31 3 2 53  
## 379 31 3 4 62  
## 380 31 3 6 73  
## 381 31 3 8 85  
## 382 31 3 10 102  
## 383 31 3 12 123  
## 384 31 3 14 138  
## 385 31 3 16 170  
## 386 31 3 18 204  
## 387 31 3 20 235  
## 388 31 3 21 256  
## 389 39 3 0 42  
## 390 39 3 2 50  
## 391 39 3 4 61  
## 392 39 3 6 78  
## 393 39 3 8 89  
## 394 39 3 10 109  
## 395 39 3 12 130  
## 396 39 3 14 146  
## 397 39 3 16 170  
## 398 39 3 18 214  
## 399 39 3 20 250  
## 400 39 3 21 272  
## 401 38 3 0 41  
## 402 38 3 2 49  
## 403 38 3 4 61  
## 404 38 3 6 74  
## 405 38 3 8 98  
## 406 38 3 10 109  
## 407 38 3 12 128  
## 408 38 3 14 154  
## 409 38 3 16 192  
## 410 38 3 18 232  
## 411 38 3 20 280  
## 412 38 3 21 290  
## 413 32 3 0 41  
## 414 32 3 2 49  
## 415 32 3 4 65  
## 416 32 3 6 82  
## 417 32 3 8 107  
## 418 32 3 10 129  
## 419 32 3 12 159  
## 420 32 3 14 179  
## 421 32 3 16 221  
## 422 32 3 18 263  
## 423 32 3 20 291  
## 424 32 3 21 305  
## 425 40 3 0 41  
## 426 40 3 2 55  
## 427 40 3 4 66  
## 428 40 3 6 79  
## 429 40 3 8 101  
## 430 40 3 10 120  
## 431 40 3 12 154  
## 432 40 3 14 182  
## 433 40 3 16 215  
## 434 40 3 18 262  
## 435 40 3 20 295  
## 436 40 3 21 321  
## 437 34 3 0 41  
## 438 34 3 2 49  
## 439 34 3 4 63  
## 440 34 3 6 85  
## 441 34 3 8 107  
## 442 34 3 10 134  
## 443 34 3 12 164  
## 444 34 3 14 186  
## 445 34 3 16 235  
## 446 34 3 18 294  
## 447 34 3 20 327  
## 448 34 3 21 341  
## 449 35 3 0 41  
## 450 35 3 2 53  
## 451 35 3 4 64  
## 452 35 3 6 87  
## 453 35 3 8 123  
## 454 35 3 10 158  
## 455 35 3 12 201  
## 456 35 3 14 238  
## 457 35 3 16 287  
## 458 35 3 18 332  
## 459 35 3 20 361  
## 460 35 3 21 373  
## 461 44 4 0 42  
## 462 44 4 2 51  
## 463 44 4 4 65  
## 464 44 4 6 86  
## 465 44 4 8 103  
## 466 44 4 10 118  
## 467 44 4 12 127  
## 468 44 4 14 138  
## 469 44 4 16 145  
## 470 44 4 18 146  
## 471 45 4 0 41  
## 472 45 4 2 50  
## 473 45 4 4 61  
## 474 45 4 6 78  
## 475 45 4 8 98  
## 476 45 4 10 117  
## 477 45 4 12 135  
## 478 45 4 14 141  
## 479 45 4 16 147  
## 480 45 4 18 174  
## 481 45 4 20 197  
## 482 45 4 21 196  
## 483 43 4 0 42  
## 484 43 4 2 55  
## 485 43 4 4 69  
## 486 43 4 6 96  
## 487 43 4 8 131  
## 488 43 4 10 157  
## 489 43 4 12 184  
## 490 43 4 14 188  
## 491 43 4 16 197  
## 492 43 4 18 198  
## 493 43 4 20 199  
## 494 43 4 21 200  
## 495 41 4 0 42  
## 496 41 4 2 51  
## 497 41 4 4 66  
## 498 41 4 6 85  
## 499 41 4 8 103  
## 500 41 4 10 124  
## 501 41 4 12 155  
## 502 41 4 14 153  
## 503 41 4 16 175  
## 504 41 4 18 184  
## 505 41 4 20 199  
## 506 41 4 21 204  
## 507 47 4 0 41  
## 508 47 4 2 53  
## 509 47 4 4 66  
## 510 47 4 6 79  
## 511 47 4 8 100  
## 512 47 4 10 123  
## 513 47 4 12 148  
## 514 47 4 14 157  
## 515 47 4 16 168  
## 516 47 4 18 185  
## 517 47 4 20 210  
## 518 47 4 21 205  
## 519 49 4 0 40  
## 520 49 4 2 53  
## 521 49 4 4 64  
## 522 49 4 6 85  
## 523 49 4 8 108  
## 524 49 4 10 128  
## 525 49 4 12 152  
## 526 49 4 14 166  
## 527 49 4 16 184  
## 528 49 4 18 203  
## 529 49 4 20 233  
## 530 49 4 21 237  
## 531 46 4 0 40  
## 532 46 4 2 52  
## 533 46 4 4 62  
## 534 46 4 6 82  
## 535 46 4 8 101  
## 536 46 4 10 120  
## 537 46 4 12 144  
## 538 46 4 14 156  
## 539 46 4 16 173  
## 540 46 4 18 210  
## 541 46 4 20 231  
## 542 46 4 21 238  
## 543 50 4 0 41  
## 544 50 4 2 54  
## 545 50 4 4 67  
## 546 50 4 6 84  
## 547 50 4 8 105  
## 548 50 4 10 122  
## 549 50 4 12 155  
## 550 50 4 14 175  
## 551 50 4 16 205  
## 552 50 4 18 234  
## 553 50 4 20 264  
## 554 50 4 21 264  
## 555 42 4 0 42  
## 556 42 4 2 49  
## 557 42 4 4 63  
## 558 42 4 6 84  
## 559 42 4 8 103  
## 560 42 4 10 126  
## 561 42 4 12 160  
## 562 42 4 14 174  
## 563 42 4 16 204  
## 564 42 4 18 234  
## 565 42 4 20 269  
## 566 42 4 21 281  
## 567 48 4 0 39  
## 568 48 4 2 50  
## 569 48 4 4 62  
## 570 48 4 6 80  
## 571 48 4 8 104  
## 572 48 4 10 125  
## 573 48 4 12 154  
## 574 48 4 14 170  
## 575 48 4 16 222  
## 576 48 4 18 261  
## 577 48 4 20 303  
## 578 48 4 21 322

chick\_clean %>%  
 mutate(weight\_per\_day = weight\_gm/age\_days)

## chick\_id diet age\_days weight\_gm weight\_per\_day  
## 1 1 1 0 42 Inf  
## 2 1 1 2 51 25.500000  
## 3 1 1 4 59 14.750000  
## 4 1 1 6 64 10.666667  
## 5 1 1 8 76 9.500000  
## 6 1 1 10 93 9.300000  
## 7 1 1 12 106 8.833333  
## 8 1 1 14 125 8.928571  
## 9 1 1 16 149 9.312500  
## 10 1 1 18 171 9.500000  
## 11 1 1 20 199 9.950000  
## 12 1 1 21 205 9.761905  
## 13 2 1 0 40 Inf  
## 14 2 1 2 49 24.500000  
## 15 2 1 4 58 14.500000  
## 16 2 1 6 72 12.000000  
## 17 2 1 8 84 10.500000  
## 18 2 1 10 103 10.300000  
## 19 2 1 12 122 10.166667  
## 20 2 1 14 138 9.857143  
## 21 2 1 16 162 10.125000  
## 22 2 1 18 187 10.388889  
## 23 2 1 20 209 10.450000  
## 24 2 1 21 215 10.238095  
## 25 3 1 0 43 Inf  
## 26 3 1 2 39 19.500000  
## 27 3 1 4 55 13.750000  
## 28 3 1 6 67 11.166667  
## 29 3 1 8 84 10.500000  
## 30 3 1 10 99 9.900000  
## 31 3 1 12 115 9.583333  
## 32 3 1 14 138 9.857143  
## 33 3 1 16 163 10.187500  
## 34 3 1 18 187 10.388889  
## 35 3 1 20 198 9.900000  
## 36 3 1 21 202 9.619048  
## 37 4 1 0 42 Inf  
## 38 4 1 2 49 24.500000  
## 39 4 1 4 56 14.000000  
## 40 4 1 6 67 11.166667  
## 41 4 1 8 74 9.250000  
## 42 4 1 10 87 8.700000  
## 43 4 1 12 102 8.500000  
## 44 4 1 14 108 7.714286  
## 45 4 1 16 136 8.500000  
## 46 4 1 18 154 8.555556  
## 47 4 1 20 160 8.000000  
## 48 4 1 21 157 7.476190  
## 49 5 1 0 41 Inf  
## 50 5 1 2 42 21.000000  
## 51 5 1 4 48 12.000000  
## 52 5 1 6 60 10.000000  
## 53 5 1 8 79 9.875000  
## 54 5 1 10 106 10.600000  
## 55 5 1 12 141 11.750000  
## 56 5 1 14 164 11.714286  
## 57 5 1 16 197 12.312500  
## 58 5 1 18 199 11.055556  
## 59 5 1 20 220 11.000000  
## 60 5 1 21 223 10.619048  
## 61 6 1 0 41 Inf  
## 62 6 1 2 49 24.500000  
## 63 6 1 4 59 14.750000  
## 64 6 1 6 74 12.333333  
## 65 6 1 8 97 12.125000  
## 66 6 1 10 124 12.400000  
## 67 6 1 12 141 11.750000  
## 68 6 1 14 148 10.571429  
## 69 6 1 16 155 9.687500  
## 70 6 1 18 160 8.888889  
## 71 6 1 20 160 8.000000  
## 72 6 1 21 157 7.476190  
## 73 7 1 0 41 Inf  
## 74 7 1 2 49 24.500000  
## 75 7 1 4 57 14.250000  
## 76 7 1 6 71 11.833333  
## 77 7 1 8 89 11.125000  
## 78 7 1 10 112 11.200000  
## 79 7 1 12 146 12.166667  
## 80 7 1 14 174 12.428571  
## 81 7 1 16 218 13.625000  
## 82 7 1 18 250 13.888889  
## 83 7 1 20 288 14.400000  
## 84 7 1 21 305 14.523810  
## 85 8 1 0 42 Inf  
## 86 8 1 2 50 25.000000  
## 87 8 1 4 61 15.250000  
## 88 8 1 6 71 11.833333  
## 89 8 1 8 84 10.500000  
## 90 8 1 10 93 9.300000  
## 91 8 1 12 110 9.166667  
## 92 8 1 14 116 8.285714  
## 93 8 1 16 126 7.875000  
## 94 8 1 18 134 7.444444  
## 95 8 1 20 125 6.250000  
## 96 9 1 0 42 Inf  
## 97 9 1 2 51 25.500000  
## 98 9 1 4 59 14.750000  
## 99 9 1 6 68 11.333333  
## 100 9 1 8 85 10.625000  
## 101 9 1 10 96 9.600000  
## 102 9 1 12 90 7.500000  
## 103 9 1 14 92 6.571429  
## 104 9 1 16 93 5.812500  
## 105 9 1 18 100 5.555556  
## 106 9 1 20 100 5.000000  
## 107 9 1 21 98 4.666667  
## 108 10 1 0 41 Inf  
## 109 10 1 2 44 22.000000  
## 110 10 1 4 52 13.000000  
## 111 10 1 6 63 10.500000  
## 112 10 1 8 74 9.250000  
## 113 10 1 10 81 8.100000  
## 114 10 1 12 89 7.416667  
## 115 10 1 14 96 6.857143  
## 116 10 1 16 101 6.312500  
## 117 10 1 18 112 6.222222  
## 118 10 1 20 120 6.000000  
## 119 10 1 21 124 5.904762  
## 120 11 1 0 43 Inf  
## 121 11 1 2 51 25.500000  
## 122 11 1 4 63 15.750000  
## 123 11 1 6 84 14.000000  
## 124 11 1 8 112 14.000000  
## 125 11 1 10 139 13.900000  
## 126 11 1 12 168 14.000000  
## 127 11 1 14 177 12.642857  
## 128 11 1 16 182 11.375000  
## 129 11 1 18 184 10.222222  
## 130 11 1 20 181 9.050000  
## 131 11 1 21 175 8.333333  
## 132 12 1 0 41 Inf  
## 133 12 1 2 49 24.500000  
## 134 12 1 4 56 14.000000  
## 135 12 1 6 62 10.333333  
## 136 12 1 8 72 9.000000  
## 137 12 1 10 88 8.800000  
## 138 12 1 12 119 9.916667  
## 139 12 1 14 135 9.642857  
## 140 12 1 16 162 10.125000  
## 141 12 1 18 185 10.277778  
## 142 12 1 20 195 9.750000  
## 143 12 1 21 205 9.761905  
## 144 13 1 0 41 Inf  
## 145 13 1 2 48 24.000000  
## 146 13 1 4 53 13.250000  
## 147 13 1 6 60 10.000000  
## 148 13 1 8 65 8.125000  
## 149 13 1 10 67 6.700000  
## 150 13 1 12 71 5.916667  
## 151 13 1 14 70 5.000000  
## 152 13 1 16 71 4.437500  
## 153 13 1 18 81 4.500000  
## 154 13 1 20 91 4.550000  
## 155 13 1 21 96 4.571429  
## 156 14 1 0 41 Inf  
## 157 14 1 2 49 24.500000  
## 158 14 1 4 62 15.500000  
## 159 14 1 6 79 13.166667  
## 160 14 1 8 101 12.625000  
## 161 14 1 10 128 12.800000  
## 162 14 1 12 164 13.666667  
## 163 14 1 14 192 13.714286  
## 164 14 1 16 227 14.187500  
## 165 14 1 18 248 13.777778  
## 166 14 1 20 259 12.950000  
## 167 14 1 21 266 12.666667  
## 168 15 1 0 41 Inf  
## 169 15 1 2 49 24.500000  
## 170 15 1 4 56 14.000000  
## 171 15 1 6 64 10.666667  
## 172 15 1 8 68 8.500000  
## 173 15 1 10 68 6.800000  
## 174 15 1 12 67 5.583333  
## 175 15 1 14 68 4.857143  
## 176 16 1 0 41 Inf  
## 177 16 1 2 45 22.500000  
## 178 16 1 4 49 12.250000  
## 179 16 1 6 51 8.500000  
## 180 16 1 8 57 7.125000  
## 181 16 1 10 51 5.100000  
## 182 16 1 12 54 4.500000  
## 183 17 1 0 42 Inf  
## 184 17 1 2 51 25.500000  
## 185 17 1 4 61 15.250000  
## 186 17 1 6 72 12.000000  
## 187 17 1 8 83 10.375000  
## 188 17 1 10 89 8.900000  
## 189 17 1 12 98 8.166667  
## 190 17 1 14 103 7.357143  
## 191 17 1 16 113 7.062500  
## 192 17 1 18 123 6.833333  
## 193 17 1 20 133 6.650000  
## 194 17 1 21 142 6.761905  
## 195 18 1 0 39 Inf  
## 196 18 1 2 35 17.500000  
## 197 19 1 0 43 Inf  
## 198 19 1 2 48 24.000000  
## 199 19 1 4 55 13.750000  
## 200 19 1 6 62 10.333333  
## 201 19 1 8 65 8.125000  
## 202 19 1 10 71 7.100000  
## 203 19 1 12 82 6.833333  
## 204 19 1 14 88 6.285714  
## 205 19 1 16 106 6.625000  
## 206 19 1 18 120 6.666667  
## 207 19 1 20 144 7.200000  
## 208 19 1 21 157 7.476190  
## 209 20 1 0 41 Inf  
## 210 20 1 2 47 23.500000  
## 211 20 1 4 54 13.500000  
## 212 20 1 6 58 9.666667  
## 213 20 1 8 65 8.125000  
## 214 20 1 10 73 7.300000  
## 215 20 1 12 77 6.416667  
## 216 20 1 14 89 6.357143  
## 217 20 1 16 98 6.125000  
## 218 20 1 18 107 5.944444  
## 219 20 1 20 115 5.750000  
## 220 20 1 21 117 5.571429  
## 221 21 2 0 40 Inf  
## 222 21 2 2 50 25.000000  
## 223 21 2 4 62 15.500000  
## 224 21 2 6 86 14.333333  
## 225 21 2 8 125 15.625000  
## 226 21 2 10 163 16.300000  
## 227 21 2 12 217 18.083333  
## 228 21 2 14 240 17.142857  
## 229 21 2 16 275 17.187500  
## 230 21 2 18 307 17.055556  
## 231 21 2 20 318 15.900000  
## 232 21 2 21 331 15.761905  
## 233 22 2 0 41 Inf  
## 234 22 2 2 55 27.500000  
## 235 22 2 4 64 16.000000  
## 236 22 2 6 77 12.833333  
## 237 22 2 8 90 11.250000  
## 238 22 2 10 95 9.500000  
## 239 22 2 12 108 9.000000  
## 240 22 2 14 111 7.928571  
## 241 22 2 16 131 8.187500  
## 242 22 2 18 148 8.222222  
## 243 22 2 20 164 8.200000  
## 244 22 2 21 167 7.952381  
## 245 23 2 0 43 Inf  
## 246 23 2 2 52 26.000000  
## 247 23 2 4 61 15.250000  
## 248 23 2 6 73 12.166667  
## 249 23 2 8 90 11.250000  
## 250 23 2 10 103 10.300000  
## 251 23 2 12 127 10.583333  
## 252 23 2 14 135 9.642857  
## 253 23 2 16 145 9.062500  
## 254 23 2 18 163 9.055556  
## 255 23 2 20 170 8.500000  
## 256 23 2 21 175 8.333333  
## 257 24 2 0 42 Inf  
## 258 24 2 2 52 26.000000  
## 259 24 2 4 58 14.500000  
## 260 24 2 6 74 12.333333  
## 261 24 2 8 66 8.250000  
## 262 24 2 10 68 6.800000  
## 263 24 2 12 70 5.833333  
## 264 24 2 14 71 5.071429  
## 265 24 2 16 72 4.500000  
## 266 24 2 18 72 4.000000  
## 267 24 2 20 76 3.800000  
## 268 24 2 21 74 3.523810  
## 269 25 2 0 40 Inf  
## 270 25 2 2 49 24.500000  
## 271 25 2 4 62 15.500000  
## 272 25 2 6 78 13.000000  
## 273 25 2 8 102 12.750000  
## 274 25 2 10 124 12.400000  
## 275 25 2 12 146 12.166667  
## 276 25 2 14 164 11.714286  
## 277 25 2 16 197 12.312500  
## 278 25 2 18 231 12.833333  
## 279 25 2 20 259 12.950000  
## 280 25 2 21 265 12.619048  
## 281 26 2 0 42 Inf  
## 282 26 2 2 48 24.000000  
## 283 26 2 4 57 14.250000  
## 284 26 2 6 74 12.333333  
## 285 26 2 8 93 11.625000  
## 286 26 2 10 114 11.400000  
## 287 26 2 12 136 11.333333  
## 288 26 2 14 147 10.500000  
## 289 26 2 16 169 10.562500  
## 290 26 2 18 205 11.388889  
## 291 26 2 20 236 11.800000  
## 292 26 2 21 251 11.952381  
## 293 27 2 0 39 Inf  
## 294 27 2 2 46 23.000000  
## 295 27 2 4 58 14.500000  
## 296 27 2 6 73 12.166667  
## 297 27 2 8 87 10.875000  
## 298 27 2 10 100 10.000000  
## 299 27 2 12 115 9.583333  
## 300 27 2 14 123 8.785714  
## 301 27 2 16 144 9.000000  
## 302 27 2 18 163 9.055556  
## 303 27 2 20 185 9.250000  
## 304 27 2 21 192 9.142857  
## 305 28 2 0 39 Inf  
## 306 28 2 2 46 23.000000  
## 307 28 2 4 58 14.500000  
## 308 28 2 6 73 12.166667  
## 309 28 2 8 92 11.500000  
## 310 28 2 10 114 11.400000  
## 311 28 2 12 145 12.083333  
## 312 28 2 14 156 11.142857  
## 313 28 2 16 184 11.500000  
## 314 28 2 18 207 11.500000  
## 315 28 2 20 212 10.600000  
## 316 28 2 21 233 11.095238  
## 317 29 2 0 39 Inf  
## 318 29 2 2 48 24.000000  
## 319 29 2 4 59 14.750000  
## 320 29 2 6 74 12.333333  
## 321 29 2 8 87 10.875000  
## 322 29 2 10 106 10.600000  
## 323 29 2 12 134 11.166667  
## 324 29 2 14 150 10.714286  
## 325 29 2 16 187 11.687500  
## 326 29 2 18 230 12.777778  
## 327 29 2 20 279 13.950000  
## 328 29 2 21 309 14.714286  
## 329 30 2 0 42 Inf  
## 330 30 2 2 48 24.000000  
## 331 30 2 4 59 14.750000  
## 332 30 2 6 72 12.000000  
## 333 30 2 8 85 10.625000  
## 334 30 2 10 98 9.800000  
## 335 30 2 12 115 9.583333  
## 336 30 2 14 122 8.714286  
## 337 30 2 16 143 8.937500  
## 338 30 2 18 151 8.388889  
## 339 30 2 20 157 7.850000  
## 340 30 2 21 150 7.142857  
## 341 31 3 0 42 Inf  
## 342 31 3 2 53 26.500000  
## 343 31 3 4 62 15.500000  
## 344 31 3 6 73 12.166667  
## 345 31 3 8 85 10.625000  
## 346 31 3 10 102 10.200000  
## 347 31 3 12 123 10.250000  
## 348 31 3 14 138 9.857143  
## 349 31 3 16 170 10.625000  
## 350 31 3 18 204 11.333333  
## 351 31 3 20 235 11.750000  
## 352 31 3 21 256 12.190476  
## 353 32 3 0 41 Inf  
## 354 32 3 2 49 24.500000  
## 355 32 3 4 65 16.250000  
## 356 32 3 6 82 13.666667  
## 357 32 3 8 107 13.375000  
## 358 32 3 10 129 12.900000  
## 359 32 3 12 159 13.250000  
## 360 32 3 14 179 12.785714  
## 361 32 3 16 221 13.812500  
## 362 32 3 18 263 14.611111  
## 363 32 3 20 291 14.550000  
## 364 32 3 21 305 14.523810  
## 365 33 3 0 39 Inf  
## 366 33 3 2 50 25.000000  
## 367 33 3 4 63 15.750000  
## 368 33 3 6 77 12.833333  
## 369 33 3 8 96 12.000000  
## 370 33 3 10 111 11.100000  
## 371 33 3 12 137 11.416667  
## 372 33 3 14 144 10.285714  
## 373 33 3 16 151 9.437500  
## 374 33 3 18 146 8.111111  
## 375 33 3 20 156 7.800000  
## 376 33 3 21 147 7.000000  
## 377 34 3 0 41 Inf  
## 378 34 3 2 49 24.500000  
## 379 34 3 4 63 15.750000  
## 380 34 3 6 85 14.166667  
## 381 34 3 8 107 13.375000  
## 382 34 3 10 134 13.400000  
## 383 34 3 12 164 13.666667  
## 384 34 3 14 186 13.285714  
## 385 34 3 16 235 14.687500  
## 386 34 3 18 294 16.333333  
## 387 34 3 20 327 16.350000  
## 388 34 3 21 341 16.238095  
## 389 35 3 0 41 Inf  
## 390 35 3 2 53 26.500000  
## 391 35 3 4 64 16.000000  
## 392 35 3 6 87 14.500000  
## 393 35 3 8 123 15.375000  
## 394 35 3 10 158 15.800000  
## 395 35 3 12 201 16.750000  
## 396 35 3 14 238 17.000000  
## 397 35 3 16 287 17.937500  
## 398 35 3 18 332 18.444444  
## 399 35 3 20 361 18.050000  
## 400 35 3 21 373 17.761905  
## 401 36 3 0 39 Inf  
## 402 36 3 2 48 24.000000  
## 403 36 3 4 61 15.250000  
## 404 36 3 6 76 12.666667  
## 405 36 3 8 98 12.250000  
## 406 36 3 10 116 11.600000  
## 407 36 3 12 145 12.083333  
## 408 36 3 14 166 11.857143  
## 409 36 3 16 198 12.375000  
## 410 36 3 18 227 12.611111  
## 411 36 3 20 225 11.250000  
## 412 36 3 21 220 10.476190  
## 413 37 3 0 41 Inf  
## 414 37 3 2 48 24.000000  
## 415 37 3 4 56 14.000000  
## 416 37 3 6 68 11.333333  
## 417 37 3 8 80 10.000000  
## 418 37 3 10 83 8.300000  
## 419 37 3 12 103 8.583333  
## 420 37 3 14 112 8.000000  
## 421 37 3 16 135 8.437500  
## 422 37 3 18 157 8.722222  
## 423 37 3 20 169 8.450000  
## 424 37 3 21 178 8.476190  
## 425 38 3 0 41 Inf  
## 426 38 3 2 49 24.500000  
## 427 38 3 4 61 15.250000  
## 428 38 3 6 74 12.333333  
## 429 38 3 8 98 12.250000  
## 430 38 3 10 109 10.900000  
## 431 38 3 12 128 10.666667  
## 432 38 3 14 154 11.000000  
## 433 38 3 16 192 12.000000  
## 434 38 3 18 232 12.888889  
## 435 38 3 20 280 14.000000  
## 436 38 3 21 290 13.809524  
## 437 39 3 0 42 Inf  
## 438 39 3 2 50 25.000000  
## 439 39 3 4 61 15.250000  
## 440 39 3 6 78 13.000000  
## 441 39 3 8 89 11.125000  
## 442 39 3 10 109 10.900000  
## 443 39 3 12 130 10.833333  
## 444 39 3 14 146 10.428571  
## 445 39 3 16 170 10.625000  
## 446 39 3 18 214 11.888889  
## 447 39 3 20 250 12.500000  
## 448 39 3 21 272 12.952381  
## 449 40 3 0 41 Inf  
## 450 40 3 2 55 27.500000  
## 451 40 3 4 66 16.500000  
## 452 40 3 6 79 13.166667  
## 453 40 3 8 101 12.625000  
## 454 40 3 10 120 12.000000  
## 455 40 3 12 154 12.833333  
## 456 40 3 14 182 13.000000  
## 457 40 3 16 215 13.437500  
## 458 40 3 18 262 14.555556  
## 459 40 3 20 295 14.750000  
## 460 40 3 21 321 15.285714  
## 461 41 4 0 42 Inf  
## 462 41 4 2 51 25.500000  
## 463 41 4 4 66 16.500000  
## 464 41 4 6 85 14.166667  
## 465 41 4 8 103 12.875000  
## 466 41 4 10 124 12.400000  
## 467 41 4 12 155 12.916667  
## 468 41 4 14 153 10.928571  
## 469 41 4 16 175 10.937500  
## 470 41 4 18 184 10.222222  
## 471 41 4 20 199 9.950000  
## 472 41 4 21 204 9.714286  
## 473 42 4 0 42 Inf  
## 474 42 4 2 49 24.500000  
## 475 42 4 4 63 15.750000  
## 476 42 4 6 84 14.000000  
## 477 42 4 8 103 12.875000  
## 478 42 4 10 126 12.600000  
## 479 42 4 12 160 13.333333  
## 480 42 4 14 174 12.428571  
## 481 42 4 16 204 12.750000  
## 482 42 4 18 234 13.000000  
## 483 42 4 20 269 13.450000  
## 484 42 4 21 281 13.380952  
## 485 43 4 0 42 Inf  
## 486 43 4 2 55 27.500000  
## 487 43 4 4 69 17.250000  
## 488 43 4 6 96 16.000000  
## 489 43 4 8 131 16.375000  
## 490 43 4 10 157 15.700000  
## 491 43 4 12 184 15.333333  
## 492 43 4 14 188 13.428571  
## 493 43 4 16 197 12.312500  
## 494 43 4 18 198 11.000000  
## 495 43 4 20 199 9.950000  
## 496 43 4 21 200 9.523810  
## 497 44 4 0 42 Inf  
## 498 44 4 2 51 25.500000  
## 499 44 4 4 65 16.250000  
## 500 44 4 6 86 14.333333  
## 501 44 4 8 103 12.875000  
## 502 44 4 10 118 11.800000  
## 503 44 4 12 127 10.583333  
## 504 44 4 14 138 9.857143  
## 505 44 4 16 145 9.062500  
## 506 44 4 18 146 8.111111  
## 507 45 4 0 41 Inf  
## 508 45 4 2 50 25.000000  
## 509 45 4 4 61 15.250000  
## 510 45 4 6 78 13.000000  
## 511 45 4 8 98 12.250000  
## 512 45 4 10 117 11.700000  
## 513 45 4 12 135 11.250000  
## 514 45 4 14 141 10.071429  
## 515 45 4 16 147 9.187500  
## 516 45 4 18 174 9.666667  
## 517 45 4 20 197 9.850000  
## 518 45 4 21 196 9.333333  
## 519 46 4 0 40 Inf  
## 520 46 4 2 52 26.000000  
## 521 46 4 4 62 15.500000  
## 522 46 4 6 82 13.666667  
## 523 46 4 8 101 12.625000  
## 524 46 4 10 120 12.000000  
## 525 46 4 12 144 12.000000  
## 526 46 4 14 156 11.142857  
## 527 46 4 16 173 10.812500  
## 528 46 4 18 210 11.666667  
## 529 46 4 20 231 11.550000  
## 530 46 4 21 238 11.333333  
## 531 47 4 0 41 Inf  
## 532 47 4 2 53 26.500000  
## 533 47 4 4 66 16.500000  
## 534 47 4 6 79 13.166667  
## 535 47 4 8 100 12.500000  
## 536 47 4 10 123 12.300000  
## 537 47 4 12 148 12.333333  
## 538 47 4 14 157 11.214286  
## 539 47 4 16 168 10.500000  
## 540 47 4 18 185 10.277778  
## 541 47 4 20 210 10.500000  
## 542 47 4 21 205 9.761905  
## 543 48 4 0 39 Inf  
## 544 48 4 2 50 25.000000  
## 545 48 4 4 62 15.500000  
## 546 48 4 6 80 13.333333  
## 547 48 4 8 104 13.000000  
## 548 48 4 10 125 12.500000  
## 549 48 4 12 154 12.833333  
## 550 48 4 14 170 12.142857  
## 551 48 4 16 222 13.875000  
## 552 48 4 18 261 14.500000  
## 553 48 4 20 303 15.150000  
## 554 48 4 21 322 15.333333  
## 555 49 4 0 40 Inf  
## 556 49 4 2 53 26.500000  
## 557 49 4 4 64 16.000000  
## 558 49 4 6 85 14.166667  
## 559 49 4 8 108 13.500000  
## 560 49 4 10 128 12.800000  
## 561 49 4 12 152 12.666667  
## 562 49 4 14 166 11.857143  
## 563 49 4 16 184 11.500000  
## 564 49 4 18 203 11.277778  
## 565 49 4 20 233 11.650000  
## 566 49 4 21 237 11.285714  
## 567 50 4 0 41 Inf  
## 568 50 4 2 54 27.000000  
## 569 50 4 4 67 16.750000  
## 570 50 4 6 84 14.000000  
## 571 50 4 8 105 13.125000  
## 572 50 4 10 122 12.200000  
## 573 50 4 12 155 12.916667  
## 574 50 4 14 175 12.500000  
## 575 50 4 16 205 12.812500  
## 576 50 4 18 234 13.000000  
## 577 50 4 20 264 13.200000  
## 578 50 4 21 264 12.571429

chick\_clean %>%  
 mutate(weight\_lbs = weight\_gm/453.6) #mutate will always add to end of dataset

## chick\_id diet age\_days weight\_gm weight\_lbs  
## 1 1 1 0 42 0.09259259  
## 2 1 1 2 51 0.11243386  
## 3 1 1 4 59 0.13007055  
## 4 1 1 6 64 0.14109347  
## 5 1 1 8 76 0.16754850  
## 6 1 1 10 93 0.20502646  
## 7 1 1 12 106 0.23368607  
## 8 1 1 14 125 0.27557319  
## 9 1 1 16 149 0.32848325  
## 10 1 1 18 171 0.37698413  
## 11 1 1 20 199 0.43871252  
## 12 1 1 21 205 0.45194004  
## 13 2 1 0 40 0.08818342  
## 14 2 1 2 49 0.10802469  
## 15 2 1 4 58 0.12786596  
## 16 2 1 6 72 0.15873016  
## 17 2 1 8 84 0.18518519  
## 18 2 1 10 103 0.22707231  
## 19 2 1 12 122 0.26895944  
## 20 2 1 14 138 0.30423280  
## 21 2 1 16 162 0.35714286  
## 22 2 1 18 187 0.41225750  
## 23 2 1 20 209 0.46075838  
## 24 2 1 21 215 0.47398589  
## 25 3 1 0 43 0.09479718  
## 26 3 1 2 39 0.08597884  
## 27 3 1 4 55 0.12125220  
## 28 3 1 6 67 0.14770723  
## 29 3 1 8 84 0.18518519  
## 30 3 1 10 99 0.21825397  
## 31 3 1 12 115 0.25352734  
## 32 3 1 14 138 0.30423280  
## 33 3 1 16 163 0.35934744  
## 34 3 1 18 187 0.41225750  
## 35 3 1 20 198 0.43650794  
## 36 3 1 21 202 0.44532628  
## 37 4 1 0 42 0.09259259  
## 38 4 1 2 49 0.10802469  
## 39 4 1 4 56 0.12345679  
## 40 4 1 6 67 0.14770723  
## 41 4 1 8 74 0.16313933  
## 42 4 1 10 87 0.19179894  
## 43 4 1 12 102 0.22486772  
## 44 4 1 14 108 0.23809524  
## 45 4 1 16 136 0.29982363  
## 46 4 1 18 154 0.33950617  
## 47 4 1 20 160 0.35273369  
## 48 4 1 21 157 0.34611993  
## 49 5 1 0 41 0.09038801  
## 50 5 1 2 42 0.09259259  
## 51 5 1 4 48 0.10582011  
## 52 5 1 6 60 0.13227513  
## 53 5 1 8 79 0.17416226  
## 54 5 1 10 106 0.23368607  
## 55 5 1 12 141 0.31084656  
## 56 5 1 14 164 0.36155203  
## 57 5 1 16 197 0.43430335  
## 58 5 1 18 199 0.43871252  
## 59 5 1 20 220 0.48500882  
## 60 5 1 21 223 0.49162257  
## 61 6 1 0 41 0.09038801  
## 62 6 1 2 49 0.10802469  
## 63 6 1 4 59 0.13007055  
## 64 6 1 6 74 0.16313933  
## 65 6 1 8 97 0.21384480  
## 66 6 1 10 124 0.27336861  
## 67 6 1 12 141 0.31084656  
## 68 6 1 14 148 0.32627866  
## 69 6 1 16 155 0.34171076  
## 70 6 1 18 160 0.35273369  
## 71 6 1 20 160 0.35273369  
## 72 6 1 21 157 0.34611993  
## 73 7 1 0 41 0.09038801  
## 74 7 1 2 49 0.10802469  
## 75 7 1 4 57 0.12566138  
## 76 7 1 6 71 0.15652557  
## 77 7 1 8 89 0.19620811  
## 78 7 1 10 112 0.24691358  
## 79 7 1 12 146 0.32186949  
## 80 7 1 14 174 0.38359788  
## 81 7 1 16 218 0.48059965  
## 82 7 1 18 250 0.55114638  
## 83 7 1 20 288 0.63492063  
## 84 7 1 21 305 0.67239859  
## 85 8 1 0 42 0.09259259  
## 86 8 1 2 50 0.11022928  
## 87 8 1 4 61 0.13447972  
## 88 8 1 6 71 0.15652557  
## 89 8 1 8 84 0.18518519  
## 90 8 1 10 93 0.20502646  
## 91 8 1 12 110 0.24250441  
## 92 8 1 14 116 0.25573192  
## 93 8 1 16 126 0.27777778  
## 94 8 1 18 134 0.29541446  
## 95 8 1 20 125 0.27557319  
## 96 9 1 0 42 0.09259259  
## 97 9 1 2 51 0.11243386  
## 98 9 1 4 59 0.13007055  
## 99 9 1 6 68 0.14991182  
## 100 9 1 8 85 0.18738977  
## 101 9 1 10 96 0.21164021  
## 102 9 1 12 90 0.19841270  
## 103 9 1 14 92 0.20282187  
## 104 9 1 16 93 0.20502646  
## 105 9 1 18 100 0.22045855  
## 106 9 1 20 100 0.22045855  
## 107 9 1 21 98 0.21604938  
## 108 10 1 0 41 0.09038801  
## 109 10 1 2 44 0.09700176  
## 110 10 1 4 52 0.11463845  
## 111 10 1 6 63 0.13888889  
## 112 10 1 8 74 0.16313933  
## 113 10 1 10 81 0.17857143  
## 114 10 1 12 89 0.19620811  
## 115 10 1 14 96 0.21164021  
## 116 10 1 16 101 0.22266314  
## 117 10 1 18 112 0.24691358  
## 118 10 1 20 120 0.26455026  
## 119 10 1 21 124 0.27336861  
## 120 11 1 0 43 0.09479718  
## 121 11 1 2 51 0.11243386  
## 122 11 1 4 63 0.13888889  
## 123 11 1 6 84 0.18518519  
## 124 11 1 8 112 0.24691358  
## 125 11 1 10 139 0.30643739  
## 126 11 1 12 168 0.37037037  
## 127 11 1 14 177 0.39021164  
## 128 11 1 16 182 0.40123457  
## 129 11 1 18 184 0.40564374  
## 130 11 1 20 181 0.39902998  
## 131 11 1 21 175 0.38580247  
## 132 12 1 0 41 0.09038801  
## 133 12 1 2 49 0.10802469  
## 134 12 1 4 56 0.12345679  
## 135 12 1 6 62 0.13668430  
## 136 12 1 8 72 0.15873016  
## 137 12 1 10 88 0.19400353  
## 138 12 1 12 119 0.26234568  
## 139 12 1 14 135 0.29761905  
## 140 12 1 16 162 0.35714286  
## 141 12 1 18 185 0.40784832  
## 142 12 1 20 195 0.42989418  
## 143 12 1 21 205 0.45194004  
## 144 13 1 0 41 0.09038801  
## 145 13 1 2 48 0.10582011  
## 146 13 1 4 53 0.11684303  
## 147 13 1 6 60 0.13227513  
## 148 13 1 8 65 0.14329806  
## 149 13 1 10 67 0.14770723  
## 150 13 1 12 71 0.15652557  
## 151 13 1 14 70 0.15432099  
## 152 13 1 16 71 0.15652557  
## 153 13 1 18 81 0.17857143  
## 154 13 1 20 91 0.20061728  
## 155 13 1 21 96 0.21164021  
## 156 14 1 0 41 0.09038801  
## 157 14 1 2 49 0.10802469  
## 158 14 1 4 62 0.13668430  
## 159 14 1 6 79 0.17416226  
## 160 14 1 8 101 0.22266314  
## 161 14 1 10 128 0.28218695  
## 162 14 1 12 164 0.36155203  
## 163 14 1 14 192 0.42328042  
## 164 14 1 16 227 0.50044092  
## 165 14 1 18 248 0.54673721  
## 166 14 1 20 259 0.57098765  
## 167 14 1 21 266 0.58641975  
## 168 15 1 0 41 0.09038801  
## 169 15 1 2 49 0.10802469  
## 170 15 1 4 56 0.12345679  
## 171 15 1 6 64 0.14109347  
## 172 15 1 8 68 0.14991182  
## 173 15 1 10 68 0.14991182  
## 174 15 1 12 67 0.14770723  
## 175 15 1 14 68 0.14991182  
## 176 16 1 0 41 0.09038801  
## 177 16 1 2 45 0.09920635  
## 178 16 1 4 49 0.10802469  
## 179 16 1 6 51 0.11243386  
## 180 16 1 8 57 0.12566138  
## 181 16 1 10 51 0.11243386  
## 182 16 1 12 54 0.11904762  
## 183 17 1 0 42 0.09259259  
## 184 17 1 2 51 0.11243386  
## 185 17 1 4 61 0.13447972  
## 186 17 1 6 72 0.15873016  
## 187 17 1 8 83 0.18298060  
## 188 17 1 10 89 0.19620811  
## 189 17 1 12 98 0.21604938  
## 190 17 1 14 103 0.22707231  
## 191 17 1 16 113 0.24911817  
## 192 17 1 18 123 0.27116402  
## 193 17 1 20 133 0.29320988  
## 194 17 1 21 142 0.31305115  
## 195 18 1 0 39 0.08597884  
## 196 18 1 2 35 0.07716049  
## 197 19 1 0 43 0.09479718  
## 198 19 1 2 48 0.10582011  
## 199 19 1 4 55 0.12125220  
## 200 19 1 6 62 0.13668430  
## 201 19 1 8 65 0.14329806  
## 202 19 1 10 71 0.15652557  
## 203 19 1 12 82 0.18077601  
## 204 19 1 14 88 0.19400353  
## 205 19 1 16 106 0.23368607  
## 206 19 1 18 120 0.26455026  
## 207 19 1 20 144 0.31746032  
## 208 19 1 21 157 0.34611993  
## 209 20 1 0 41 0.09038801  
## 210 20 1 2 47 0.10361552  
## 211 20 1 4 54 0.11904762  
## 212 20 1 6 58 0.12786596  
## 213 20 1 8 65 0.14329806  
## 214 20 1 10 73 0.16093474  
## 215 20 1 12 77 0.16975309  
## 216 20 1 14 89 0.19620811  
## 217 20 1 16 98 0.21604938  
## 218 20 1 18 107 0.23589065  
## 219 20 1 20 115 0.25352734  
## 220 20 1 21 117 0.25793651  
## 221 21 2 0 40 0.08818342  
## 222 21 2 2 50 0.11022928  
## 223 21 2 4 62 0.13668430  
## 224 21 2 6 86 0.18959436  
## 225 21 2 8 125 0.27557319  
## 226 21 2 10 163 0.35934744  
## 227 21 2 12 217 0.47839506  
## 228 21 2 14 240 0.52910053  
## 229 21 2 16 275 0.60626102  
## 230 21 2 18 307 0.67680776  
## 231 21 2 20 318 0.70105820  
## 232 21 2 21 331 0.72971781  
## 233 22 2 0 41 0.09038801  
## 234 22 2 2 55 0.12125220  
## 235 22 2 4 64 0.14109347  
## 236 22 2 6 77 0.16975309  
## 237 22 2 8 90 0.19841270  
## 238 22 2 10 95 0.20943563  
## 239 22 2 12 108 0.23809524  
## 240 22 2 14 111 0.24470899  
## 241 22 2 16 131 0.28880071  
## 242 22 2 18 148 0.32627866  
## 243 22 2 20 164 0.36155203  
## 244 22 2 21 167 0.36816578  
## 245 23 2 0 43 0.09479718  
## 246 23 2 2 52 0.11463845  
## 247 23 2 4 61 0.13447972  
## 248 23 2 6 73 0.16093474  
## 249 23 2 8 90 0.19841270  
## 250 23 2 10 103 0.22707231  
## 251 23 2 12 127 0.27998236  
## 252 23 2 14 135 0.29761905  
## 253 23 2 16 145 0.31966490  
## 254 23 2 18 163 0.35934744  
## 255 23 2 20 170 0.37477954  
## 256 23 2 21 175 0.38580247  
## 257 24 2 0 42 0.09259259  
## 258 24 2 2 52 0.11463845  
## 259 24 2 4 58 0.12786596  
## 260 24 2 6 74 0.16313933  
## 261 24 2 8 66 0.14550265  
## 262 24 2 10 68 0.14991182  
## 263 24 2 12 70 0.15432099  
## 264 24 2 14 71 0.15652557  
## 265 24 2 16 72 0.15873016  
## 266 24 2 18 72 0.15873016  
## 267 24 2 20 76 0.16754850  
## 268 24 2 21 74 0.16313933  
## 269 25 2 0 40 0.08818342  
## 270 25 2 2 49 0.10802469  
## 271 25 2 4 62 0.13668430  
## 272 25 2 6 78 0.17195767  
## 273 25 2 8 102 0.22486772  
## 274 25 2 10 124 0.27336861  
## 275 25 2 12 146 0.32186949  
## 276 25 2 14 164 0.36155203  
## 277 25 2 16 197 0.43430335  
## 278 25 2 18 231 0.50925926  
## 279 25 2 20 259 0.57098765  
## 280 25 2 21 265 0.58421517  
## 281 26 2 0 42 0.09259259  
## 282 26 2 2 48 0.10582011  
## 283 26 2 4 57 0.12566138  
## 284 26 2 6 74 0.16313933  
## 285 26 2 8 93 0.20502646  
## 286 26 2 10 114 0.25132275  
## 287 26 2 12 136 0.29982363  
## 288 26 2 14 147 0.32407407  
## 289 26 2 16 169 0.37257496  
## 290 26 2 18 205 0.45194004  
## 291 26 2 20 236 0.52028219  
## 292 26 2 21 251 0.55335097  
## 293 27 2 0 39 0.08597884  
## 294 27 2 2 46 0.10141093  
## 295 27 2 4 58 0.12786596  
## 296 27 2 6 73 0.16093474  
## 297 27 2 8 87 0.19179894  
## 298 27 2 10 100 0.22045855  
## 299 27 2 12 115 0.25352734  
## 300 27 2 14 123 0.27116402  
## 301 27 2 16 144 0.31746032  
## 302 27 2 18 163 0.35934744  
## 303 27 2 20 185 0.40784832  
## 304 27 2 21 192 0.42328042  
## 305 28 2 0 39 0.08597884  
## 306 28 2 2 46 0.10141093  
## 307 28 2 4 58 0.12786596  
## 308 28 2 6 73 0.16093474  
## 309 28 2 8 92 0.20282187  
## 310 28 2 10 114 0.25132275  
## 311 28 2 12 145 0.31966490  
## 312 28 2 14 156 0.34391534  
## 313 28 2 16 184 0.40564374  
## 314 28 2 18 207 0.45634921  
## 315 28 2 20 212 0.46737213  
## 316 28 2 21 233 0.51366843  
## 317 29 2 0 39 0.08597884  
## 318 29 2 2 48 0.10582011  
## 319 29 2 4 59 0.13007055  
## 320 29 2 6 74 0.16313933  
## 321 29 2 8 87 0.19179894  
## 322 29 2 10 106 0.23368607  
## 323 29 2 12 134 0.29541446  
## 324 29 2 14 150 0.33068783  
## 325 29 2 16 187 0.41225750  
## 326 29 2 18 230 0.50705467  
## 327 29 2 20 279 0.61507937  
## 328 29 2 21 309 0.68121693  
## 329 30 2 0 42 0.09259259  
## 330 30 2 2 48 0.10582011  
## 331 30 2 4 59 0.13007055  
## 332 30 2 6 72 0.15873016  
## 333 30 2 8 85 0.18738977  
## 334 30 2 10 98 0.21604938  
## 335 30 2 12 115 0.25352734  
## 336 30 2 14 122 0.26895944  
## 337 30 2 16 143 0.31525573  
## 338 30 2 18 151 0.33289242  
## 339 30 2 20 157 0.34611993  
## 340 30 2 21 150 0.33068783  
## 341 31 3 0 42 0.09259259  
## 342 31 3 2 53 0.11684303  
## 343 31 3 4 62 0.13668430  
## 344 31 3 6 73 0.16093474  
## 345 31 3 8 85 0.18738977  
## 346 31 3 10 102 0.22486772  
## 347 31 3 12 123 0.27116402  
## 348 31 3 14 138 0.30423280  
## 349 31 3 16 170 0.37477954  
## 350 31 3 18 204 0.44973545  
## 351 31 3 20 235 0.51807760  
## 352 31 3 21 256 0.56437390  
## 353 32 3 0 41 0.09038801  
## 354 32 3 2 49 0.10802469  
## 355 32 3 4 65 0.14329806  
## 356 32 3 6 82 0.18077601  
## 357 32 3 8 107 0.23589065  
## 358 32 3 10 129 0.28439153  
## 359 32 3 12 159 0.35052910  
## 360 32 3 14 179 0.39462081  
## 361 32 3 16 221 0.48721340  
## 362 32 3 18 263 0.57980600  
## 363 32 3 20 291 0.64153439  
## 364 32 3 21 305 0.67239859  
## 365 33 3 0 39 0.08597884  
## 366 33 3 2 50 0.11022928  
## 367 33 3 4 63 0.13888889  
## 368 33 3 6 77 0.16975309  
## 369 33 3 8 96 0.21164021  
## 370 33 3 10 111 0.24470899  
## 371 33 3 12 137 0.30202822  
## 372 33 3 14 144 0.31746032  
## 373 33 3 16 151 0.33289242  
## 374 33 3 18 146 0.32186949  
## 375 33 3 20 156 0.34391534  
## 376 33 3 21 147 0.32407407  
## 377 34 3 0 41 0.09038801  
## 378 34 3 2 49 0.10802469  
## 379 34 3 4 63 0.13888889  
## 380 34 3 6 85 0.18738977  
## 381 34 3 8 107 0.23589065  
## 382 34 3 10 134 0.29541446  
## 383 34 3 12 164 0.36155203  
## 384 34 3 14 186 0.41005291  
## 385 34 3 16 235 0.51807760  
## 386 34 3 18 294 0.64814815  
## 387 34 3 20 327 0.72089947  
## 388 34 3 21 341 0.75176367  
## 389 35 3 0 41 0.09038801  
## 390 35 3 2 53 0.11684303  
## 391 35 3 4 64 0.14109347  
## 392 35 3 6 87 0.19179894  
## 393 35 3 8 123 0.27116402  
## 394 35 3 10 158 0.34832451  
## 395 35 3 12 201 0.44312169  
## 396 35 3 14 238 0.52469136  
## 397 35 3 16 287 0.63271605  
## 398 35 3 18 332 0.73192240  
## 399 35 3 20 361 0.79585538  
## 400 35 3 21 373 0.82231041  
## 401 36 3 0 39 0.08597884  
## 402 36 3 2 48 0.10582011  
## 403 36 3 4 61 0.13447972  
## 404 36 3 6 76 0.16754850  
## 405 36 3 8 98 0.21604938  
## 406 36 3 10 116 0.25573192  
## 407 36 3 12 145 0.31966490  
## 408 36 3 14 166 0.36596120  
## 409 36 3 16 198 0.43650794  
## 410 36 3 18 227 0.50044092  
## 411 36 3 20 225 0.49603175  
## 412 36 3 21 220 0.48500882  
## 413 37 3 0 41 0.09038801  
## 414 37 3 2 48 0.10582011  
## 415 37 3 4 56 0.12345679  
## 416 37 3 6 68 0.14991182  
## 417 37 3 8 80 0.17636684  
## 418 37 3 10 83 0.18298060  
## 419 37 3 12 103 0.22707231  
## 420 37 3 14 112 0.24691358  
## 421 37 3 16 135 0.29761905  
## 422 37 3 18 157 0.34611993  
## 423 37 3 20 169 0.37257496  
## 424 37 3 21 178 0.39241623  
## 425 38 3 0 41 0.09038801  
## 426 38 3 2 49 0.10802469  
## 427 38 3 4 61 0.13447972  
## 428 38 3 6 74 0.16313933  
## 429 38 3 8 98 0.21604938  
## 430 38 3 10 109 0.24029982  
## 431 38 3 12 128 0.28218695  
## 432 38 3 14 154 0.33950617  
## 433 38 3 16 192 0.42328042  
## 434 38 3 18 232 0.51146384  
## 435 38 3 20 280 0.61728395  
## 436 38 3 21 290 0.63932981  
## 437 39 3 0 42 0.09259259  
## 438 39 3 2 50 0.11022928  
## 439 39 3 4 61 0.13447972  
## 440 39 3 6 78 0.17195767  
## 441 39 3 8 89 0.19620811  
## 442 39 3 10 109 0.24029982  
## 443 39 3 12 130 0.28659612  
## 444 39 3 14 146 0.32186949  
## 445 39 3 16 170 0.37477954  
## 446 39 3 18 214 0.47178131  
## 447 39 3 20 250 0.55114638  
## 448 39 3 21 272 0.59964727  
## 449 40 3 0 41 0.09038801  
## 450 40 3 2 55 0.12125220  
## 451 40 3 4 66 0.14550265  
## 452 40 3 6 79 0.17416226  
## 453 40 3 8 101 0.22266314  
## 454 40 3 10 120 0.26455026  
## 455 40 3 12 154 0.33950617  
## 456 40 3 14 182 0.40123457  
## 457 40 3 16 215 0.47398589  
## 458 40 3 18 262 0.57760141  
## 459 40 3 20 295 0.65035273  
## 460 40 3 21 321 0.70767196  
## 461 41 4 0 42 0.09259259  
## 462 41 4 2 51 0.11243386  
## 463 41 4 4 66 0.14550265  
## 464 41 4 6 85 0.18738977  
## 465 41 4 8 103 0.22707231  
## 466 41 4 10 124 0.27336861  
## 467 41 4 12 155 0.34171076  
## 468 41 4 14 153 0.33730159  
## 469 41 4 16 175 0.38580247  
## 470 41 4 18 184 0.40564374  
## 471 41 4 20 199 0.43871252  
## 472 41 4 21 204 0.44973545  
## 473 42 4 0 42 0.09259259  
## 474 42 4 2 49 0.10802469  
## 475 42 4 4 63 0.13888889  
## 476 42 4 6 84 0.18518519  
## 477 42 4 8 103 0.22707231  
## 478 42 4 10 126 0.27777778  
## 479 42 4 12 160 0.35273369  
## 480 42 4 14 174 0.38359788  
## 481 42 4 16 204 0.44973545  
## 482 42 4 18 234 0.51587302  
## 483 42 4 20 269 0.59303351  
## 484 42 4 21 281 0.61948854  
## 485 43 4 0 42 0.09259259  
## 486 43 4 2 55 0.12125220  
## 487 43 4 4 69 0.15211640  
## 488 43 4 6 96 0.21164021  
## 489 43 4 8 131 0.28880071  
## 490 43 4 10 157 0.34611993  
## 491 43 4 12 184 0.40564374  
## 492 43 4 14 188 0.41446208  
## 493 43 4 16 197 0.43430335  
## 494 43 4 18 198 0.43650794  
## 495 43 4 20 199 0.43871252  
## 496 43 4 21 200 0.44091711  
## 497 44 4 0 42 0.09259259  
## 498 44 4 2 51 0.11243386  
## 499 44 4 4 65 0.14329806  
## 500 44 4 6 86 0.18959436  
## 501 44 4 8 103 0.22707231  
## 502 44 4 10 118 0.26014109  
## 503 44 4 12 127 0.27998236  
## 504 44 4 14 138 0.30423280  
## 505 44 4 16 145 0.31966490  
## 506 44 4 18 146 0.32186949  
## 507 45 4 0 41 0.09038801  
## 508 45 4 2 50 0.11022928  
## 509 45 4 4 61 0.13447972  
## 510 45 4 6 78 0.17195767  
## 511 45 4 8 98 0.21604938  
## 512 45 4 10 117 0.25793651  
## 513 45 4 12 135 0.29761905  
## 514 45 4 14 141 0.31084656  
## 515 45 4 16 147 0.32407407  
## 516 45 4 18 174 0.38359788  
## 517 45 4 20 197 0.43430335  
## 518 45 4 21 196 0.43209877  
## 519 46 4 0 40 0.08818342  
## 520 46 4 2 52 0.11463845  
## 521 46 4 4 62 0.13668430  
## 522 46 4 6 82 0.18077601  
## 523 46 4 8 101 0.22266314  
## 524 46 4 10 120 0.26455026  
## 525 46 4 12 144 0.31746032  
## 526 46 4 14 156 0.34391534  
## 527 46 4 16 173 0.38139330  
## 528 46 4 18 210 0.46296296  
## 529 46 4 20 231 0.50925926  
## 530 46 4 21 238 0.52469136  
## 531 47 4 0 41 0.09038801  
## 532 47 4 2 53 0.11684303  
## 533 47 4 4 66 0.14550265  
## 534 47 4 6 79 0.17416226  
## 535 47 4 8 100 0.22045855  
## 536 47 4 10 123 0.27116402  
## 537 47 4 12 148 0.32627866  
## 538 47 4 14 157 0.34611993  
## 539 47 4 16 168 0.37037037  
## 540 47 4 18 185 0.40784832  
## 541 47 4 20 210 0.46296296  
## 542 47 4 21 205 0.45194004  
## 543 48 4 0 39 0.08597884  
## 544 48 4 2 50 0.11022928  
## 545 48 4 4 62 0.13668430  
## 546 48 4 6 80 0.17636684  
## 547 48 4 8 104 0.22927690  
## 548 48 4 10 125 0.27557319  
## 549 48 4 12 154 0.33950617  
## 550 48 4 14 170 0.37477954  
## 551 48 4 16 222 0.48941799  
## 552 48 4 18 261 0.57539683  
## 553 48 4 20 303 0.66798942  
## 554 48 4 21 322 0.70987654  
## 555 49 4 0 40 0.08818342  
## 556 49 4 2 53 0.11684303  
## 557 49 4 4 64 0.14109347  
## 558 49 4 6 85 0.18738977  
## 559 49 4 8 108 0.23809524  
## 560 49 4 10 128 0.28218695  
## 561 49 4 12 152 0.33509700  
## 562 49 4 14 166 0.36596120  
## 563 49 4 16 184 0.40564374  
## 564 49 4 18 203 0.44753086  
## 565 49 4 20 233 0.51366843  
## 566 49 4 21 237 0.52248677  
## 567 50 4 0 41 0.09038801  
## 568 50 4 2 54 0.11904762  
## 569 50 4 4 67 0.14770723  
## 570 50 4 6 84 0.18518519  
## 571 50 4 8 105 0.23148148  
## 572 50 4 10 122 0.26895944  
## 573 50 4 12 155 0.34171076  
## 574 50 4 14 175 0.38580247  
## 575 50 4 16 205 0.45194004  
## 576 50 4 18 234 0.51587302  
## 577 50 4 20 264 0.58201058  
## 578 50 4 21 264 0.58201058

# doubles will go to infinite within computer bounds

42/453.6

## [1] 0.09259259

chick\_clean %>%  
 mutate(weight\_lbs = weight\_gm/453.6) %>%  
 group\_by(diet, age\_days) %>%  
 summarize(n = n(),  
 mean\_wgt\_gm = mean(weight\_gm,),  
 mean\_wgt\_lbs =mean(weight\_lbs)) %>%  
 filter(age\_days == 20)

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

## # A tibble: 4 × 5  
## # Groups: diet [4]  
## diet age\_days n mean\_wgt\_gm mean\_wgt\_lbs  
## <fct> <dbl> <int> <dbl> <dbl>  
## 1 1 20 17 170. 0.376  
## 2 2 20 10 206. 0.453  
## 3 3 20 10 259. 0.571  
## 4 4 20 9 234. 0.516

3 things: for each diet group number of chicks, avg gm weight, avg lbs weight on day 20

chick\_clean %>%  
 filter(age\_days == 20) %>%  
 mutate(weight\_lbs = weight\_gm/453.6) %>%  
 group\_by(diet) %>%  
 summarize(n = n(),  
 mean\_wgt\_gm = mean(weight\_gm,),  
 mean\_wgt\_lbs = mean(weight\_lbs))

## # A tibble: 4 × 4  
## diet n mean\_wgt\_gm mean\_wgt\_lbs  
## <fct> <int> <dbl> <dbl>  
## 1 1 17 170. 0.376  
## 2 2 10 206. 0.453  
## 3 3 10 259. 0.571  
## 4 4 9 234. 0.516

mtcars\_modified <- read\_csv("mtcars\_fromR\_rownames.csv")

## Rows: 32 Columns: 12  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): rowname  
## dbl (11): mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb  
##   
## 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.

glimpse(mtcars\_modified)

## Rows: 32  
## Columns: 12  
## $ rowname <chr> "Mazda RX4", "Mazda RX4 Wag", "Datsun 710", "Hornet 4 Drive", …  
## $ mpg <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.8, 19.2, 17…  
## $ cyl <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8, 8, 8, 8, 8, 8, 4, 4, 4, 4,…  
## $ disp <dbl> 160.0, 160.0, 108.0, 258.0, 360.0, 225.0, 360.0, 146.7, 140.8,…  
## $ hp <dbl> 110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123, 180, 180, …  
## $ drat <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.92, 3.92, 3.…  
## $ wt <dbl> 2.620, 2.875, 2.320, 3.215, 3.440, 3.460, 3.570, 3.190, 3.150,…  
## $ qsec <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 20.22, 15.84, 20.00, 22.90,…  
## $ vs <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,…  
## $ am <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0,…  
## $ gear <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 4, 4, 4, 3,…  
## $ carb <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3, 3, 3, 4, 4, 4, 1, 2, 1, 1,…

# ICMA 9/23/2024

billboard

## # A tibble: 317 × 79  
## artist track date.entered wk1 wk2 wk3 wk4 wk5 wk6 wk7 wk8  
## <chr> <chr> <date> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 Pac Baby… 2000-02-26 87 82 72 77 87 94 99 NA  
## 2 2Ge+her The … 2000-09-02 91 87 92 NA NA NA NA NA  
## 3 3 Doors D… Kryp… 2000-04-08 81 70 68 67 66 57 54 53  
## 4 3 Doors D… Loser 2000-10-21 76 76 72 69 67 65 55 59  
## 5 504 Boyz Wobb… 2000-04-15 57 34 25 17 17 31 36 49  
## 6 98^0 Give… 2000-08-19 51 39 34 26 26 19 2 2  
## 7 A\*Teens Danc… 2000-07-08 97 97 96 95 100 NA NA NA  
## 8 Aaliyah I Do… 2000-01-29 84 62 51 41 38 35 35 38  
## 9 Aaliyah Try … 2000-03-18 59 53 38 28 21 18 16 14  
## 10 Adams, Yo… Open… 2000-08-26 76 76 74 69 68 67 61 58  
## # ℹ 307 more rows  
## # ℹ 68 more variables: wk9 <dbl>, wk10 <dbl>, wk11 <dbl>, wk12 <dbl>,  
## # wk13 <dbl>, wk14 <dbl>, wk15 <dbl>, wk16 <dbl>, wk17 <dbl>, wk18 <dbl>,  
## # wk19 <dbl>, wk20 <dbl>, wk21 <dbl>, wk22 <dbl>, wk23 <dbl>, wk24 <dbl>,  
## # wk25 <dbl>, wk26 <dbl>, wk27 <dbl>, wk28 <dbl>, wk29 <dbl>, wk30 <dbl>,  
## # wk31 <dbl>, wk32 <dbl>, wk33 <dbl>, wk34 <dbl>, wk35 <dbl>, wk36 <dbl>,  
## # wk37 <dbl>, wk38 <dbl>, wk39 <dbl>, wk40 <dbl>, wk41 <dbl>, wk42 <dbl>, …

billboard %>%   
 pivot\_longer(wk1:wk76)

## # A tibble: 24,092 × 5  
## artist track date.entered name value  
## <chr> <chr> <date> <chr> <dbl>  
## 1 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk1 87  
## 2 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk2 82  
## 3 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk3 72  
## 4 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk4 77  
## 5 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk5 87  
## 6 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk6 94  
## 7 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk7 99  
## 8 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk8 NA  
## 9 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk9 NA  
## 10 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk10 NA  
## # ℹ 24,082 more rows

billboard\_longer <- billboard %>%  
 pivot\_longer(cols = starts\_with("wk"),   
 names\_to = "week",  
 values\_to = "position",  
 names\_prefix = "wk",  
 values\_drop\_na = TRUE)  
  
billboard\_longer

## # A tibble: 5,307 × 5  
## artist track date.entered week position  
## <chr> <chr> <date> <chr> <dbl>  
## 1 2 Pac Baby Don't Cry (Keep... 2000-02-26 1 87  
## 2 2 Pac Baby Don't Cry (Keep... 2000-02-26 2 82  
## 3 2 Pac Baby Don't Cry (Keep... 2000-02-26 3 72  
## 4 2 Pac Baby Don't Cry (Keep... 2000-02-26 4 77  
## 5 2 Pac Baby Don't Cry (Keep... 2000-02-26 5 87  
## 6 2 Pac Baby Don't Cry (Keep... 2000-02-26 6 94  
## 7 2 Pac Baby Don't Cry (Keep... 2000-02-26 7 99  
## 8 2Ge+her The Hardest Part Of ... 2000-09-02 1 91  
## 9 2Ge+her The Hardest Part Of ... 2000-09-02 2 87  
## 10 2Ge+her The Hardest Part Of ... 2000-09-02 3 92  
## # ℹ 5,297 more rows

billboard\_longer %>%   
 pivot\_wider(names\_from = "week",   
 values\_from = "position")

## # A tibble: 317 × 68  
## artist track date.entered `1` `2` `3` `4` `5` `6` `7` `8`  
## <chr> <chr> <date> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 Pac Baby… 2000-02-26 87 82 72 77 87 94 99 NA  
## 2 2Ge+her The … 2000-09-02 91 87 92 NA NA NA NA NA  
## 3 3 Doors D… Kryp… 2000-04-08 81 70 68 67 66 57 54 53  
## 4 3 Doors D… Loser 2000-10-21 76 76 72 69 67 65 55 59  
## 5 504 Boyz Wobb… 2000-04-15 57 34 25 17 17 31 36 49  
## 6 98^0 Give… 2000-08-19 51 39 34 26 26 19 2 2  
## 7 A\*Teens Danc… 2000-07-08 97 97 96 95 100 NA NA NA  
## 8 Aaliyah I Do… 2000-01-29 84 62 51 41 38 35 35 38  
## 9 Aaliyah Try … 2000-03-18 59 53 38 28 21 18 16 14  
## 10 Adams, Yo… Open… 2000-08-26 76 76 74 69 68 67 61 58  
## # ℹ 307 more rows  
## # ℹ 57 more variables: `9` <dbl>, `10` <dbl>, `11` <dbl>, `12` <dbl>,  
## # `13` <dbl>, `14` <dbl>, `15` <dbl>, `16` <dbl>, `17` <dbl>, `18` <dbl>,  
## # `19` <dbl>, `20` <dbl>, `21` <dbl>, `22` <dbl>, `23` <dbl>, `24` <dbl>,  
## # `25` <dbl>, `26` <dbl>, `27` <dbl>, `28` <dbl>, `29` <dbl>, `30` <dbl>,  
## # `31` <dbl>, `32` <dbl>, `33` <dbl>, `34` <dbl>, `35` <dbl>, `36` <dbl>,  
## # `37` <dbl>, `38` <dbl>, `39` <dbl>, `40` <dbl>, `41` <dbl>, `42` <dbl>, …

?pivot\_longer

billboard %>%   
 separate(col = date.entered,  
 into = c("year", "month", "day"))

## # A tibble: 317 × 81  
## artist track year month day wk1 wk2 wk3 wk4 wk5 wk6 wk7  
## <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 Pac Baby… 2000 02 26 87 82 72 77 87 94 99  
## 2 2Ge+her The … 2000 09 02 91 87 92 NA NA NA NA  
## 3 3 Doors Do… Kryp… 2000 04 08 81 70 68 67 66 57 54  
## 4 3 Doors Do… Loser 2000 10 21 76 76 72 69 67 65 55  
## 5 504 Boyz Wobb… 2000 04 15 57 34 25 17 17 31 36  
## 6 98^0 Give… 2000 08 19 51 39 34 26 26 19 2  
## 7 A\*Teens Danc… 2000 07 08 97 97 96 95 100 NA NA  
## 8 Aaliyah I Do… 2000 01 29 84 62 51 41 38 35 35  
## 9 Aaliyah Try … 2000 03 18 59 53 38 28 21 18 16  
## 10 Adams, Yol… Open… 2000 08 26 76 76 74 69 68 67 61  
## # ℹ 307 more rows  
## # ℹ 69 more variables: wk8 <dbl>, wk9 <dbl>, wk10 <dbl>, wk11 <dbl>,  
## # wk12 <dbl>, wk13 <dbl>, wk14 <dbl>, wk15 <dbl>, wk16 <dbl>, wk17 <dbl>,  
## # wk18 <dbl>, wk19 <dbl>, wk20 <dbl>, wk21 <dbl>, wk22 <dbl>, wk23 <dbl>,  
## # wk24 <dbl>, wk25 <dbl>, wk26 <dbl>, wk27 <dbl>, wk28 <dbl>, wk29 <dbl>,  
## # wk30 <dbl>, wk31 <dbl>, wk32 <dbl>, wk33 <dbl>, wk34 <dbl>, wk35 <dbl>,  
## # wk36 <dbl>, wk37 <dbl>, wk38 <dbl>, wk39 <dbl>, wk40 <dbl>, wk41 <dbl>, …

billboard %>%  
 unite(col = "date",  
 into = c(year, month, day))

# ICMA 9/25/2024

ChickWeight %>%  
 ggplot(aes(x = weight)) + # provide the aesthetic mapping   
 geom\_histogram()

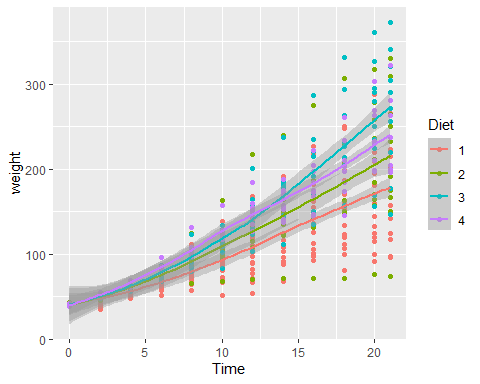
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



cant use pipe functions for geom histogram

ChickWeight %>%  
 ggplot(aes(y = weight,  
 x = Time,  
 color = Diet)) +  
 geom\_point() + geom\_smooth()

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

 kind of diet missing, need more information ^will add above noise = random variability

ChickWeight %>%  
 ggplot(aes(y = weight,  
 x = Time,  
 color = Diet)) +  
 geom\_boxplot(alpha = 0.5) + # make a lil transparent  
 #geom\_smooth(se = FALSE) + #remove error bands  
 theme\_minimal() + # using a premade theme to change appearance  
 facet\_grid(~Diet) +  
 labs(y = "weight in grams",   
 x = "days since birth",  
 title = "Weight Over Time Across Diets")   
  
ggsave("chick\_weightovertime.png", width = 9 , length = 8)

# ICMA 10/7/2024

types: numeric, character, logical shared functions: ypeof() - find type of object numeric: integers, doubles (r default)

doubles do not need decimals be careful if you do as.integer(“hello”) -> will not give error will give warning, it will convert to NA (can’t parse as number)

as.numeric("12") #digits can parse into numbers

## [1] 12

my\_double <- 12.9  
typeof(my\_double)

## [1] "double"

as.integer(my\_double) #does not round, cuts off decimal part (truncate)

## [1] 12

as.character(my\_double)

## [1] "12.9"

NA introduced by coercion NA is logical used as missing value indicator 0 is a value, null is a null object

“missing” in sense of does not exist in R

2==NA #result is NA (does not know) - does not know if it is equal to 2 because it does not know if it is

## [1] NA

NA==NA # us is.na()

## [1] NA

my\_values <- c(1,3,5,NA,8,10)  
mean(my\_values)

## [1] NA

#typically can override this by adding a na.rm = TRUE argument to remove the NAs  
mean(my\_values, na.rm = TRUE) #ignore NA

## [1] 5.4

single or/and will evaulate all double or/and will evaulate until one

prediction: true true true true true

(3 / 4) > 0 && (8 + 1 - 7) == 2 # evaluates until one

## [1] TRUE

(5 / 2) == 1 | (2 / 1) > 0 # evaluates all

## [1] TRUE

(5 \* (5-3)) == 10 || (10 / 2) < 4 # evaluates until one

## [1] TRUE

is.numeric(mean(c(38,12,123.7,4.32)))

## [1] TRUE

#TRUE would mean that the mean of those numbers is a numeric in R  
  
((7/2) == 3.5) & TRUE | NA

## [1] TRUE

NA & TRUE # NA since NA is neither TRUE or FALSE

## [1] NA

NA & NA # NA

## [1] NA

# ICMA 10/9/2024

scalars, vectors, lists, data frames, tibbles

scalar: single value, like a number or string can me numeric (int, dbl), char, string

vector: a series of elements that have the same type - numbers can be stored as strings, how to know –> quotations

* because vectors have to have the same type, R will coerce things to be the same type

c("1", 2, 3 ,4) #takes value as a string

## [1] "1" "2" "3" "4"

c(2, 3, 4) #takes values as ints

## [1] 2 3 4

string has higher precedence than ints during coercion

lists: objects that can store multiple things of different types, including additional vectors –> use list()

my\_list <- list("letters" = c("A", "L", "J", "H"),  
 "numbers" = 4:8,  
 "more\_numbers" = 10:30)   
  
my\_list

## $letters  
## [1] "A" "L" "J" "H"  
##   
## $numbers  
## [1] 4 5 6 7 8  
##   
## $more\_numbers  
## [1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

single brackets [] will return the list double brackets [[]] will return the values within the list; can also use $

data frame: each element is of the same length and are bound together –> will get error if diff length

use data.frame

also use [], [[]], and $ to access elements

tibble:a modern reimagining of the data.frame - lazy and surly

create a tibble: 1. using a list of vectors using tibble() 2. type out row wise using tribble()

age\_data <- tribble(  
 ~id, ~age,  
 1, 8,   
 2, 10,   
 3, 8,  
 5, 9  
   
)  
  
gender\_data <- tribble(  
 ~id, ~gender,  
 1, "f",  
 2, "m",  
 3, "nb",  
 4, "m",  
 6, "f"  
)  
  
age\_data

## # A tibble: 4 × 2  
## id age  
## <dbl> <dbl>  
## 1 1 8  
## 2 2 10  
## 3 3 8  
## 4 5 9

gender\_data

## # A tibble: 5 × 2  
## id gender  
## <dbl> <chr>   
## 1 1 f   
## 2 2 m   
## 3 3 nb   
## 4 4 m   
## 5 6 f

full\_join(age\_data, gender\_data)

## Joining with `by = join\_by(id)`

## # A tibble: 6 × 3  
## id age gender  
## <dbl> <dbl> <chr>   
## 1 1 8 f   
## 2 2 10 m   
## 3 3 8 nb   
## 4 5 9 <NA>   
## 5 4 NA m   
## 6 6 NA f

bind\_cols(age\_data, gender\_data)

bind\_rows(age\_data, gender\_data)

## # A tibble: 9 × 3  
## id age gender  
## <dbl> <dbl> <chr>   
## 1 1 8 <NA>   
## 2 2 10 <NA>   
## 3 3 8 <NA>   
## 4 5 9 <NA>   
## 5 1 NA f   
## 6 2 NA m   
## 7 3 NA nb   
## 8 4 NA m   
## 9 6 NA f

# ICMA 10/16/2024

ITERATIONS AND LOOPS - often need to do the same thing over and over again - functions help solve this program by packaging together a bunch of code - loops help run the same code over and over again through iterations

mean(c(1,2,3))

## [1] 2

(1+2+3)/3

## [1] 2

vec <- c(1,2,3)  
  
(vec[1]+vec[2]+vec[3])/length(vec)

## [1] 2

for loop iterates over a vector and executes R code after every iteration general syntax:

for(var in seq){  
 expr  
}

example

for (i in 1:10) {  
   
 i <- i^2  
 print(i)  
}

## [1] 1  
## [1] 4  
## [1] 9  
## [1] 16  
## [1] 25  
## [1] 36  
## [1] 49  
## [1] 64  
## [1] 81  
## [1] 100

fave\_number <- tribble(~name, ~number,  
 "MC", 8,  
 "AP", 9,  
 "NG", 12,  
 "VE", 6,  
 "AL", 17,  
 "AR", 3,  
 "MC", 7,  
 "RK", 8,  
 "AC", 4,  
 "SC", 21,  
 "KK", 16,  
 "RK", 28,  
 "VK", 17,  
 "MH", 12  
 )

1. for loop to print fave number

for (i in fave\_number$number) {  
   
 print(i)  
   
}

## [1] 8  
## [1] 9  
## [1] 12  
## [1] 6  
## [1] 17  
## [1] 3  
## [1] 7  
## [1] 8  
## [1] 4  
## [1] 21  
## [1] 16  
## [1] 28  
## [1] 17  
## [1] 12

fave\_number$number is different than fave\_number[“number”]… why?

typeof(fave\_number$number) #type of the vector

## [1] "double"

typeof(fave\_number["number"]) #type of the list

## [1] "list"

one square bracket means pepper is in packaging in jar two square brackt means actual pepper is in jar

length(fave\_number)

## [1] 2

1. output for each person –> **’s fave number is**

for (i in seq\_along(fave\_number$name)) {  
   
 print(str\_c(fave\_number$name[i], "'s favorite number is ", fave\_number$number[i]))  
   
}

## [1] "MC's favorite number is 8"  
## [1] "AP's favorite number is 9"  
## [1] "NG's favorite number is 12"  
## [1] "VE's favorite number is 6"  
## [1] "AL's favorite number is 17"  
## [1] "AR's favorite number is 3"  
## [1] "MC's favorite number is 7"  
## [1] "RK's favorite number is 8"  
## [1] "AC's favorite number is 4"  
## [1] "SC's favorite number is 21"  
## [1] "KK's favorite number is 16"  
## [1] "RK's favorite number is 28"  
## [1] "VK's favorite number is 17"  
## [1] "MH's favorite number is 12"

?seq\_along

mtcars

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2

everything(), map(), across()

while(cond) –> while something is true do this thing

# ICMA 10/30/2024