Notebook3

Managing Data Frames with the dplyr package

```
#install.packages("dplyr")
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

select()

- The select() function can be used to select columns of a data frame.

```
df = read.csv("data/iris.txt")
head(df)
```

```
  Sepal.length Sepal.width Petal.length Petal.width       Class
1          5.1         3.5          1.4         0.2 Iris-setosa
2          4.9         3.0          1.4         0.2 Iris-setosa
3          4.7         3.2          1.3         0.2 Iris-setosa
4          4.6         3.1          1.5         0.2 Iris-setosa
5          5.0         3.6          1.4         0.2 Iris-setosa
6          5.4         3.9          1.7         0.4 Iris-setosa
```

```
head(df[2:4])
```

```
  Sepal.width Petal.length Petal.width
1         3.5          1.4         0.2
2         3.0          1.4         0.2
3         3.2          1.3         0.2
4         3.1          1.5         0.2
5         3.6          1.4         0.2
6         3.9          1.7         0.4
```

```
#head(select(df, (2:4)))
head(select(df, (Sepal.length:Petal.length)))
```

```
  Sepal.length Sepal.width Petal.length
1          5.1         3.5          1.4
2          4.9         3.0          1.4
3          4.7         3.2          1.3
4          4.6         3.1          1.5
5          5.0         3.6          1.4
6          5.4         3.9          1.7
```

- You can also omit variables using the select() function by using the negative sign

```
head(select(df, -(Sepal.length:Petal.width)))
```

```
       Class
1 Iris-setosa
2 Iris-setosa
3 Iris-setosa
4 Iris-setosa
5 Iris-setosa
6 Iris-setosa
```

The select() function also allows a special syntax that allows you to specify variable names based on patterns (ends_width, start_width, contains).

```
head(select(df, starts_with("p")))
```

```
  Petal.length Petal.width
1          1.4         0.2
2          1.4         0.2
3          1.3         0.2
4          1.5         0.2
5          1.4         0.2
6          1.7         0.4
```

```
head(select(df, ends_with("h")))
```

```
  Sepal.length Sepal.width Petal.length Petal.width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
4          4.6         3.1          1.5         0.2
5          5.0         3.6          1.4         0.2
6          5.4         3.9          1.7         0.4
```

fliter()

- The filter() function is used to extract subsets of rows from a data frame.

```
head(filter(df, Sepal.length>5))
```

```
  Sepal.length Sepal.width Petal.length Petal.width       Class
1          5.1         3.5          1.4         0.2 Iris-setosa
2          5.4         3.9          1.7         0.4 Iris-setosa
3          5.4         3.7          1.5         0.2 Iris-setosa
4          5.8         4.0          1.2         0.2 Iris-setosa
5          5.7         4.4          1.5         0.4 Iris-setosa
6          5.4         3.9          1.3         0.4 Iris-setosa
```

arrange()

- The arrange() function is used to reorder rows of a data frame according to one of the variables/- columns.

```
head(arrange(df, Sepal.width))
```

```
  Sepal.length Sepal.width Petal.length Petal.width           Class
1          5.0         2.0          3.5         1.0 Iris-versicolor
2          6.0         2.2          4.0         1.0 Iris-versicolor
3          6.2         2.2          4.5         1.5 Iris-versicolor
4          6.0         2.2          5.0         1.5  Iris-virginica
5          4.5         2.3          1.3         0.3     Iris-setosa
6          5.5         2.3          4.0         1.3 Iris-versicolor
```

```
head(arrange(df, desc(Sepal.width)))
```

```
  Sepal.length Sepal.width Petal.length Petal.width       Class
1          5.7         4.4          1.5         0.4 Iris-setosa
2          5.5         4.2          1.4         0.2 Iris-setosa
3          5.2         4.1          1.5         0.1 Iris-setosa
4          5.8         4.0          1.2         0.2 Iris-setosa
5          5.4         3.9          1.7         0.4 Iris-setosa
6          5.4         3.9          1.3         0.4 Iris-setosa
```

rename()

Renaming a variable in a data frame in R.

```
head(rename(df, s1 = Sepal.width))
```

```
  Sepal.length  s1 Petal.length Petal.width       Class
1          5.1 3.5          1.4         0.2 Iris-setosa
2          4.9 3.0          1.4         0.2 Iris-setosa
3          4.7 3.2          1.3         0.2 Iris-setosa
4          4.6 3.1          1.5         0.2 Iris-setosa
5          5.0 3.6          1.4         0.2 Iris-setosa
6          5.4 3.9          1.7         0.4 Iris-setosa
```

mutate()

- The mutate() function exists to compute transformations of variables in a data frame.
- to create new variables that are derived from existing variables

```
head(mutate(df, S1 = Sepal.width - mean(Sepal.width)))
```

```
  Sepal.length Sepal.width Petal.length Petal.width       Class          S1
1          5.1         3.5          1.4         0.2 Iris-setosa  0.44266667
2          4.9         3.0          1.4         0.2 Iris-setosa -0.05733333
3          4.7         3.2          1.3         0.2 Iris-setosa  0.14266667
4          4.6         3.1          1.5         0.2 Iris-setosa  0.04266667
5          5.0         3.6          1.4         0.2 Iris-setosa  0.54266667
6          5.4         3.9          1.7         0.4 Iris-setosa  0.84266667
```

```
head(transmute(df, S1 = Sepal.width - mean(Sepal.width)))
```

```
          S1
1  0.44266667
2 -0.05733333
3  0.14266667
4  0.04266667
5  0.54266667
6  0.84266667
```

%>%

The pipeline operater %>% is very handy for stringing together multiple dplyr functions in a sequence of operations

third(second(first(x)))

first(x) %>% second %>% third

```
head(df)
```

```
  Sepal.length Sepal.width Petal.length Petal.width       Class
1          5.1         3.5          1.4         0.2 Iris-setosa
2          4.9         3.0          1.4         0.2 Iris-setosa
3          4.7         3.2          1.3         0.2 Iris-setosa
4          4.6         3.1          1.5         0.2 Iris-setosa
5          5.0         3.6          1.4         0.2 Iris-setosa
6          5.4         3.9          1.7         0.4 Iris-setosa
```

```
df1 <- df %>%
  filter(df[["Class"]] == "Iris-setosa") %>%
  select(c('Class','Sepal.length'))
```

```
head(df1)
```

```
       Class Sepal.length
1 Iris-setosa          5.1
2 Iris-setosa          4.9
3 Iris-setosa          4.7
4 Iris-setosa          4.6
5 Iris-setosa          5.0
6 Iris-setosa          5.4
```

group_by(), summarize()

```
group_by(df, Class)
```

```
# A tibble: 150 x 5
# Groups:   Class [3]
   Sepal.length Sepal.width Petal.length Petal.width Class
          <dbl>       <dbl>        <dbl>       <dbl> <chr>
 1          5.1         3.5          1.4         0.2 Iris-setosa
 2          4.9         3            1.4         0.2 Iris-setosa
 3          4.7         3.2          1.3         0.2 Iris-setosa
 4          4.6         3.1          1.5         0.2 Iris-setosa
 5          5           3.6          1.4         0.2 Iris-setosa
 6          5.4         3.9          1.7         0.4 Iris-setosa
 7          4.6         3.4          1.4         0.3 Iris-setosa
 8          5           3.4          1.5         0.2 Iris-setosa
 9          4.4         2.9          1.4         0.2 Iris-setosa
10          4.9         3.1          1.5         0.1 Iris-setosa
# i 140 more rows
```

- Group_by() function alone will not give any output. It should be followed by summarise() function with an appropriate action to perform.

https://media.geeksforgeeks.org/wp-content/cdn-ploads/20210802175750/Sample_Superstore.csv

```
df = read.csv("data/Sample_Superstore.csv");
head(df)
```

```
  Row.ID      State Region        Category Sub.Category    Sales    Profit
1      1   Kentucky  South       Furniture    Bookcases 261.9600   41.9136
```

```
2      2   Kentucky  South       Furniture       Chairs 731.9400  219.5820
3      3 California   West Office Supplies       Labels  14.6200    6.8714
4      4    Florida  South       Furniture       Tables 957.5775 -383.0310
5      5    Florida  South Office Supplies      Storage  22.3680    2.5164
6      6 California   West       Furniture  Furnishings  48.8600   14.1694
```

```
df_grp_region = df %>% group_by(Region)  %>%
                summarise(total_sales = sum(Sales),
                          .groups = 'drop')

df_grp_region
```

```
# A tibble: 4 x 3
  Region  total_sales total_profits
  <chr>         <dbl>         <dbl>
1 Central      45502.         -850.
2 East         33798.         1712.
3 South        12344.         -424.
4 West         37782.         4925.
```

Grouping multiple columns

```
df_grp_reg_cat = df %>% group_by(Region, Category) %>%
df_grp_reg_cat
```

```
# A tibble: 12 x 4
   Region  Category        total_Sales total_Profit
   <chr>   <chr>                 <dbl>        <dbl>
 1 Central Furniture            13596.       -688.
 2 Central Office Supplies       7762.        318.
 3 Central Technology           24144.       -479.
 4 East    Furniture            11658.      -1784.
 5 East    Office Supplies      12794.       2537.
 6 East    Technology            9346.        959.
 7 South   Furniture             6329.       -552.
 8 South   Office Supplies       3401.       -198.
 9 South   Technology            2614.        326.
10 West    Furniture            14334.        -93.9
11 West    Office Supplies       7510.       1575.
12 West    Technology           15938.       3444.
```

```r
df %>% group_by(Region) %>%
  summarise(n = n(), .groups = "keep") %>%
  group_vars()
```

```
[1] "Region"
```

```r
df %>% group_by(Region) %>%
  summarise(n = n(), .groups = "drop") %>%
  group_vars()
```

```
character(0)
```

```r
grp <- df %>% group_by(Region)
grp %>% group_keys()
```

```
# A tibble: 4 x 1
  Region
  <chr>
1 Central
2 East
3 South
4 West
```

```r
grp1 <- df %>% group_by(Region, Category)
grp1 %>%  group_keys()
```

```
# A tibble: 12 x 2
   Region  Category
   <chr>   <chr>
 1 Central Furniture
 2 Central Office Supplies
 3 Central Technology
 4 East    Furniture
 5 East    Office Supplies
 6 East    Technology
 7 South   Furniture
 8 South   Office Supplies
```

```
 9 South   Technology
10 West    Furniture
11 West    Office Supplies
12 West    Technology
```

```
# You can see which group each row belongs
grp %>% group_indices()
```

```
  [1] 3 3 4 3 3 4 4 4 4 4 4 4 3 4 1 1 1 4 4 4 4 1 1 2 4 4 4 2 2 2 2 2 2 2 1 1 1
 [38] 1 1 1 1 1 4 3 1 1 1 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 4 4 4 4 1 4 4 3 2 1 3 3
 [75] 3 1 1 1 1 3 3 4 4 3 1 3 1 1 1 4 4 4 1 1 1 4 2 4 1 1 1 1 1 4 4 4 3 3 3 1 2
[112] 1 1 2 2 2 2 4 3 2 2 2 2 2 1 1 4 4 4 4 2 2 2 4 4 4 4 4 4 4 2 4 4 4 1 4 2 1
[149] 1 1 1 4 4 4 4 4 4 4 1 1 4 2 4 4 1 1 1 1 1 1 1 4 4 4 1 1 1 2 2 2 1 1 3 3 3
[186] 2 4 1 1 2 2 2 2 2 4 2 2 2 2 2 2 4 4 1 3 4 1 1 1 1 1 1 2 2 2 2 2 4 4 2 2 2
[223] 2 2 2 1 1 1 3 3 3 3 3 3 3 3 4 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 2 2 1 1 1 1 2
[260] 2 2 1 1 1 1 4 3 2 2 2 4 4 4 4 4 4 2 2 2 2 1 1 4 4 4 4 3 3 3 2 2 2 2 4 4 4
[297] 4 4 2 2 2 2 2 2 1 2 1 2 2 3 4 4 4 4 2 3 3 2 2 2 2 2 2 4 4 4 3 2 2 2 2 2 2 2
[334] 2 4 4 4 4 4 4 4 2 2 2 2 1 4 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 2 2 2 2 4 2 2 2 2
[371] 1 1 4 4 4 4 1 2 1 1 1 4 4 1 3 3 2 2 2 2 2 4 4 1 4 2 2 1 1 1 1 1 3 3 2 4 4
[408] 4 4 4 4 4 4 4 4 4 4 4 3 4 4 4 2 2 3 1 1 2 1 3 3 3 3 2 3 3 1 1 1 2 1 1 1 1
[445] 1 1 1 2 2 2 2 2 2 2 2 1 4 1 1 4 4 4 4 4 4 4 4 1 1 1 2 4 4 4 4 4 4 4 2 2 2
[482] 4 2 2 4 4 4 1 1 1 1 2 2 4 3 3 4 4 4 4
```

```
#which rows each group contains
grp %>% group_rows()
```

```
<list_of<integer>[4]>
[[1]]
  [1]  15  16  17  22  23  35  36  37  38  39  40  41  42  45  46  47  50  51
 [19]  52  53  67  72  76  77  78  79  85  87  88  89  93  94  95  99 100 101
 [37] 102 103 110 112 113 125 126 145 148 149 150 151 159 160 165 166 167 168
 [55] 169 170 171 175 176 177 181 182 188 189 204 207 208 209 210 211 212 226
 [73] 227 228 238 239 240 241 242 243 244 245 246 247 248 249 255 256 257 258
 [91] 262 263 264 265 281 282 304 306 345 371 372 377 379 380 381 384 394 398
[109] 399 400 401 402 426 427 429 437 438 439 441 442 443 444 445 446 447 456
[127] 458 459 468 469 470 488 489 490 491

[[2]]
  [1]  24  28  29  30  31  32  33  34  48  49  54  55  56  57  58  59  60  61
 [19]  62  71  97 111 114 115 116 117 120 121 122 123 124 131 132 133 141 147
```

```
 [37] 162 178 179 180 186 190 191 192 193 194 196 197 198 199 200 201 213 214
 [55] 215 216 217 220 221 222 223 224 225 253 254 259 260 261 268 269 270 277
 [73] 278 279 280 290 291 292 293 299 300 301 302 303 305 307 308 314 317 318
 [91] 319 320 321 322 327 328 329 330 331 332 333 334 341 342 343 344 347 348
[109] 349 350 351 352 353 354 355 356 357 358 362 363 364 365 367 368 369 370
[127] 378 387 388 389 390 391 396 397 405 423 424 428 434 440 448 449 450 451
[145] 452 453 454 455 471 479 480 481 483 484 492 493

[[3]]
 [1]   1   2   4   5  13  44  70  73  74  75  80  81  84  86 107 108 109 119 183
[20] 184 185 205 229 230 231 232 233 234 235 236 267 287 288 289 309 315 316 326
[39] 359 360 361 385 386 403 404 419 425 430 431 432 433 435 436 495 496

[[4]]
  [1]   3   6   7   8   9  10  11  12  14  18  19  20  21  25  26  27  43  63
 [19]  64  65  66  68  69  82  83  90  91  92  96  98 104 105 106 118 127 128
 [37] 129 130 134 135 136 137 138 139 140 142 143 144 146 152 153 154 155 156
 [55] 157 158 161 163 164 172 173 174 187 195 202 203 206 218 219 237 250 251
 [73] 252 266 271 272 273 274 275 276 283 284 285 286 294 295 296 297 298 310
 [91] 311 312 313 323 324 325 335 336 337 338 339 340 346 366 373 374 375 376
[109] 382 383 392 393 395 406 407 408 409 410 411 412 413 414 415 416 417 418
[127] 420 421 422 457 460 461 462 463 464 465 466 467 472 473 474 475 476 477
[145] 478 482 485 486 487 494 497 498 499 500
```

```
# names of the grouping variables
grp %>% group_vars()
```

```
[1] "Region"
```

```
grp1 %>% group_vars()
```

```
[1] "Region"   "Category"
```

Changing and adding to grouping variables

```
grp %>%
  group_by(Category) %>%
  tally()
```

```
# A tibble: 3 x 2
  Category            n
  <chr>           <int>
1 Furniture         110
2 Office Supplies   293
3 Technology         97
```

```
grp %>%
  group_by(Category, .add = TRUE) %>%
  tally()
```

```
# A tibble: 12 x 3
# Groups:   Region [4]
   Region  Category            n
   <chr>   <chr>           <int>
 1 Central Furniture          29
 2 Central Office Supplies    78
 3 Central Technology         28
 4 East    Furniture          30
 5 East    Office Supplies   100
 6 East    Technology         26
 7 South   Furniture          16
 8 South   Office Supplies    28
 9 South   Technology         11
10 West    Furniture          35
11 West    Office Supplies    87
12 West    Technology         32
```

```
grp %>%
  ungroup() %>%
  tally()
```

```
# A tibble: 1 x 1
      n
  <int>
1   500
```

```
grp1 %>%
  ungroup(Category) %>%
```

```
  tally()
```

```
# A tibble: 4 x 2
  Region      n
  <chr>   <int>
1 Central   135
2 East      156
3 South      55
4 West      154
```

slice()- select rows within a group

```
grp %>%
  relocate(Region) %>%
  slice(2)
```

```
# A tibble: 4 x 7
# Groups:   Region [4]
  Region  Row.ID State        Category        Sub.Category   Sales   Profit
  <chr>    <int> <chr>        <chr>           <chr>          <dbl>    <dbl>
1 Central     16 Texas        Office Supplies Binders         2.54    -3.82
2 East        28 Pennsylvania Furniture       Bookcases    3083.   -1665.
3 South        2 Kentucky     Furniture       Chairs        732.     220.
4 West         6 California   Furniture       Furnishings    48.9     14.2
```

`slice_min()` to select the smallest `n` values of a variable:

`slice_max()`to select the largest `n` values of a variable:

```
grp %>%
  relocate(Region,Sales) %>%
  slice_min(Sales,n =1)
```

```
# A tibble: 4 x 7
# Groups:   Region [4]
  Region  Sales Row.ID State          Category        Sub.Category Profit
  <chr>   <dbl>  <int> <chr>          <chr>           <chr>         <dbl>
1 Central  1.25     76 Texas          Office Supplies Binders       -1.93
2 East     1.68     62 New York       Office Supplies Art            0.84
3 South    3.30    109 North Carolina Office Supplies Art            1.07
4 West     2.39    127 Arizona        Office Supplies Binders       -1.83
```

```
grp %>%
  relocate(Region,Sales) %>%
  slice_head(n=2)
```

```
# A tibble: 8 x 7
# Groups:   Region [4]
  Region    Sales Row.ID State        Category        Sub.Category   Profit
  <chr>     <dbl>  <int> <chr>        <chr>           <chr>           <dbl>
1 Central   68.8      15 Texas        Office Supplies Appliances    -124.
2 Central    2.54     16 Texas        Office Supplies Binders         -3.82
3 East      71.4      24 Pennsylvania Furniture       Chairs          -1.02
4 East    3083.       28 Pennsylvania Furniture       Bookcases    -1665.
5 South    262.        1 Kentucky     Furniture       Bookcases       41.9
6 South    732.        2 Kentucky     Furniture       Chairs         220.
7 West      14.6       3 California   Office Supplies Labels           6.87
8 West      48.9       6 California   Furniture       Furnishings     14.2
```

Useful Summary Functions

n() - returns the size of the current group.

sum(!is.na(x)) - To count the number of non-missing values.

n_distinct(x) - To count the number of distinct (unique) values.

mean(), median(), sd(), interquartile range- ***IQR()***, mean absolute deviation - ***mad()***

min(), max()

first(x), nth(), last(x)

min_rank(),

```
df %>%
  count(Category)
```

```
         Category   n
1       Furniture 110
2 Office Supplies 293
3      Technology  97
```

```r
df %>%
group_by(Category) %>%
summarize(Count = n())
```

```
# A tibble: 3 x 2
  Category        Count
  <chr>           <int>
1 Furniture         110
2 Office Supplies   293
3 Technology         97
```

```r
df %>%
group_by(Category) %>%
tally()
```

```
# A tibble: 3 x 2
  Category            n
  <chr>           <int>
1 Furniture         110
2 Office Supplies   293
3 Technology         97
```

```r
df %>%
group_by(Category) %>%
tally(sort = TRUE)
```

```
# A tibble: 3 x 2
  Category            n
  <chr>           <int>
1 Office Supplies   293
2 Furniture         110
3 Technology         97
```

```r
df %>%
  group_by(Category) %>%
  summarise(count = sum(Sales > 100))
```

```
# A tibble: 3 x 2
  Category        count
  <chr>           <int>
1 Furniture          72
2 Office Supplies    61
3 Technology         58
```

Ranking

```
  x <- c(1,2,2,5)

  row_number(x)
```

```
[1] 1 2 3 4
```

```
  min_rank(x)
```

```
[1] 1 2 2 4
```

```
  dense_rank(x)
```

```
[1] 1 2 2 3
```

```
  df %>%
    select(Region, Sales) %>%
    group_by(Region) %>%
    mutate(rank = min_rank(Sales)) %>%
    arrange(Region,Sales)
```

```
# A tibble: 500 x 3
# Groups:   Region [4]
   Region  Sales  rank
   <chr>   <dbl> <int>
 1 Central  1.25     1
 2 Central  1.62     2
 3 Central  1.79     3
 4 Central  2.08     4
```

```
 5 Central  2.2      5
 6 Central  2.54     6
 7 Central  2.91     7
 8 Central  4.04     8
 9 Central  4.79     9
10 Central  6.16    10
# i 490 more rows
```

```r
df %>%
  select(Region, Sales) %>%
  group_by(Region) %>%
  mutate(rank = dense_rank(Sales)) %>%
  arrange(Region,Sales)
```

```
# A tibble: 500 x 3
# Groups:   Region [4]
   Region  Sales  rank
   <chr>   <dbl> <int>
 1 Central  1.25     1
 2 Central  1.62     2
 3 Central  1.79     3
 4 Central  2.08     4
 5 Central  2.2      5
 6 Central  2.54     6
 7 Central  2.91     7
 8 Central  4.04     8
 9 Central  4.79     9
10 Central  6.16    10
# i 490 more rows
```

n_distinct(x)

Q. Find group with max rows.

```r
df %>%
group_by(Category) %>%
summarize(Sub.Categories = n_distinct(Sub.Category))
```

```
# A tibble: 3 x 2
  Category        Sub.Categories
  <chr>                    <int>
```

```
1 Furniture                    4
2 Office Supplies              9
3 Technology                   4
```

Q. Find max sales Category wise.

```
df %>%
  group_by(Category) %>%
  summarize(count = max(Sales))
```

```
# A tibble: 3 x 2
  Category        count
  <chr>           <dbl>
1 Furniture       3083.
2 Office Supplies 4355.
3 Technology      8160.
```

```
names(df)
```

```
[1] "Row.ID"       "State"        "Region"       "Category"     "Sub.Category"
[6] "Sales"        "Profit"
```

```
head(df[c("Category","Sales")])
```

```
         Category    Sales
1       Furniture 261.9600
2       Furniture 731.9400
3 Office Supplies  14.6200
4       Furniture 957.5775
5 Office Supplies  22.3680
6       Furniture  48.8600
```

```
df %>%
  group_by(Category) %>%
  summarize(first = first(Sales))
```

```
# A tibble: 3 x 2
  Category        first
  <chr>           <dbl>
1 Furniture       262.
2 Office Supplies  14.6
3 Technology      907.
```

Grouping in R Variants

- **Group_by_all:** Allows the user to use every field in the data set.
- **Group_by_if:** Allows you to use an if function to group certain fields.
- **Group_Split:** This allows the user to separate the data into a list of Tibbles.
- **Group_Nest:** This returns a Tibble containing the grouped columns and the data from those respective groups.

```
df %>%
 group_by_all() %>%
 summarise(my_cnt = n(), .groups = 'keep') %>%
 arrange(desc(my_cnt)) %>% group_vars()
```

```
[1] "Row.ID"       "State"        "Region"       "Category"     "Sub.Category"
[6] "Sales"        "Profit"
```

group_split()

```
df_split <- df %>%
  group_by(Category) %>%
  group_split()

df_split[[3]]
```

```
# A tibble: 97 x 7
   Row.ID State      Region  Category   Sub.Category  Sales Profit
    <int> <chr>      <chr>   <chr>      <chr>         <dbl>  <dbl>
 1      8 California West    Technology Phones        907.   90.7
 2     12 California West    Technology Phones        911.   68.4
 3     20 California West    Technology Phones        213.   16.0
 4     27 California West    Technology Accessories    90.6  11.8
```

```
 5     36 Texas      Central Technology Phones       1098.  123.
 6     41 Texas      Central Technology Phones        371.   41.8
 7     42 Illinois   Central Technology Phones        147.   16.6
 8     45 Minnesota  Central Technology Accessories    46.0  19.8
 9     48 Delaware   East    Technology Accessories    45     4.95
10     49 Delaware   East    Technology Phones         21.8   6.10
# i 87 more rows
```

```
df_nest <- df %>%
 group_nest(Region);

df_nest
```

```
# A tibble: 4 x 2
  Region                data
  <chr>   <list<tibble[,6]>>
1 Central          [135 x 6]
2 East             [156 x 6]
3 South             [55 x 6]
4 West             [154 x 6]
```