TABLES TO ROWS

CONVERTING DATA FROM TABLES TO ROWS

- compact data - often in data: lists categorical variables in combinations > put number of cases/observations that fall into each combination
- small efficient way of storing data - often presenting it
- BUT can be problem for analysis
- soln: take data out of table format > put it into long, row by row format

INSTALL AND LOAD PACKAGES

```
pacman::p_load(datasets, pacman, tidyverse)
```

LOAD AND EXPLORE DATA

```
?UCBAdmission        # info on dataset
```

- University of California at Berkeley
- well known dataset, bc looks at differences of associations that can happen at different LEVELS of observation
- a 3 dimensional array
- shows whether a person who applied was admitted, rejected / gender (m, f) / department (a, b,c, d,e)

No	Name	Levels
1	admit	admitted, rejected
2	gender	male, female
3	dept	A, B, C, D, E (anonymously labels as A thru F)

```
str(UCBAdmission)          # 3D table, N = 4526
```

- have 3 character variables / numbers that tell us how many people in each combination
- console:

```
'table' num [1:2, 1:2, 1:6] 512 313 89 19 353 207 17 8 120 205 ...
- attr(*, "dimnames")=List of 3
 ..$ Admit : chr [1:2] "Admitted" "Rejected"
 ..$ Gender: chr [1:2] "Male" "Female"
 ..$ Dept  : chr [1:6] "A" "B" "C" "D" ...
```

```
UCBAdmissions              # prints 6 tables
```

6 tables = 6 depts / each has 2x2 table m/f - admitted/rejected
adv: elegant compact way to represent the data
disad: dn work for other approaches > want 1 row per observation

```
, , Dept = A

          Gender
Admit      Male Female
  Admitted  512     89
  Rejected  313     19

, , Dept = B

          Gender
Admit      Male Female
  Admitted  353     17
  Rejected  207      8

, , Dept = C

          Gender
Admit      Male Female
  Admitted  120    202
  Rejected  205    391
```

```
, , Dept = D

          Gender
Admit      Male Female
  Admitted  138    131
  Rejected  279    244

, , Dept = E

          Gender
Admit      Male Female
  Admitted   53     94
  Rejected  138    299

, , Dept = F

          Gender
Admit      Male Female
  Admitted   22     24
  Rejected  351    317
```

THE WRONG WAY (bc not many other options)

1.Coerce table to data frame

```
admit1 <- as.data.frame.table(UCBAdmissions)
admit1  # show results
```

Data	
admit1	24 obs. of 4 variables

```
> admit1  # Show results
      Admit Gender Dept Freq
1  Admitted   Male    A  512
2  Rejected   Male    A  313
3  Admitted Female    A   89
4  Rejected Female    A   19
5  Admitted   Male    B  353
6  Rejected   Male    B  207
7  Admitted Female    B   17
8  Rejected Female    B    8
9  Admitted   Male    C  120
10 Rejected   Male    C  205
11 Admitted Female    C  202
12 Rejected Female    C  391
13 Admitted   Male    D  138
14 Rejected   Male    D  279
15 Admitted Female    D  131
16 Rejected Female    D  244
17 Admitted   Male    E   53
18 Rejected   Male    E  138
19 Admitted Female    E   94
20 Rejected Female    E  299
21 Admitted   Male    F   22
22 Rejected   Male    F  351
23 Admitted Female    F   24
24 Rejected Female    F  317
```

2.Repeat each row to math value in Freq (makes a list)

```
admit2 <-lapply(admit1, function(x)rep(x, admit1$Freq))
admit2  # show results
```

admit2	List of 4

```
> admit2  # Show results
$Admit
   [1] Admitted Admitted Admitted
   [9] Admitted Admitted Admitted

$Gender
   [1] Male   Male   Male   Male
  [11] Male   Male   Male   Male

$Dept
   [1] A A A A A A A A A A
  [37] A A A A A A A A A A

$Freq
   [1] 512 512 512 512 512
  [19] 512 512 512 512 512
```

3.Convert from list back to data frame

```
admit3 <- as.data.frame(admit2)
admit3  # show results
```

admit3	4526 obs. of 4 variables

```
> admit3  # Show results
       Admit Gender Dept Freq
1   Admitted   Male    A  512
2   Admitted   Male    A  512
3   Admitted   Male    A  512
4   Admitted   Male    A  512
```

4.Remove the last column, which has the frequencies

```
admit4 <- admit3[, -4]
admit4   # show results
```

admit4 | 4526 obs. of 3 variables

```
> admit4  # Show results
      Admit Gender Dept
1  Admitted   Male    A
2  Admitted   Male    A
```

or- combine all of the steps into a single line

```
admit.rows <-
as.data.frame(lapply(as.data.frame.table(UCBAdmissions)….
```

admit.rows | 4526 obs. of 3 variables

```
> admit.rows  # Show results
      Admit Gender Dept
1  Admitted   Male    A
2  Admitted   Male    A
```

THE RIGHT WAY

```
ucb <- UCBAdmissions %>%     # start with tabular data
       as.tibble() %>%       # convert to tibble with rows (flattens it out)
       uncount(n) %>%        # convert from summary values
       print()               # show results
```

- relatively new command, **`uncount`**:
  - take those frequencies, and split it up, and repeat it however many times need -
- results:

tibble 4,526 x 3
admit <chr> / gender <chr> / dept <chr>

```
# A tibble: 4,526 × 3
   Admit    Gender Dept
   <chr>    <chr>  <chr>
 1 Admitted Male   A
 2 Admitted Male   A
 3 Admitted Male   A
 4 Admitted Male   A
 5 Admitted Male   A
 6 Admitted Male   A
 7 Admitted Male   A
 8 Admitted Male   A
 9 Admitted Male   A
10 Admitted Male   A
# ℹ 4,516 more rows
# ℹ Use `print(n = ...)` to see more rows
```

ucb | 4526 obs. of 3 variables

do in single line:

```
ucb <- UCBAdmissions %>% as.tibble() %>% uncount(n)
```

ANOTHER EXAMPLE

*Use tabular dataset **HairColor***

```
?HairEyeColor  # get info
HairEyeColor   # see data tables
```

- results:

, , Sex = Male

eye colour vs- hair colour

```
> HairEyeColor   # See data tables
, , Sex = Male

       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8

, , Sex = Female

       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
```

Convert table to rows:

```
df <- HairColor %>%                 # start with tabular data
   as.tibble %>%                    # convert to tibble with rows
  uncount(n) %>%                    # convert from summary values
   mutate_all(as_factor) %>%        # convert all variables to factors
   mutate_all(fct_infreq) %>%       # order by descending frequencies
   print()                          # show results
```

- results:

tibble 592 x 3

Hair <fct> / Eye <fct> / Sex <fct>

```
# A tibble: 592 × 3
   Hair  Eye   Sex
   <fct> <fct> <fct>
 1 Black Brown Male
 2 Black Brown Male
 3 Black Brown Male
 4 Black Brown Male
 5 Black Brown Male
 6 Black Brown Male
 7 Black Brown Male
 8 Black Brown Male
 9 Black Brown Male
10 Black Brown Male
# ℹ 582 more rows
# ℹ Use `print(n = ...)` to see more rows
```

df	592 obs. of 3 variables

⇒ ready for ggplot to make graphics and other analysis