Table One

...the easiest way

Paloma Rojas S.

What do we use **Table 1** for?

Table 1. Characteristics of the Patients at Baseline.

Characteristic	Patients Enrolled (N=38)		Patients Treated (N=28)	
	Follicular Lymphoma (N=15)	Diffuse Large B-Cell Lymphoma (N=23)	Follicular Lymphoma (N=14)	Diffuse Large B-Cell Lymphoma (N=14)
Age — yr				
Median	62	56	59	58
Range	43–72	25–77	43–72	25–77
Female sex — no. (%)	8 (53)	7 (30)	7 (50)	3 (21)
Previous therapies				
Median	5	3	5	3
Range	2–10	1–8	2–10	1–8
Advanced stage disease — no. (%)*	13 (87)	17 (74)	12 (86)	9 (64)
Bone marrow involved — no./total no. (%)	4/15 (27)	4/21 (19)	4/14 (28)	3/14 (21)
Elevated lactate dehydrogenase — no. (%)	10 (67)	16 (70)	9 (64)	8 (57)
ECOG performance-status score†				
Median	0	1	0	1
Range	0–1	0–1	0–1	0–1
Refractory diffuse large B-cell lymphoma — no. (%)‡	—	21 (91)	—	12 (86)
Double refractory follicular lymphoma — no. (%)§	9 (60)	—	8 (57)	—
Previous stem-cell transplantation — no. (%)				
Autologous	3 (20)	9 (39)	3 (21)	7 (50)
Allogeneic	1 (7)	0	1 (7)	0

Disclaimers

Before building your Table 1. be sure you know your data

- You already cleaned and prepared your data
- You know which variables are categorical, binary, continuous, discrete and..
- You know how you want to present them according to their properties (ex. mean vs. median)

Let's get started!

Get **tableone**

Install and open the **tableone** package

```
install.packages(tableone)
```

```
library(tableone)
```

Open the example dataset

```
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
```

```
db <- starwars
```

```
db
```

```
## # A tibble: 87 x 13
##    name  height  mass hair_color skin_color eye_color birth_year gender
##    <chr>  <int> <dbl> <chr>      <chr>      <chr>           <dbl> <chr>
##  1 Luke~    172    77 blond      fair       blue             19   male
##  2 C-3PO    167    75 <NA>       gold       yellow          112   <NA>
##  3 R2-D2     96    32 <NA>       white, bl~ red              33   <NA>
##  4 Dart~    202   136 none       white      yellow           41.9 male
##  5 Leia~    150    49 brown      light      brown            19   female
##  6 Owen~    178   120 brown, gr~ light      blue             52   male
##  7 Beru~    165    75 brown      light      blue             47   female
##  8 R5-D4     97    32 <NA>       white, red red              NA   <NA>
##  9 Bigg~    183    84 black      light      brown            24   male
## 10 Obi-~    182    77 auburn, w~ fair       blue-gray        57   male
## # ... with 77 more rows, and 5 more variables: homeworld <chr>,
## #   species <chr>, films <list>, vehicles <list>, starships <list>
```

```
glimpse(db)
```

```
## Observations: 87
## Variables: 13
## $ name       <chr> "Luke Skywalker", "C-3PO", "R2-D2", "Darth Vader", ...
## $ height     <int> 172, 167, 96, 202, 150, 178, 165, 97, 183, 182, 188...
## $ mass       <dbl> 77.0, 75.0, 32.0, 136.0, 49.0, 120.0, 75.0, 32.0, 8...
## $ hair_color <chr> "blond", NA, NA, "none", "brown", "brown, grey", "b...
## $ skin_color <chr> "fair", "gold", "white, blue", "white", "light", "l...
## $ eye_color  <chr> "blue", "yellow", "red", "yellow", "brown", "blue",...
## $ birth_year <dbl> 19.0, 112.0, 33.0, 41.9, 19.0, 52.0, 47.0, NA, 24.0...
## $ gender     <chr> "male", NA, NA, "male", "female", "male", "female",...
## $ homeworld  <chr> "Tatooine", "Tatooine", "Naboo", "Tatooine", "Alder...
## $ species    <chr> "Human", "Droid", "Droid", "Human", "Human", "Human...
## $ films      <list> [<"Revenge of the Sith", "Return of the Jedi", "Th...
## $ vehicles   <list> [<"Snowspeeder", "Imperial Speeder Bike">, <>, <>,...
## $ starships  <list> [<"X-wing", "Imperial shuttle">, <>, <>, "TIE Adva...
```

```
db <- db %>%
  select(-name)
```

First case

You have a dataframe with all the variables to include in your Table1, and they are ready to be summarized (not this case).

```
CreateTableOne(data = db)
```

```
## Warning in CreateTableOne(data = db): Dropping variable(s) films vehicles starships  due to unsupported cl
```

```
##
##                          Overall
##   n                         87
##   height (mean (sd))     174.36 (34.77)
##   mass (mean (sd))        97.31 (169.46)
##   hair_color (%)
##      auburn                  1 ( 1.2)
##      auburn, grey            1 ( 1.2)
##      auburn, white           1 ( 1.2)
##      black                  13 (15.9)
##      blond                   3 ( 3.7)
##      blonde                  1 ( 1.2)
##      brown                  18 (22.0)
##      brown, grey             1 ( 1.2)
##      grey                    1 ( 1.2)
##      none                   37 (45.1)
##      unknown                 1 ( 1.2)
##      white                   4 ( 4.9)
##   skin_color (%)
##      blue                    2 ( 2.3)
##      blue, grey              2 ( 2.3)
##      brown                   4 ( 4.6)
##      brown mottle            1 ( 1.1)
##      brown, white            1 ( 1.1)
##      dark                    6 ( 6.9)
##      fair                   17 (19.5)
##      fair, green, yellow     1 ( 1.1)
##      gold                    1 ( 1.1)
```

Second case

You want only some variables from your dataframe in your table 1

- Save your variable names in a vector
- Define which variables should be presented as categorical. Since they should be factors, the `CreateTableOne` function includes an option `factorVars`.

```
db <- db %>%
  mutate(human = case_when(
    species == "Human" ~ "Human",
    TRUE ~ "Other"))

myVars<- c("height", "mass", "birth_year", "gender", "human")

catVars <- c("gender", "human")
```

Second case

```
table2 <-CreateTableOne(vars = myVars, data = db, factorVars = catVars)
table2
```

```
##
##                            Overall
##   n                           87
##   height (mean (sd))      174.36 (34.77)
##   mass (mean (sd))         97.31 (169.46)
##   birth_year (mean (sd))   87.57 (154.69)
##   gender (%)
##      female                   19 (22.6)
##      hermaphrodite             1 ( 1.2)
##      male                     62 (73.8)
##      none                      2 ( 2.4)
##   human = Other (%)           52 (59.8)
```

Options

- In case you want all levels for your categorical variables, use the option `showAllLevels`
- If you have skewed numerical variables, and want to present as median and IQR, use option `nonnormal`.To do this this save the names of the variables as a vector.

```
skewed <- c("mass", "birth_year")

print(table2, showAllLevels = TRUE, nonnormal = skewed, explain = FALSE)
```

```
##
##              level         Overall
##    n                           87
##    height                  174.36 (34.77)
##    mass                     79.00 [55.60, 84.50]
##    birth_year               52.00 [35.00, 72.00]
##    gender     female           19 (22.6)
##               hermaphrodite     1 ( 1.2)
##               male             62 (73.8)
##               none              2 ( 2.4)
##    human      Human            35 (40.2)
##               Other            52 (59.8)
```

- If you don't want the description (mean, sd, %), set `explain = FALSE`

For more options to print, `?print.TableOne`

Third case, by groups

- We want to see columns for each treatment. Use option `strata`
- By default, the table will include p-values that compare the characteristics between groups.
- It will use chi2 for categorical variables, tttest/Anova for numerical variables.

```
table3 <-CreateTableOne(vars = myVars, strata = "human",
                        data = db, factorVars = catVars)
```

```
table3
```

```
##                          Stratified by human
##                           Human          Other           p      test
##   n                           35             52
##   height (mean (sd))     176.65 (12.54) 172.94 (43.27)   0.644
##   mass (mean (sd))        82.78 (19.38) 105.95 (214.10)  0.616
##   birth_year (mean (sd))  53.41 (24.65) 135.00 (232.73)  0.088
##   gender (%)                                             0.494
##      female                9 (25.7)      10 ( 20.4)
##      hermaphrodite         0 ( 0.0)       1 (  2.0)
##      male                 26 (74.3)      36 ( 73.5)
##      none                  0 ( 0.0)       2 (  4.1)
##   human = Other (%)        0 ( 0.0)      52 (100.0)  <0.001
```

- If you set variables as nonnormal, it will use Wilcoxon or Kruskal Wallis (in print fx)
- For categorical variables with small cell counts, use the option exact (in print fx)

```
print(table3, nonnormal = skewed, exact = "gender", explain = FALSE)
```

```
##                    Stratified by human
##                     Human                  Other                  p
##   n                     35                     52
##   height            176.65 (12.54)         172.94 (43.27)          0.644
##   mass               79.00 [77.00, 84.00]   74.00 [50.00, 87.00]   0.177
##   birth_year         48.00 [31.50, 72.00]   52.50 [40.25, 84.50]   0.649
##   gender                                                           0.712
##      female              9 (25.7)              10 ( 20.4)
##      hermaphrodite       0 ( 0.0)               1 (  2.0)
##      male               26 (74.3)              36 ( 73.5)
##      none                0 ( 0.0)               2 (  4.1)
##   human = Other          0 ( 0.0)              52 (100.0)         <0.001
##                    Stratified by human
##                     test
##   n
##   height
##   mass              nonnorm
##   birth_year        nonnorm
##   gender            exact
##      female
##      hermaphrodite
##      male
##      none
##   human = Other
```

Cool extra to check missing data

```
summary(table2)
```

```
##
##      ### Summary of continuous variables ###
##
## strata: Overall
##             n miss p.miss mean  sd median p25 p75 min  max skew kurt
## height     87    6      7  174  35    180 167 191  66  264   -1    2
## mass       87   28     32   97 169     79  56  84  15 1358    7   55
## birth_year 87   44     51   88 155     52  35  72   8  896    4   21
##
## ============================================================================
##
##      ### Summary of categorical variables ###
##
## strata: Overall
##     var  n miss p.miss         level freq percent cum.percent
##  gender 87    3    3.4        female   19    22.6        22.6
##                       hermaphrodite    1     1.2        23.8
##                                male   62    73.8        97.6
##                                none    2     2.4       100.0
##
##   human 87    0    0.0         Human   35    40.2        40.2
##                                Other   52    59.8       100.0
##
```

Exporting

If your table is going to word file, you can export int a .csv and make it pretty your way.

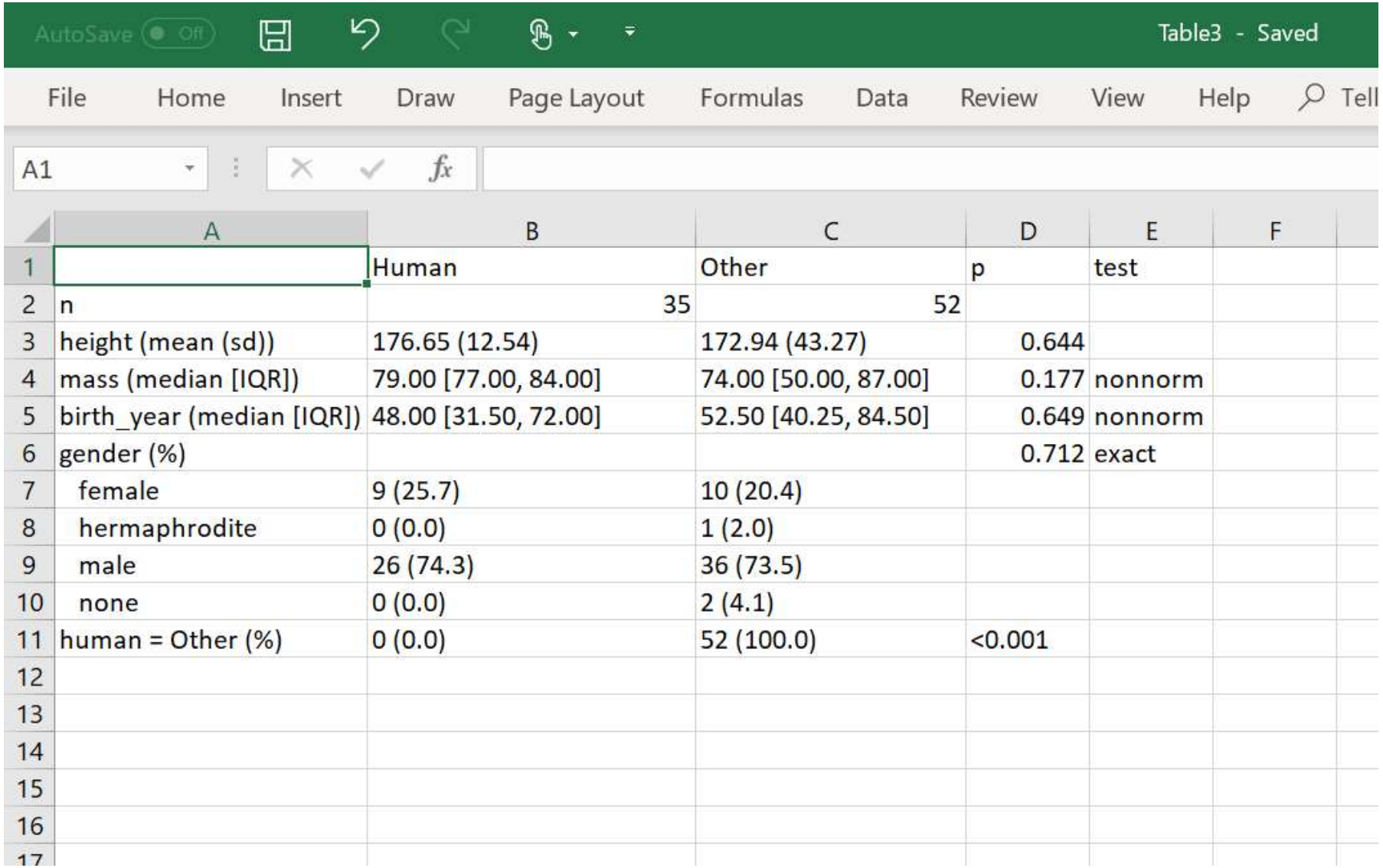
- Save your `print` code adding a few extra statements:

```
table3_exp<- print(table3, nonnormal = skewed,
                   exact = "gender", explain = TRUE,
                   test = TRUE, quote = FALSE, noSpaces = TRUE,
                   printToggle = FALSE)
```

- Export as a CSV file

```
write.csv(table3_exp, file = "Table3.csv")
```

Ready to be tunned



	A	B	C	D	E	F
1		Human	Other	p	test	
2	n	35	52			
3	height (mean (sd))	176.65 (12.54)	172.94 (43.27)	0.644		
4	mass (median [IQR])	79.00 [77.00, 84.00]	74.00 [50.00, 87.00]	0.177	nonnorm	
5	birth_year (median [IQR])	48.00 [31.50, 72.00]	52.50 [40.25, 84.50]	0.649	nonnorm	
6	gender (%)			0.712	exact	
7	female	9 (25.7)	10 (20.4)			
8	hermaphrodite	0 (0.0)	1 (2.0)			
9	male	26 (74.3)	36 (73.5)			
10	none	0 (0.0)	2 (4.1)			
11	human = Other (%)	0 (0.0)	52 (100.0)	<0.001		
12						
13						
14						
15						
16						
17						

If you are use RMarkdown

```
install.packages(kableExtra)
```

```
library(kableExtra)
```

Use kableExtra and have fun!!

```
library(kableExtra)
kable(table3_exp) %>%
  kable_styling(bootstrap_options = c("hover", "condensed", "responsive"),
                full_width = F, font_size = 16)
```

	Human	Other	p	test
n	35	52		
height (mean (sd))	176.65 (12.54)	172.94 (43.27)	0.644	
mass (median [IQR])	79.00 [77.00, 84.00]	74.00 [50.00, 87.00]	0.177	nonnorm
birth_year (median [IQR])	48.00 [31.50, 72.00]	52.50 [40.25, 84.50]	0.649	nonnorm
gender (%)			0.712	exact
female	9 (25.7)	10 (20.4)		
hermaphrodite	0 (0.0)	1 (2.0)		
male	26 (74.3)	36 (73.5)		
none	0 (0.0)	2 (4.1)		
human = Other (%)	0 (0.0)	52 (100.0)	<0.001	

More on style!

```
table_kable<- kable(table3_exp) %>%
  kable_styling(bootstrap_options = c("hover", "condensed", "responsive"),
                full_width = F, font_size = 16) %>%
  row_spec(0, bold = T, color = " white", background = "#562457") %>%
  column_spec(1, bold = T, color = " white",
              border_right = T, background = "#88398A") %>%
  add_header_above(c("Star Wars character's description" = 5)) %>%
  footnote(general = "Data from dplyr package",
           symbol = "nonnorm refers to Mann Whitney test",
           alphabet = "exact refers to Fisher exact test")
```

```
table_kable
```

Star Wars character's description				
	Human	Other	p	test
n	35	52		
height (mean (sd))	176.65 (12.54)	172.94 (43.27)	0.644	
mass (median [IQR])	79.00 [77.00, 84.00]	74.00 [50.00, 87.00]	0.177	nonnorm
birth_year (median [IQR])	48.00 [31.50, 72.00]	52.50 [40.25, 84.50]	0.649	nonnorm
gender (%)			0.712	exact
female	9 (25.7)	10 (20.4)		
hermaphrodite	0 (0.0)	1 (2.0)		
male	26 (74.3)	36 (73.5)		
none	0 (0.0)	2 (4.1)		
human = Other (%)	0 (0.0)	52 (100.0)	<0.001	
Note:				
Data from dplyr package				
[a] exact refers to Fisher exact test				
[*] nonnorm refers to Mann Whitney test				

All info is available at:

Table one package description: https://cran.r-project.org/web/packages/tableone/vignettes/introduction.html

KableExtra features: https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html

Slides created via the R package: https://github.com/yihui/xaringan


Thank
you!
Pusheen.com