Tables With Arsenal Package

Melinda Higgins

2/28/2021

Arsenal Package

The `arsenal` package is described on CRAN as:

> "An Arsenal of 'R' functions for large-scale statistical summaries, which are streamlined to work within the latest reporting tools in 'R' and 'RStudio' and which use formulas and versatile summary statistics for summary tables and models. The primary functions include tableby(), a Table-1-like summary of multiple variable types 'by' the levels of one or more categorical variables; paired(), a Table-1-like summary of multiple variable types paired across two time points; modelsum(), which performs simple model fits on one or more endpoints for many variables (univariate or adjusted for covariates); freqlist(), a powerful frequency table across many categorical variables; comparedf(), a function for comparing data.frames; and write2(), a function to output tables to a document."

The documentation for the `arsenal` package can be found at:

- CRAN https://cran.r-project.org/web/packages/arsenal/index.html
- custom `pkgdown` website https://mayoverse.github.io/arsenal/
- on Github https://github.com/mayoverse/arsenal

But the real beauty of the `arsenal` package is that it "knits" to HTML, PDF and DOCX nicely to all 3 formats!! The `knitr::kable()` function also "knits" to these 3 formats well, but `kable()` only makes nicely formatted tables for "rectangular" data that is already compiled like these object formats:

- `table`
- `data.frame`
- `matrix`
- and *(I think)* `data.table`

The `arsenal` package takes this table formatting a step further by creating "smarter" tables that provide summary statistics in a table format by variables of choice. You can also create tables by groupings with statistical comparison tests performed with p-values added to those tables.

Example Dataset - Palmer Penguins

Let's load the Palmer Penguins dataset from the `palmerpenguins` package.

```
library(palmerpenguins)
library(dplyr)
```

Let's use the `arsenal` package to get some summary statistics for the body measurements.

First let's use the basic `summary()` and the `knitr::kable()` functions to make a simple summary table.

```
penguins %>%
  select(bill_length_mm, bill_depth_mm,
         flipper_length_mm, body_mass_g) %>%
  summary() %>%
  knitr::kable()
```

bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
Min. :32.10	Min. :13.10	Min. :172.0	Min. :2700
1st Qu.:39.23	1st Qu.:15.60	1st Qu.:190.0	1st Qu.:3550
Median :44.45	Median :17.30	Median :197.0	Median :4050
Mean :43.92	Mean :17.15	Mean :200.9	Mean :4202
3rd Qu.:48.50	3rd Qu.:18.70	3rd Qu.:213.0	3rd Qu.:4750
Max. :59.60	Max. :21.50	Max. :231.0	Max. :6300
NA's :2	NA's :2	NA's :2	NA's :2

This code works without too much trouble or added "knitr" options.

Here is a similar table using the `tableby()` function from the `arsenal` package.

After loading the package, you have to first make a "table" using the `tableby()` function. For now we will NOT list a "grouping" variable before the `~` symbol. After the `~` symbol, simply list the variables you want a summary of adding each using the plus `+` operator.

```
# load package
library(arsenal)

# make table with tableby
tab1 <- tableby(~ bill_length_mm + bill_depth_mm +
                  flipper_length_mm + body_mass_g,
                data = penguins)
```

Now that the table is created and saved as an object `tab1`, we can then create the table using the `summary()` function - technically this is calling the `summary.tableby()` function from the `arsenal` package.

You'll notice that the output is actually showing the markdown synatx for creating the table which is not what we want to see.

```
summary(tab1)
```

```
##
##
```

```
## |                           |   Overall (N=344)   |
## |:--------------------------|:-------------------:|
## |**bill_length_mm**         |                     |
## |   N-Miss   |          2          |
## |   Mean (SD)|   43.922 (5.460)    |
## |   Range    |   32.100 - 59.600   |
## |**bill_depth_mm**          |                     |
## |   N-Miss   |          2          |
## |   Mean (SD)|   17.151 (1.975)    |
## |   Range    |   13.100 - 21.500   |
## |**flipper_length_mm**      |                     |
## |   N-Miss   |          2          |
## |   Mean (SD)|  200.915 (14.062)   |
## |   Range    |  172.000 - 231.000  |
## |**body_mass_g**            |                     |
## |   N-Miss   |          2          |
## |   Mean (SD)| 4201.754 (801.955)  |
## |   Range    | 2700.000 - 6300.000 |
```

So, BEFORE this will "print" correctly for each format when "knitted", you must add `results="asis"` to the R code chunk options - see below.

```
```{r results="asis"}
summary(tab1)
```
```

Now the table should look correctly formatted.

	Overall (N=344)
bill_length_mm	
N-Miss	2
Mean (SD)	43.922 (5.460)
Range	32.100 - 59.600
bill_depth_mm	
N-Miss	2
Mean (SD)	17.151 (1.975)
Range	13.100 - 21.500
flipper_length_mm	
N-Miss	2
Mean (SD)	200.915 (14.062)
Range	172.000 - 231.000
body_mass_g	
N-Miss	2
Mean (SD)	4201.754 (801.955)
Range	2700.000 - 6300.000

""

Add better labels

Let's clean this table up a little bit. We'll add labels that are better to read than the native variable name using the `attr()` attributes function.

```
attr(penguins$bill_length_mm,'label') <-
  'Bill Length (mm)'
attr(penguins$bill_depth_mm,'label') <-
  'Bill Depth (mm)'
attr(penguins$flipper_length_mm,'label') <-
  'Flipper Length (mm)'
attr(penguins$body_mass_g,'label') <-
  'Body Mass (g)'
```

Now make the table again.

```
# make table with tableby
tab1 <- tableby(~ bill_length_mm + bill_depth_mm +
                  flipper_length_mm + body_mass_g,
                data = penguins)
summary(tab1)
```

	Overall (N=344)
Bill Length (mm)	
N-Miss	2
Mean (SD)	43.922 (5.460)
Range	32.100 - 59.600
Bill Depth (mm)	
N-Miss	2
Mean (SD)	17.151 (1.975)
Range	13.100 - 21.500
Flipper Length (mm)	
N-Miss	2
Mean (SD)	200.915 (14.062)
Range	172.000 - 231.000
Body Mass (g)	
N-Miss	2
Mean (SD)	4201.754 (801.955)
Range	2700.000 - 6300.000

Use non-parametric summary statistics - median and IQR

Let's update these stats to include non-parametric statisics:

- N - amount of non-missing data
- Nmiss - number of missing
- Median
- q1q3 - the 25th and 75th quartiles
- range

See more options by reading the help page for `tableby.stats()`.

```
# make table with tableby
tab1 <- tableby(~ bill_length_mm + bill_depth_mm +
                  flipper_length_mm + body_mass_g,
                numeric.stats=c("N","Nmiss",
                                "median","q1q3",
                                "range"),
                data = penguins)
summary(tab1)
```

	Overall (N=344)
Bill Length (mm)	
N	342
N-Miss	2
Median	44.450
Q1, Q3	39.225, 48.500
Range	32.100 - 59.600
Bill Depth (mm)	
N	342
N-Miss	2
Median	17.300
Q1, Q3	15.600, 18.700
Range	13.100 - 21.500
Flipper Length (mm)	
N	342
N-Miss	2
Median	197.000
Q1, Q3	190.000, 213.000
Range	172.000 - 231.000
Body Mass (g)	
N	342
N-Miss	2
Median	4050.000
Q1, Q3	3550.000, 4750.000
Range	2700.000 - 6300.000

Categorical data

For variables that are character or Factor type variables, the `arsenal` package will create a table of counts/frequencies and percentages.

Let's make a table for the counts and percentages for the categorical variables:

- `species`
- `island`
- `sex`

Notice that the percents are the percentage of the total number of non-missing. So for `sex`, there are 11 penguins who don't have `sex` data. So the percentages are computed for 344-11 = 333.

Missing data is listed by default.

```
tab2 <- tableby(~ species + island + sex,
                data = penguins)
summary(tab2)
```

	Overall (N=344)
species	
Adelie	152 (44.2%)
Chinstrap	68 (19.8%)
Gentoo	124 (36.0%)
island	
Biscoe	168 (48.8%)
Dream	124 (36.0%)
Torgersen	52 (15.1%)
sex	
N-Miss	11
female	165 (49.5%)
male	168 (50.5%)

Let's look at statistics by group

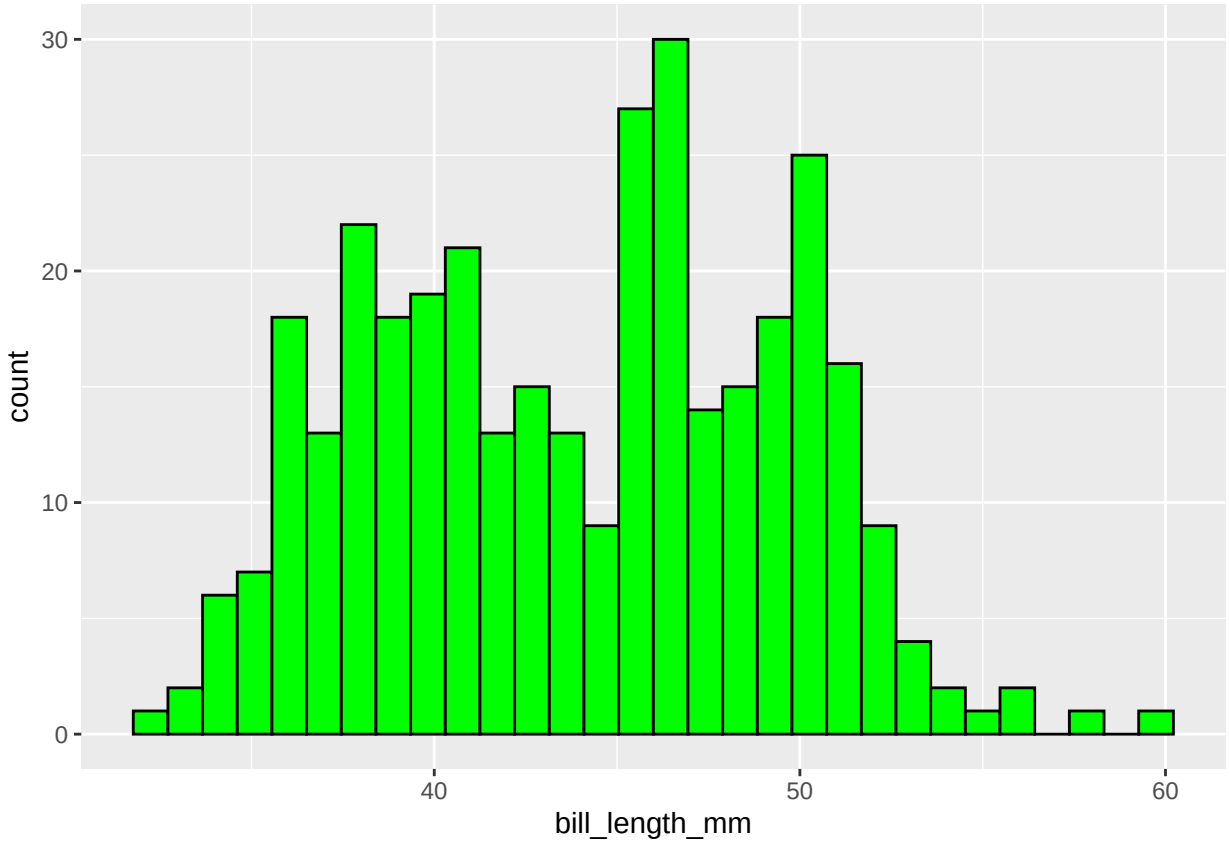
Another feature of the `tableby()` function is that it easily creates tables of statistics by group. This function also adds a "test of group differences" automatically. The test can be turned off as needed and customized for either parametric (anova) or non-parametric (kwt = Kruskal-Wallis) tests.

Test for normality assumptions first

Let's look at the body size measurements and see if they meet the assumptions of normality first. We can look at histograms and Q-Q Plots (using `qqPlot()` from the `car` package). We could also perform the Shapiro-Wilks test for normality.

Here is an example for `bill_length_mm`.

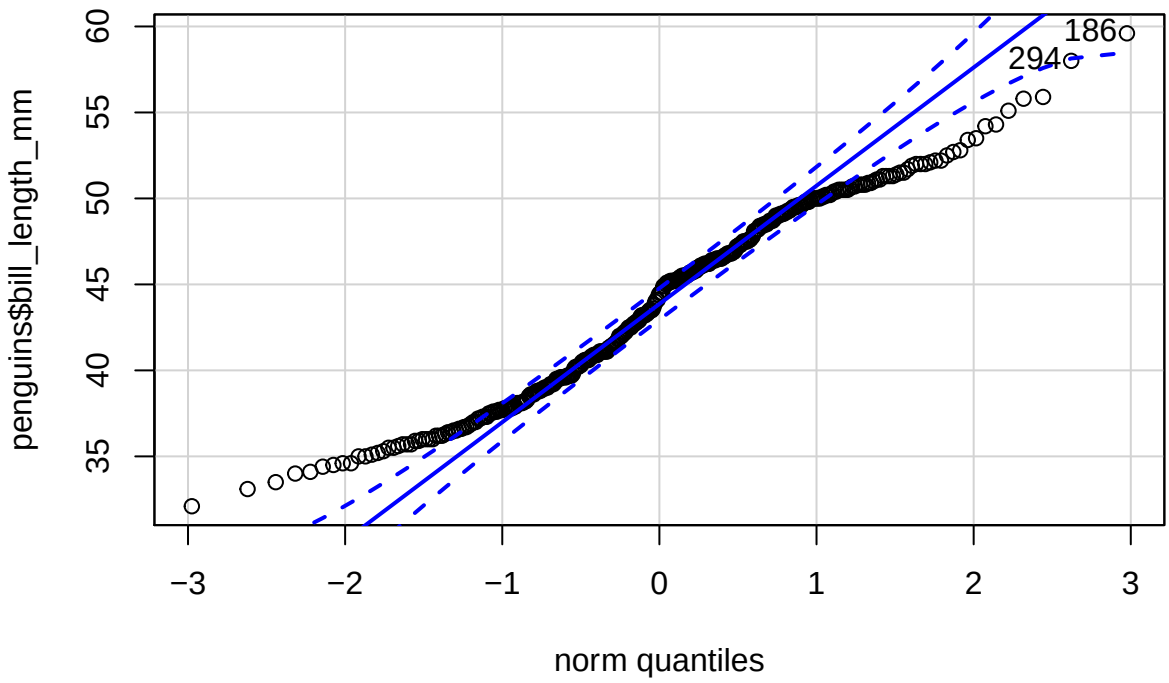
```
library(ggplot2)
ggplot(penguins, aes(bill_length_mm)) +
  geom_histogram(color = "black", fill = "green")
```



Notice that this is somewhat bi-modal.

Here is the Q-Q Plot - this is reverse S-shaped with points falling off the line indicating deviations from normality.

```
library(car)
car::qqPlot(penguins$bill_length_mm)
```




```
## [1] 186 294
```

Here is the Shapiro-Wilks' test.

```
shapiro.test(penguins$bill_length_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  penguins$bill_length_mm
## W = 0.97485, p-value = 1.12e-05
```

This is significant - also indicating deviations from normality.

Let's look at the other body size variables. I have grouped these for easier printing using the `patchwork` package.

NOTE: For the `patchwork` package to work, I had to switch to the `ggpubr` package to make the Q-Q Plot using `ggqqplot()` to break a ggplot plot object and save it.

```
library(patchwork)
library(ggpubr)

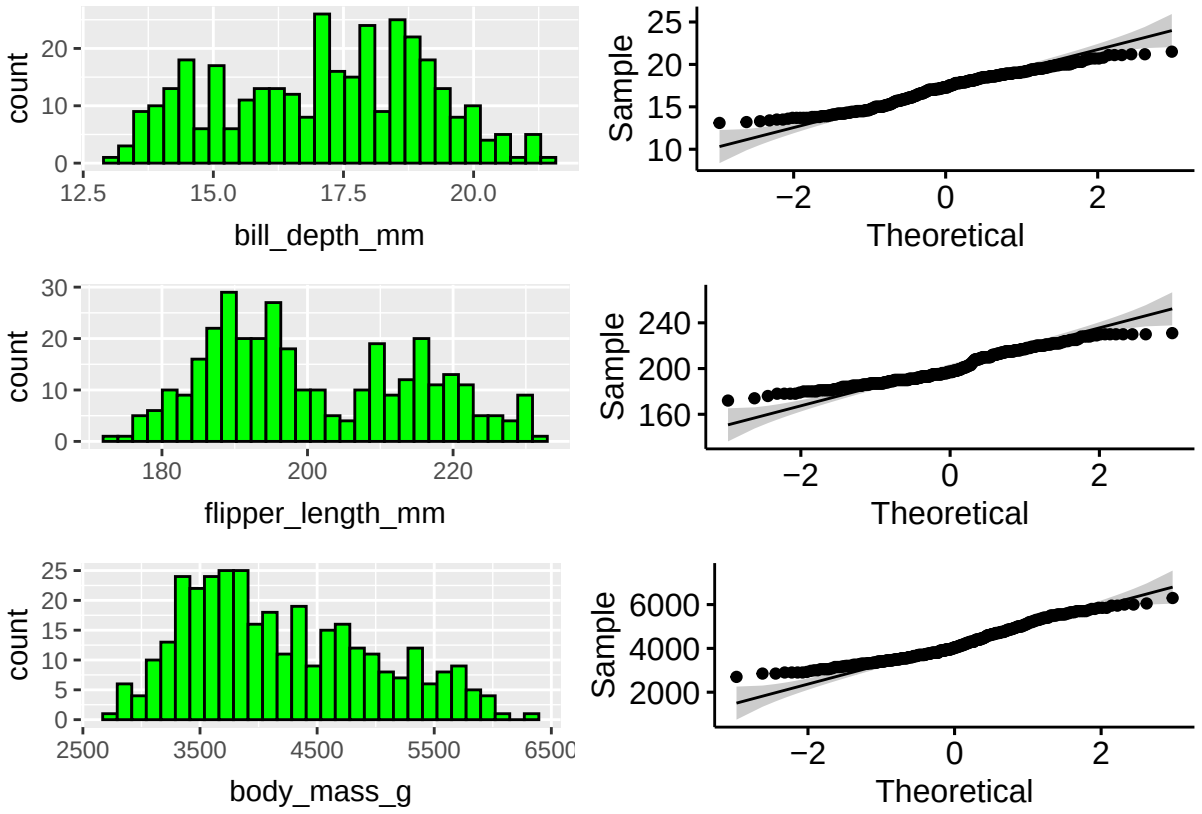
p1 <- ggplot(penguins, aes(bill_depth_mm)) +
  geom_histogram(color = "black", fill = "green")
```

```
p2 <- ggqqplot(penguins$bill_depth_mm)

p3 <- ggplot(penguins, aes(flipper_length_mm)) +
  geom_histogram(color = "black", fill = "green")
p4 <- ggqqplot(penguins$flipper_length_mm)

p5 <- ggplot(penguins, aes(body_mass_g)) +
  geom_histogram(color = "black", fill = "green")
p6 <- ggqqplot(penguins$body_mass_g)

(p1 | p2) / (p3 | p4) / (p5 | p6)
```



And for completeness, here are the Shapiro-Wilks tests for the rest of these variables.

```
shapiro.test(penguins$bill_depth_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  penguins$bill_depth_mm
## W = 0.97258, p-value = 4.419e-06
```

```
shapiro.test(penguins$flipper_length_mm)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  penguins$flipper_length_mm
## W = 0.95155, p-value = 3.54e-09
```

```
shapiro.test(penguins$body_mass_g)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  penguins$body_mass_g
## W = 0.95921, p-value = 3.679e-08
```

These are all significant. So, we should probably run non-parametric statistics and tests. Although the sample size here is good (>300) and the deviations are not too bad. We could probably be ok with either ANOVA or Krusal-Wallis Tests. Let's run both and compare.

Parametric statistics and ANOVA tests for group differences

We'll use the code above as our guide. Let's look at the body size measurements by species and perform ANOVA tests for the group differences.

Notice I put `species` before the `~` operator in the "formula" below.

```
tab1g <- tableby(species ~ bill_length_mm +
                    bill_depth_mm +
                    flipper_length_mm + body_mass_g,
                 data = penguins)
summary(tab1g)
```

	Adelie (N=152)	Chinstrap (N=68)	Gentoo (N=124)	Total (N=344)	p value
Bill Length (mm)					< 0.001
N-Miss	1	0	1	2	
Mean (SD)	38.791 (2.663)	48.834 (3.339)	47.505 (3.082)	43.922 (5.460)	
Range	32.100 - 46.000	40.900 - 58.000	40.900 - 59.600	32.100 - 59.600	
Bill Depth (mm)					< 0.001
N-Miss	1	0	1	2	
Mean (SD)	18.346 (1.217)	18.421 (1.135)	14.982 (0.981)	17.151 (1.975)	
Range	15.500 - 21.500	16.400 - 20.800	13.100 - 17.300	13.100 - 21.500	
Flipper Length (mm)					< 0.001
N-Miss	1	0	1	2	
Mean (SD)	189.954 (6.539)	195.824 (7.132)	217.187 (6.485)	200.915 (14.062)	
Range	172.000 - 210.000	178.000 - 212.000	203.000 - 231.000	172.000 - 231.000	
Body Mass (g)					< 0.001
N-Miss	1	0	1	2	
Mean (SD)	3700.662 (458.566)	3733.088 (384.335)	5076.016 (504.116)	4201.754 (801.955)	

	Adelie (N=152)	Chinstrap (N=68)	Gentoo (N=124)	Total (N=344)	p value
Range	2850.000 - 4775.000	2700.000 - 4800.000	3950.000 - 6300.000	2700.000 - 6300.000	

Non-parametric statistics and the KW tests for group differences

Now let's run this again but using the non-parametric KW (Kruskal-Wallis) tests. To do this we add `numeric.test = "kwt"` as an option.

In addition to running a different statistical tests, I also updated the default summary statistics to the `"median"` and IQR (`"q1q3"`) along with `"Nmiss"` and `"range"`.

```
tab1np <- tableby(species ~ bill_length_mm +
                    bill_depth_mm +
                    flipper_length_mm + body_mass_g,
                  numeric.stats=c("Nmiss","median","q1q3",
                                  "range"),
                  numeric.test = "kwt",
                  data = penguins)
summary(tab1np)
```

	Adelie (N=152)	Chinstrap (N=68)	Gentoo (N=124)	Total (N=344)	p value
Bill Length (mm)					< 0.001
N-Miss	1	0	1	2	
Median	38.800	49.550	47.300	44.450	
Q1, Q3	36.750, 40.750	46.350, 51.075	45.300, 49.550	39.225, 48.500	
Range	32.100 - 46.000	40.900 - 58.000	40.900 - 59.600	32.100 - 59.600	
Bill Depth (mm)					< 0.001
N-Miss	1	0	1	2	
Median	18.400	18.450	15.000	17.300	
Q1, Q3	17.500, 19.000	17.500, 19.400	14.200, 15.700	15.600, 18.700	
Range	15.500 - 21.500	16.400 - 20.800	13.100 - 17.300	13.100 - 21.500	
Flipper Length (mm)					< 0.001
N-Miss	1	0	1	2	
Median	190.000	196.000	216.000	197.000	
Q1, Q3	186.000, 195.000	191.000, 201.000	212.000, 221.000	190.000, 213.000	
Range	172.000 - 210.000	178.000 - 212.000	203.000 - 231.000	172.000 - 231.000	
Body Mass (g)					< 0.001
N-Miss	1	0	1	2	
Median	3700.000	3700.000	5000.000	4050.000	
Q1, Q3	3350.000, 4000.000	3487.500, 3950.000	4700.000, 5500.000	3550.000, 4750.000	
Range	2850.000 - 4775.000	2700.000 - 4800.000	3950.000 - 6300.000	2700.000 - 6300.000	

Regardless of our approach, we can see that there are significant differences between these body size measurements by species.

Categorical data by groups - Chi-square tests

Let's look at the breakdown of the penguins by species for their location (`island`) and `sex`.

By default, the statistical tests runs for categorical data is the Chi-square test of independence.

```
tab2g <- tableby(species ~ island + sex,
                 data = penguins)
summary(tab2g)
```

	Adelie (N=152)	Chinstrap (N=68)	Gentoo (N=124)	Total (N=344)	p value
island					< 0.001
Biscoe	44 (28.9%)	0 (0.0%)	124 (100.0%)	168 (48.8%)	
Dream	56 (36.8%)	68 (100.0%)	0 (0.0%)	124 (36.0%)	
Torgersen	52 (34.2%)	0 (0.0%)	0 (0.0%)	52 (15.1%)	
sex					0.976
N-Miss	6	0	5	11	
female	73 (50.0%)	34 (50.0%)	58 (48.7%)	165 (49.5%)	
male	73 (50.0%)	34 (50.0%)	61 (51.3%)	168 (50.5%)	

We could also force it to run the Fisher's Exact tests using the "fe" option. Learn more for `help("tableby.control")`. Also see the vignette at https://cran.r-project.org/web/packages/arsenal/vignettes/tableby.html.

In the code below, I customized the table to show the results of a Chi-square test for the `species ~ island` cross table and the Fisher's exact tests for the `species ~ sex` cross table results.

```
tab2g <- tableby(species ~ chisq(island) + fe(sex),
                 data = penguins)
summary(tab2g)
```

	Adelie (N=152)	Chinstrap (N=68)	Gentoo (N=124)	Total (N=344)	p value
island					< 0.001
Biscoe	44 (28.9%)	0 (0.0%)	124 (100.0%)	168 (48.8%)	
Dream	56 (36.8%)	68 (100.0%)	0 (0.0%)	124 (36.0%)	
Torgersen	52 (34.2%)	0 (0.0%)	0 (0.0%)	52 (15.1%)	
sex					0.979
N-Miss	6	0	5	11	
female	73 (50.0%)	34 (50.0%)	58 (48.7%)	165 (49.5%)	
male	73 (50.0%)	34 (50.0%)	61 (51.3%)	168 (50.5%)	

Make a better table - title and footnotes

We could customize this table a little more by adding a title and setting `pfootnote` to `TRUE` to add details on the custom statistical tests.

```
tab2g <- tableby(species ~ chisq(island) + fe(sex),
                 data = penguins)
summary(tab2g,
        title = "Location and Sex by Species",
        pfootnote = TRUE)
```

Table 10: Location and Sex by Species

	Adelie (N=152)	Chinstrap (N=68)	Gentoo (N=124)	Total (N=344)	p value
island					< 0.001[1]
Biscoe	44 (28.9%)	0 (0.0%)	124 (100.0%)	168 (48.8%)	
Dream	56 (36.8%)	68 (100.0%)	0 (0.0%)	124 (36.0%)	
Torgersen	52 (34.2%)	0 (0.0%)	0 (0.0%)	52 (15.1%)	
sex					0.979[2]
N-Miss	6	0	5	11	
female	73 (50.0%)	34 (50.0%)	58 (48.7%)	165 (49.5%)	
male	73 (50.0%)	34 (50.0%)	61 (51.3%)	168 (50.5%)	

1. Pearson's Chi-squared test
2. Fisher's Exact Test for Count Data