ODSC: ggplot2 Graph Gallery

Hierarchies|proportions: part-to-whole relationships

by Martin Frigaard & Peter Spangler

Written: September 21 2021

Updated: April 03 2022

Resources :

The graphs

- The `ggplot2` book by Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen

- Data Visualization: A Practical Introduction by Kieran Healy (2018)

- R Graphics Cookbook, 2nd edition by Winston Chang (2022)

Graph Categories

- Fundamentals of Data Visualization by Claus O. Wilke (2019)

- Data Visualisation: A Handbook for Data Driven Design by Andy Kirk (2019)

- Data Points by Nathan Yau (2013)

Graph Categories: *The 'CHRTS' Families of Chart Types*

From *"Data Visualisation: A Handbook for Data Driven Design"*, Andy Kirk (2019)

Comparing categories and distributions

Hierarchies/proportions

Correlations and connections

Trends and intervals over time

Maps, overlays, and/or distortions

Graph Categories: Directory of Visualizations

From *"Fundamentals of Data Visualization"*, Claus O. Wilke (2019)

Amounts

Distributions

Proportions

X–Y relationships

Geospatial Data

Uncertainty

Comprehensive Graph Gallery

Comparing categories and values

- Amounts
- Distributions

Hierarchies and proportions

- *Part-to-whole relationships*

Trends, correlations and connections

- X–Y relationships

Maps, overlays, and distortions

- Geospatial Data

Statistical measures

- Uncertainty

Data

Data come from the following packages:

- **`palmerpenguins`**

- **`fivethirtyeight`**

- **`ggplot2movies`**

Or created using `tribble()`

```
tribble(
  ~`variable 1`, ~`variable 2`,
            "a",             1,
            "b",             2,
            "c",             3)
```

variable 1	variable 2
<chr>	<dbl>
a	1
b	2
c	3

3 rows

Load data packages

```r
library(palmerpenguins)
library(fivethirtyeight)
library(ggplot2movies)
```

palmerpenguins

palmerpenguins package website

```
palmerpenguins::penguins -> penguins
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
<fct>	<fct>	<dbl>	<dbl>	<int>	<int>	<fct>	<int>
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	*NA*	*NA*	*NA*	*NA*	*NA*	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	*NA*	2007
Adelie	Torgersen	42.0	20.2	190	4250	*NA*	2007

1-10 of 344 rows

Previous **1** 2 3 4 5 6 ... 35 Next

fivethirtyeight

fivethirtyeight package website

All datasets are listed below with descriptions

```
datasets("fivethirtyeight")
```

dataset <chr>
US_births_1994_2003
US_births_2000_2014
ahca_polls
airline_safety
antiquities_act
august_senate_polls
avengers
bachelorette
bad_drivers
bechdel

1-10 of 129 rows | 1-1 of 2 columns   Previous **1** 2 3 4 5 6 ... 13 Next

`ggplot2movies`

`ggplot2movies` package website

We're using `movies_data` (derived version of the `ggplot2movies::movies`)

```
movies_data
```

title <chr>	year <int>	length <int>	budget <int>	rating <dbl>	mpaa <fct>
100 Mile Rule	2002	98	1100000	5.6	R
13 Going On 30	2004	98	37000000	6.4	PG-13
15 Minutes	2001	120	42000000	6.1	R
2 Fast 2 Furious	2003	107	76000000	5.1	PG-13
2046	2004	129	12000000	7.6	R
21 Grams	2003	124	20000000	8.0	R
25th Hour	2002	135	15000000	7.8	R
3000 Miles to Graceland	2001	125	62000000	5.4	R
40 Days and 40 Nights	2002	96	17000000	5.4	R
50 First Dates	2004	99	75000000	6.8	PG-13

1-10 of 751 rows | 1-6 of 7 columns

Previous **1** 2 3 4 5 6 ... 76 Next

Hierarchies/proportions

Part-to-whole relationships:

Part-to-whole relationships: *Pie charts*

Pie-charts (`ggpubr::ggpie`) are ideal for comparing the proportions of categorical variable values.

"In general, pie charts work well when the goal is to emphasize simple fractions, such as one-half, one-third, or one-quarter."

"They also work well when we have very small datasets." - Claus O. Wilke, Fundamentals of Data Visualization (2019)

Part-to-whole relationships: *Pie charts*

```
movies_mpaa_avg_rating <- tibble::tribble(
    ~mpaa,             ~avg,
     "PG", 5.72621359223301,
  "PG-13", 5.95468164794007,
      "R", 6.04015748031496
  )
movies_mpaa_avg_rating <- mutate(movies_mpaa_avg_rating,
       mpaa = factor(mpaa, levels = c("PG", "PG-13", "R")))
```

mpaa	**avg**
<fct>	<dbl>
PG	5.726214
PG-13	5.954682
R	6.040157

3 rows

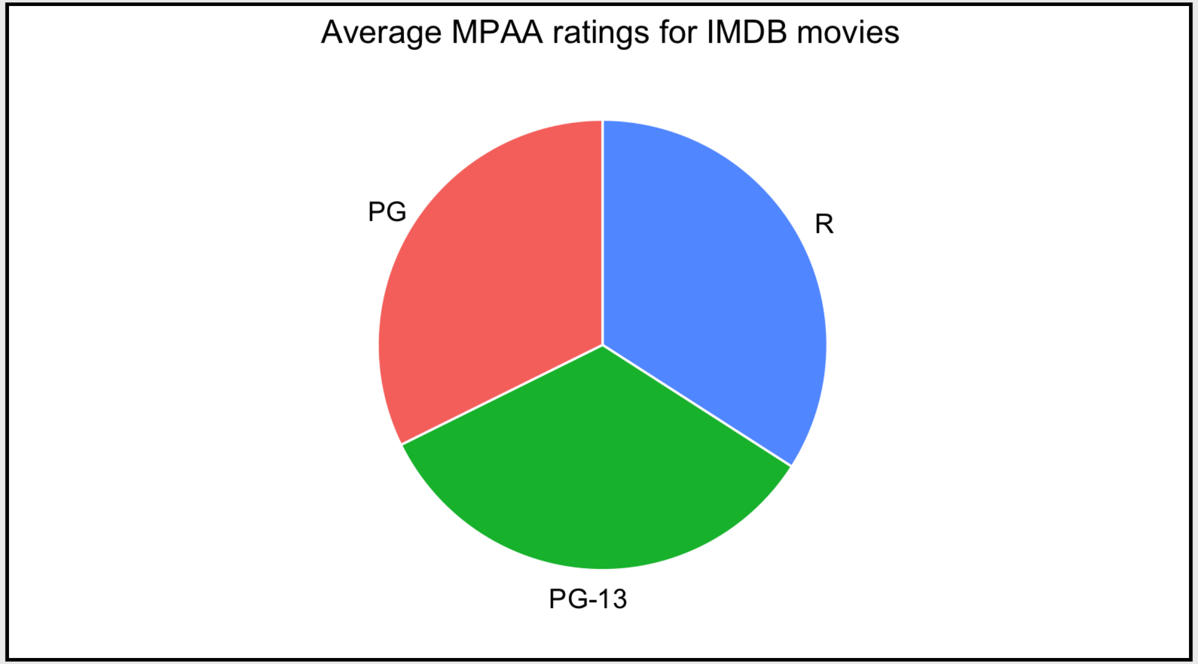
Part-to-whole relationships: *Pie charts*

Map `avg` to the `x` axis, `mpaa` to `label` and `fill`, "white" to `color`, remove the legend and add the labels.

```
labs_pie <- labs(
  x = "Average MPAA rating",
  title = "Average MPAA ratings for IMDB movies")
```

Note that we do not add a geom inside the `ggpie()` function.

```
ggpubr::ggpie(data = movies_mpaa_avg_rating,
                    x = "avg",
                label = "mpaa",
                fill = "mpaa",
                color = "white") +
    # remove legend
    theme(legend.position = "none") +
    labs_pie
```



Part-to-whole relationships: *Stacked-density*

We previously used density graphs to visualize the distribution of a single variable, but stacked density graphs are great for visualizing how proportions vary across numeric (continuous) variables.

Part-to-whole relationships: *Stacked-density*

```
penguins <- palmerpenguins::penguins
penguins_stacked_density <- filter(penguins, !is.na(species))
```

species <fct>	island <fct>	bill_length_mm <dbl>	bill_depth_mm <dbl>	flipper_length_mm <int>	body_mass_g <int>	sex <fct>	year <int>
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	*NA*	*NA*	*NA*	*NA*	*NA*	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	*NA*	2007
Adelie	Torgersen	42.0	20.2	190	4250	*NA*	2007

1-10 of 344 rows

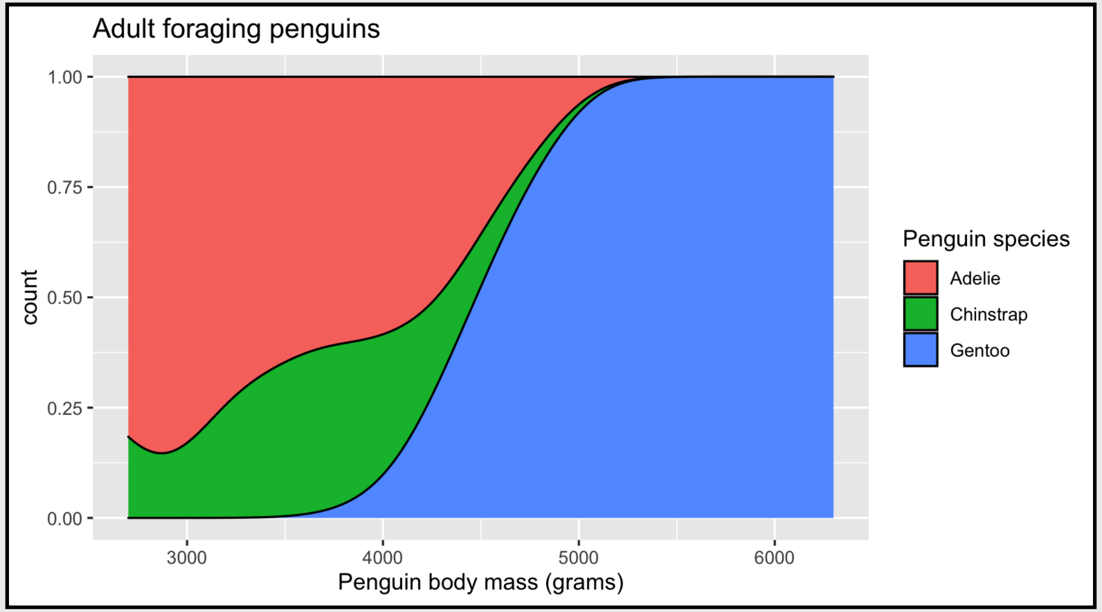
Previous **1** 2 3 4 5 6 ... 35 Next

Part-to-whole relationships: *Stacked-density*

To create a stacked-density graph, map the continuous variable to the `x` variable, and the categorical variable to both the `group` and `fill` aesthetic. We also map `y` to `..count..` to see each relative distributions across a scale of `0.00` to `1.00`, and add the `adjust` and `position` arguments to the `geom_density()`

```
labs_stacked_density <- labs(
  x = "Penguin body mass (grams)",
  title = "Adult foraging penguins",
  fill = "Penguin species")
```

```
ggplot(data = penguins_stacked_density,
       aes(x = body_mass_g,
           y = ..count..,
           group = species,
           fill = species)) +
    geom_density(adjust = 1.5,
                 position = "fill") +
    labs_stacked_density
```



Part-to-whole relationships: *Waffle chart*

Waffle charts use color to display the levels that make up the values in a categorical variable. The counts for each level are divided into separate colors into a square or grid display.

Part-to-whole relationships: *Waffle chart*

Waffle charts require a special data transformation with `ggwaffle::waffle_iron()`. Set the `group` argument in `aes_d()` as the categorical variable you want to see the relative counts for.

```r
penguins <- palmerpenguins::penguins
penguins <- mutate(penguins, species =
as.character(species))
waffle_penguins <- waffle_iron(penguins,
                               aes_d(group = species))
```

	y <int>	x <int>	group <chr>
1	1	1	Adelie
2	2	1	Adelie
3	3	1	Adelie
4	4	1	Adelie
5	5	1	Adelie
6	6	1	Adelie
7	7	1	Adelie
8	8	1	Adelie
9	1	2	Adelie
10	2	2	Adelie

1-10 of 344 rows   Previous **1** 2 3 4 5 6 ... 35 Next

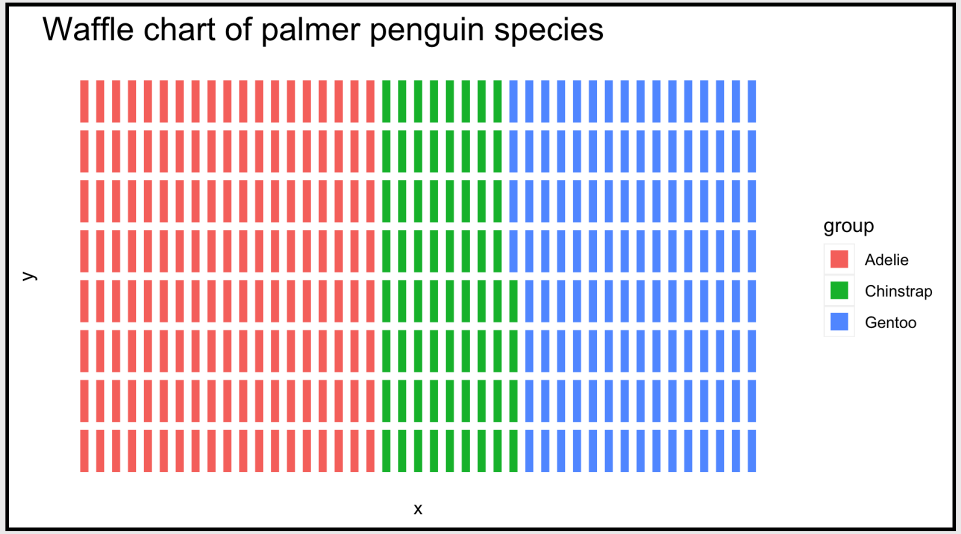
Part-to-whole relationships: *Waffle chart*

Map the `x` and `y` to the `x` and `y` axes, `group` to `fill`, and the labels

We'll also add the `ggwaffle::theme_waffle()` to our plot to remove some of the axis text and ticks.

```r
labs_waffle <- labs(
  title = "Waffle chart of palmer penguin species")
```

```r
ggplot(waffle_penguins,
       aes(x = x,
           y = y,
           fill = group)) +
  geom_waffle() +
  labs_waffle +
  # 'flavour your waffles with a standard theme'
  ggwaffle::theme_waffle()
```



Part-to-whole relationships: *Mosaic plot*

A mosaic plot is similar to a stacked bar-graph, but instead of only relying on height and color to display the relative amount for each value, mosaic plots also use width.

Part-to-whole relationships: *Mosaic plot*

```
flying <- fivethirtyeight::flying
# remove missing from baby
# and unruly_child
flying_mosaic <- filter(flying,
                        !is.na(baby) &
                 !is.na(unruly_child))
```

respondent_id	gender	age	height	children_under_18	household_income
<dbl>	<chr>	<ord>	<ord>	<lgl>	<ord>
3434278696	Male	30-44	6'3"	TRUE	*NA*
3434275578	Male	30-44	5'8"	FALSE	$100,000 - $149,999
3434268208	Male	30-44	5'11"	FALSE	$0 - $24,999
3434250245	Male	30-44	5'7"	FALSE	$50,000 - $99,999
3434245875	Male	30-44	5'9"	TRUE	$25,000 - $49,999
3434235351	Male	30-44	6'2"	TRUE	*NA*
3434218031	Male	30-44	6'0"	TRUE	$0 - $24,999
3434172894	Male	30-44	5'6"	FALSE	$0 - $24,999
3434165659	Male	30-44	6'0"	FALSE	$50,000 - $99,999
3434131535	Male	18-29	6'0"	FALSE	*NA*

1-10 of 849 rows | 1-6 of 27 columns

Previous **1** 2 3 4 5 6 ... 85 Next

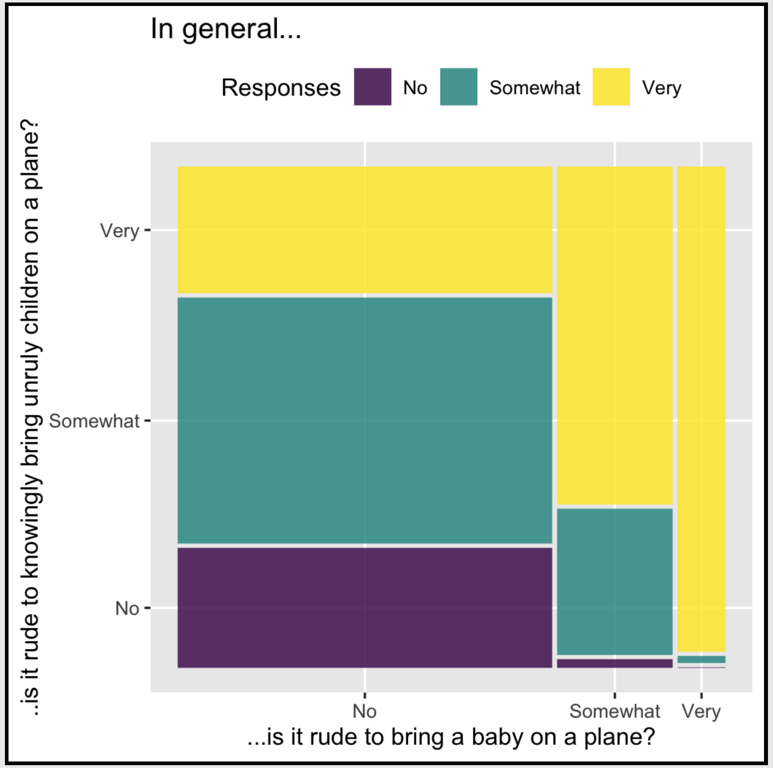
Part-to-whole relationships: *Mosaic plot*

Map the `product()` of `unruly_child` and `baby` to the `x` axis, `unruly_child` to `fill`, and add the labels.

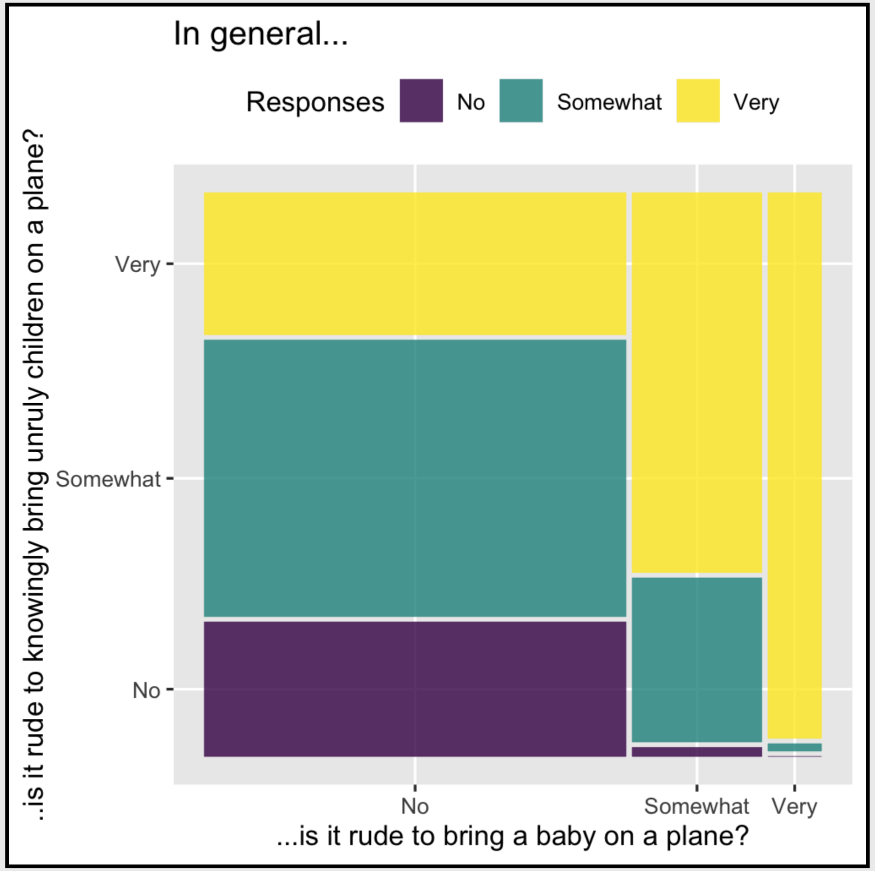
```r
mosaic_labs <- labs(
  title = "In general...",
  x = "...is it rude to bring a baby on a plane?",
  y = "..is it rude to knowingly bring unruly children on a plane?",
  fill = "Responses")
```

Move the legend to the top of the graph with `theme(legend.position = "top")`

```r
ggplot(data = flying_mosaic) +
  geom_mosaic(
    aes(x = product(unruly_child, baby),
                fill = unruly_child)
    ) +
  mosaic_labs +
  theme(legend.position = "top")
```



Part-to-whole relationships: *Mosaic plot*



As we can see, the widths of each rectangle are proportional to the responses to the x axis survey item

and the heights are proportional to the responses to the y axis survey item.

Part-to-whole relationships: *Treemaps*

Treemaps display how numerical hierarchical values make up a whole in a rectangular layout, often referred to as 'squarified', which represents of the 100% total values. We can guild treemaps in ggplot2 with the `treemapify` package

Part-to-whole relationships: *Treemaps*

```
treemap_penguins <- filter(penguins, !is.na(sex))
treemap_penguins_grouped <- group_by(treemap_penguins,
                                    species, island, sex)
treemap_penguins_counts <- ungroup(
                               count(treemap_penguins_grouped,
                                species, island))
```

species <chr>	island <fct>	sex <fct>	n <int>
Adelie	Biscoe	female	22
Adelie	Biscoe	male	22
Adelie	Dream	female	27
Adelie	Dream	male	28
Adelie	Torgersen	female	24
Adelie	Torgersen	male	23
Chinstrap	Dream	female	34
Chinstrap	Dream	male	34
Gentoo	Biscoe	female	58
Gentoo	Biscoe	male	61

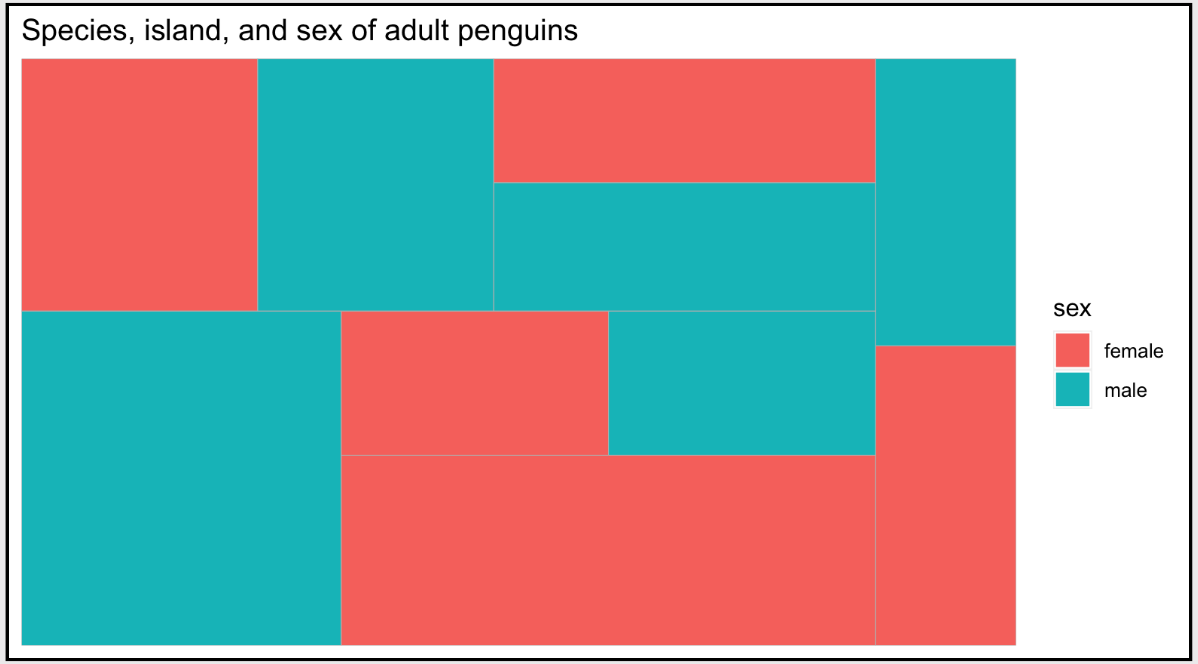
1-10 of 10 rows

Part-to-whole relationships: *Treemaps*

Map the `n` to `area`, `sex` to `fill`, `species` to `label`, `island` to `subgroup` and add the labels

```
labs_treemap <- labs(
  title = "Species, island, and sex of adult penguins")
```

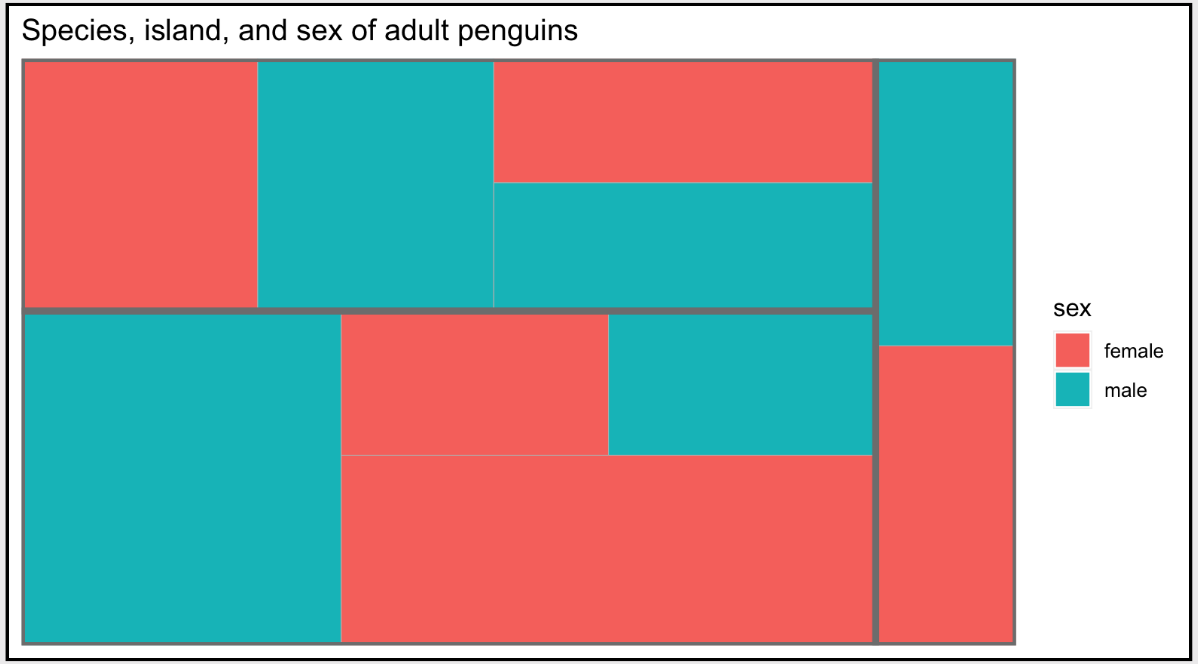
```
ggplot(treemap_penguins_counts,
       aes(area = n,
           fill = sex,
           label = species,
           subgroup = island)) +
       geom_treemap() +
       labs_treemap
```



Part-to-whole relationships: *Treemaps*

Add the borders with `geom_treemap_subgroup_border()`

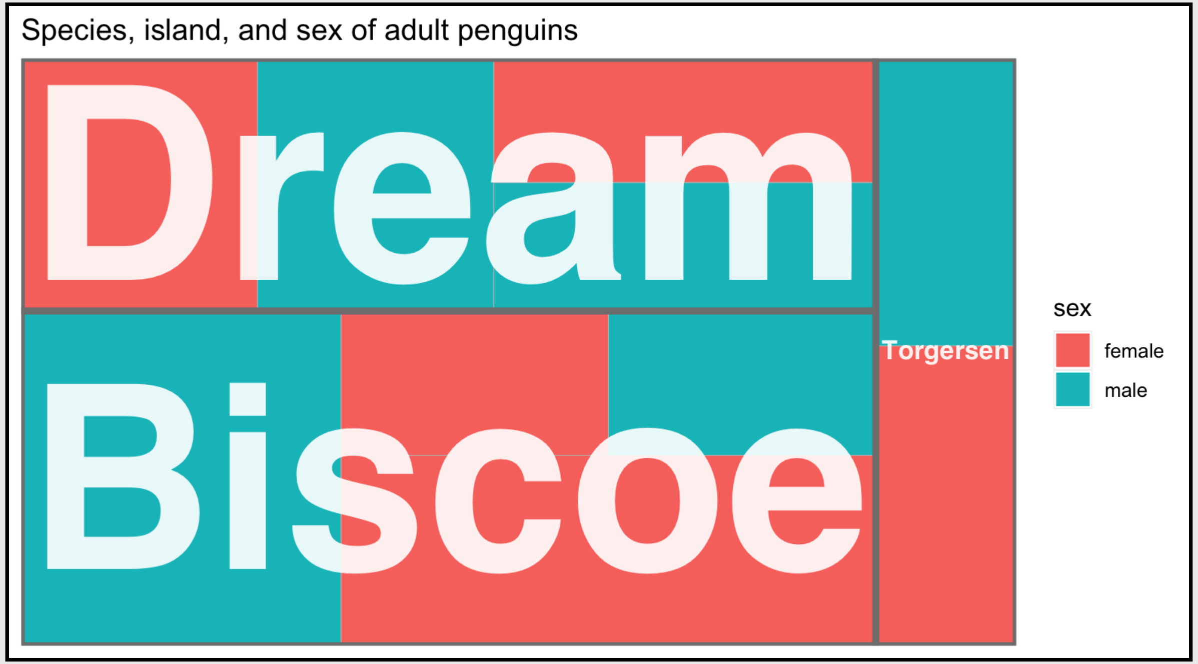
```
ggplot(treemap_penguins_counts,
       aes(area = n,
           fill = sex,
           label = species,
           subgroup = island)) +
       geom_treemap() +
       geom_treemap_subgroup_border() +
       labs_treemap
```



Part-to-whole relationships: *Treemaps*

Include labels for subgroup with `geom_treemap_subgroup_text()` (see full list of arguments here)

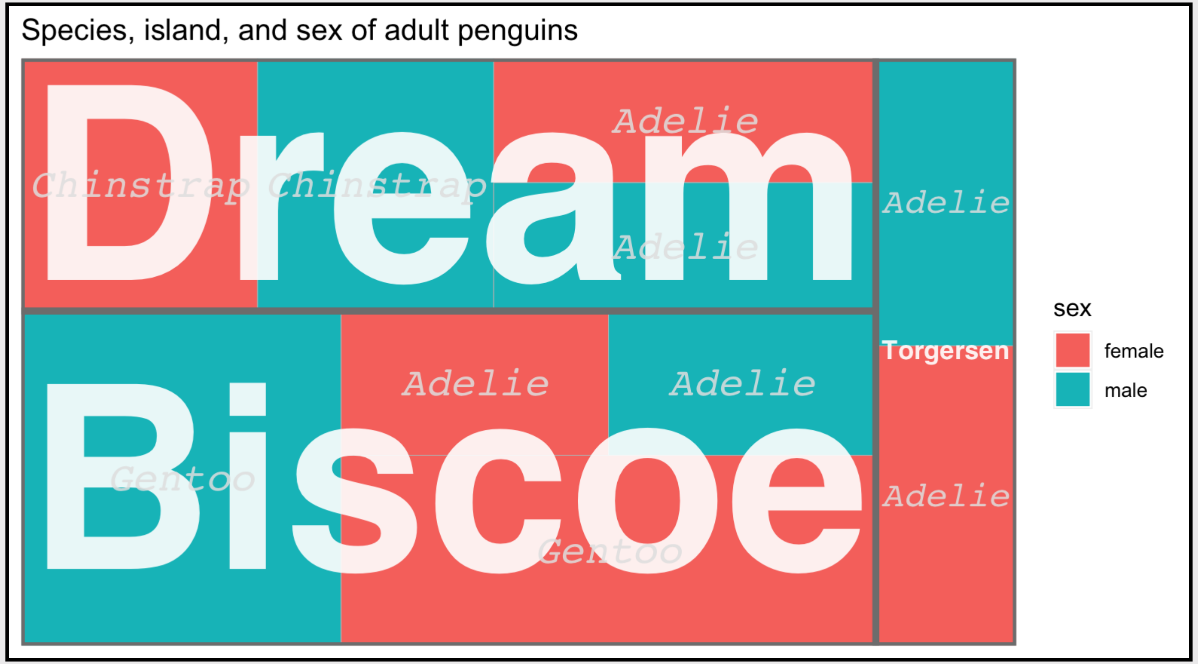
```
ggplot(treemap_penguins_counts,
       aes(area = n,
         fill = sex,
         label = species,
         subgroup = island)) +
       geom_treemap() +
       geom_treemap_subgroup_border() +
       geom_treemap_subgroup_text(
         place = "center",
         grow = TRUE,
         alpha = 0.9,
         color = "white",
         fontface = "bold",
         family = "sans",
         min.size = 0) +
       labs_treemap
```



Part-to-whole relationships: *Treemaps*

Include labels for additional subgroup with `geom_treemap_text()` (see full list of arguments here)

```
ggplot(treemap_penguins_counts,
       aes(area = n,
         fill = sex,
         label = species,
         subgroup = island)) +
       geom_treemap() +
       geom_treemap_subgroup_border() +
       geom_treemap_subgroup_text(
         place = "center",
         grow = TRUE,
         alpha = 0.9,
         color = "white",
         fontface = "bold",
         family = "sans",
         min.size = 0) +
       geom_treemap_text(
         colour = "gray90",
         place = "center",
         alpha = 0.85,
         family = "mono",
         fontface = "italic",
         reflow = TRUE) +
       labs_treemap
```


Species, island, and sex of adult penguins
Dream
Chinstrap
Chinstrap
Adelie
Adelie
Adelie
Torgersen
Adelie
Biscoe
Adelie
Adelie
Gentoo
Gentoo
sex
female
male