STAT 118: Notes C

Aggregating data with `summarize`, `group_by()`

Emily Malcolm-White

```
#LOAD PACKAGES
library(tidyverse)
```

Today's Dataset: `palmerpenguins` Size measurements, clutch observations, and blood isotope ratios for adult foraging Adélie, Chinstrap, and Gentoo penguins observed on islands in the Palmer Archipelago near Palmer Station, Antarctica. Data were collected and made available by Dr. Kristen Gorman and the Palmer Station Long Term Ecological Research (LTER) Program.

```
#LOAD DATA
library(palmerpenguins)
data(penguins)
```

Remove rows with missing data with `drop_na()`

```
penguins <- penguins %>%
  drop_na()
```

> ⚠ Warning
>
> Is it appropriate to remove rows with missing data? How many rows have missing data? Do the missing rows have something in common?
> Removing rows can affect the validity and generalizability of your analysis!

`summarize` Function or `summarise` Function (either works)

Suppose we are interested in the average bill length of all Adelie penguins:

```
penguins %>%
  filter(species == "Adelie") %>%
  summarize(average_bill_lenth = mean(bill_length_mm))
```

```
# A tibble: 1 x 1
  average_bill_lenth
               <dbl>
1               38.8
```

Suppose we are interested in the average bill length AND average bill depth of all Adelie penguins:

```
penguins %>%
  filter(species == "Adelie") %>%
  summarize(average_bill_lenth = mean(bill_length_mm),
            average_bill_depth = mean(bill_depth_mm))
```

```
# A tibble: 1 x 2
  average_bill_lenth average_bill_depth
               <dbl>              <dbl>
1               38.8               18.3
```

There are lots of other functions available:

- `min`: minimum value
- `max`: maximum value
- `mean`: average or mean value
- `median`: median value
- `var`: variance
- `sd`: standard deviation
- `n`: count or number of values
- `n_distinct`: counts number of distinct values

Suppose we are interested in the average bill length AND the median bill length of all Adelie penguins:

```r
penguins %>%
  filter(species == "Adelie") %>%
  summarise(average_bill_lenth = mean(bill_length_mm),
            median_bill_length = median(bill_length_mm))
```

```
# A tibble: 1 x 2
  average_bill_lenth median_bill_length
               <dbl>              <dbl>
1               38.8               38.8
```

group_by

Let's say we were interested in the average bill length and bill depth of all penguin species in this dataset. We could repeat this for the other species (Gentoo and Chinstrap). This would be a fair amount of work AND the results would not end up in the same table.

OR we could use the `group_by` command!

```r
penguins %>%
  group_by(species) %>%
  summarise(average_bill_lenth = mean(bill_length_mm),
            average_bill_depth = mean(bill_depth_mm))
```

```
# A tibble: 3 x 3
  species   average_bill_lenth average_bill_depth
  <fct>                  <dbl>              <dbl>
1 Adelie                  38.8               18.3
2 Chinstrap               48.8               18.4
3 Gentoo                  47.6               15.0
```

Multiple Groups

Suppose we wish to have the average bill length and average bill depth broken down by sex AND species:

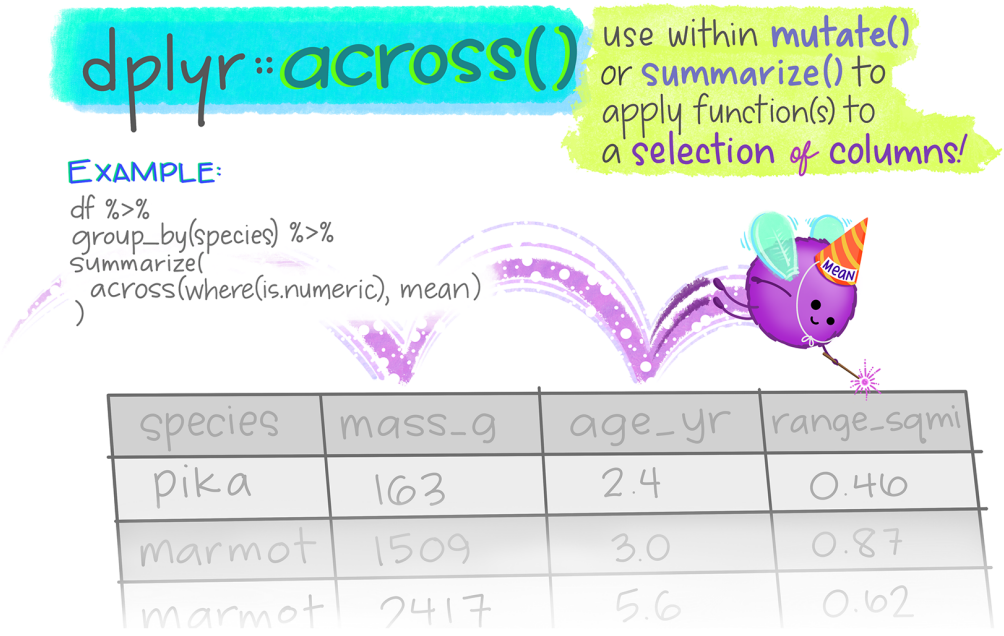
```r
penguins %>%
  group_by(species, sex) %>%
  summarise(average_bill_length = mean(bill_length_mm),
            average_bill_depth = mean(bill_depth_mm))
```

```
`summarise()` has grouped output by 'species'. You can override using the
`.groups` argument.
```

```
# A tibble: 6 x 4
# Groups:   species [3]
  species   sex    average_bill_length average_bill_depth
  <fct>     <fct>                <dbl>              <dbl>
1 Adelie    female                37.3               17.6
2 Adelie    male                  40.4               19.1
3 Chinstrap female                46.6               17.6
4 Chinstrap male                  51.1               19.3
5 Gentoo    female                45.6               14.2
6 Gentoo    male                  49.5               15.7
```

(Optional)`across`

If you wish to apply the same calculation to many columns, you may wish to check out the



`across` function.

More Examples

Suppose we want to calculate the number of distinct islands each species is found on:

```
penguins %>%
  group_by(species) %>%
  summarise(number_islands = n_distinct(island))
```

```
# A tibble: 3 x 2
  species   number_islands
  <fct>              <int>
1 Adelie                 3
2 Chinstrap              1
3 Gentoo                 1
```

Suppose we are interested in how many penguins of each species are on each island in the year 2007:

```
penguins %>%
  filter(year == "2007") %>%
  group_by(species, island) %>%
  summarise(number_penguins = n())
```

```
`summarise()` has grouped output by 'species'. You can override using the
`.groups` argument.
```

```
# A tibble: 5 x 3
# Groups:   species [3]
  species   island    number_penguins
  <fct>     <fct>               <int>
1 Adelie    Biscoe                 10
2 Adelie    Dream                  19
3 Adelie    Torgersen              15
4 Chinstrap Dream                  26
5 Gentoo    Biscoe                 33
```

> **Warning**
>
> Remember when we deleted rows with missing data earlier? The above is only the number of penguins which we have full data for! There could be more penguins on those islands who didn't have a complete data available for them!

Brain Break

This is a story about Jinjing the South American Magellanic Penguin, that swims 5,000 miles each year to be reunited with the man who saved his life. The rescued Penguin was saved by João Pereira de Souza, a 73 year old part-time fisherman, who lives in an island village just outside Rio de Janeiro, Brazil. https://youtu.be/oks2R4LqWtE