

## Exploring data 2

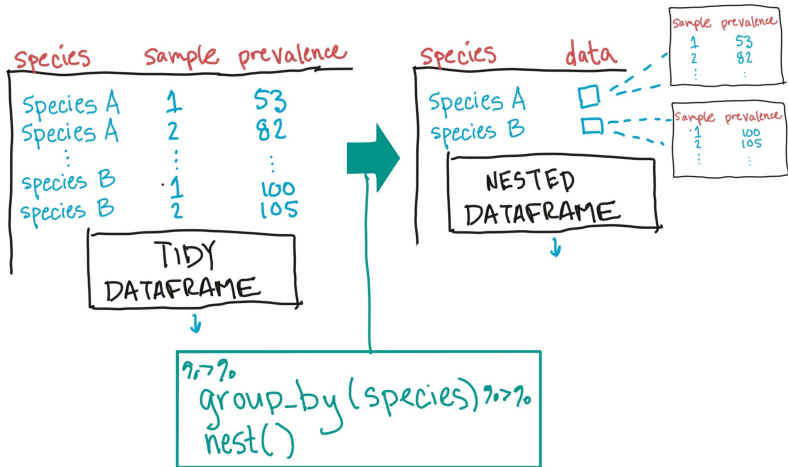
---

## More on nesting and mapping

---

# Nested dataframe

To create a nested dataframe, you can `group_by` a grouping value (e.g., bacteria species) and then `nest`:



## Nested dataframe

Here's the code to do that:

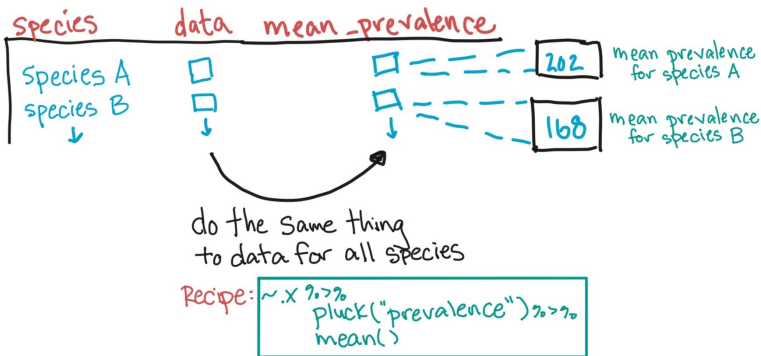
```
nested_samples <- tidy_samples %>%  
  group_by(species) %>%  
  nest()
```

```
nested_samples %>% head(3)
```

```
## # A tibble: 3 x 2  
## # Groups:   species [3]  
##   species      data  
##   <chr>      <list>  
## 1 Actinomycetaceae <tibble [1,151 x 2]>  
## 2 Aerococcus      <tibble [1,151 x 2]>  
## 3 Aeromonas       <tibble [1,151 x 2]>
```

# Mapping with a nested dataframe

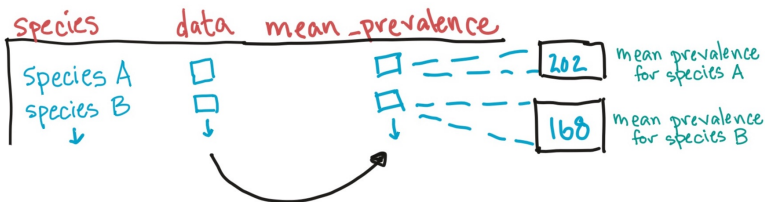
Now, you want to run the “recipe” you figured out on each bacteria species’ dataframe.



The goal is to create a new list-column with the results for each species.

# Mapping with a nested dataframe

You can use the `map` function from the `purrr` package within a `mutate` function to run the “recipe” you figured out on each bacteria species’ dataframe.



```
library(purrr)
mutate(mean_prevalence =
  map(data, ~.x %>%
    pluck("prevalence") %>%
    mean()))
```

# Mapping with a nested dataframe

Here is what this looks like in code:

```
library(purrr)
nested_samples2 <- nested_samples %>%
  mutate(mean_prevalence = map(data, ~ .x %>%
                                pull("prevalence") %>%
                                mean()))

nested_samples2 %>%
  head(3)

## # A tibble: 3 x 3
## # Groups:   species [3]
##   species          data          mean_prevalence
##   <chr>          <list>          <list>
## 1 Actinomycetaceae <tibble [1,151 x 2]> <dbl [1]>
## 2 Aerococcus      <tibble [1,151 x 2]> <dbl [1]>
## 3 Aeromonas       <tibble [1,151 x 2]> <dbl [1]>
```

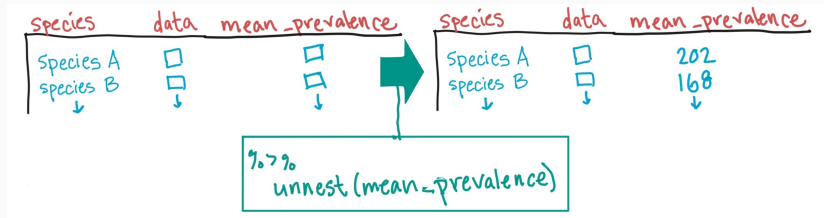
## Unnesting a nested dataframe

You'll usually want to unnest the new column so you can use it for creating plots, tables, and other output. You can use `unnest` to convert this column back to a regular column or columns (e.g., a vector or vectors) in a dataframe.



# Unnesting a nested dataframe

You'll need to specify which list-column to "unnest" when you use unnest:



# Unnesting a nested dataframe

Here's an example in code:

```
unnested_samples <- nested_samples2 %>%  
  unnest(mean_prevalence)
```

```
unnested_samples %>%  
  head(3)
```

```
## # A tibble: 3 x 3  
## # Groups:   species [3]  
##   species          data          mean_prevalence  
##   <chr>          <list>          <dbl>  
## 1 Actinomycetaceae <tibble [1,151 x 2]>      0.368  
## 2 Aerococcus       <tibble [1,151 x 2]>      0.00521  
## 3 Aeromonas        <tibble [1,151 x 2]>      0.00174
```

# Nesting and mapping

Here's what the whole process looks like, if you're doing it in a single piece of code:

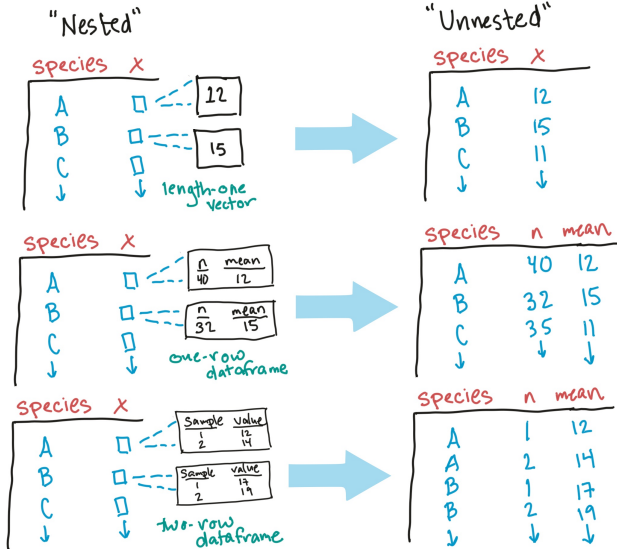
```
atlas1006 %>%  
  get_sample() %>%  
  as.data.frame() %>%  
  tibble::rownames_to_column(var = "species") %>%  
  tidyr::pivot_longer(- species,  
                      names_to = "sample",  
                      values_to = "prevalence") %>%  
  group_by(species) %>%  
  nest() %>%  
  mutate(mean_prevalence = map(data, ~ .x %>%  
                                pull("prevalence") %>%  
                                mean())) %>%  
  unnest(mean_prevalence)
```

## Nesting and mapping

So far, this might not seem too exciting, since everything we just did could have been done more easily with `group_by` and `summarize`.

However, this approach allows you to do more complex things. You can expand because you can unnest much more complicated list-columns than ones with length-one vectors for each grouping value.

# Nesting and mapping



## Nesting and mapping

For example, you can use this approach to run Shapiro-Wilk tests for all bacteria's prevalence samples:

```
sample_norm_test <- atlas1006 %>%  
  get_sample() %>%  
  as.data.frame() %>%  
  tibble::rownames_to_column(var = "species") %>%  
  tidyr::pivot_longer(- species,  
                      names_to = "sample",  
                      values_to = "prevalence") %>%  
  group_by(species) %>%  
  nest() %>%  
  mutate(norm_test = map(data, ~ .x %>%  
                        pull("prevalence") %>%  
                        shapiro.test() %>%  
                        tidy())) %>%  
  unnest(norm_test)
```

# Nesting and mapping

```
sample_norm_test %>%  
  head(4)
```

```
## # A tibble: 4 x 5
```

```
## # Groups:   species [4]
```

##	species	data	statistic	p.value	method
##	<chr>	<list>	<dbl>	<dbl>	<chr>
## 1	Actinomycetace~	<tibble [1,151 x ~	0.343	2.42e-53	Shapi
## 2	Aerococcus	<tibble [1,151 x ~	0.0362	6.49e-60	Shapi
## 3	Aeromonas	<tibble [1,151 x ~	0.0185	3.09e-60	Shapi
## 4	Akkermansia	<tibble [1,151 x ~	0.510	1.49e-48	Shapi

# Nesting and mapping

```
sample_norm_test %>%  
  ggplot(aes(x = log10(p.value))) +  
  geom_histogram()
```

