Introduction to R for Biologists

Day 3 – Data transformation with dplyr

Developed by Rachael Cox

*Several slides have been contributed by or modified from slides by Dr. Claus Wilke

Tidy data

Three rules:

1. Each variable forms a column
2. Each observation forms a row
3. Each type of observational unit forms a table

Class Outline

- Refresher from Day 2 (ggplot)
- Lecture on combining tables
  - Demonstration #1
- Lecture on filter() and select()
  - Demonstration #2
- Lecture on group_by and summarize()
  - Demonstration #3

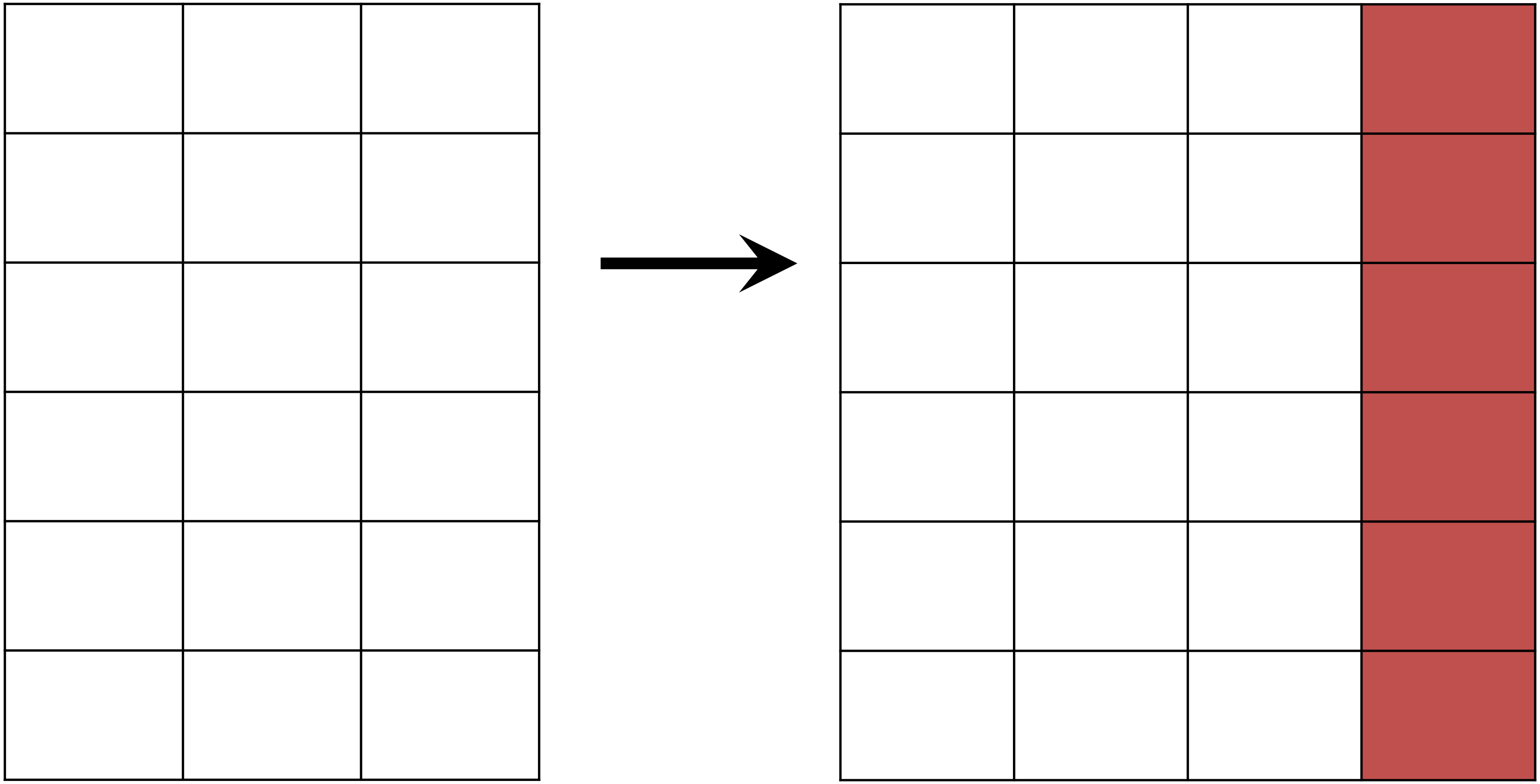
Working with tidy data in R: tidyverse

Fundamental actions on data tables:

- make new columns — `mutate()`
- combine tables, adding columns — `left_join()`
- combine tables, adding rows — `bind_rows()`
- choose rows — `filter()`
- choose columns — `select()`
- arrange rows — `arrange()`
- calculate summary statistics — `summarize()`
- work on groups of data — `group_by()`

`mutate()`: make new columns

`mutate()`: make new columns



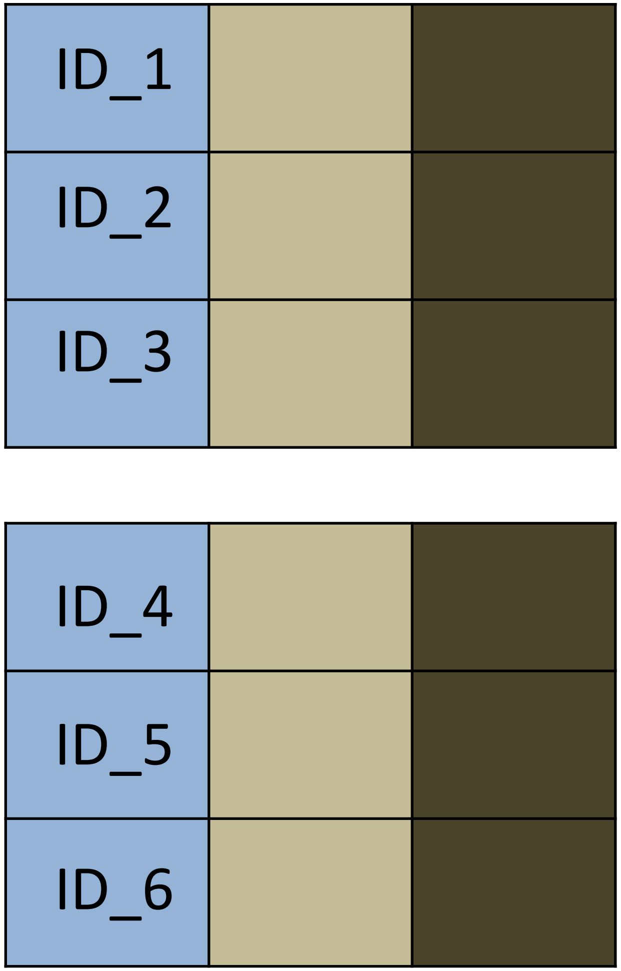
Make new column with ratio of Sepal.Length to Sepal.Width

```
> mutate(iris, sepal_length_to_width = Sepal.Length/Sepal.Width)
```

Make new column with ratio of Sepal.Length to Sepal.Width

```
> mutate(iris, sepal_length_to_width = Sepal.Length/Sepal.Width)
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species sepal_length_to_width
1            5.1         3.5          1.4         0.2     setosa              1.457143
2            4.9         3.0          1.4         0.2     setosa              1.633333
3            4.7         3.2          1.3         0.2     setosa              1.468750
4            4.6         3.1          1.5         0.2     setosa              1.483871
5            5.0         3.6          1.4         0.2     setosa              1.388889
6            5.4         3.9          1.7         0.4     setosa              1.384615
7            4.6         3.4          1.4         0.3     setosa              1.352941
8            5.0         3.4          1.5         0.2     setosa              1.470588
9            4.4         2.9          1.4         0.2     setosa              1.517241
10           4.9         3.1          1.5         0.1     setosa              1.580645
11           5.4         3.7          1.5         0.2     setosa              1.459459
12           4.8         3.4          1.6         0.2     setosa              1.411765
13           4.8         3.0          1.4         0.1     setosa              1.600000
14           4.3         3.0          1.1         0.1     setosa              1.433333
15           5.8         4.0          1.2         0.2     setosa              1.450000
16           5.7         4.4          1.5         0.4     setosa              1.295455
17           5.4         3.9          1.3         0.4     setosa              1.384615
18           5.1         3.5          1.4         0.3     setosa              1.457143
19           5.7         3.8          1.7         0.3     setosa              1.500000
20           5.1         3.8          1.5         0.3     setosa              1.342105
```

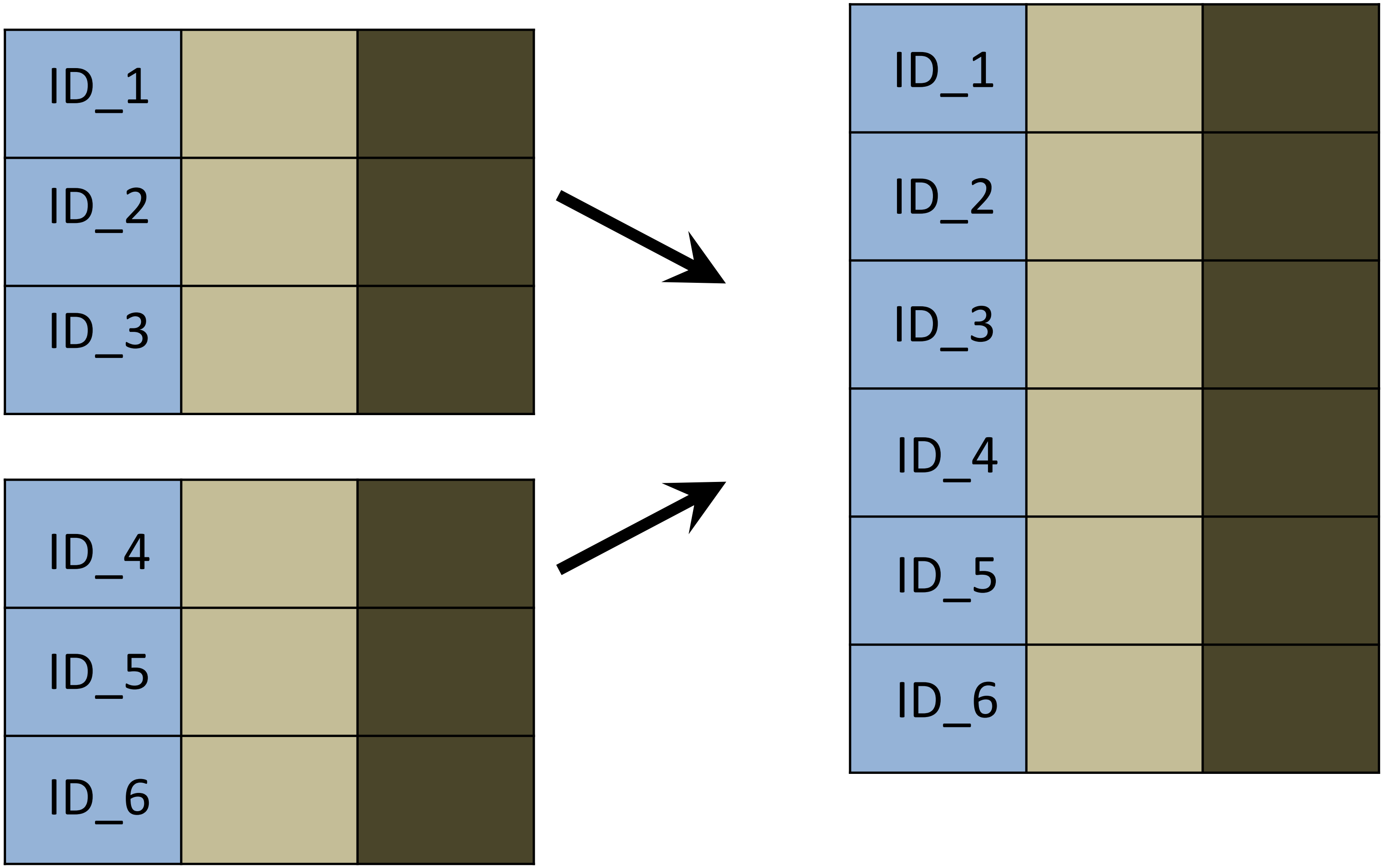
rbind() or bind_rows()



ID_1		
ID_2		
ID_3		

ID_4		
ID_5		
ID_6		

`rbind()` or `bind_rows()`: Stack tables

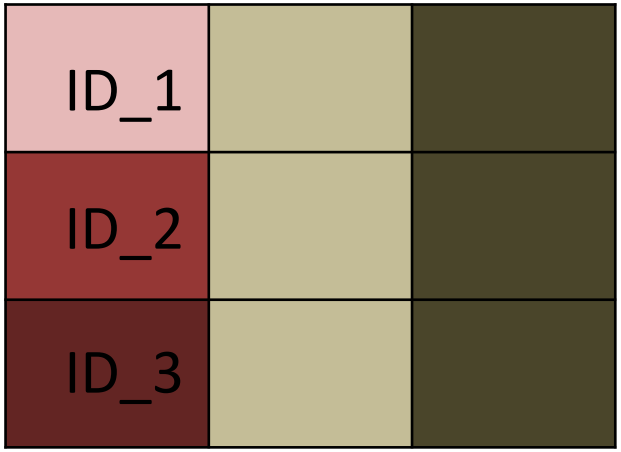


ID_1
ID_2
ID_3

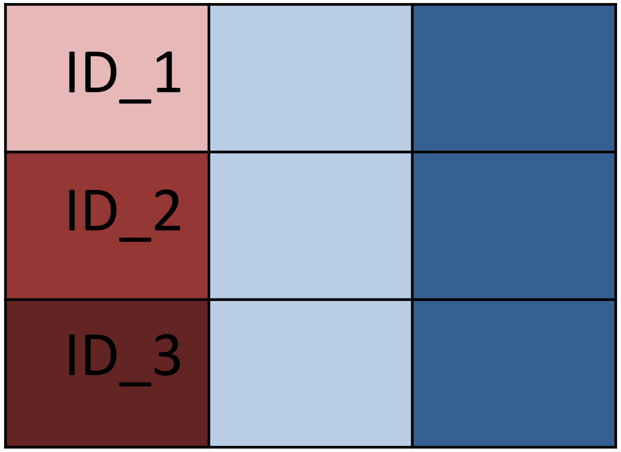
ID_4
ID_5
ID_6

ID_1
ID_2
ID_3
ID_4
ID_5
ID_6

`left_join()`: combine two tables

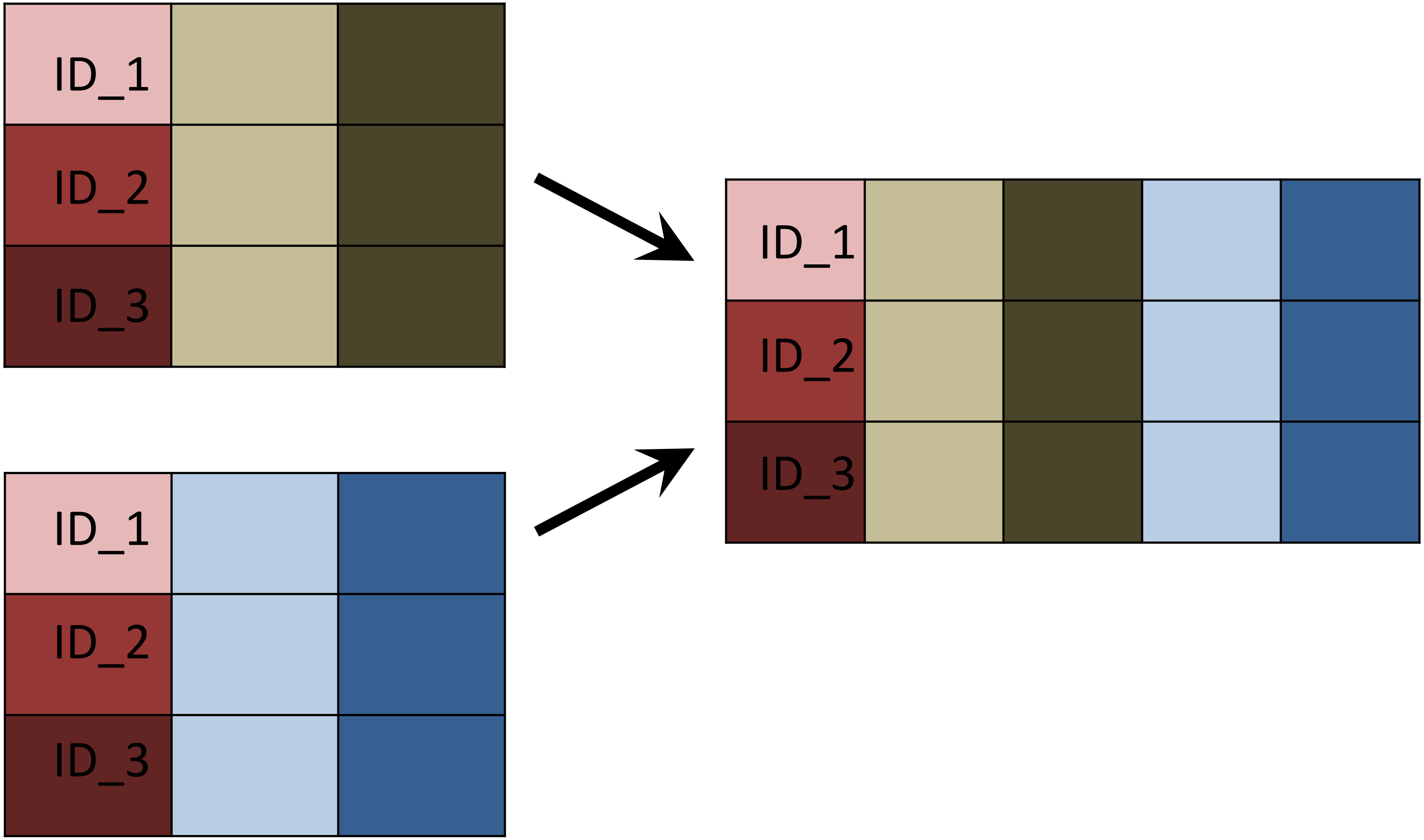


ID_1		
ID_2		
ID_3		



ID_1		
ID_2		
ID_3		

left_join(): combine two tables

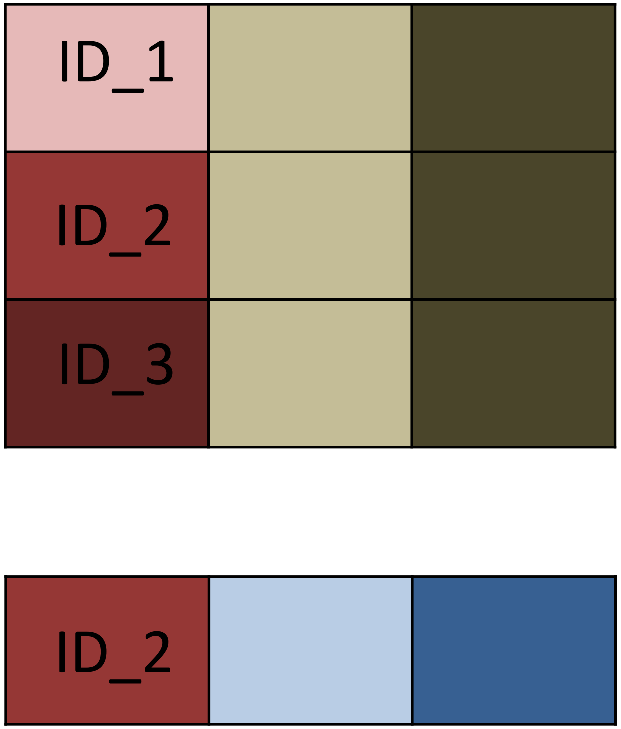


ID_1
ID_2
ID_3

ID_1
ID_2
ID_3

ID_1
ID_2
ID_3

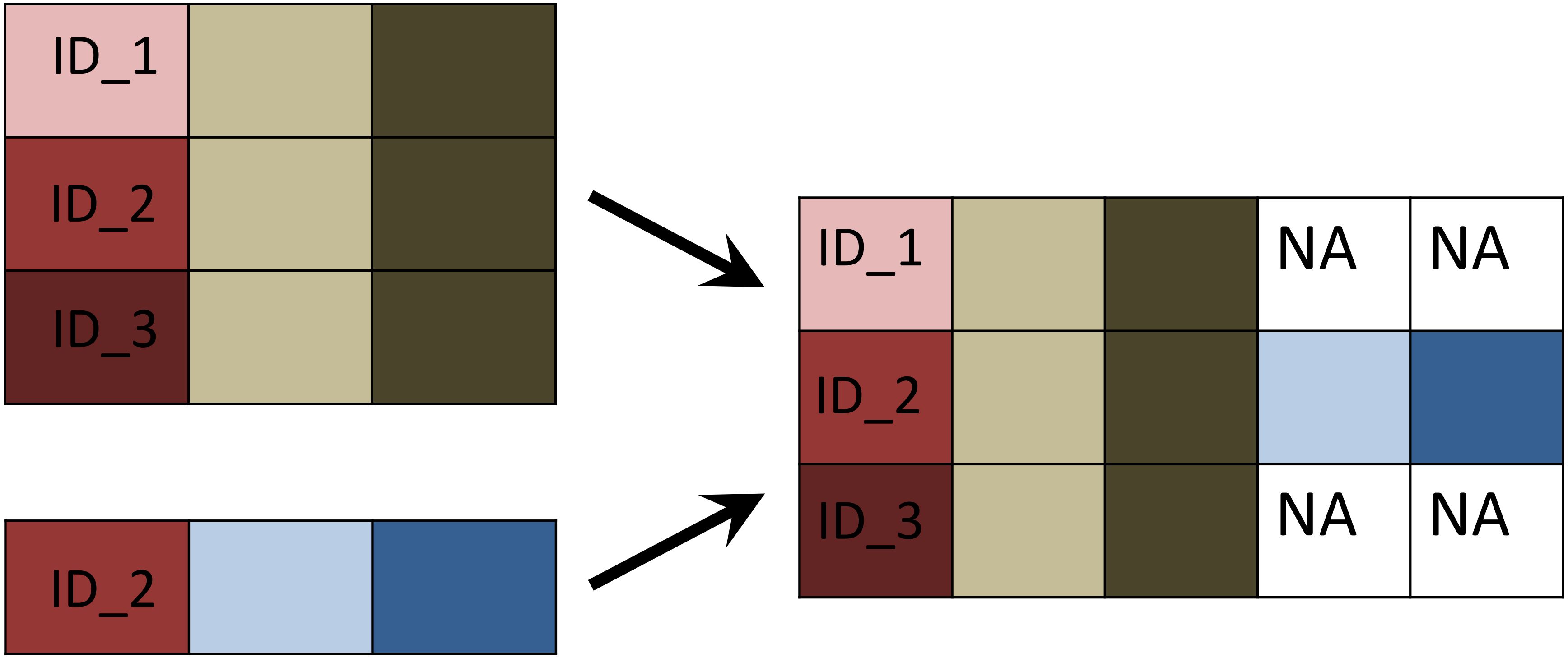
`left_join()`: missing values in 2nd table are set to NA



ID_1		
ID_2		
ID_3		

ID_2		

`left_join()`: missing values in 2nd table are set to NA



ID_1

ID_2

ID_3

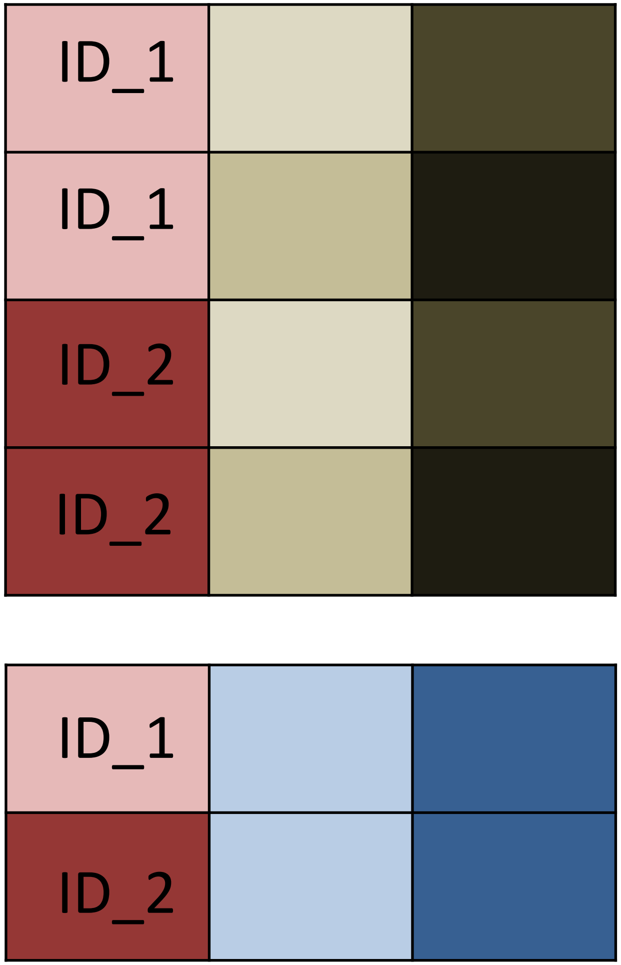
ID_2

ID_1 NA NA

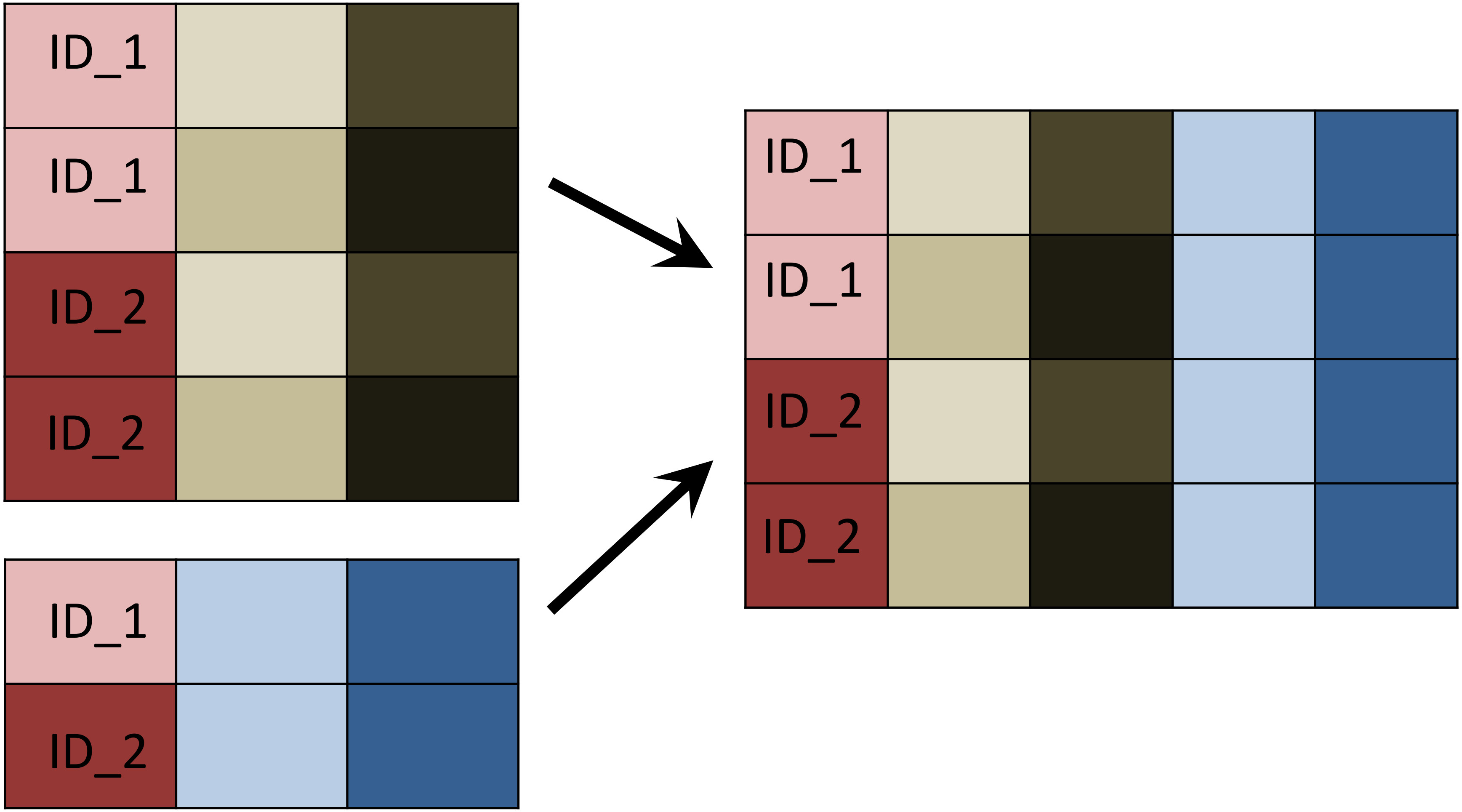
ID_2

ID_3 NA NA

`left_join()`: values from 2nd table are duplicated where necessary



`left_join()`: values from 2nd table are duplicated where necessary



Example: Joining tables

Let’s extract two tables from msleep:

Example: Joining tables

Let’s extract two tables from msleep:

```
> order_table <- select(msleep, name, order)
> order_table
                             name           order
1                         Cheetah       Carnivora
2                      Owl monkey        Primates
3                 Mountain beaver        Rodentia
4      Greater short-tailed shrew    Soricomorpha
5                             Cow    Artiodactyla
6                Three-toed sloth          Pilosa
7               Northern fur seal       Carnivora
8                    Vesper mouse        Rodentia
9                             Dog       Carnivora
10                       Roe deer    Artiodactyla
```

Example: Joining tables

Let’s extract two tables from msleep:

```
> awake_table <- select(msleep, name, awake)
> awake_table
                             name awake
1                         Cheetah 11.90
2                      Owl monkey  7.00
3                 Mountain beaver  9.60
4      Greater short-tailed shrew  9.10
5                             Cow 20.00
6                Three-toed sloth  9.60
7               Northern fur seal 15.30
8                    Vesper mouse 17.00
9                             Dog 13.90
10                       Roe deer 21.00
```

Example: Joining tables

And put them back together:

```
> left_join(order_table, awake_table)
```

Example: Joining tables

And put them back together:

```
> left_join(order_table, awake_table)
Joining by: "name"
                             name           order awake
1                         Cheetah       Carnivora 11.90
2                      Owl monkey        Primates  7.00
3                 Mountain beaver        Rodentia  9.60
4      Greater short-tailed shrew    Soricomorpha  9.10
5                             Cow    Artiodactyla 20.00
6                Three-toed sloth          Pilosa  9.60
7               Northern fur seal       Carnivora 15.30
8                    Vesper mouse        Rodentia 17.00
9                             Dog       Carnivora 13.90
10                       Roe deer    Artiodactyla 21.00
```

Several different join functions are available

- `left_join()`
- `right_join()`
- `inner_join()`
- `semi_join()`
- `full_join()`
- `anti_join()`

Demonstration Time!

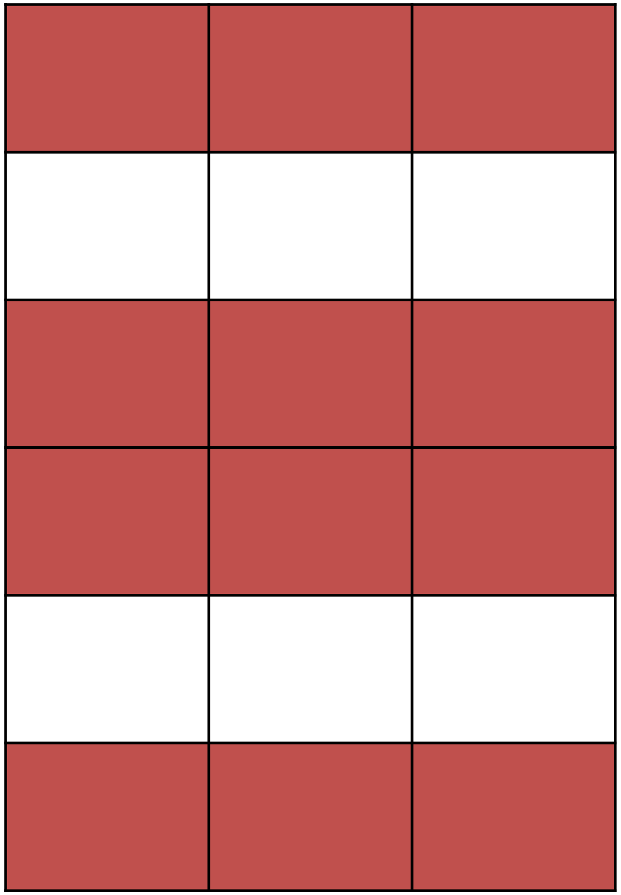
Work on Section 1.1 , 1.2 and 1.3

Working with tidy data in R: tidyverse

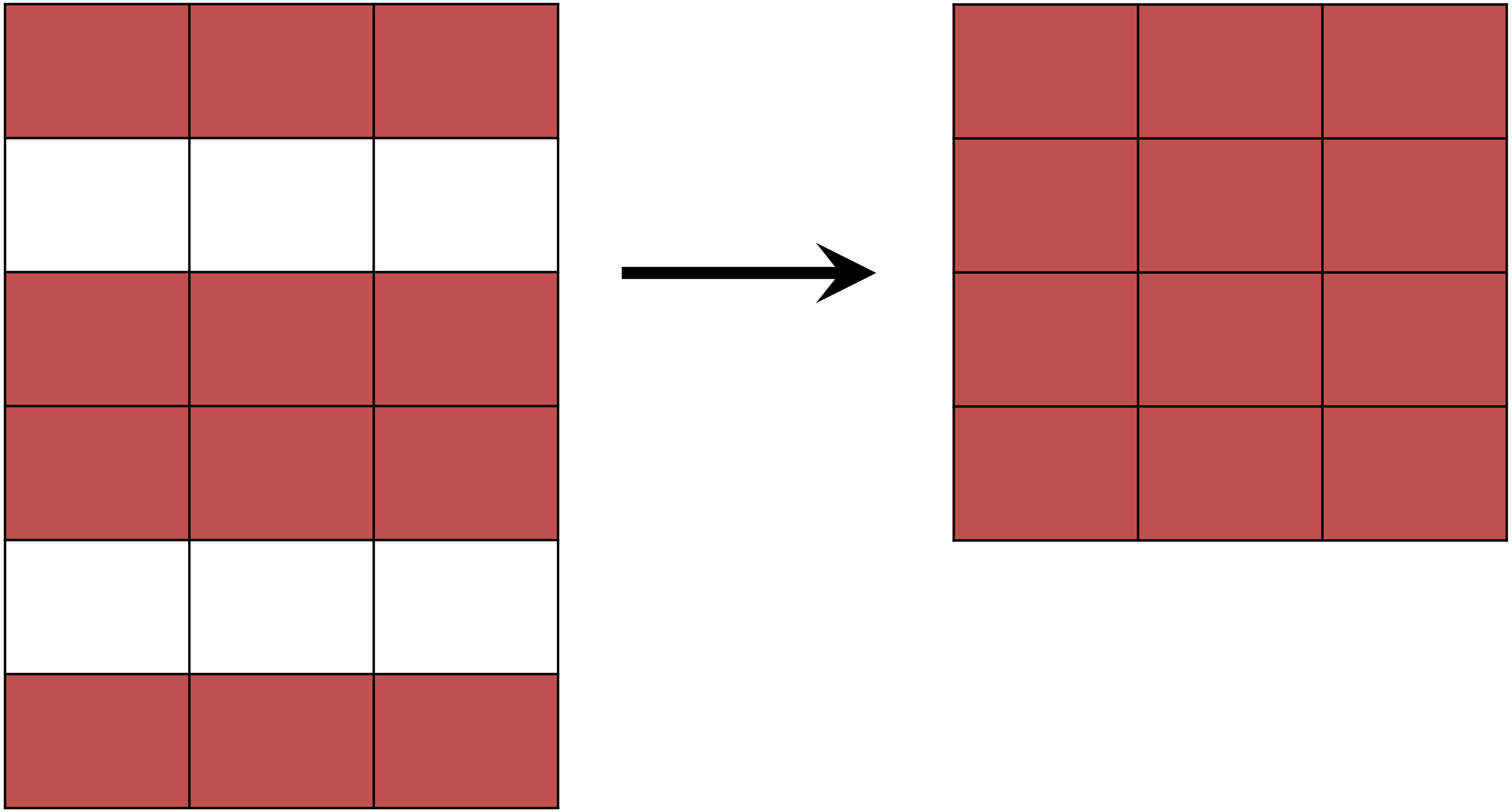
Fundamental actions on data tables:

- make new columns — `mutate()`
- combine tables, adding columns — `left_join()`
- combine tables, adding rows — `bind_rows()`
- choose rows — `filter()`
- choose columns — `select()`
- arrange rows — `arrange()`
- calculate summary statistics — `summarize()`
- work on groups of data — `group_by()`

filter(): pick rows



`filter()`: pick rows



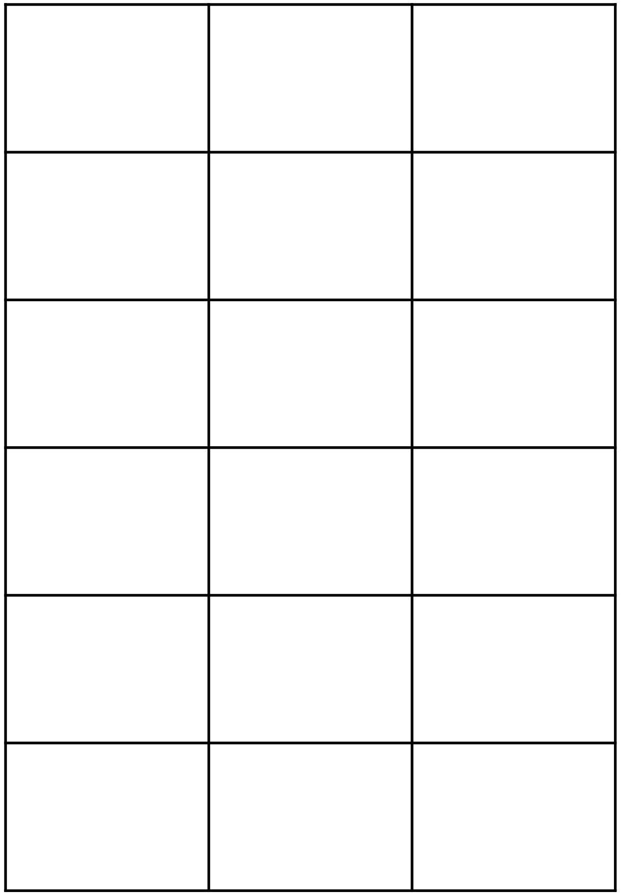
Choose rows with Sepal.Width > 4

```
> filter(iris, Sepal.Width > 4)
```

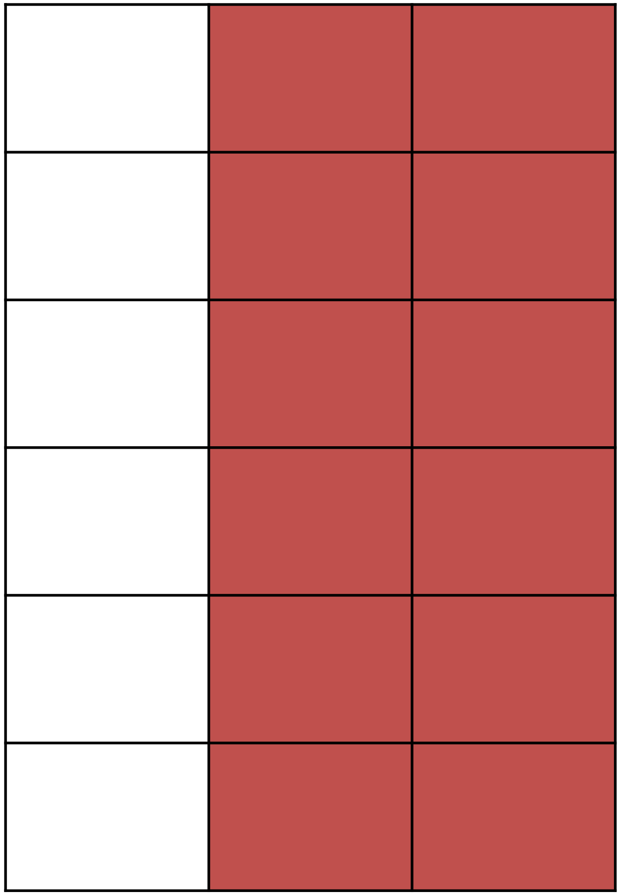
Choose rows with Sepal.Width > 4

```
> filter(iris, Sepal.Width > 4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.7         4.4          1.5         0.4  setosa
2          5.2         4.1          1.5         0.1  setosa
3          5.5         4.2          1.4         0.2  setosa
```

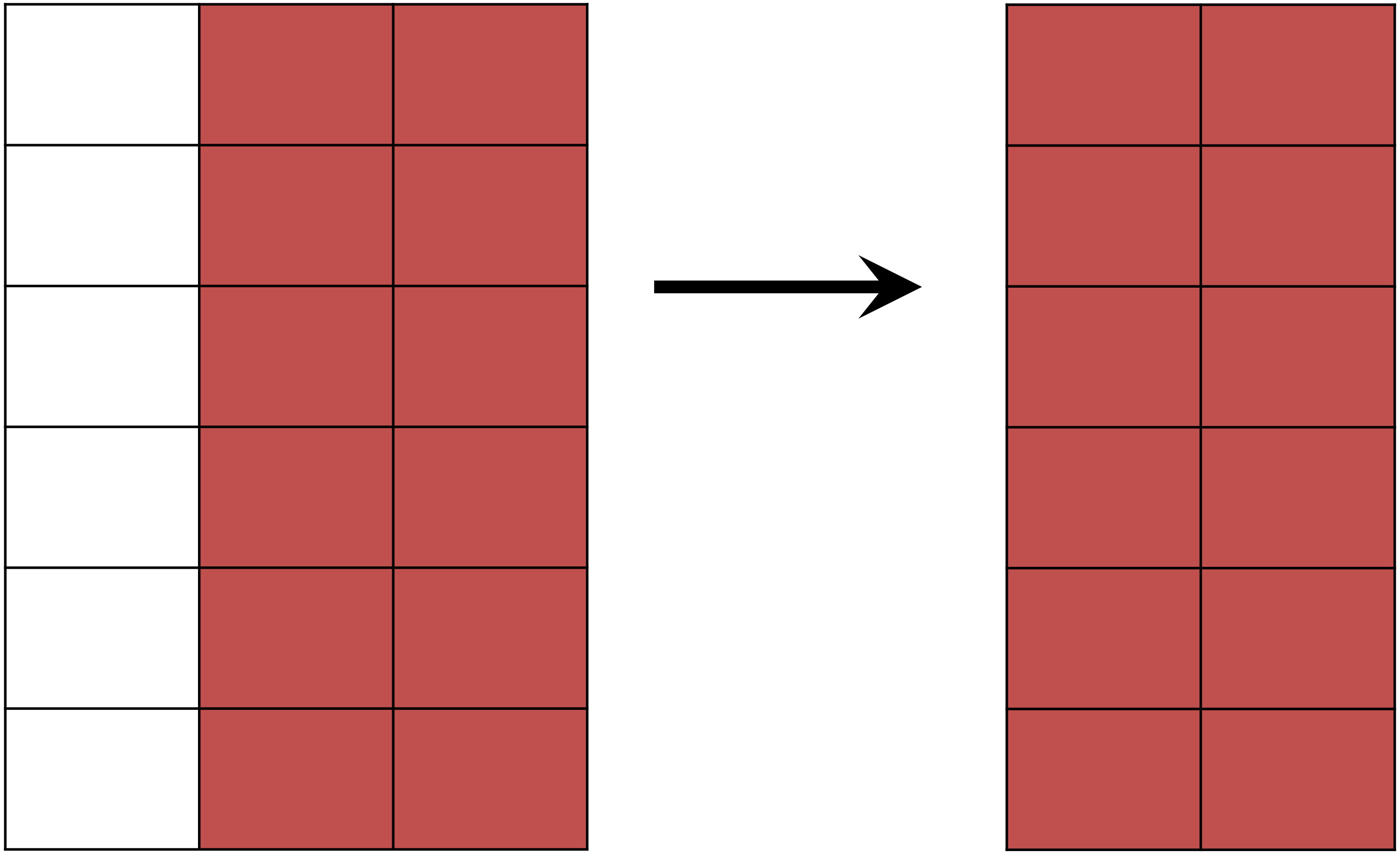
select(): pick columns



`select()`: pick columns



`select()`: pick columns



Choose the two columns Species and Sepal.Width

```
> select(iris, Species, Sepal.Width)
```

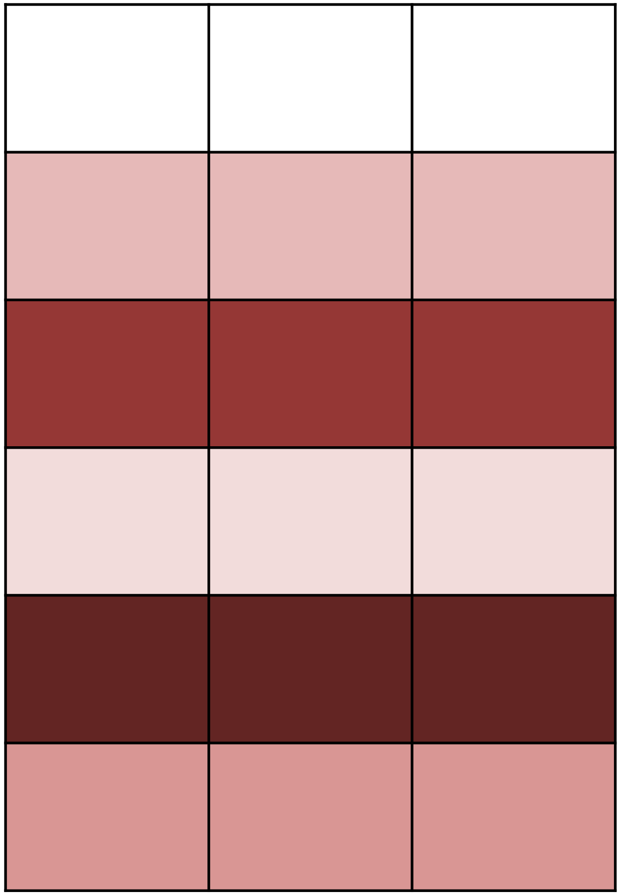
Choose the two columns Species and Sepal.Width

```
> select(iris, Species, Sepal.Width)
       Species Sepal.Width
1       setosa         3.5
2       setosa         3.0
3       setosa         3.2
4       setosa         3.1
5       setosa         3.6
6       setosa         3.9
7       setosa         3.4
8       setosa         3.4
9       setosa         2.9
10      setosa         3.1
11      setosa         3.7
12      setosa         3.4
13      setosa         3.0
14      setosa         3.0
```

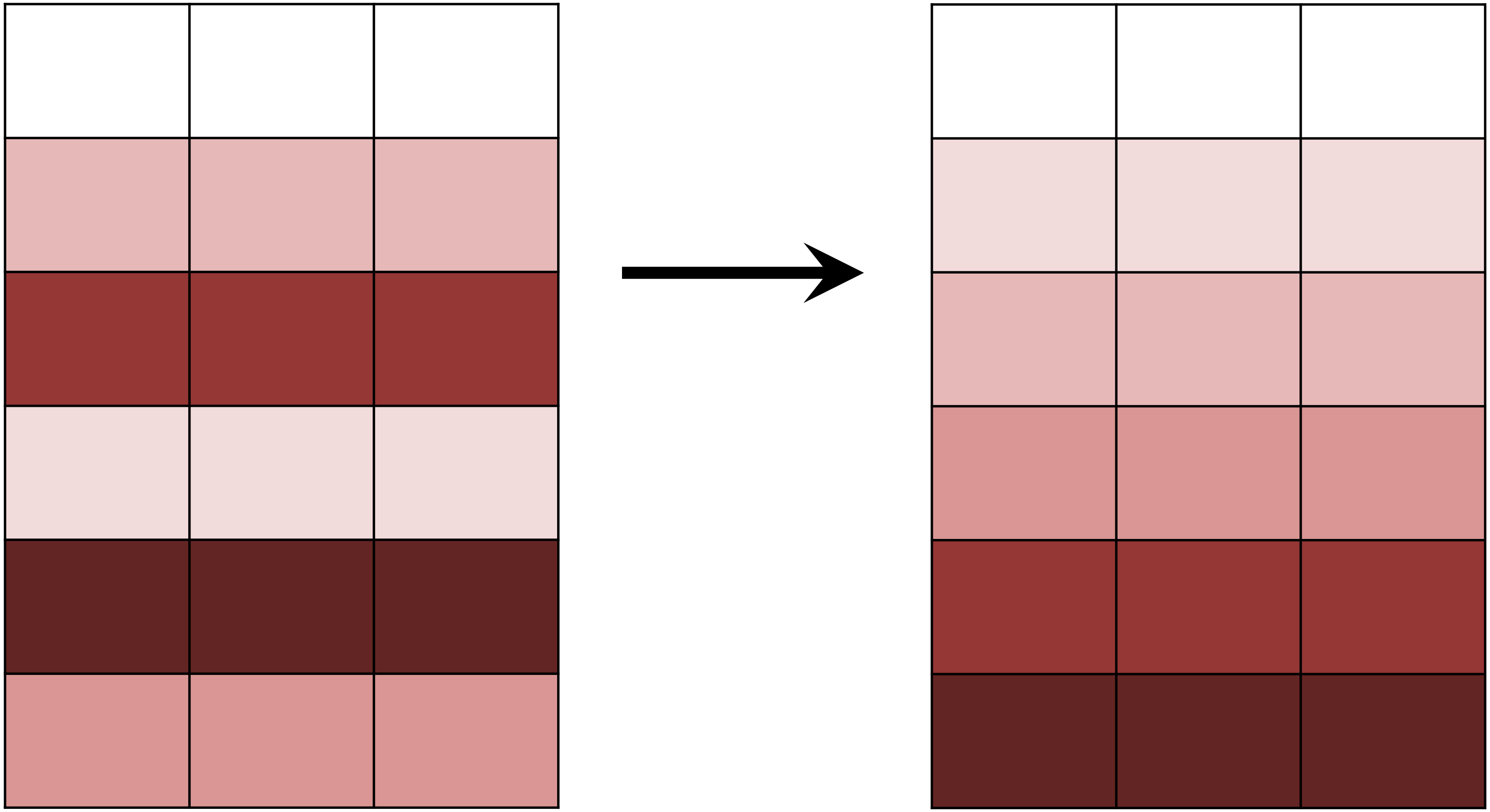
Demonstration Time!

Work on Section #2

`arrange()`: change row order



`arrange()`: change row order



Sort by increasing order of Sepal.Width

```
> arrange(iris, Sepal.Width)
```

Sort by increasing order of Sepal.Width

```
> arrange(iris, Sepal.Width)
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.0         2.0          3.5         1.0 versicolor
2            6.0         2.2          4.0         1.0 versicolor
3            6.2         2.2          4.5         1.5 versicolor
4            6.0         2.2          5.0         1.5  virginica
5            4.5         2.3          1.3         0.3     setosa
6            5.5         2.3          4.0         1.3 versicolor
7            6.3         2.3          4.4         1.3 versicolor
8            5.0         2.3          3.3         1.0 versicolor
9            4.9         2.4          3.3         1.0 versicolor
10           5.5         2.4          3.8         1.1 versicolor
11           5.5         2.4          3.7         1.0 versicolor
12           5.6         2.5          3.9         1.1 versicolor
```

Sort by decreasing order of Sepal.Length

```
> arrange(iris, desc(Sepal.Length))
```

Sort by decreasing order of Sepal.Length

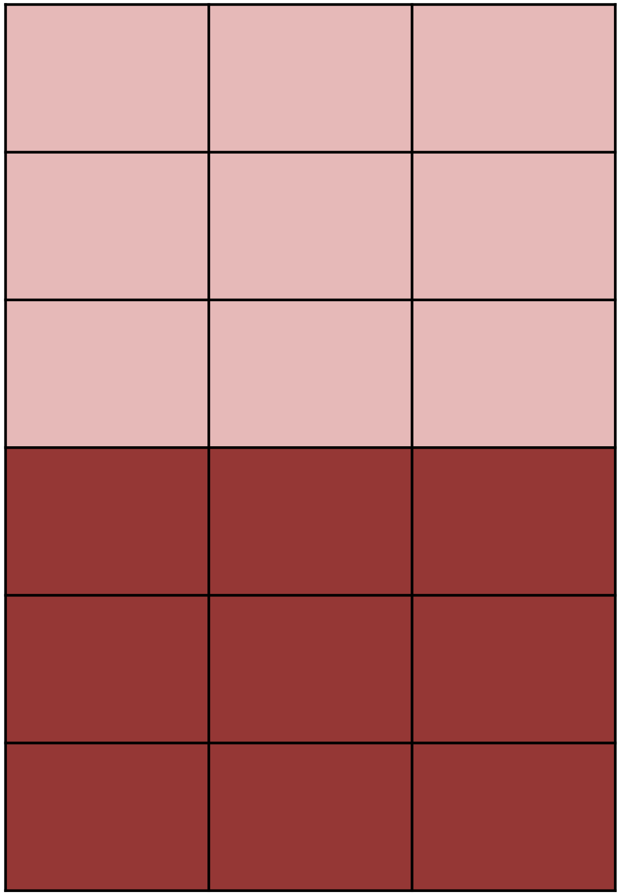
```
> arrange(iris, desc(Sepal.Length))
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1            7.9         3.8          6.4         2.0 virginica
2            7.7         3.8          6.7         2.2 virginica
3            7.7         2.6          6.9         2.3 virginica
4            7.7         2.8          6.7         2.0 virginica
5            7.7         3.0          6.1         2.3 virginica
6            7.6         3.0          6.6         2.1 virginica
7            7.4         2.8          6.1         1.9 virginica
8            7.3         2.9          6.3         1.8 virginica
9            7.2         3.6          6.1         2.5 virginica
10           7.2         3.2          6.0         1.8 virginica
11           7.2         3.0          5.8         1.6 virginica
12           7.1         3.0          5.9         2.1 virginica
```

Working with tidy data in R: tidyverse

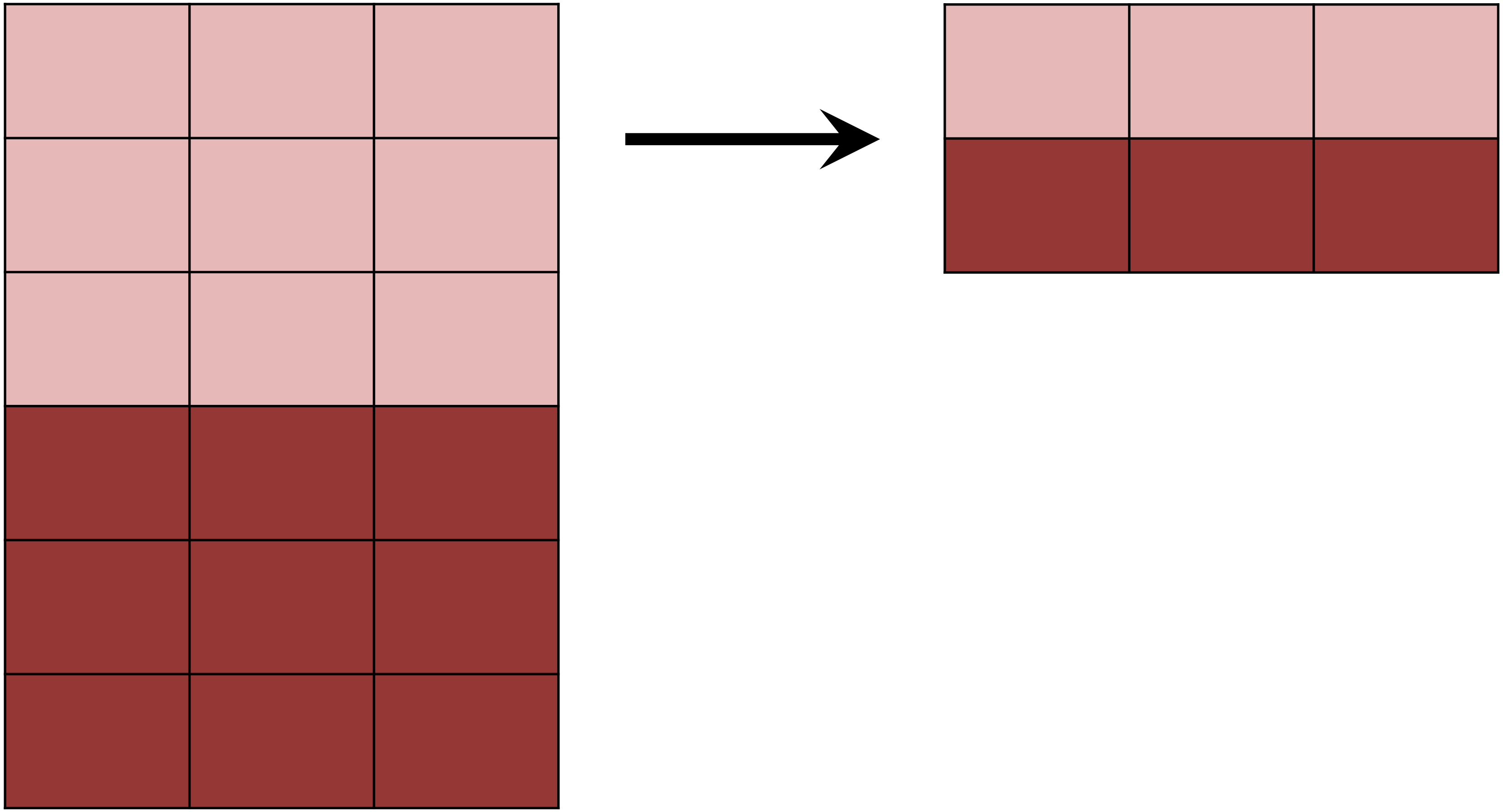
Fundamental actions on data tables:

- make new columns — `mutate()`
- combine tables, adding columns — `left_join()`
- combine tables, adding rows — `bind_rows()`
- choose rows — `filter()`
- choose columns — `select()`
- arrange rows — `arrange()`
- calculate summary statistics — `summarize()`
- work on groups of data — `group_by()`

`summarize()`: collapse multiple rows



`summarize()`: collapse multiple rows



Calculate mean and standard deviation of Sepal.Length

```
> summarize(iris, mean_sepal_length = mean(Sepal.Length),
                  sd_sepal_length   = sd(Sepal.Length))
```

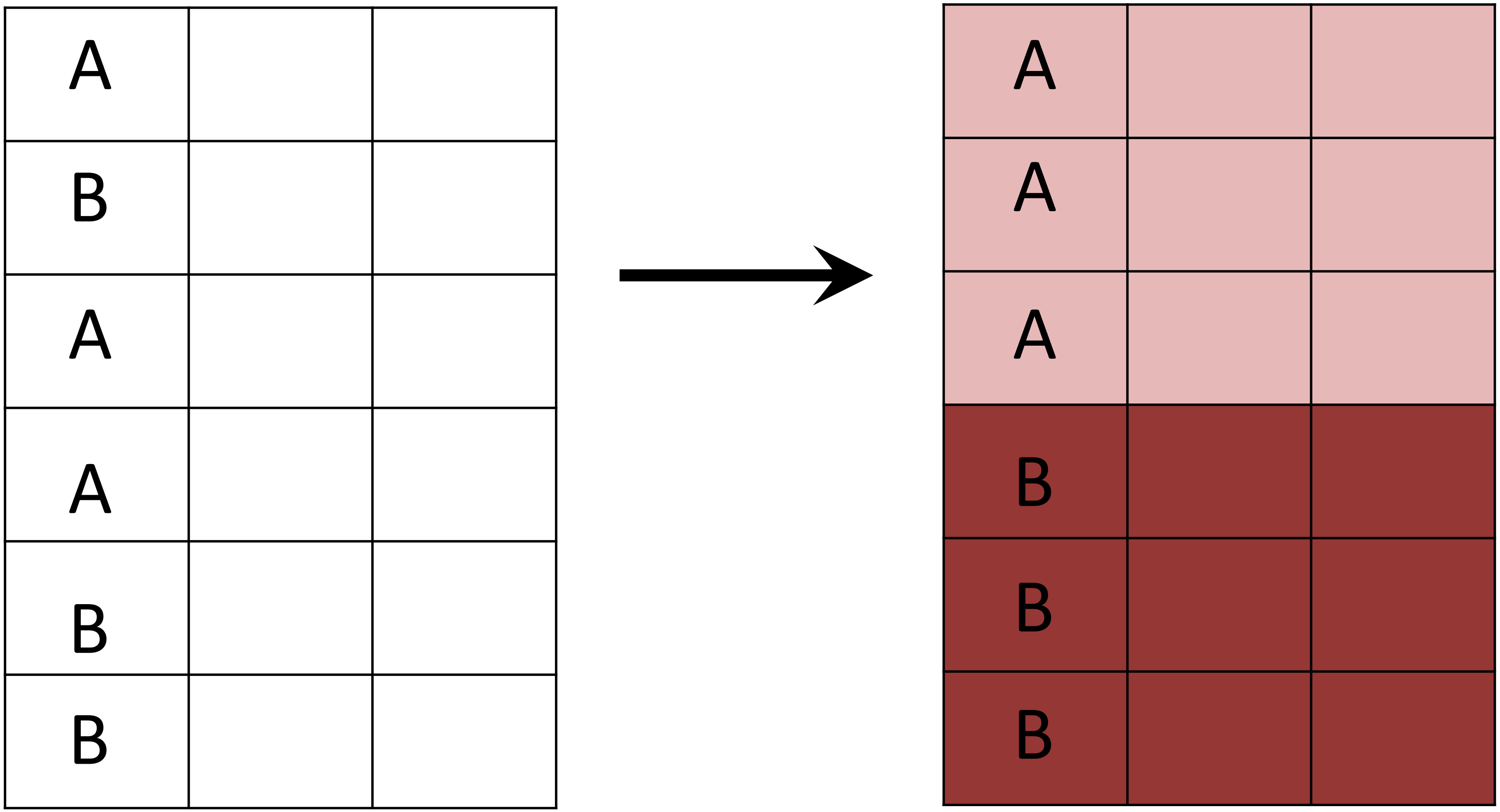
Calculate mean and standard deviation of Sepal.Length

```
> summarize(iris, mean_sepal_length = mean(Sepal.Length),
                  sd_sepal_length   = sd(Sepal.Length))
  mean_sepal_length sd_sepal_length
1          5.843333       0.8280661
```

`group_by()`: set up groupings

A		
B		
A		
A		
B		
B		

`group_by()`: set up groupings



A		
B		
A		
A		
B		
B		

A		
A		
A		
B		
B		
B		

Calculate mean and standard deviation of Sepal.Length, grouped by Species

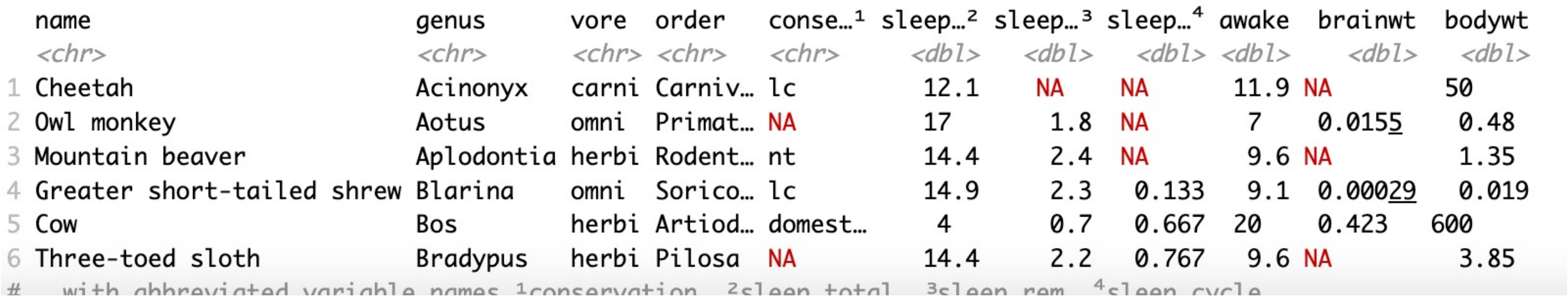
```
> summarize(group_by(iris, Species),
                  mean_sepal_length = mean(Sepal.Length),
                  sd_sepal_length   = sd(Sepal.Length))
```

Calculate mean and standard deviation of Sepal.Length, grouped by Species

```
> summarize(group_by(iris, Species),
                  mean_sepal_length = mean(Sepal.Length),
                  sd_sepal_length   = sd(Sepal.Length))
Source: local data frame [3 x 3]

     Species mean_sepal_length sd_sepal_length
1     setosa             5.006       0.3524897
2 versicolor             5.936       0.5161711
3  virginica             6.588       0.6358796
```

Pipe example 1: count how many herbivores of different orders there are in `msleep`



```
  name                       genus      vore  order    conse…¹ sleep…² sleep…³ sleep…⁴ awake  brainwt  bodywt
  <chr>                      <chr>      <chr> <chr>    <chr>     <dbl>   <dbl>   <dbl> <dbl>    <dbl>   <dbl>
1 Cheetah                    Acinonyx   carni Carniv… lc         12.1    NA     NA      11.9 NA        50
2 Owl monkey                 Aotus      omni  Primat… NA         17       1.8   NA       7    0.0155    0.48
3 Mountain beaver            Aplodontia herbi Rodent… nt         14.4     2.4   NA       9.6 NA         1.35
4 Greater short-tailed shrew Blarina    omni  Sorico… lc         14.9     2.3    0.133   9.1  0.00029   0.019
5 Cow                        Bos        herbi Artiod… domest…     4       0.7    0.667  20    0.423   600
6 Three-toed sloth           Bradypus   herbi Pilosa  NA         14.4     2.2    0.767   9.6 NA         3.85
# … with abbreviated variable names ¹conservation, ²sleep_total, ³sleep_rem, ⁴sleep_cycle
```

Pipe example 1: count how many herbivores of different orders there are in `msleep`

```
msleep %>%
  filter(vore == "herbi")
```

Pipe example 1: count how many herbivores of different orders there are in `msleep`

```
msleep %>%
  filter(vore == "herbi") %>%
  group_by(order)
```

Pipe example 1: count how many herbivores of different orders there are in `msleep`

```
msleep %>%
  filter(vore == "herbi") %>%
  group_by(order) %>%
  summarize(count = n())
```

Pipe example 1: count how many herbivores of different orders there are in `msleep`

```
msleep %>%
  filter(vore == "herbi") %>%
  group_by(order) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
```

Pipe example 1: count how many herbivores of different orders there are in msleep

```
msleep %>%
  filter(vore == "herbi") %>%
  group_by(order) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
```

```
           order count
1       Rodentia    16
2   Artiodactyla     5
3 Perissodactyla     3
4     Hyracoidea     2
5    Proboscidea     2
6  Diprotodontia     1
7     Lagomorpha     1
8         Pilosa     1
9       Primates     1
```

Pipe example 2: What is the median awake time of different orders in `msleep`?

Pipe example 2: What is the median awake time of different orders in msleep?

```
msleep %>%
  group_by(order)
```

Pipe example 2: What is the median awake time of different orders in `msleep`?

```
msleep %>%
  group_by(order) %>%
  summarize(med_awake = median(awake))
```

Pipe example 2: What is the median awake time of different orders in `msleep`?

```
msleep %>%
  group_by(order) %>%
  summarize(med_awake = median(awake)) %>%
  arrange(med_awake)
```

Pipe example 2: What is the median awake time of different orders in `msleep`?

```
msleep %>%
  group_by(order) %>%
  summarize(med_awake = median(awake)) %>%
  arrange(med_awake)
```

```
              order med_awake
1        Chiroptera      4.20
2   Didelphimorphia      5.30
3         Cingulata      6.25
4      Afrosoricida      8.40
5            Pilosa      9.60
6          Rodentia     11.10
7     Diprotodontia     11.60
8      Soricomorpha     13.70
9         Carnivora     13.75
10  Erinaceomorpha     13.80
```

Demonstration Time!

Work on Section #3