dplyr::mutate and dplyr::summarize

Running example

So that we have a running example to work with, here are the first 5 rows of the `iris` data set that comes with R:

```
iris_short
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
```

Logical Operations on Vectors

We can compare vectors to each other using:

- `<` less than
- `<=` less than or equal to
- `==` equal to
- `>=` greater than or equal to
- `>` greater than
- `!=` not equal to

```
iris_short$Sepal.Length > iris_short$Sepal.Width
```

```
## [1] TRUE TRUE TRUE TRUE TRUE
```

```
iris_short$Sepal.Length >= 5.0
```

```
## [1]  TRUE FALSE FALSE FALSE  TRUE
```

```
iris_short$Sepal.Length != 5.0
```

```
## [1]  TRUE  TRUE  TRUE  TRUE FALSE
```

We can combine multiple conditions using:

- `&` is TRUE if both conditions are TRUE
- `|` is TRUE if either (or both) condition is TRUE

```
iris_short$Sepal.Length > 4.6 & iris_short$Sepal.Length < 5.0
```

```
## [1] FALSE  TRUE  TRUE FALSE FALSE
```

```
iris_short$Sepal.Length <= 4.6 | iris_short$Sepal.Length >= 5.0
```

```
## [1]  TRUE FALSE FALSE  TRUE  TRUE
```

When you perform arithmetic operations on a logical vector, `TRUE` is converted to 1 and `FALSE` is converted to 0:

```
sum(iris_short$Sepal.Length <= 4.6 | iris_short$Sepal.Length >= 5.0)
```

```
## [1] 3
```

```
mean(iris_short$Sepal.Length <= 4.6 | iris_short$Sepal.Length >= 5.0)
```

```
## [1] 0.6
```

Defining a data frame

The format of the `mutate` and `summarize` commands is less confusing if we have the way to create a data frame firmly in mind:

```
example_df <- data.frame(
  a = c("v", "w", "x", "y", "z"),
  b = 10 * sqrt(1:5)
)

example_df
```

```
##   a        b
## 1 v 10.00000
## 2 w 14.14214
## 3 x 17.32051
## 4 y 20.00000
## 5 z 22.36068
```

dplyr package

The `mutate` and `summarize` functions are provided by the `dplyr` package.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

`mutate`: add a new variable or modify an existing variable in a data frame

- First argument: name of data frame to modify
- Remaining arguments: variables and how to compute them. Refer to variables in the data frame directly.

```
mutate(iris_short,
  a = c("v", "w", "x", "y", "z"),
  b = 10 * sqrt(1:5),
  sepal_sum = Sepal.Length + Sepal.Width
)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species a        b
## 1          5.1         3.5          1.4         0.2  setosa v 10.00000
## 2          4.9         3.0          1.4         0.2  setosa w 14.14214
## 3          4.7         3.2          1.3         0.2  setosa x 17.32051
## 4          4.6         3.1          1.5         0.2  setosa y 20.00000
## 5          5.0         3.6          1.4         0.2  setosa z 22.36068
##   sepal_sum
## 1       8.6
## 2       7.9
## 3       7.9
## 4       7.7
## 5       8.6
```

- The original data frame is not changed, a copy is made
- The pipe `%>%` calls the function on the right using the expresson on the left as the first argument

```
iris_updated <- iris_short %>% mutate(
  a = c("v", "w", "x", "y", "z"),
  b = 10 * sqrt(1:5),
  sepal_sum = Sepal.Length + Sepal.Width
)

iris_updated
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species a        b
## 1          5.1         3.5          1.4         0.2  setosa v 10.00000
## 2          4.9         3.0          1.4         0.2  setosa w 14.14214
## 3          4.7         3.2          1.3         0.2  setosa x 17.32051
## 4          4.6         3.1          1.5         0.2  setosa y 20.00000
## 5          5.0         3.6          1.4         0.2  setosa z 22.36068
##   sepal_sum
## 1       8.6
## 2       7.9
## 3       7.9
## 4       7.7
## 5       8.6
```

```
iris_short
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
```

summarize: compute length-1 summaries of a data frame

```
mean(iris_short$Sepal.Length)
```

```
## [1] 4.86
```

```
sd(iris_short$Sepal.Length)
```

```
## [1] 0.2073644
```

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

```
##   mean_sepal_length sd_sepal_length
## 1              4.86       0.2073644
```

- Longer-length summaries result in errors:

```
quantile(iris_short$Sepal.Length)
```

```
##   0%  25%  50%  75% 100%
##  4.6  4.7  4.9  5.0  5.1
```

```
iris_short %>%
  summarize(
    quantiles_sepal_length = quantile(Sepal.Length)
  )
```

```
## Error in summarise_impl(.data, dots): Column `quantiles_sepal_length` must be length 1 (a summary value),
```