R internal data sets

- A list of built-in data sets in R can be obtained by entering libary(help = "datasets")
- There appear to be around 2 dozen
- Per sthda.com, the most commonly used are:
  - Mtcars
  - Iris
  - ToothGrowth
  - PlantGrowth
  - USArrests

Obtain Mean and SE with dplyr

Script

```
sepal_means <- Iris %>%
  group_by(Species) %>%
  summarise(nObs = n(),
            mn = mean(Sepal.Length),
            sem =
FSA::se(Sepal.Length))
```

Before (6 of 150 obs)

```
head(Iris)
  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
6          5.4         3.9          1.7         0.4  setosa
```

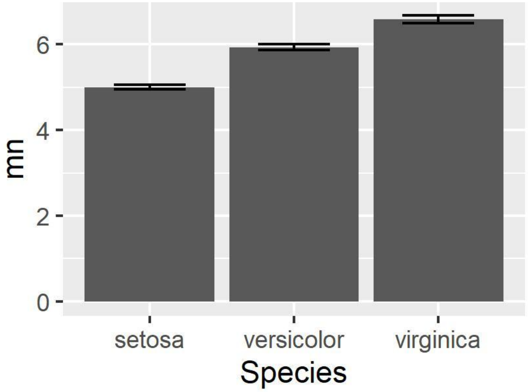
After (data frame sepal_means)

```
# A tibble: 3 × 4
  Species     nObs    mn    sem
  <fct>      <int> <dbl>  <dbl>
1 setosa        50  5.01 0.0498
2 versicolor    50  5.94 0.0730
3 virginica     50  6.59 0.0899
```

Basic ggplot–vertical bar chart

```
p1 <- ggplot(sepal_means, aes(x = Species, y = mn)) +
  geom_col() +
  geom_errorbar(aes(ymin = mn - sem,
                          ymax = mn + sem),
                          width = 0.5)
p1

ggsave(filename = "iris-sepal-means-default.jpg", plot = p1,
device = "jpg",
       dpi = 300, width = 2.83, height = 2.1225, units = "in")
```

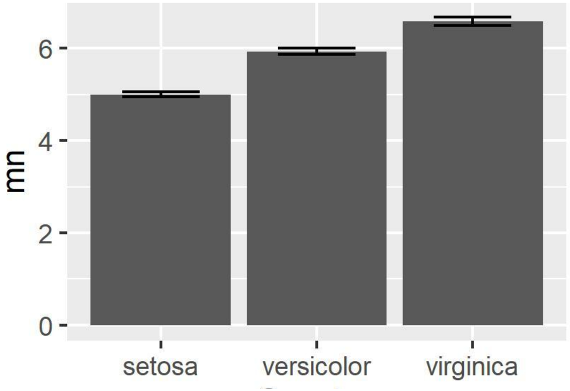


The script above provide a very basic ggplot using the data source ("sepal_means") the aesthetic (aes–what is x and what is y?), and a couple of geom_ statements. To save this plot for future use it is necessary to assign the output of the script to an object(p1) for use in a subsequent ggsave statement. This plot is almost entirely default. Notice that what it seen in a plot window in RStudio is not a reliable guide to the appearance when saved and retrieved. Note that the appearance is dependent on size information given in ggsave.

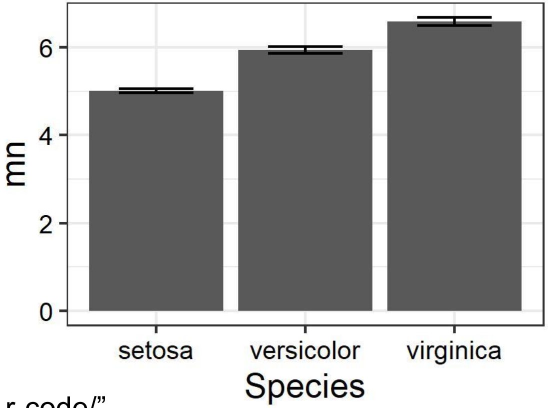
Using a function to create a theme

```
p1 <- ggplot(sepal_means, aes(x = Species, y = mn)) +
  geom_col() +
  geom_errorbar(aes(ymin = mn - sem,
                    ymax = mn + sem),
                width = 0.5)
  # + theme_burks_halfwidth()

  theme_burks_halfwidth <- function(){
    theme_bw() +
    theme(
      axis.text.x = element_text(color = "black", size = 9),
      axis.text.y = element_text(color = "black", size = 9),
      axis.title.x = element_text(color = "black", size = 12),
      axis.title.y = element_text(color = "black", size = 12),
      legend.title = element_text(color = "black", size = 9),
      legend.text = element_text(color = "black", size = 8))
  }
```



Before addition of custom theme



After addition of custom theme

From:
“https://www.r-bloggers.com/2023/01/tips-for-organising-your-r-code/”