Data Visualization - Exercise 4: Plotting Multiple Data Sets and Fitting Linear Models
Solution

Luna L Sanchez Reyes

2023-03-06

There are 3 mains steps to solve this exercise.

1) Read the data in with quality control

```
trees <- read_tsv("../data-raw/TREE_SURVEYS.txt",
              col_types = list(HEIGHT = col_double(),
                               AXIS_2 = col_double()))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
acacia <- read.csv(file = "../data-raw/ACACIA_DREPANOLOBIUM_SURVEY.txt",
         sep = "\t",
         na.strings = "dead")
```

2) Quality Assurance

Visualize the data to assure it is good:

```
str(trees)
```

```
## spc_tbl_ [7,508 x 16] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ SURVEY      : num [1:7508] 1 2 3 4 5 1 2 3 4 5 ...
##  $ YEAR        : num [1:7508] 2009 2010 2011 2012 2013 ...
##  $ SITE        : chr [1:7508] "SOUTH" "SOUTH" "SOUTH" "SOUTH" ...
##  $ TREATMENT   : chr [1:7508] "TOTAL" "TOTAL" "TOTAL" "TOTAL" ...
##  $ BLOCK       : num [1:7508] 2 2 2 2 2 2 2 2 2 2 ...
##  $ PLOT        : chr [1:7508] "S2TOTAL" "S2TOTAL" "S2TOTAL" "S2TOTAL" ...
##  $ SPECIES     : chr [1:7508] "Acacia_etbaica" "Acacia_etbaica" "Acacia_etbaica" "Acacia_etbaica" ..
##  $ ORIGINAL_TAG: num [1:7508] 1 1 1 1 1 2 2 2 2 2 ...
##  $ NEW_TAG     : num [1:7508] NA NA NA NA NA NA NA NA NA NA ...
##  $ DEAD        : chr [1:7508] "N" "N" "N" "N" ...
##  $ HEIGHT      : num [1:7508] 3.4 3.32 3.65 3.74 3.59 2.3 2.32 2.75 NA 2.86 ...
##  $ AXIS_1      : num [1:7508] 6.1 8.25 8.85 5.5 5 2.2 2.75 3.3 NA 3.7 ...
##  $ AXIS_2      : num [1:7508] 5 8.45 9 7.1 8.15 2.8 2.65 3.8 NA 2.6 ...
##  $ CIRC        : num [1:7508] 37.8 18.8 57 60 55 14.2 18.4 25 NA 31 ...
##  $ MEASUREMENT : chr [1:7508] "D" "D" "C" "C" ...
##  $ STEMS       : chr [1:7508] "1" "1" "1" "1" ...
```

```
##  - attr(*, "spec")=
##   .. cols(
##   ..   SURVEY = col_double(),
##   ..   YEAR = col_double(),
##   ..   SITE = col_character(),
##   ..   TREATMENT = col_character(),
##   ..   BLOCK = col_double(),
##   ..   PLOT = col_character(),
##   ..   SPECIES = col_character(),
##   ..   ORIGINAL_TAG = col_double(),
##   ..   NEW_TAG = col_double(),
##   ..   DEAD = col_character(),
##   ..   HEIGHT = col_double(),
##   ..   AXIS_1 = col_double(),
##   ..   AXIS_2 = col_double(),
##   ..   CIRC = col_double(),
##   ..   MEASUREMENT = col_character(),
##   ..   STEMS = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(acacia)
```

```
## 'data.frame':    157 obs. of  15 variables:
##  $ SURVEY   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ YEAR     : int  2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...
##  $ SITE     : chr  "SOUTH" "SOUTH" "SOUTH" "SOUTH" ...
##  $ BLOCK    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ TREATMENT: chr  "TOTAL" "TOTAL" "TOTAL" "TOTAL" ...
##  $ PLOT     : chr  "S1TOTAL" "S1TOTAL" "S1TOTAL" "S1TOTAL" ...
##  $ ID       : int  581 582 3111 3112 3113 3114 3115 3199 941 942 ...
##  $ HEIGHT   : num  2.25 2.65 1.5 2.01 1.75 1.65 1.2 1.45 1.87 2.38 ...
##  $ AXIS1    : num  2.75 4.1 1.7 1.8 1.84 1.62 1.95 2 2.15 5.55 ...
##  $ AXIS2    : num  2.15 3.9 0.85 1.6 1.42 0.85 0.9 1.75 1.82 4.82 ...
##  $ CIRC     : num  20 28 17 12 13 15 9 12.2 13 35 ...
##  $ FLOWERS  : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ BUDS     : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ FRUITS   : int  10 150 50 75 20 0 0 25 0 50 ...
##  $ ANT      : chr  "CS" "TP" "TP" "CS" ...
```

3) Plot the two data sets on the same plot and fit linear models.

The trick to do this is to call the `ggplot()` function with no data set. This allows us to provide a different data set for any and each function we call after. In this way we can plot data from different data set unto the same plot. The downside is that we have to specify the data set for **EVERY** layer that we want to add. This is different from what happens when we specify the data within the base layer `ggplot()`.

```
ggplot() +
  geom_point(data = trees, mapping = aes(x = CIRC, y = HEIGHT), color = "gray", alpha = 0.5) +
  scale_x_log10() +
  scale_y_log10() +
  geom_smooth(data = trees, mapping = aes(x = CIRC, y = HEIGHT), method = "lm", color = "black") +
  geom_point(data = acacia, mapping = aes(x = CIRC, y = HEIGHT), color = "red", alpha = 0.8) +
  geom_smooth(data = acacia, mapping = aes(x = CIRC, y = HEIGHT), method = "lm", color = "red") +
  labs(x = "Tree circumference", y = "Tree height")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 414 rows containing non-finite values (`stat_smooth()`).
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 4 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 414 rows containing missing values (`geom_point()`).
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```