Making a simple dataframe

Nathan L. Brouwer | brouwern@gmail.com | @lobrowR

October 11, 2017

In this walkthrough we'll take a simple dataset that is presented as lists of seperate numbers (eg 1,2,3,4) and put them into vectors. We'll then take these vectors and make simply dataframe. We'll then make a boxplot using ggplot2 and ggpubr.

Working with data in 2 seperate vectors

- Data often come in 2 seperate sets of numbers, such as "the control group had lengths 4, 5, 6, 3, and 3, while the treatment group had length 11, 12, 6, 4 and 7".
- One way to work with data like this in R is to load each set of numbers into a "vector" using the c() function

```
control <-   c(4,  5,6,3,3)
treatment <- c(11,12,6,4,7)
```

Summary Statistics

We can calculate the mean and other summary statistics of each vector on its own

```
mean(control)
```

```
## [1] 4.2
```

```
sd(control)
```

```
## [1] 1.30384
```

```
summary(control)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     3.0     3.0     4.0     4.2     5.0     6.0
```

Making a dataframe

To plot data its is almost always best to set it up in a dataframe.

Starting with the raw data we could make a dataframe like this

```
dat <- data.frame(length = c(4,  5,6,3,3,
                             11,12,6,4,7),
                  group = c("C","C","C","C","C",
                            "T","T","T","T","T"))
```

We can make a dataframe from the two vectors we've already typed up like this by surrounding them with a c() like this " c(control, treatment) "

```
dat <- data.frame(length = c(control,
                             treatment),
                  group = c("C","C","C","C","C",
                            "T","T","T","T","T"))
```
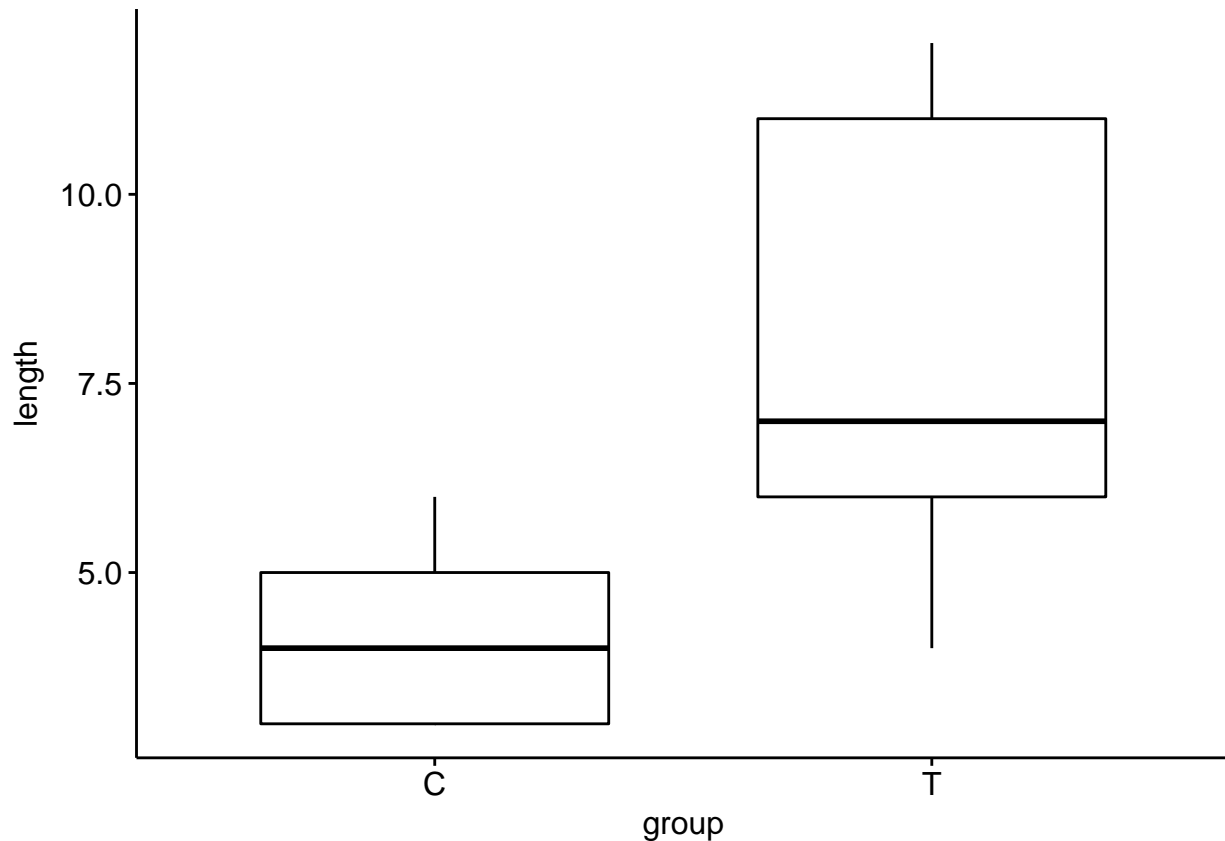
Making a boxplot

We can make a boxplot from the data like this using ggpubr

```
library(ggplot2)
library(ggpubr)
```

```
## Loading required package: magrittr
```

```
ggboxplot(data = dat,
          y = "length",
          x = "group")
```
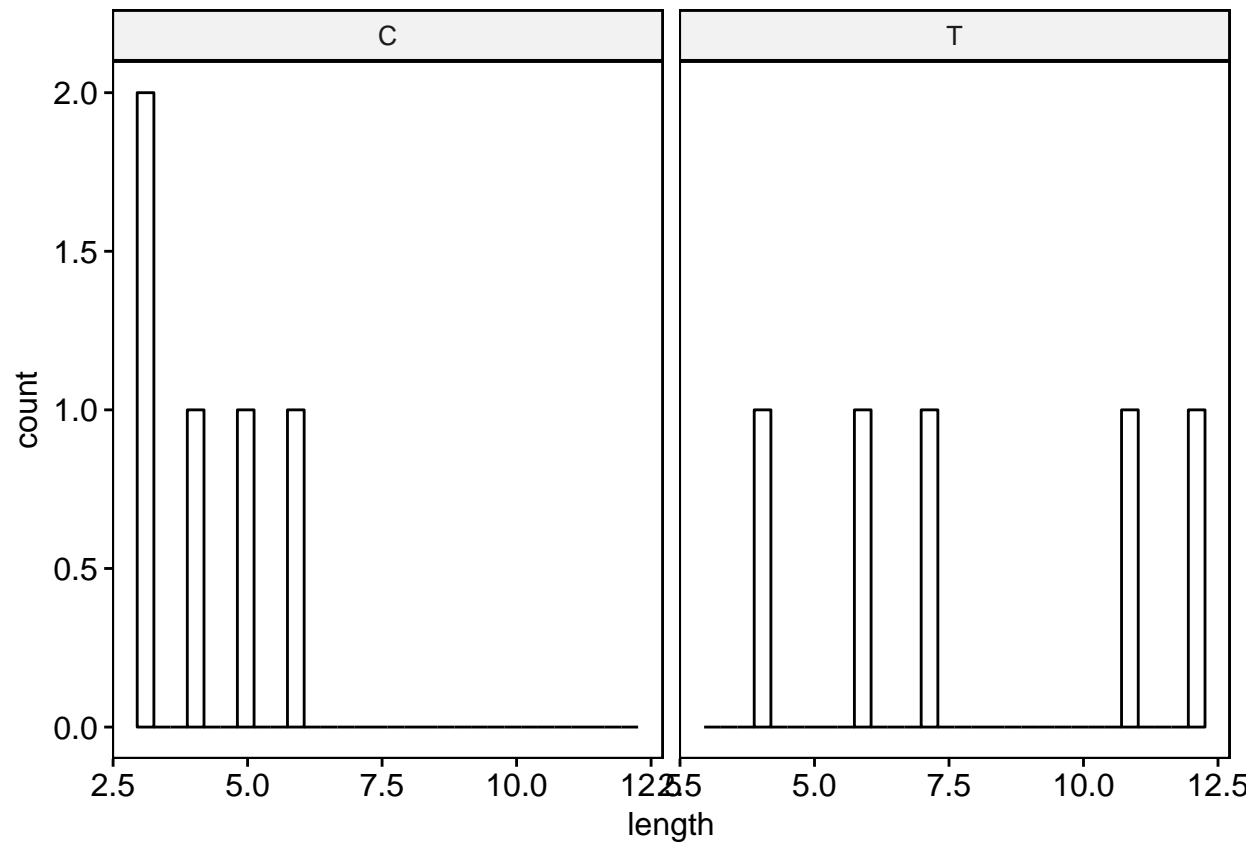

Making a histogram

- An alternative to using a boxplot is a histogram.
- We can split up the data by treatment using facet.by = "group"
- Because my sample dataset has so few datapoints this doesn't look very good.

```
gghistogram(data = dat,
            x = "length",
            facet.by = "group")
```
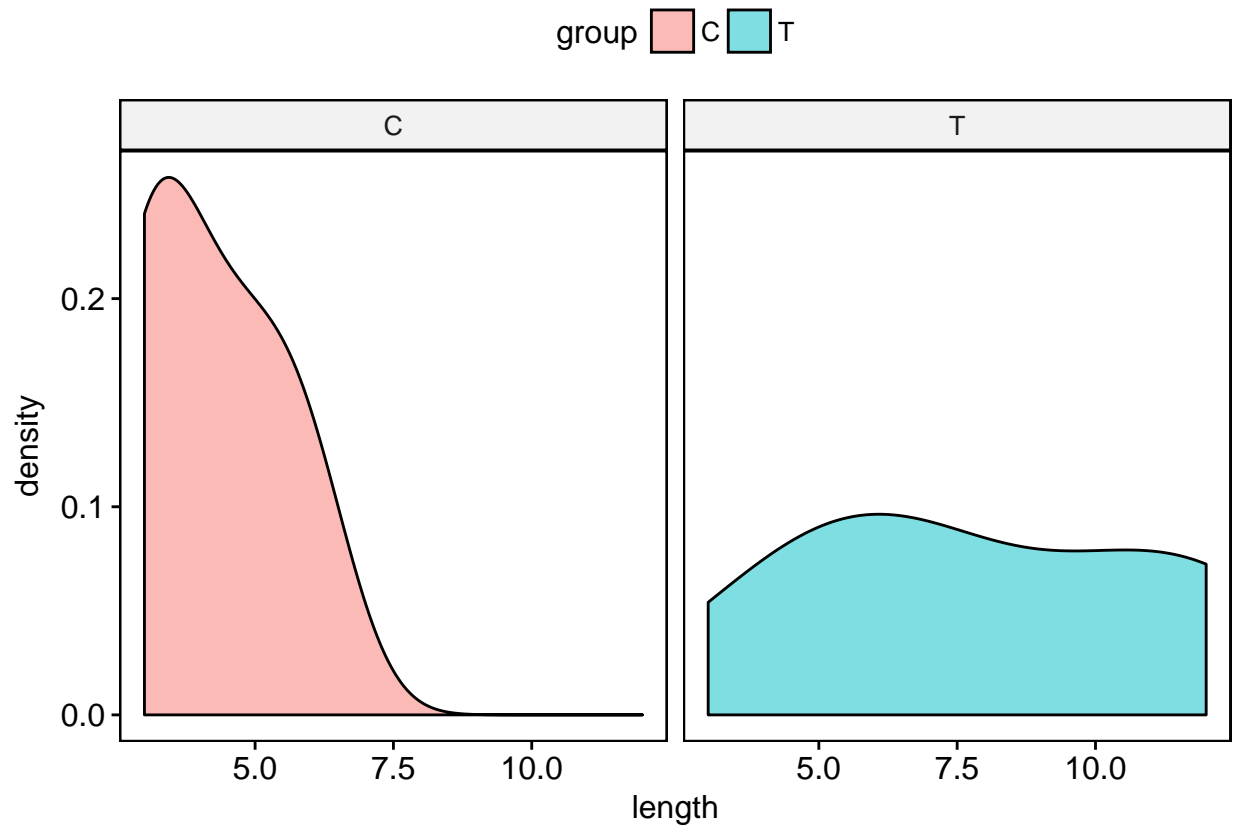
```
## Warning: Using `bins = 30` by default. Pick better value with the argument
## `bins`.
```

Making a density plot

A plot that is similar to a histogram is a density plot.

```
ggdensity(data = dat,
          x = "length",
          facet.by = "group",
          fill = "group")
```

Advanced

You can save yourself some typing by using the rep() command when making your dataframe since the group codes "C" and "T" get repeated. IF you do this you need to make sure you keep track of all your parenetheses.

```r
dat <- data.frame(length = c(control,
                             treatment),
                  group = c(rep("C",5),
                            rep("T",5)))
```
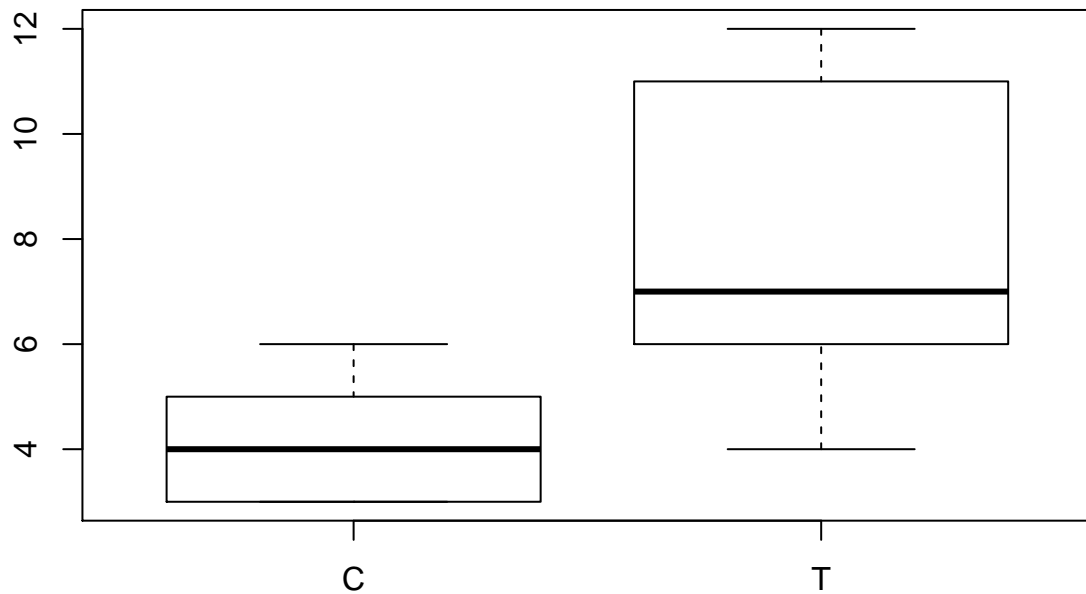
Historical reference

Base R

In the old days (eg 2008. . .) we used to make our boxplot using the boxplot function.

```r
boxplot(length ~ group, data = dat)
```


In the less old days (eg 2015) we used to make our boxplots using regular qplot or ggplot

ggplot's qplot

```r
qplot(data = dat,
      y = length,
      x = group,
      geom = "boxplot")
```

Standard ggplot

```r
ggplot(data = dat,
       aes(y = length,
       x = group)) +
         geom_boxplot()
```