

Scatterplots in R using `qplot()`

Nathan Brouwer

2017-09-11

Scatterplots: 2 continuous variables

In this lab we'll explore how to make scatterplots using the `qplot()` function in `ggplot2`.

R Preliminaries

- We'll use the `qplot()` function in the *ggplot* package
- The *cowplot* package provides nice defaults for `ggplot` IMHO

```
library(ggplot2)
library(cowplot)
```

Scatterplot of Iris data

- Let's make a scatter plot, where we plot 2 continuous, numeric variables against each other
- that is, both x and y variables are numbers; not categories

I've forgotten the names of all the iris variables, so I'll use the `names()` command to see what they are

```
names(iris)
```

I'll plot the sepals against the petals

```
qplot(y = Sepal.Length,
      x = Petal.Length,
      data = iris)
```

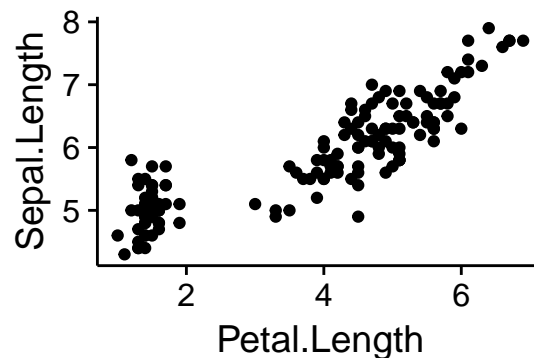


Figure 1: Sepals vs. Petals

Scatter plot of mammal brain data

Let's look at another dataset

Preliminaries

Get the data from the ggplot2 package

```
data(msleep)
```

Look at the data

#How much data is there?

```
dim(msleep)
```

#What does the data look like

```
head(msleep)
```

#Summary of the data

```
summary(msleep)
```

There are a number of “categorical” variables in this dataset

- genus
- vore = carnivore, omnivore et
- order = taxonomic order
- conservation = conservation status (endangered, etc)

For some reason they don’t load as “factor” variables (better known as categorical or grouping variables, but called “Factors” in R-land)

We can make these factors using the factor() command

```
msleep$vore <- factor(msleep$vore)
```

Now see what happens when you call summary()

```
summary(msleep)
```

Do the same for “order”

```
msleep$order <- factor(msleep$order)
```

```
summary(msleep)
```

And “conservation”

```
msleep$conservation <- factor(msleep$conservation)
```

```
summary(msleep)
```

Make a basic scatterplot

```
qplot(y = sleep_total,  
      x = brainwt,  
      data = msleep)
```

That looks really really ugly. It will work better if we “log transform the axes”

```
qplot(y = log(sleep_rem),  
      x = log(brainwt),  
      data = msleep)
```

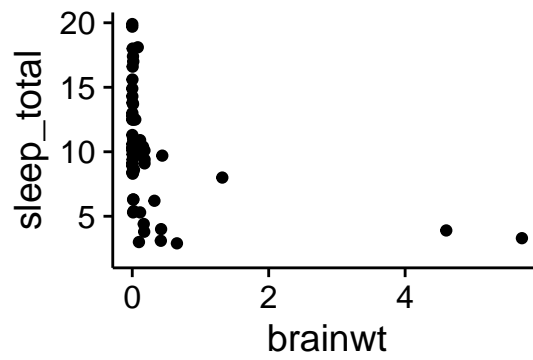


Figure 2: Mammal sleep, raw data

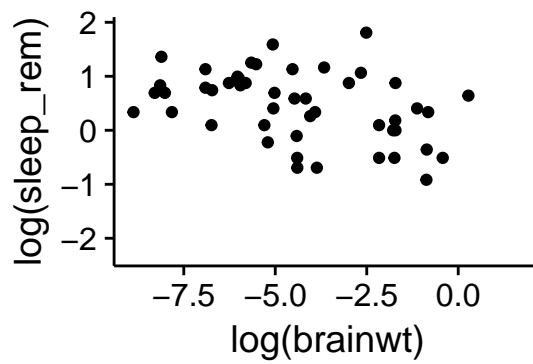


Figure 3: Mammal sleep, logged data

Things get logged all the time in stats. We'll talk more about that later.

Add color coding to scatterplot

```
qplot(y = log(sleep_rem),  
      x = log(brainwt),  
      data = msleep,  
      color = vore)
```

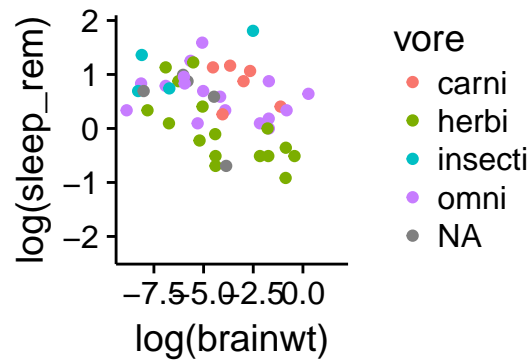


Figure 4: Add colors with color =

Add color & shape coding to scatterplot

```
qplot(y = log(sleep_rem),  
      x = log(brainwt),  
      data = msleep,  
      color = vore,  
      shape = vore)
```

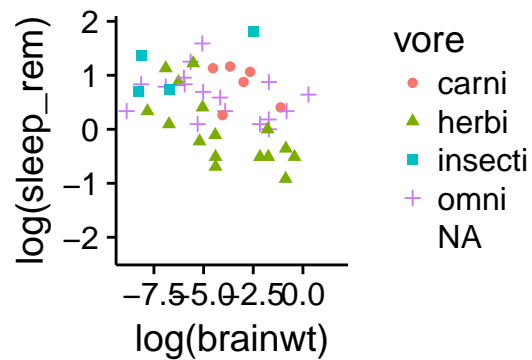


Figure 5: Add shapes with shape =

Put different “vores” in separate panels

- Separate panels can be made using the “facet” argument withing qplot

```
qplot(y = log(sleep_rem),
      x = log(brainwt),
      data = msleep,
      color = vore,
      shape = vore,
      facets = vore ~ .)
```

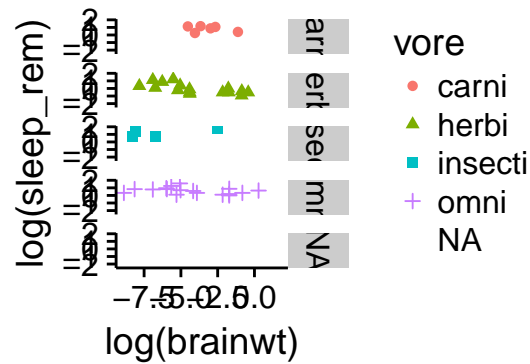


Figure 6: Split into different panels w/ facets =

Add a “trend line” to a scatterplot

- Add the `geom_smooth()` function after the initial `qplot()` command
- This works best if we remove the “color = vore” command, but you can see what happens if you leave it

```
qplot(y = log(sleep_rem),
      x = log(brainwt),
      data = msleep) +
  geom_smooth()
```

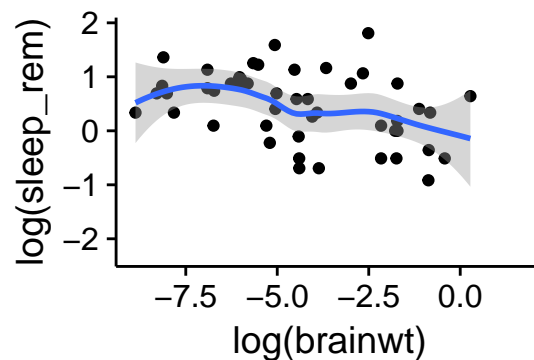


Figure 7: Add trendline with + `geom_smooth()`

Exercise: Modify mammal brain code to do the following things

- Change the axes labels
- Add a title

- Use `names(msleep)` to see what other variables are in the dataset
- Use `summary(msleep)` to whether they are continuous or categorical
- Pick another continuous variable and plot it instead of `sleep_total`
- Try this with and without logging