

TOOL

Changing Scales With ggplot()

When creating a data visualization, you'll presumably want to control aspects like which colors are used in the plot and what tick marks appear on the x- and y-axes. You can do so using scale functions, which allow you to adjust the scales, or values, used for colors, shapes, sizes, numbers, and more. The ggplot2 package offers a variety of scale functions; we summarize some of the most commonly used scale functions here.

For each of the following examples, we've used data from the **storms.csv** file you used during the course. To set that data up for use with this tool, use the following code:

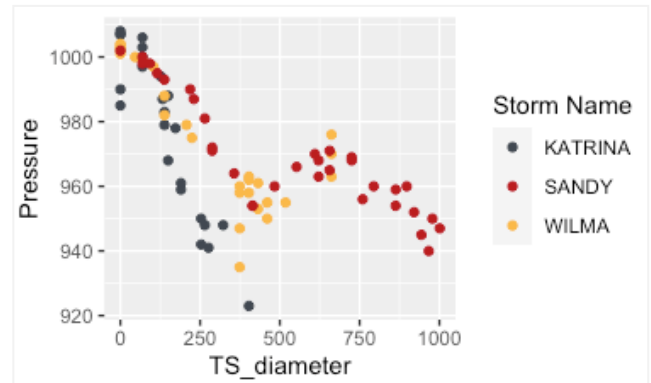
```
library(tidyverse)
# Read in the storm data
storms <- read.csv("storms.csv")
# Set the storm category to be a factor
storms$Category <- factor(storms$Category, levels = -1:5)
# Set the measurement date/time to be a factor
storms$Date <- factor(storms$Date, levels = unique(storms$Date))
# Create data set with observations only for Hurricanes Katrina, Sandy,
# and Wilma
sampleStorms <- storms %>% filter(Name %in% c("Katrina", "Sandy", "Wilma"))
```

Changing Color, Fill, and Shape for Categorical Variables

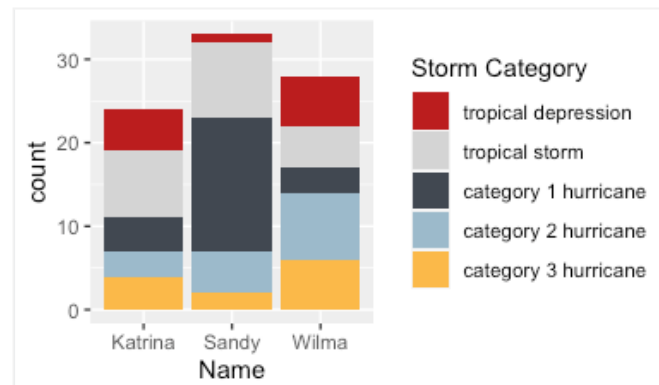
When you create an aesthetic mapping that distinguishes categories, R will apply default colors or shapes, but you may want to change them. You can use **scale_color_manual()** to change the colors of categories within a scatterplot, **scale_fill_manual()** to change the colors of categories within a bar chart, and **scale_shape_manual()** to change the shape of categories within a scatterplot. The characteristics you can manually change via arguments to these functions are **value**, **labels**, and **name**. Together, these arguments allow you to specify the color or shape associated with each category, change the legend title, and change the legend labels. You can specify colors via their hex codes, which are six-digit codes used to represent colors based on their red, blue, and green components, or via a string (e.g., "red," "black," "blue"). You can specify shapes by their "point characteristic," or **pch**, which you can look up in the help menu or online.



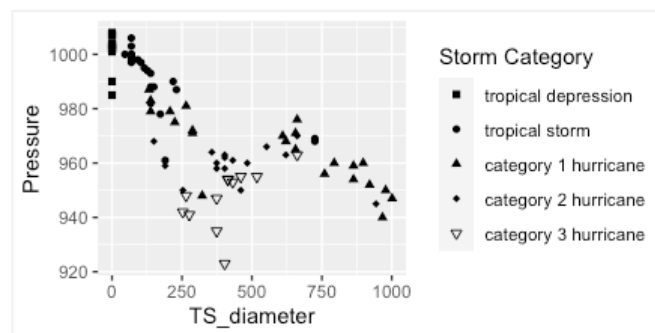
```
ggplot(data = sampleStorms, aes(x = TS_
diameter, y = Pressure, color = Name)) +
  geom_point() +
  scale_color_manual(values = c('Katrina' =
'#393f47', 'Sandy' = '#b31b1b', 'Wilma'
= '#fbb040')),
  labels = c('Katrina' = "KATRINA", 'Sandy'
= "SANDY", 'Wilma' = "WILMA"),
  name = "Storm Name")
```



```
ggplot(sampleStorms, aes(x = Name, fill =
Category)) +
  geom_bar() +
  scale_fill_manual(values = c('-1' =
'#b31b1b', '0' = '#cecece', '1' =
'#393f47', '2' = '#92b2c4', '3' =
'#fbb040')),
  labels = c('-1' = 'tropical depression',
'0' = 'tropical storm', '1' = 'category
1 hurricane', '2' = 'category 2
hurricane', '3' = 'category 3
hurricane'),
  name = 'Storm Category')
```



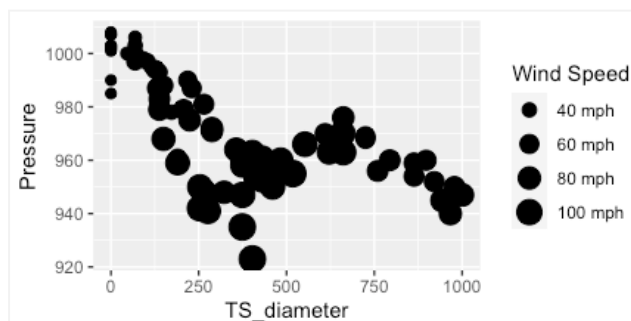
```
ggplot(data = sampleStorms, aes(x = TS_
diameter, y = Pressure, shape = Category)) +
  geom_point() +
  scale_shape_manual(values = c('-1' = 15,
'0' = 16, '1' = 17, '2' = 18, '3' = 25),
  labels = c('-1' = 'tropical depression',
'0' = 'tropical storm', '1' = 'category 1
hurricane', '2' = 'category 2 hurricane',
'3' = 'category 3 hurricane'),
  name = 'Storm Category')
```



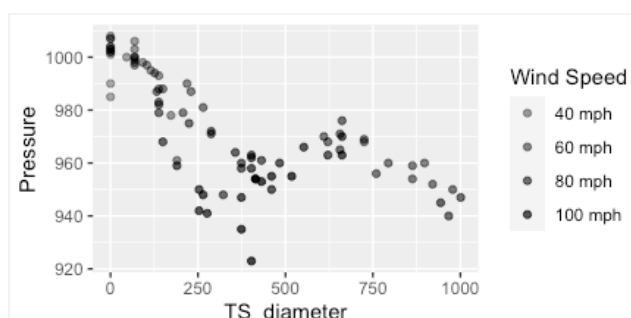
Changing Size and Transparency for Quantitative Variables

When you want to distinguish characteristics within a gradient, you may want R to scale a particular characteristic based on the value of the variable. When these values exist along a spectrum, you can use `scale_size()` to scale the point sizes, `scale_alpha()` to scale the point transparencies, and `scale_color_gradient()` to scale the point colors. These functions use the arguments `names` and `labels` to adjust the legend, just as the previous functions did. These functions, however, also take the argument `breaks` to identify which sizes, transparencies, or colors should be marked in the legend. Additionally, `scale_size()` and `scale_alpha()` take the argument `range` to specify the high and low ends of the scales, while `scale_color_gradient()` takes the arguments `low` and `high` to do the same.

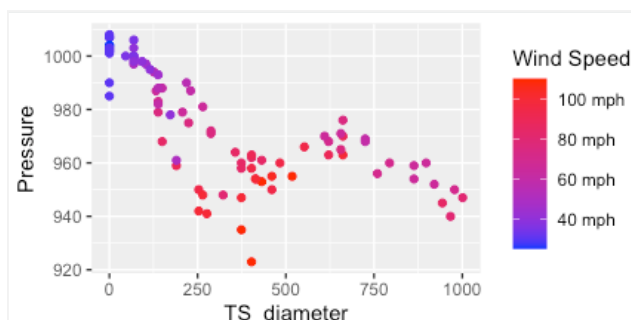
```
ggplot(data = sampleStorms, aes(x = TS_
diameter, y = Pressure, size = Wind)) +
  geom_point() +
  scale_size(breaks = c(40, 60, 80, 100),
    labels = c("40 mph", "60 mph", "80 mph",
      "100 mph"),
    name = "Wind Speed",
    range = c(1,6))
```



```
ggplot(data = sampleStorms, aes(x = TS_
diameter, y = Pressure, alpha = Wind)) +
  geom_point() +
  scale_alpha(breaks = c(40, 60, 80, 100),
    labels = c("40 mph", "60 mph", "80 mph",
      "100 mph"),
    name = "Wind Speed",
    range = c(0.4,0.8))
```



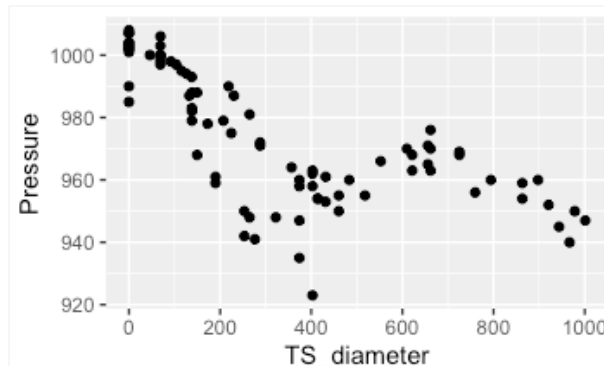
```
ggplot(data = sampleStorms, aes(x = TS_
diameter, y = Pressure, color = Wind)) +
  geom_point() +
  scale_color_gradient(low = "blue",
    high = "red",
    breaks = c(40, 60, 80, 100),
    labels = c("40 mph", "60 mph", "80 mph",
      "100 mph"),
    name = "Wind Speed")
```



Changing Axis Scales

R creates scales for the x- and y-axes based on the range of values of the corresponding variables. For example, the default scale for `TS_diameter` runs from 0 to 1000 in increments of 250. You might want to adjust these scales. For example, if the variable on the x-axis is continuous, you can adjust the x-axis scale with the function `scale_x_continuous()`. Below, the `TS_diameter` scale is adjusted so that it runs from 0 to 1000 but in increments of 200.

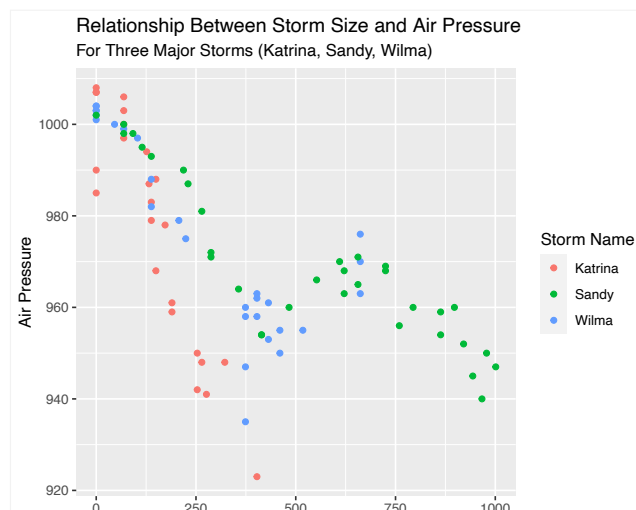
```
ggplot(data = sampleStorms, aes(x = TS_
diameter, y = Pressure)) +
  geom_point() +
  scale_x_continuous(breaks = c(0, 200,
400, 600, 800, 1000),
labels = c("0", "200", "400", "600",
"800", "1000"))
```



Setting Axis and Plot Titles

You can use the `labs()` function to set the axis names, the plot title, and even a subtitle. You can also adjust the name(s) of any legend(s) you created. In the example below, the points are colored according to storm (`color = "Name"`). Inside `labs()`, we can set `color = "Storm Name"` so that the legend is titled "Storm Name."

```
ggplot(data = sampleStorms, aes(x = TS_
diameter, y = Pressure, color = Name)) +
  geom_point() +
  labs(x = "Tropical Storm Diameter",
y = "Air Pressure",
color = "Storm Name",
title = "Relationship Between Storm
Size and Air Pressure",
subtitle = "For Three Major Storms
(Katrina, Sandy, Wilma)")
```



Combining Scales

You can use any combination of the functions above to produce visualizations that are more advanced.

```
ggplot(data = sampleStorms, aes(x = TS_
diameter, y = Pressure, color = Wind,
shape = Name)) +
  geom_point() +
  scale_color_gradient(low = "blue", high =
"red",
breaks = c(40, 60, 80, 100),
labels = c("40 mph", "60 mph", "80 mph",
"100 mph")) +
  scale_shape_manual(values = c('Katrina' =
15, 'Sandy' = 16, 'Wilma' = 17)) +
  labs(x = "Tropical Storm Diameter",
y = "Air Pressure",
color = "Wind Speed",
shape = "Storm Name",
title = "Relationship Between Storm
Size, Air Pressure, and Wind Speed",
subtitle = "For Three Major Storms
(Katrina, Sandy, Wilma)")
```

