Convey Y About X Data Set

Step 1: Set the theme and load your data.

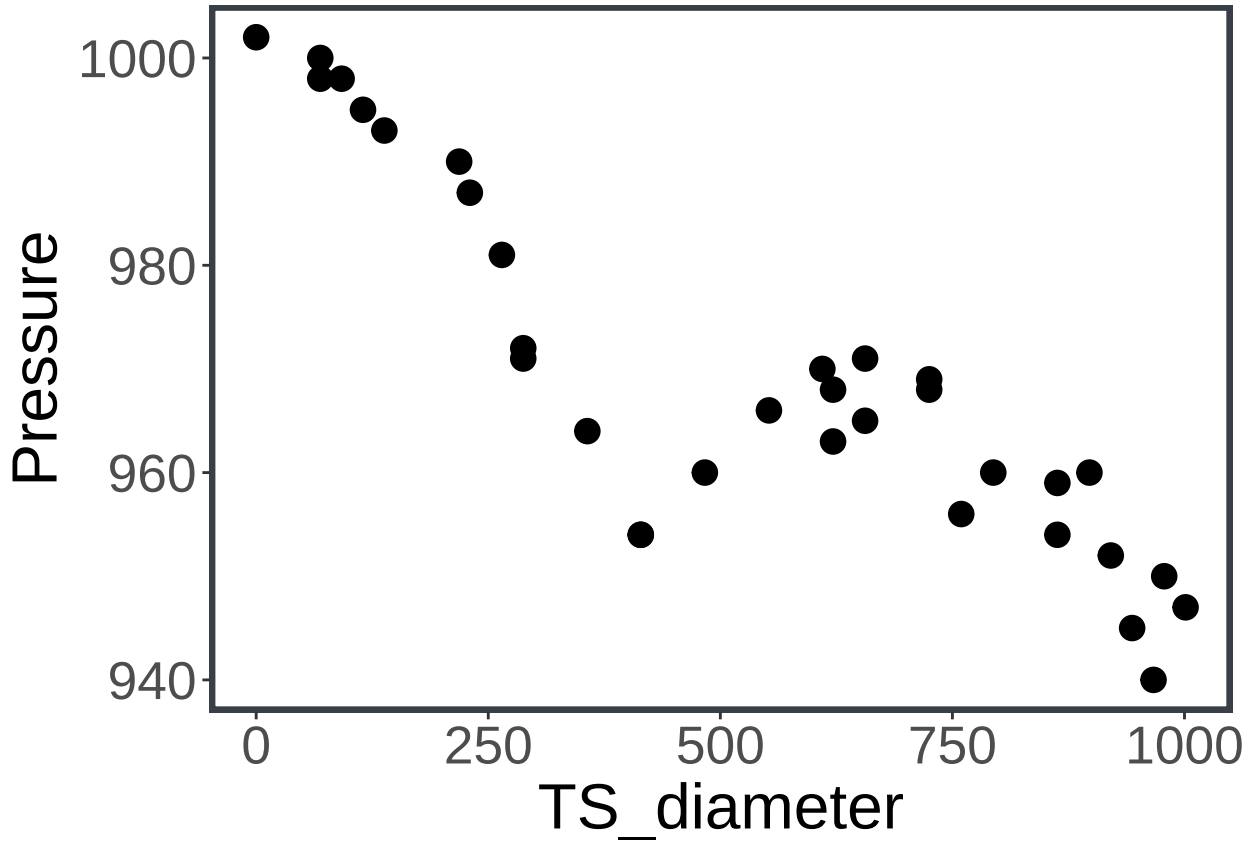
```
# 1) Set up the theme used in these videos:
library(ggplot2)

theme <-  theme(plot.margin = margin(5, 5, 5, 5, "pt"),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.background = element_blank(),
          panel.border = element_rect(colour = "#393f47", fill = NA, size = 2),

          axis.text = element_text(size = 20),
          axis.title.x = element_text(size = 24),
          axis.title.y = element_text(size = 24),
          plot.title = element_text(face = "bold", size = 30))

ourTheme <- list(theme, scale_color_manual(values = c('#393f47',
       '#b31b1b', '#fbb040', '#92b2c4', '#cecece')))

# 2) Load data from the National Oceanic and Atmospheric Administration's
# Atlantic hurricane database and convert variables to factors where necessary.

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))

# View(storms) # Look at the storms data

# 3) Filter the data to only use observations for Hurricane Sandy

sandy <- storms %>% filter(Name == "Sandy")
```

Step 2: Make a scatterplot of TS_diameter versus air pressure for Hurricane Sandy.

When you don't specify a color, all the points on the plot will be black:

```
# basic scatterplot (all points black)
ggplot(data = sandy, aes(x = TS_diameter, y = Pressure)) +
```

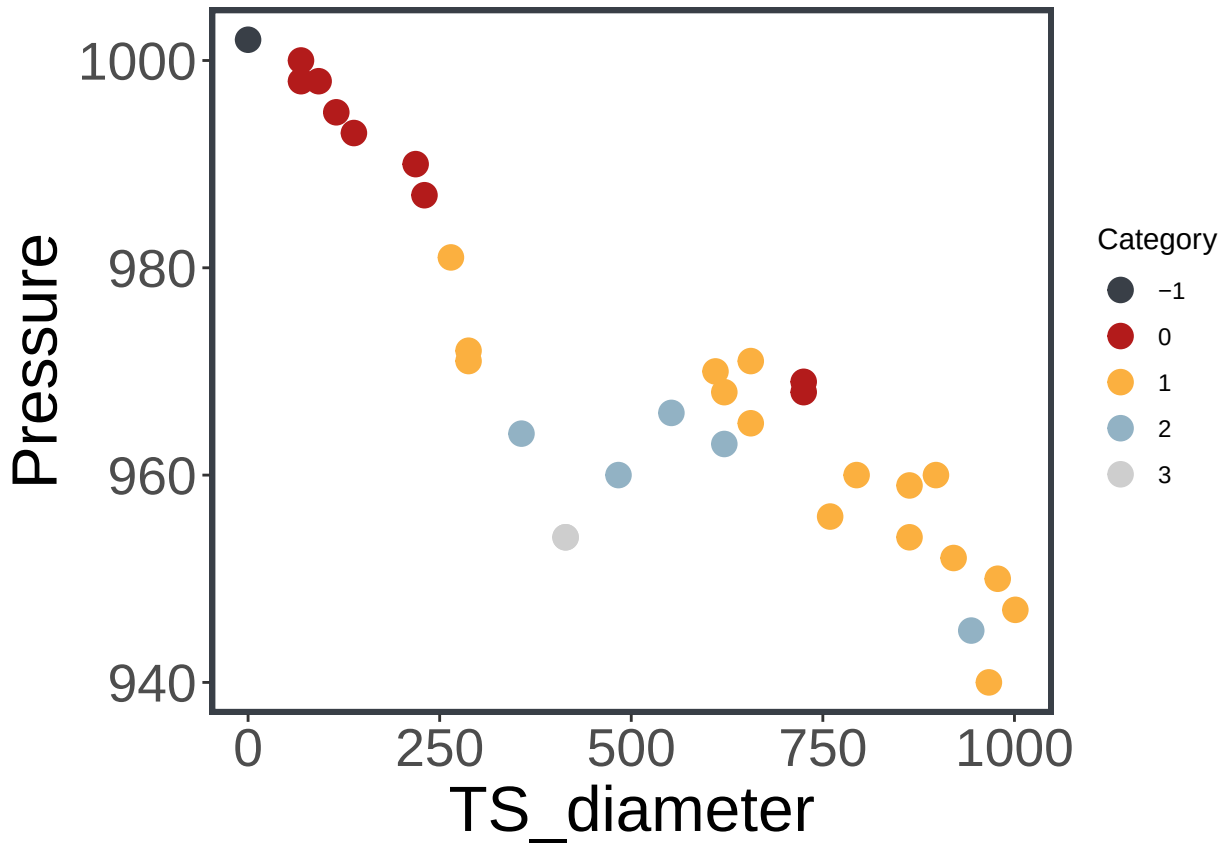
```
        geom_point(size = 4) +
        ourTheme
```



Step 3: Adjust point color to reflect storm category.

When you adjust points within the aes() function, the change will affect every point. Here, setting color = Category means point color depends on the storm category:

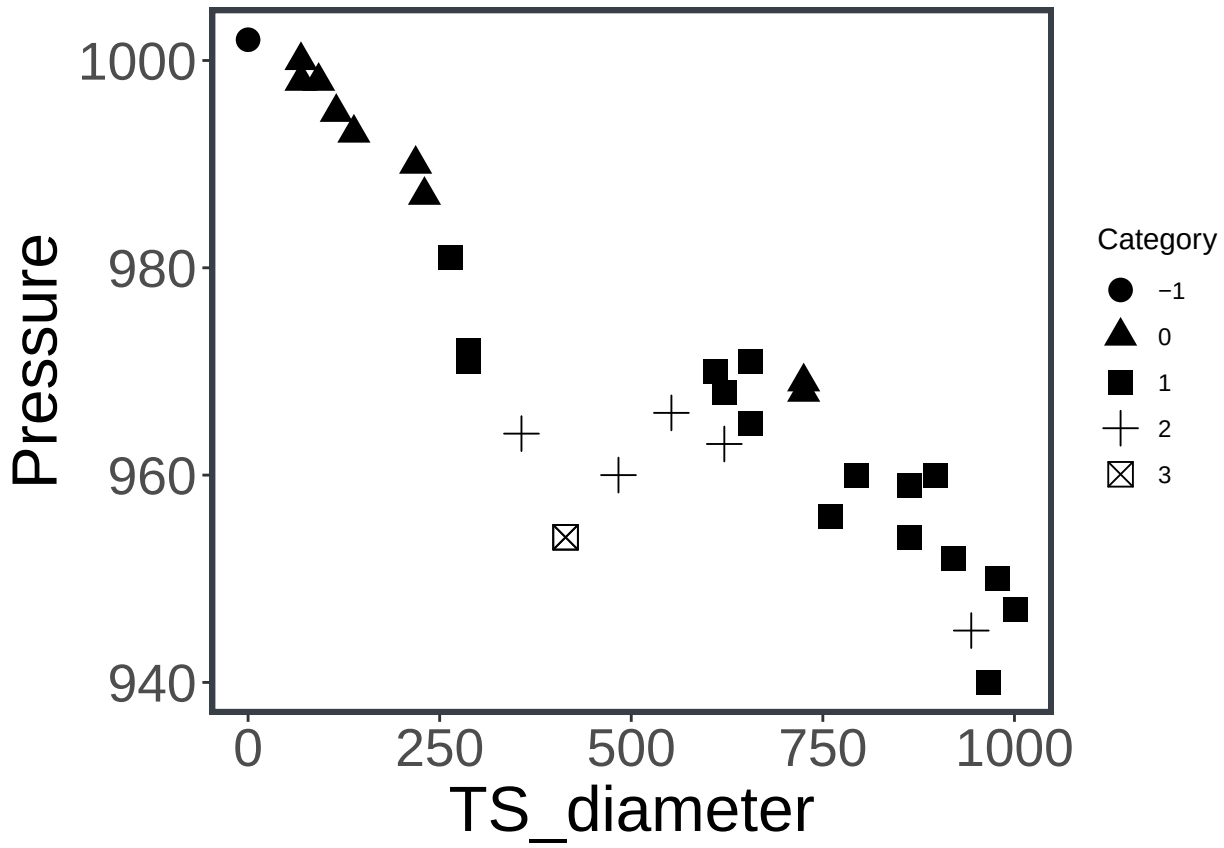
```
# color of points depends on storm category
ggplot(data = sandy, aes(x = TS_diameter, y = Pressure, color = Category)) +
        geom_point(size = 4) +
        ourTheme
```



Step 4: Change the point shapes to reflect category.

When you adjust points within the aes() function, the change will affect every point. Here, setting shape = Category means point shape depends on the storm category:

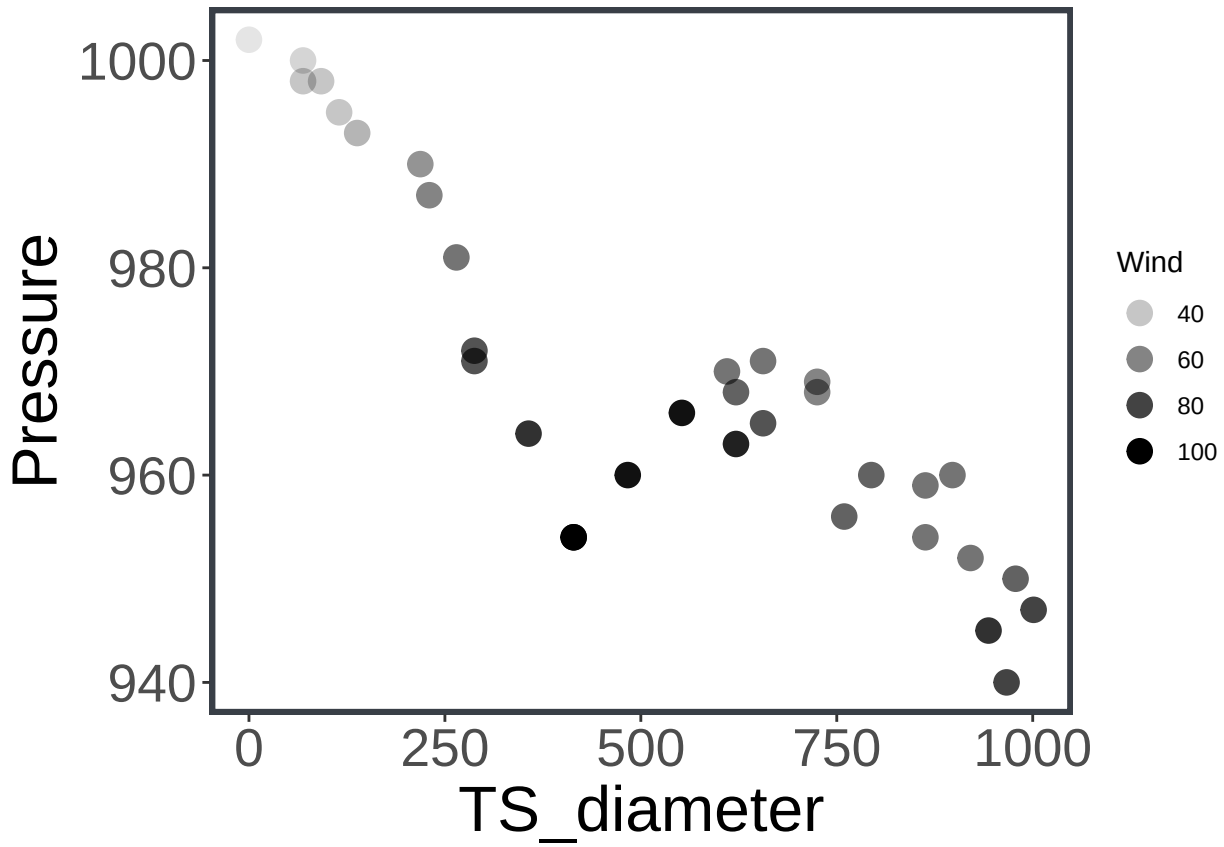
```
# shape of points depends on storm category
ggplot(data = sandy, aes(x = TS_diameter, y = Pressure, shape = Category)) +
        geom_point(size = 4) +
        ourTheme
```



Step 5: Change the point transparency to reflect wind speed.

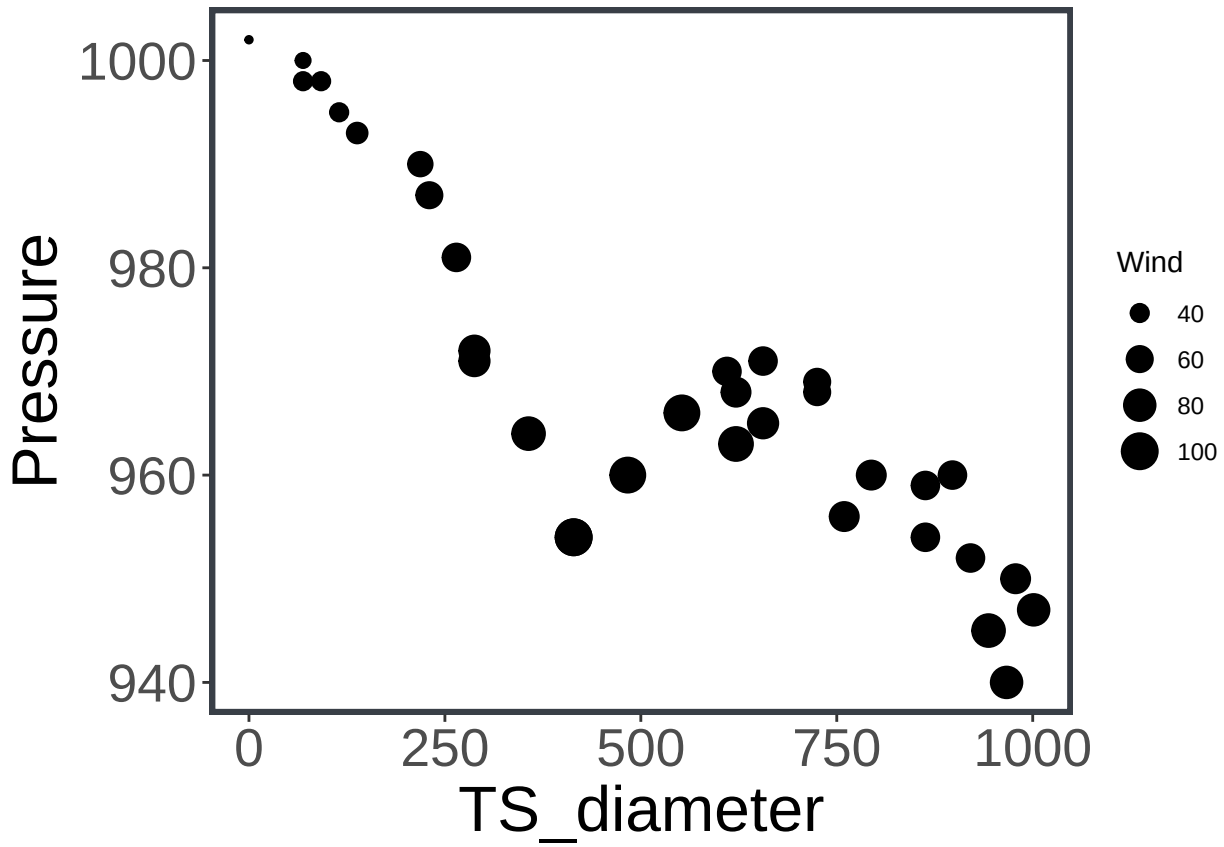
When you adjust points within the aes() function, the change will affect every point. Here, setting alpha = Wind means point transparency depends on the wind speed:

```
# transparency of points depends on wind speed
ggplot(data = sandy, aes(x = TS_diameter, y = Pressure, alpha = Wind)) +
        geom_point(size = 4) +
        ourTheme
```



Step 6: Change the point size to reflect wind speed.

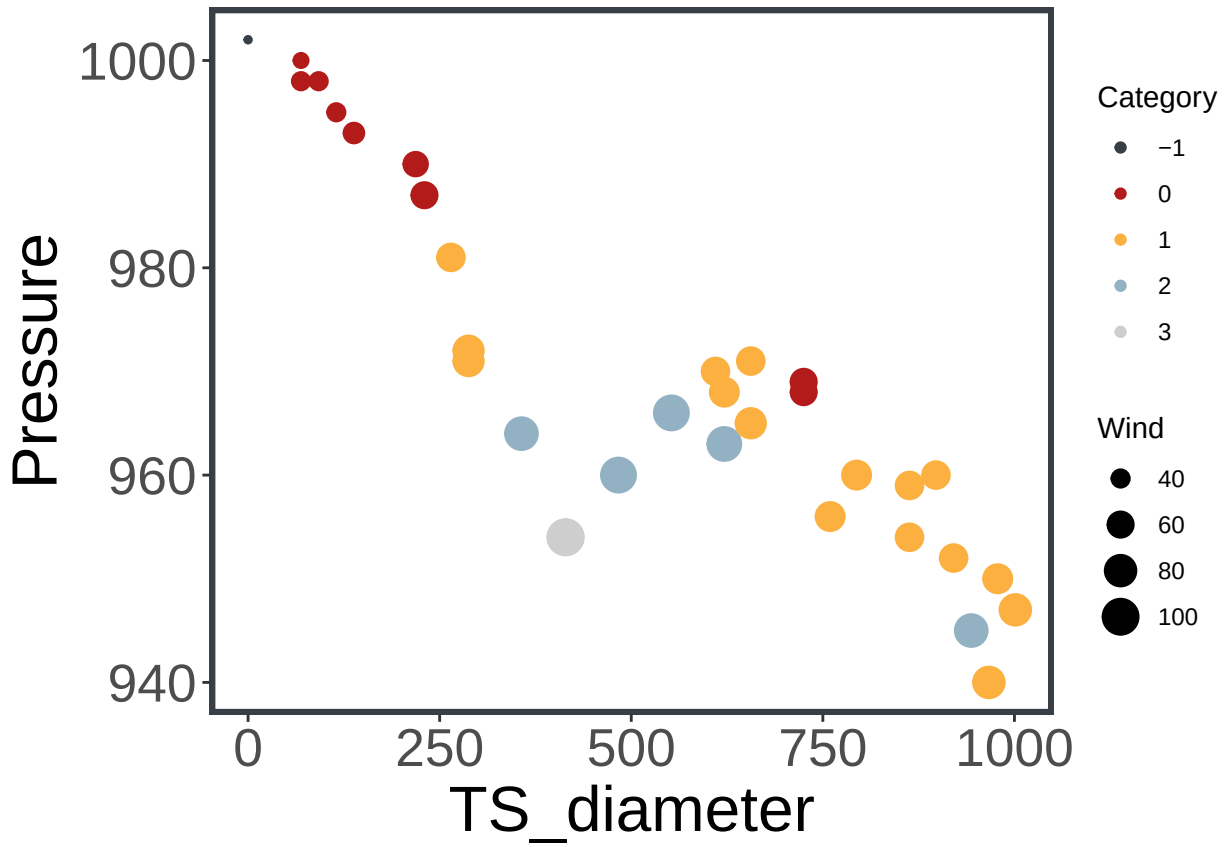
```
# size of points depends on wind speed
ggplot(data = sandy, aes(x = TS_diameter, y = Pressure, size = Wind)) +
        geom_point() +
        ourTheme
```



Step 7: Combine aesthetic adjustments.

You can adjust multiple aesthetics at once. Here, setting color = Category and size = Wind means point color depends on Category and point size depends on the wind speed:

```
# color of points depends on storm category, size of points depends on wind speed
ggplot(data = sandy, aes(x = TS_diameter, y = Pressure, color = Category, size = Wind)) +
        geom_point() +
        ourTheme
```


Pressure
1000
980
960
940
0
250
500
750
1000
TS_diameter
Category
−1
0
1
2
3
Wind
40
60
80
100