Create Groups of Plots

Step 1: Set the theme and load your data.

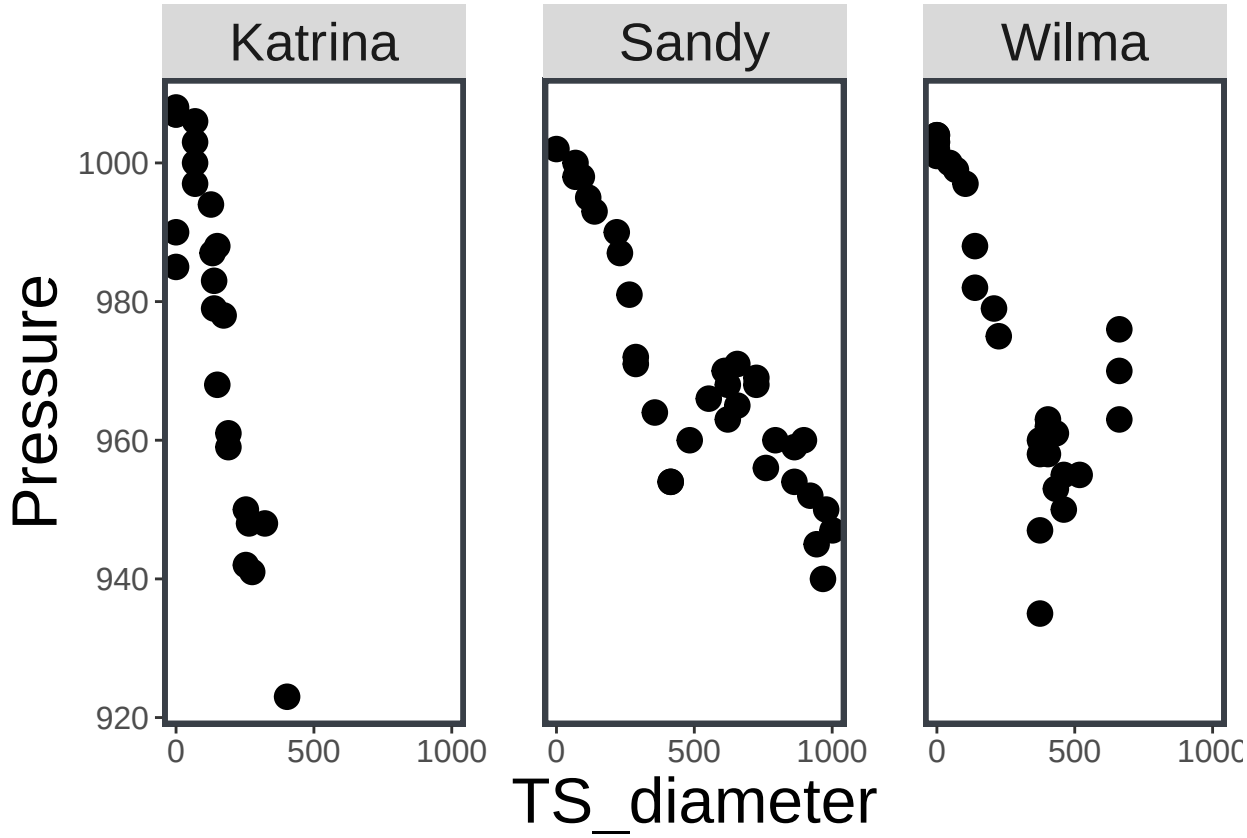
```
# 1) Set up the themes used in these videos:
library(tidyverse)
library(ggplot2)

theme1 <-  theme(plot.margin = margin(6, 6, 6, 6, "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 = 12),
        axis.title.x = element_text(size = 24),
        axis.title.y = element_text(size = 24),
        plot.title = element_text(face = "bold", size = 30),
        strip.text = element_text(size = 20),
        panel.spacing = unit(2, "lines"))
ourTheme1 <- list(theme1, scale_color_manual(values =
    c('#393f47', '#b31b1b', '#fbb040', '#92b2c4')),
    scale_x_continuous(breaks = c(0, 500, 1000),
    labels = c(0, 500, 1000)))

theme2 <-  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 = 12),
        axis.title.x = element_text(size = 24),
        axis.title.y = element_text(size = 24),
        plot.title = element_text(face = "bold", size = 30),
        strip.text = element_text(size = 20),
        panel.spacing = unit(1, "lines"))
ourTheme2 <- list(theme2, scale_color_manual(values =
        c('#393f47', '#b31b1b', '#fbb040', '#92b2c4')),
        scale_x_continuous(breaks = c(0, 250, 500, 750, 1000),
        labels = c(0, 250, 500, 750, 1000)))

theme3 <-  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 = 15),
        axis.title.x = element_text(size = 24),
```

```
        axis.title.y = element_text(size = 24),
        plot.title = element_text(face = "bold", size = 30),
        strip.text = element_text(size = 20),
        panel.spacing = unit(1.3, "lines"))
ourTheme3 <- list(theme3, scale_color_manual(values =
                c('#393f47', '#b31b1b', '#fbb040', '#92b2c4')),
                scale_x_continuous(breaks = c(0, 500, 1000),
                                   labels = c(0, 500, 1000)),
                scale_y_continuous(breaks = c(920, 1000),
                                   labels = c(920, 1000)))

# 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))

# look at the storm data:
view(storms)

# 3) Filter the data to only use observations for Hurricane Sandy:
sandy <- storms %>% filter(Name == "Sandy")

# 4) Create a data set with observations only for
#    Hurricanes Katrina, Sandy, and Wilma:
sampleStorms <- storms %>%

  filter(Name %in% c("Katrina", "Sandy", "Wilma"))
```

Step 2: Create a separate scatterplot for each category.

Use facet_wrap() to create panels of plots based on a single categorical variable. By default, each plot has the same axis scales:

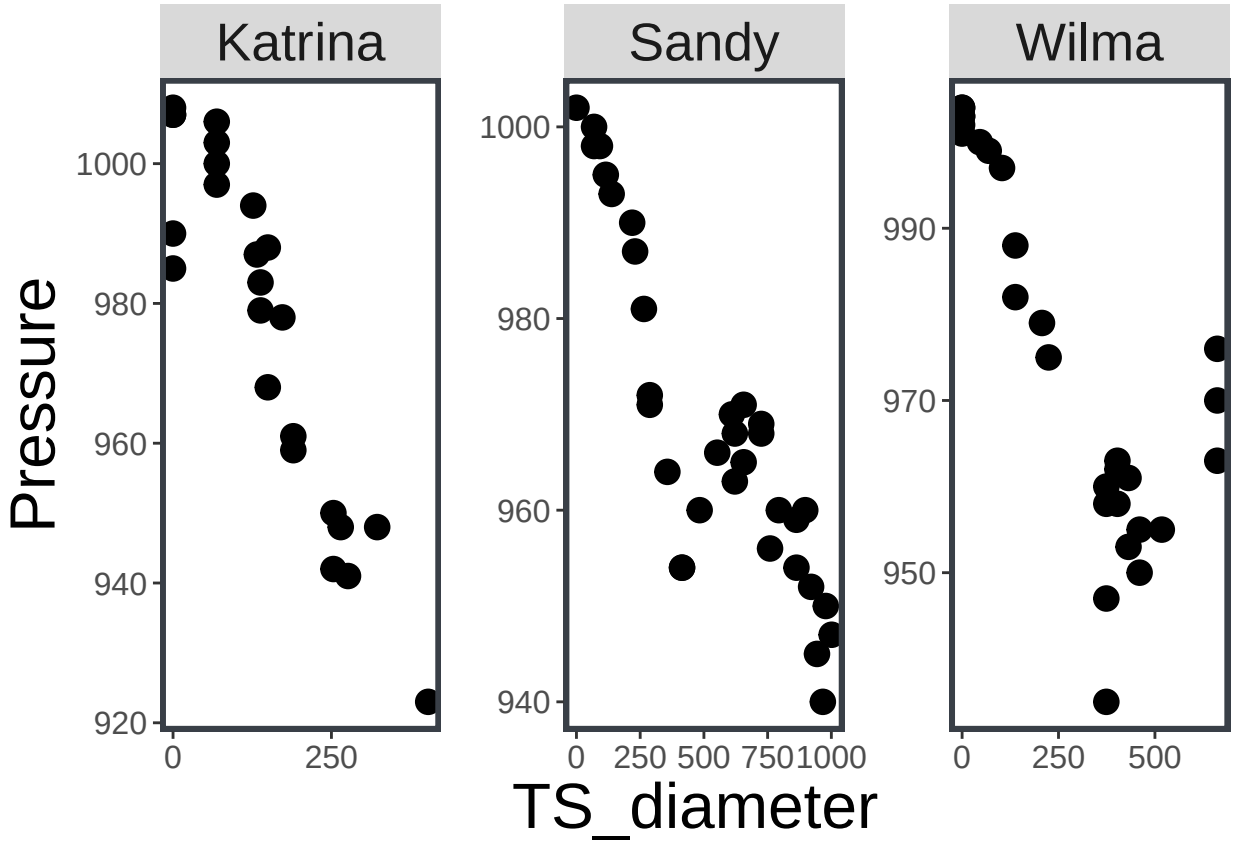
```
ggplot(data = sampleStorms, aes(x = TS_diameter, y = Pressure)) +
        geom_point(size = 4) +
        facet_wrap(~ Name) +
        ourTheme1
```



Step 3: Create separate scatterplots with different axis scales.

Use facet_wrap() to create panels of plots based on a single categorical variable. Here, adding the argument scales = "free" to the facet_wrap() function allows R to create different axes for each plot:

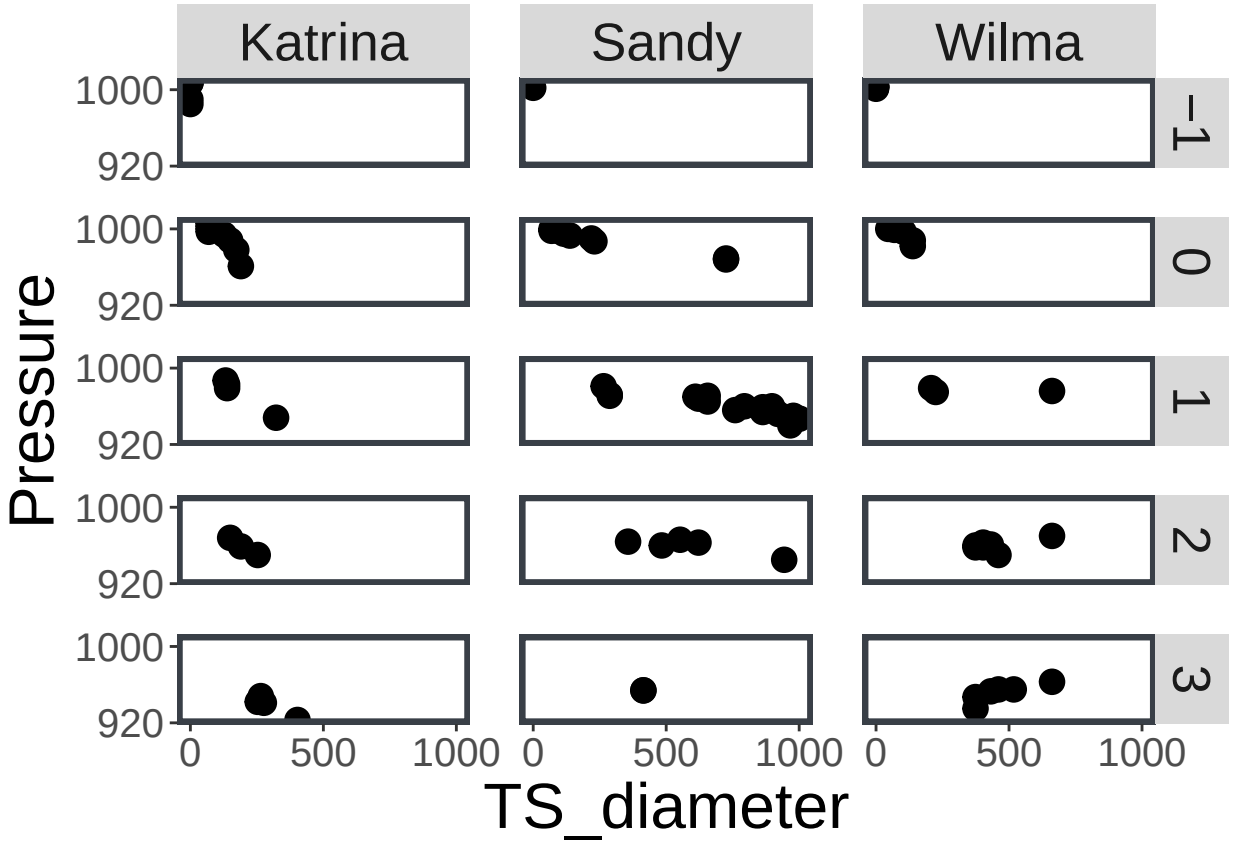
```
ggplot(data = sampleStorms, aes(x = TS_diameter, y = Pressure)) +
        geom_point(size = 4) +
        facet_wrap(~ Name, scales = "free") +
        ourTheme2
```



Step 4: Create separate scatterplots for each combination of two categorical variables.

Use facet_grid() to create a grid of plots. This function uses the syntax RowVariable ~ ColumnVariable to determine which variable should be on the rows and which variable should be on the columns of the grid:

```
ggplot(data = sampleStorms, aes(x = TS_diameter, y = Pressure)) +
        geom_point(size = 4) +
        facet_grid(Category ~ Name) +
        ourTheme3
```


Katrina
Sandy
Wilma
−1
0
1
2
3
Pressure
1000
920
0
500
1000
TS_diameter