Welcome to Tidyverse & More!

Presented by Megan Hirni

2024-10-10

Introduction

My name is Megan Hirni, and you can find my main work through GitHub (github.com/hirnia). I am a 3rd-ish year in Statistics + Education. You can find me through email (mj.hirni@missouri.edu), GitHub, and LinkedIn.

Figure 1: Me (Megan Hirni)

This Presentation Assumes You Know:

- How to access R
- Object Oriented Programming Language Foundations
- You have patience to troubleshoot and try new things

If you do not meet the prequisites, there are resources available on and off campus to support you where you are at! Feel free to ask for any direction. :)

You can follow along at: https://github.com/hirnia/for_you

Welcome to Tidyverse

Tidyverse (Wickham, 2023) is a set of packages that follows a similar coding style and output throughout. (https://www.tidyverse.org). Several packages—including `ggplot2`, `dplyr`, and `purrr` are housed under `tidyverse`, so installing `tidyverse` provides access to all housed core `tidyverse` packages.

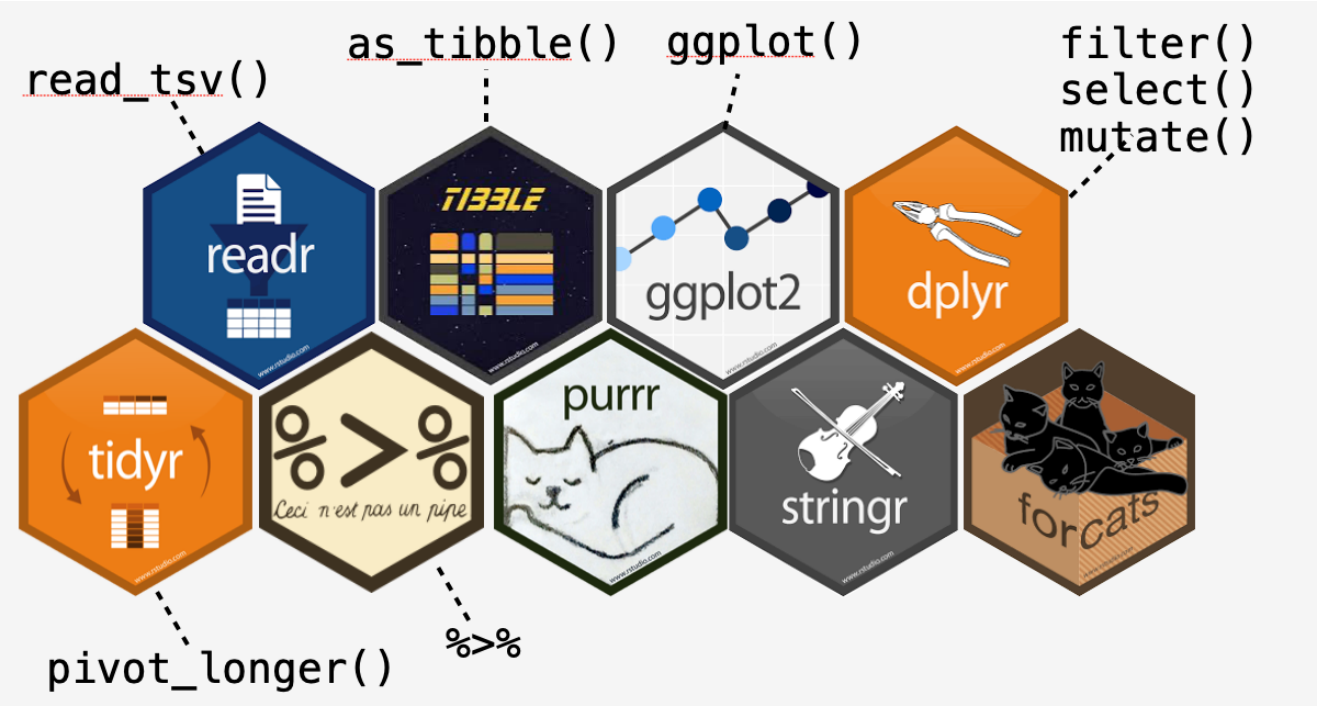


Figure 2: tidyverse core packages

A Bit About Tidyverse Syntax

Base R uses nested functions whereas Tidyverse uses `piped` functions.

```
x <- 1:5
mean(x) #base R
```

```
[1] 3
```

```
library(dplyr)
x %>% mean() #dplyr
```

```
[1] 3
```

Essentially, the pipe operator %>% takes the output of the function on the left and feeds it into the function on the right. This can make code more readable and easier to understand.

ggplot Intro

`ggplot2` will create beautiful visuals with a few lines of code. It is a powerful tool for data visualization. If you are willing to work through the learning curve, `ggplot2` can be a great tool for your data visualization needs and provide presentable visuals for reports or presentations.

`ggplot2` is within the tidyverse base packages, so loading `library(tideverse)` will also load `ggplot2.` `ggplot2` stacks on layers to create visuals, so you can add "layers" (using the + operator) to your plot to customize it to your needs. Some layers must be in a certain order while others can be in any order.

Airquality Example

The `airquality` dataset is a built-in `R` dataset that containing daily air quality measurements in New York City May to September 1973 (Chambers et al., 1983) with variables: observation was made (a value between 5 and 9).

`Ozone`: The ozone variable represents the maximum daily ozone concentration in parts per billion (ppb) measured at a monitoring station in New York.

`Solar - R`: The solar.R variable represents the daily solar radiation in Langleys (a measure of solar energy) measured at the same monitoring station.

`Wind`: The wind variable represents the average daily wind speed in miles per hour (mph).

`Temp`: The temp variable represents the average daily temperature in degrees Fahrenheit.

`Month`: The month variable represents the month in which the observation was made (a value between 5 and 9).

Airquality Table

Using `knitr` (Yihui, 2024) and other R add-on packages, you can create automated tables that are easy to read and understand. The `kable` function in the `knitr` package is used to create tables from data frames in R. For other good table generating packages, try the packages `gt` and `flextable`.

```
#library(knitr) #requires LaTeX
kable(head(airquality))
```

Ozone	Solar.R	Wind	Temp	Month	Day
41	190	7.4	67	5	1
36	118	8.0	72	5	2
12	149	12.6	74	5	3
18	313	11.5	62	5	4
NA	NA	14.3	56	5	5
28	NA	14.9	66	5	6

Figure 3: Airquality Table

Basic ggplot

Using `airquality`, We plot a histogram of the `Wind` variable.

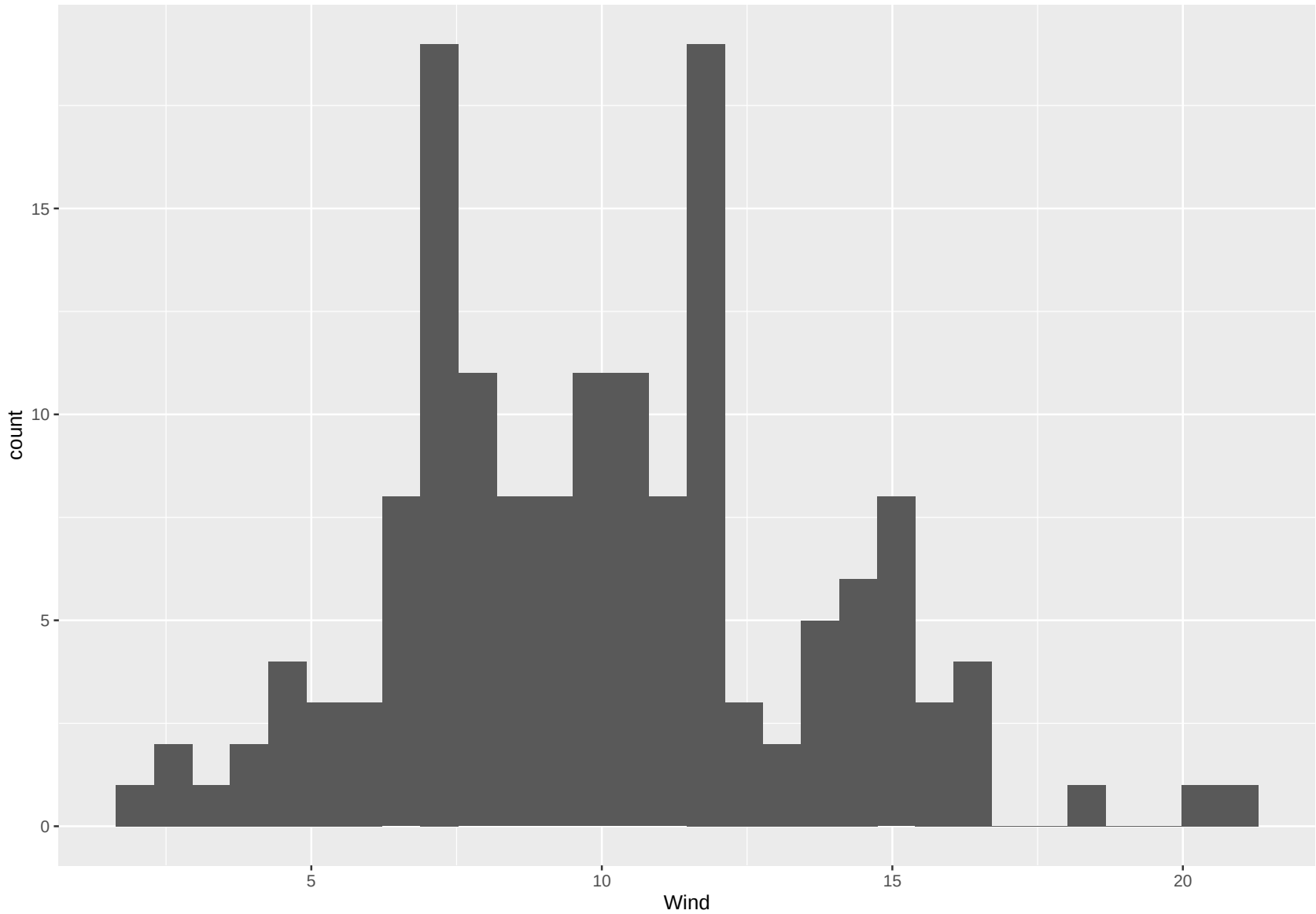
```
library(ggplot2)
basic_hist <- ggplot(data = airquality) +
  geom_histogram(mapping = aes(x = Wind))
```

This plot includes a `geom_histogram` layer that creates the histogram and a `mapping` argument that tells `ggplot2` what variable to use for the x-axis. We save the `ggplot` object into the name `basic_hist` to call later (not a required step, but handy later on. We see the + syntax in action!

The aesthetics provided by ggplot are more presentation ready than base R plots and more easily customizable.

Our Initial ggplot Visual

```
basic_hist
```



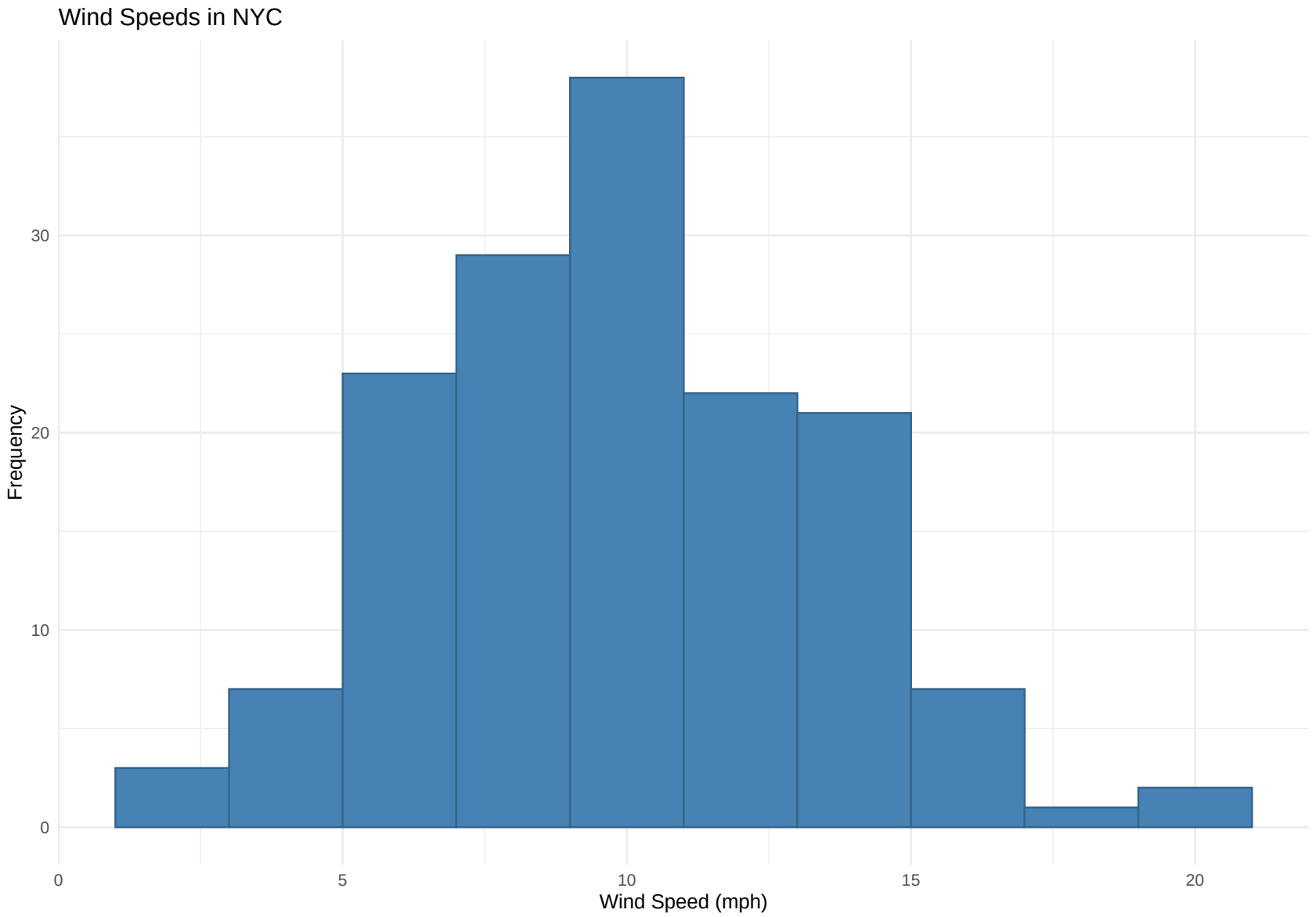
Adding onto ggplot visuals

Here, we will add color, labels, and a theme to the plot to make it more presentable.

```
wind_hist <- ggplot(data = airquality) +
  #adding colors plus how wide bins are within hist
  geom_histogram(mapping = aes(x = Wind),
                 fill = "steelblue",
                 color = "steelblue4",
                 binwidth = 2) +
  #adding title labels
  labs(title = "Wind Speeds in NYC",
       x = "Wind Speed (mph)", y = "Frequency") +
  #adding theme: theme_minimal removes the grey cast
  theme_minimal()
```

Wind Histogram

```
wind_hist
```



A Fun Example, Ridgeline Plots

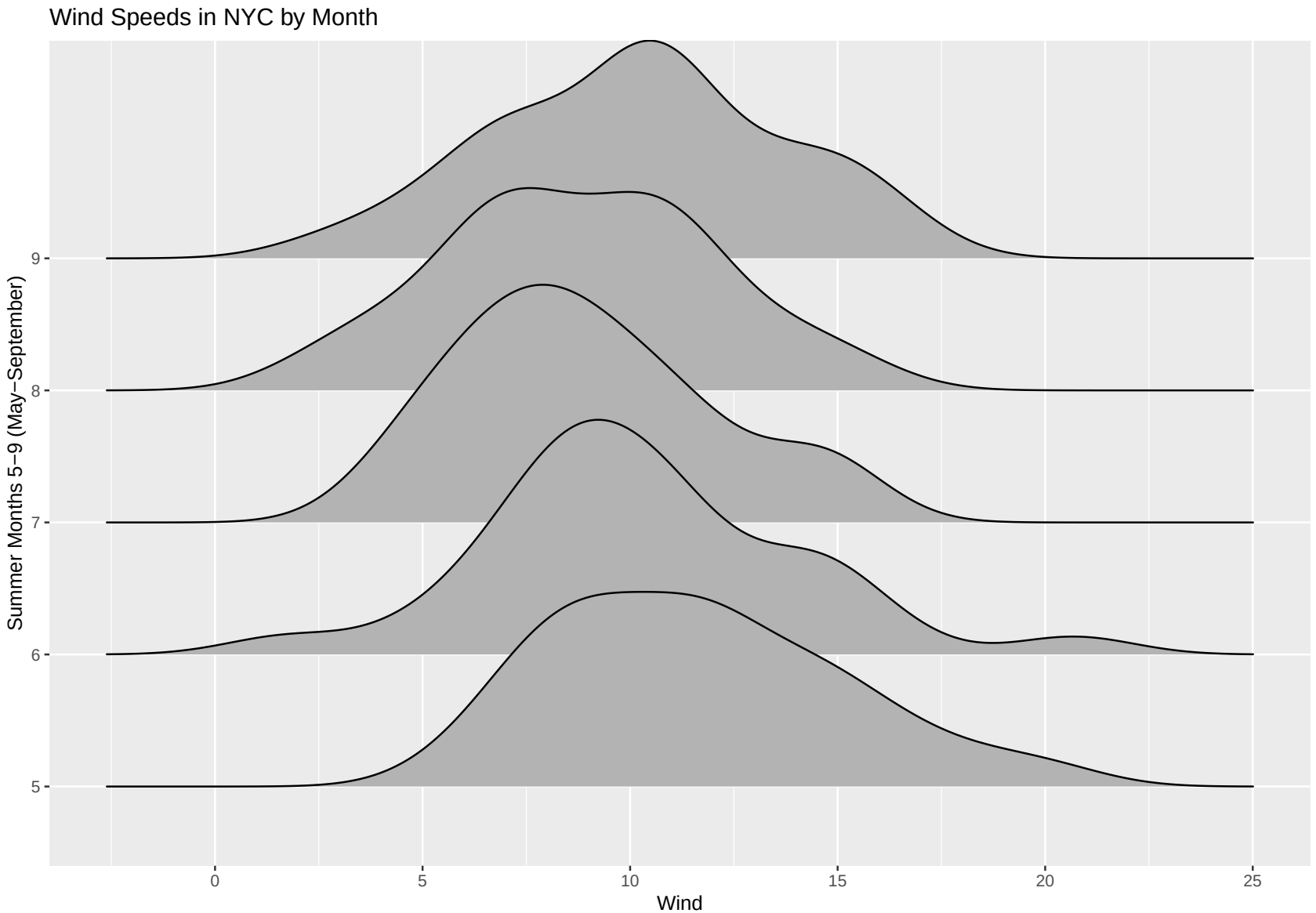
There's extensions to ggplot2 that can create unique visuals. Here, we use the ggridges package to create a ridgeline plot of the Wind variable in the airquality dataset. This plot shows the distribution of wind speeds in New York City over the summer of 1973 (May (5) to September (9)).

```
#install.packages("ggridges")
library(ggridges); library(viridis)

wind_ridgeline <- ggplot(data = airquality,
                         aes(x = Wind,
                         y = as.factor(Month))) +
  geom_density_ridges() +
  theme(legend.position = "none") +
  ylab("Summer Months 5-9 (May-September)") +
  ggtitle("Wind Speeds in NYC by Month")
```

Wind Ridgeline

```
wind_ridgeline
```



ggplot Add-Ons

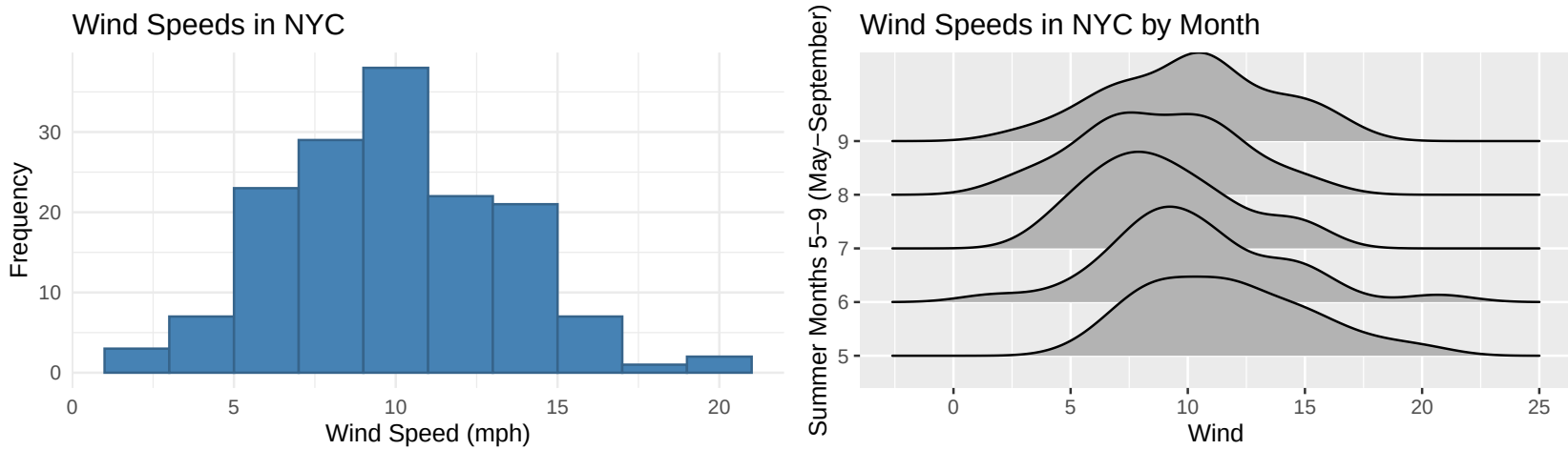
Figure 4: ggplot2 logo

As you saw with the ridgeline plot, there are many add-ons to ggplot2 that can create unique visuals. Some popular add-ons include gganimate for creating animated plots, ggrepel for adding labels to plots, and ggthemes for adding themes to plots. The site https://r-graph-gallery.com/ is a great resource for learning how to create different types of visuals using ggplot2.

ggplot Add-On: patchwork

One cool add on is `patchwork`, (Pederson, 2020) which allows you to combine multiple plots into one plot. Below is an example of how to use `patchwork` to combine the histogram and ridgeline plots we created earlier.

```
#install.packages("patchwork")
library(patchwork)
wind_hist + wind_ridgeline
```



The + operator is used to combine the two plots into one plot. You can also use the / operator to combine the plots vertically.

Back to tidyverse:

dplyr The ggplot capabilities are just small components of the tidyverse universe. The dplyr package is another powerful tool within the tidyverse that allows you to manipulate and summarize data. Below is an example of how to use dplyr to filter the airquality dataset to only include observations where the Wind speed is greater than 10 mph.

```
library(dplyr)
nrow(airquality) #153 rows
```

```
[1] 153
```

```
airquality_filtered <- airquality %>%
  filter(Wind > 10)
nrow(airquality_filtered) # 72 rows
```

```
[1] 72
```

Cleaning data with dplyr

Within our airquality dataset, we have missing values in the Ozone and Solar.R variable. We can use dplyr to filter out these missing values and create a new dataset called airquality_clean.

```
airquality_clean <- airquality %>%
  filter(!is.na(Ozone) & !is.na(Solar.R))
```

We "pipe" the airquality dataset into the filter function and use the !is.na() function to filter out rows where the Ozone or Solar.R variable is missing. The resulting dataset airquality_clean will only contain rows where both Ozone and Solar.R are not missing (filter(!is.na(Ozone) & !is.na(Solar.R))).

More on Beautification

In addition to all the work you can do in packages such as `ggplot2` and `dplyr` to beautify data, there are ways to beautify your reporting. `Quarto` and `rmarkdown` are packages in R that allows you to create beautiful reports/presentations in `R` that can be exported to `HTML`, `PDF`, and `Word` formats (uses LaTeX for compilation). You can both run and compile this presentation yourself using the `Welcome_to_tidy.qmd` file from GitHub. Check out sites such as https://quarto.org/ and https://rmarkdown.rstudio.com/ for more information on how to create beautiful reports in `R`.

Thank You!

Figure 5: GitHub Logo

You can find all content from today on GitHub:

https://github.com/hirnia/for_you.

I welcome any questions or feedback! :)