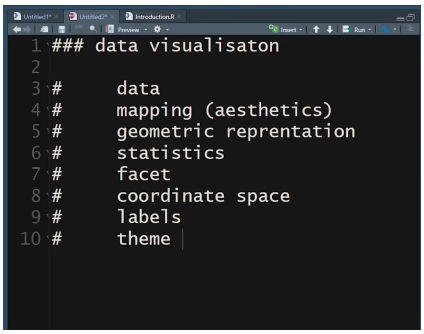
Grammar of Graphics

Ananda Biswas

When we talk about **Grammar of Graphics**, we talk about defining some parameters of a given data visualization.




```
# install.packages('tidyverse')

library(tidyverse)
```

```
## Warning:  package 'tidyverse' was built under R version 4.2.3
## Warning:  package 'ggplot2' was built under R version 4.2.2
## Warning:  package 'tibble' was built under R version 4.2.3
## Warning:  package 'tidyr' was built under R version 4.2.3
## Warning:  package 'readr' was built under R version 4.2.2
## Warning:  package 'purrr' was built under R version 4.2.3
## Warning:  package 'dplyr' was built under R version 4.2.3
## Warning:  package 'stringr' was built under R version 4.2.3
## Warning:  package 'forcats' was built under R version 4.2.2
## Warning:  package 'lubridate' was built under R version 4.2.2
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0
--
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts()
--
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

The function **data()** gives list of all data sets that are in-built in R.

```
data()
```

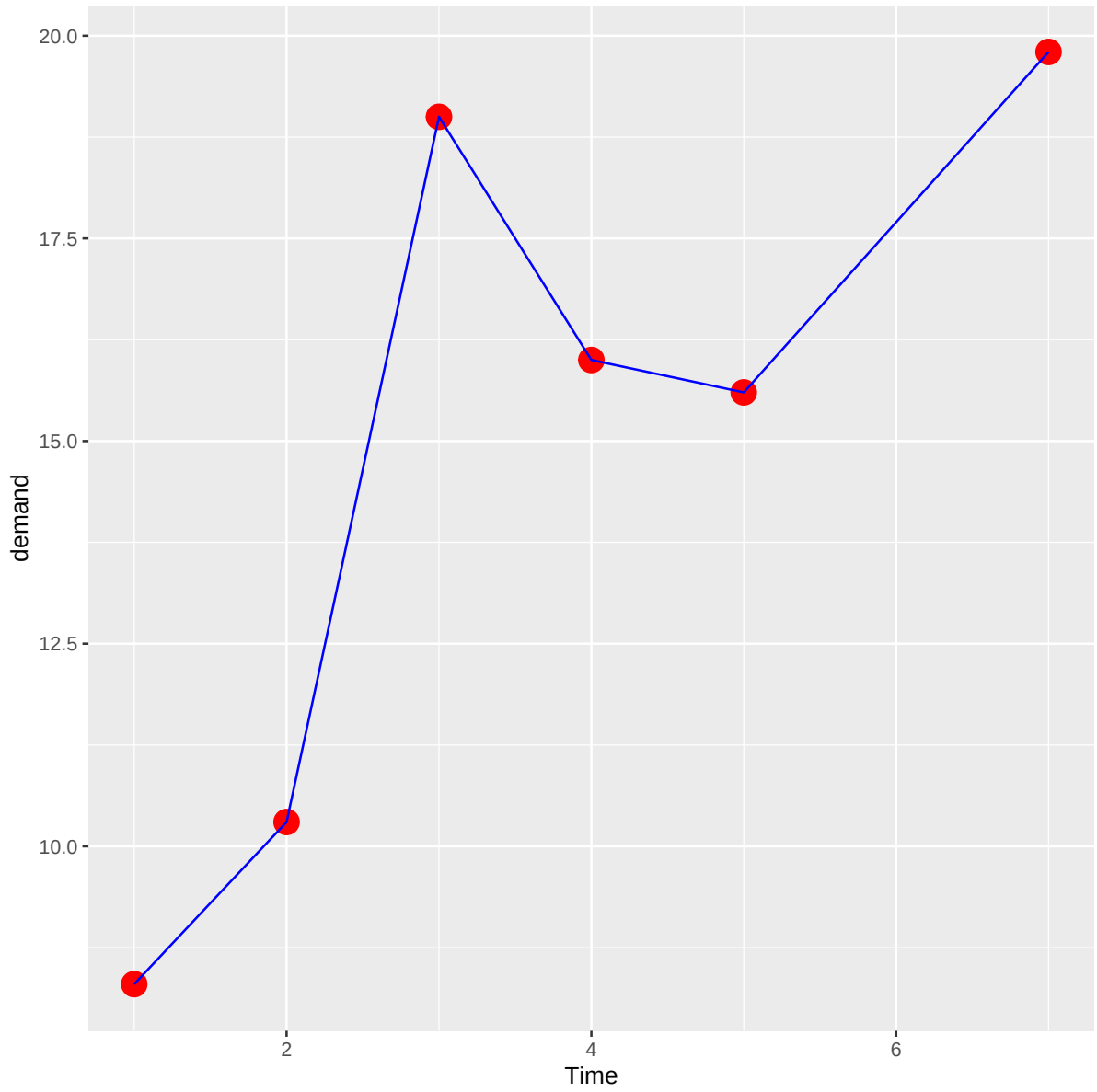
We shall use **BOD** data set.

```
# ?BOD
```

```
BOD
```

```
##   Time demand
## 1    1    8.3
## 2    2   10.3
## 3    3   19.0
## 4    4   16.0
## 5    5   15.6
## 6    7   19.8
```

```
ggplot(data = BOD, mapping = aes(x = Time, y = demand)) +
    geom_point(size = 5, colour = "red") + geom_line(color = "blue")
```



Now we shall use **CO2** data set.

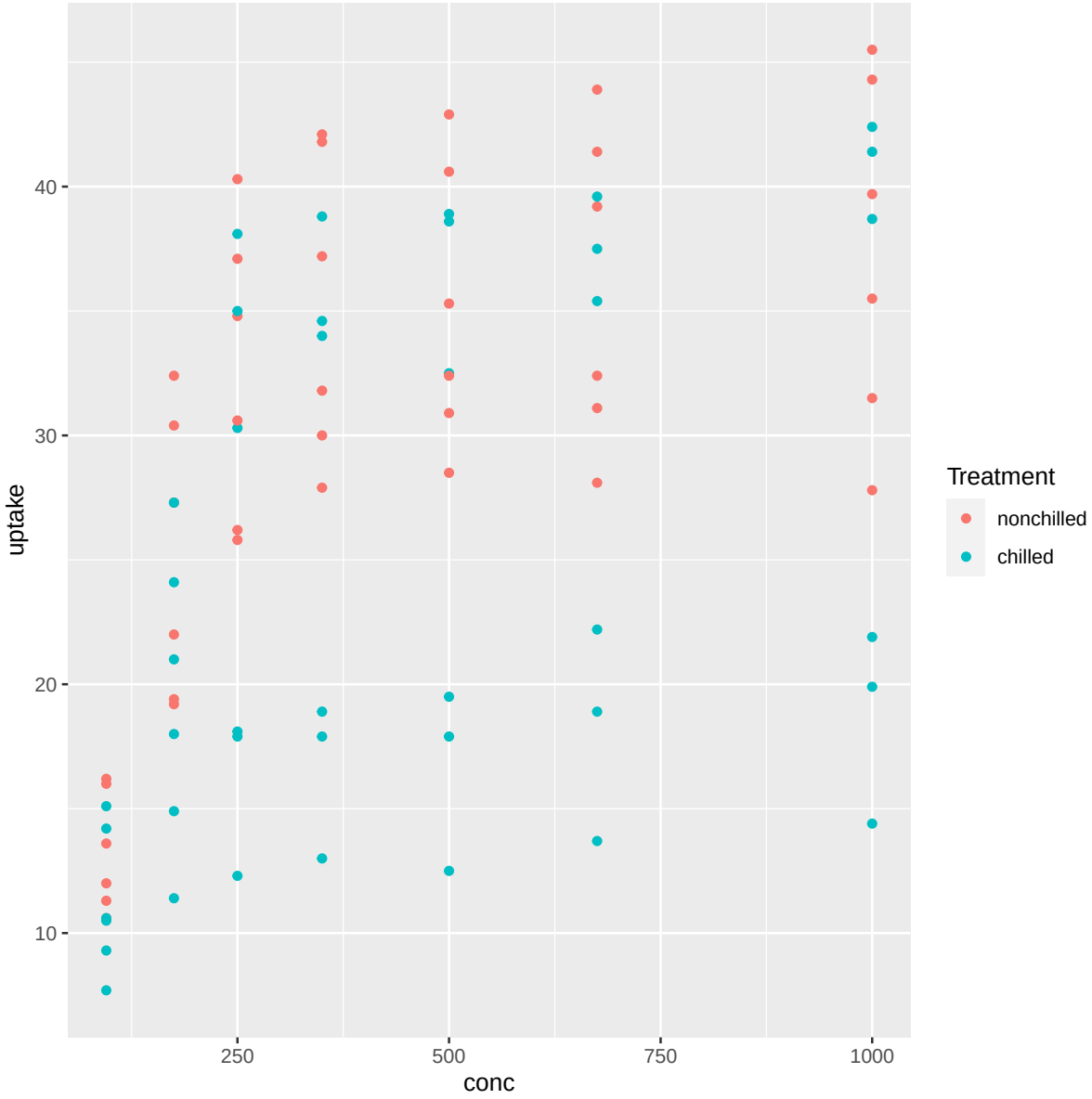
```
view(CO2)
```

```
# ?CO2
```

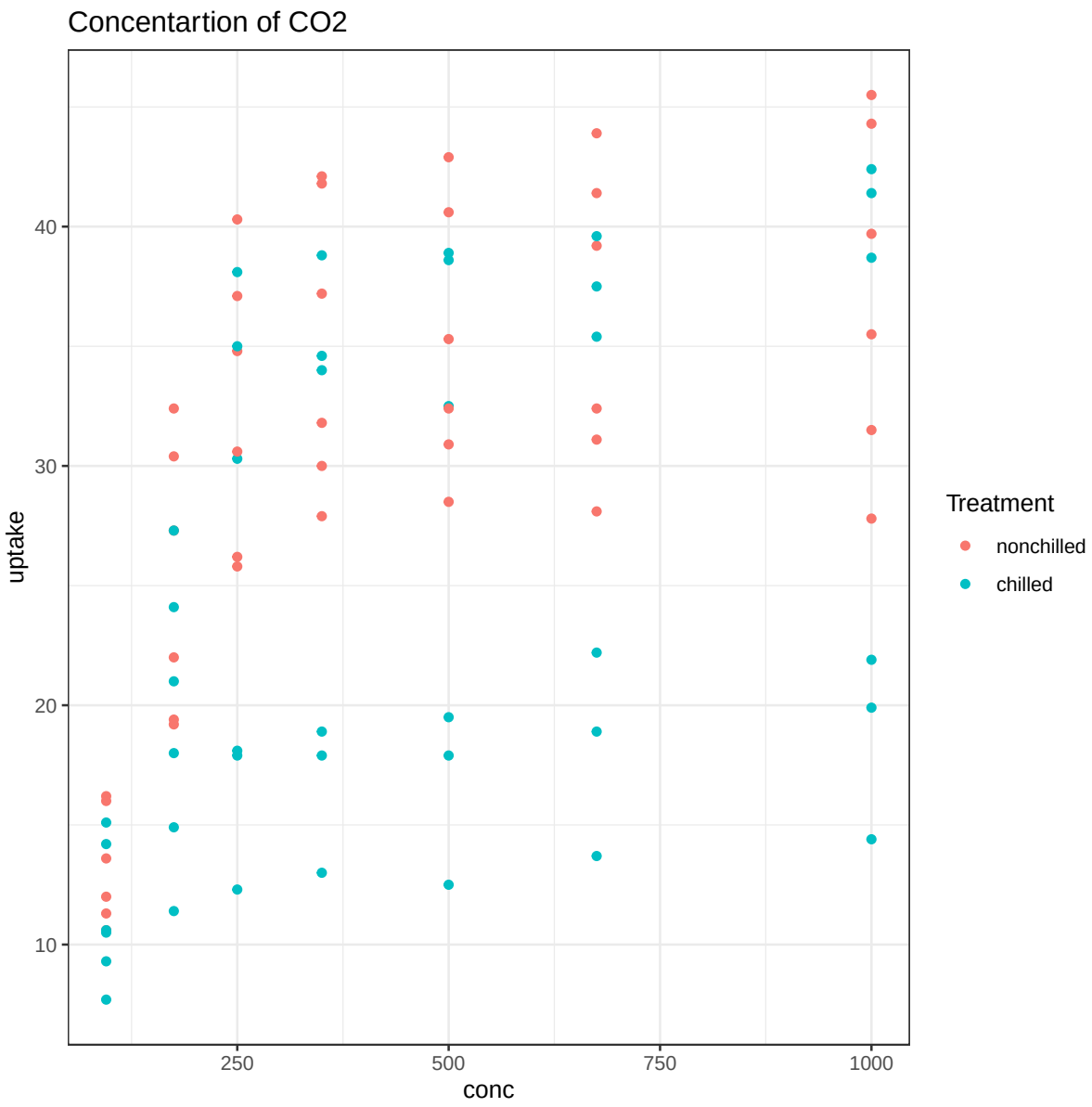
```
names(CO2)

## [1] "Plant"     "Type"      "Treatment" "conc"      "uptake"
```

%>% is an operator in this context; often called pipe operator. The operand to its left is piped into as the first argument of the function ggplot(), which is to its right.

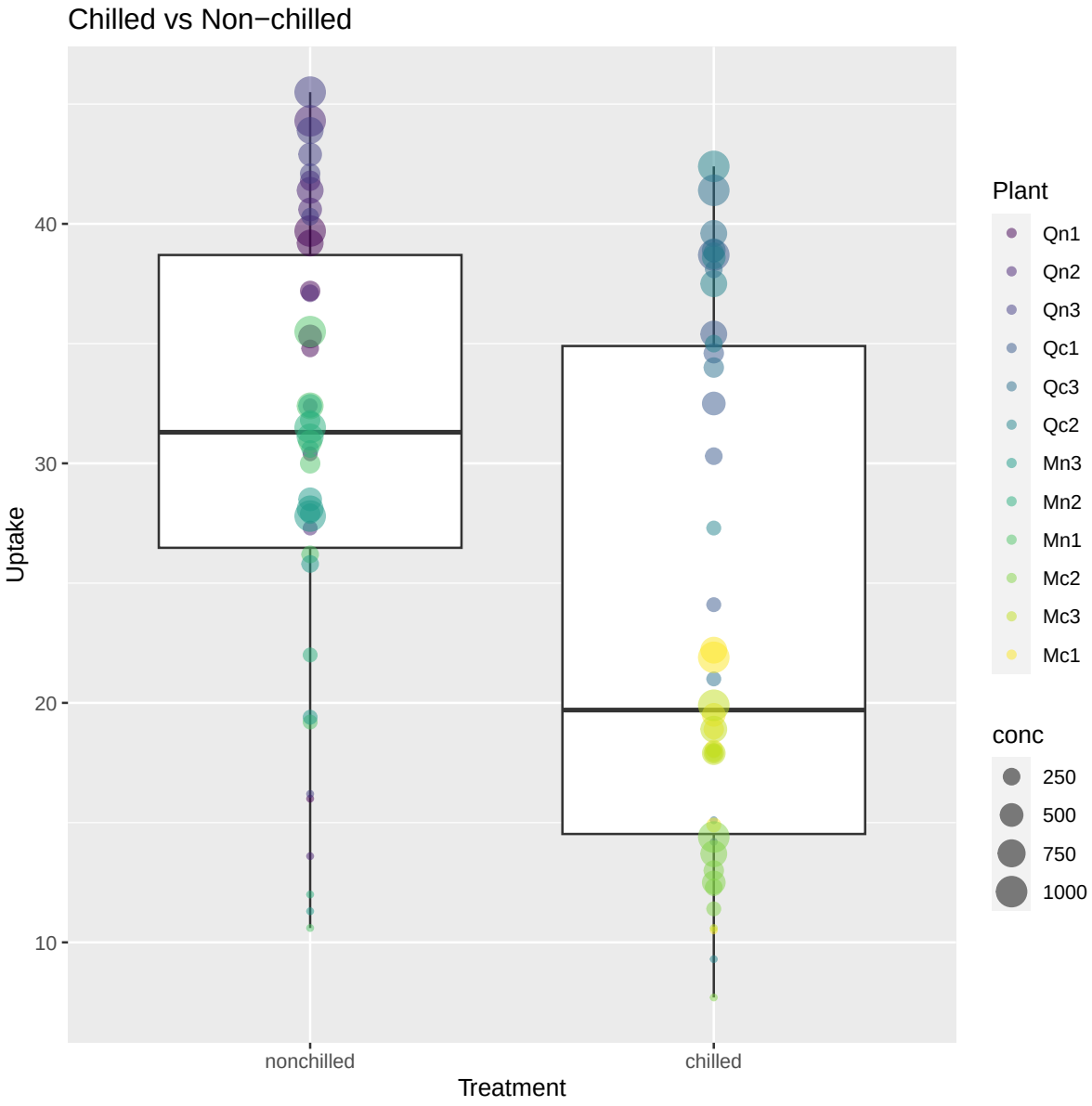
```
CO2 %>%
    ggplot(aes(x = conc, y = uptake, colour = Treatment)) +
    geom_point()
```



```r
CO2 %>%
    ggplot(aes(x = conc, y = uptake, color = Treatment)) +
    geom_point() + labs(title = "Concentartion of CO2") +
    theme_bw()
```



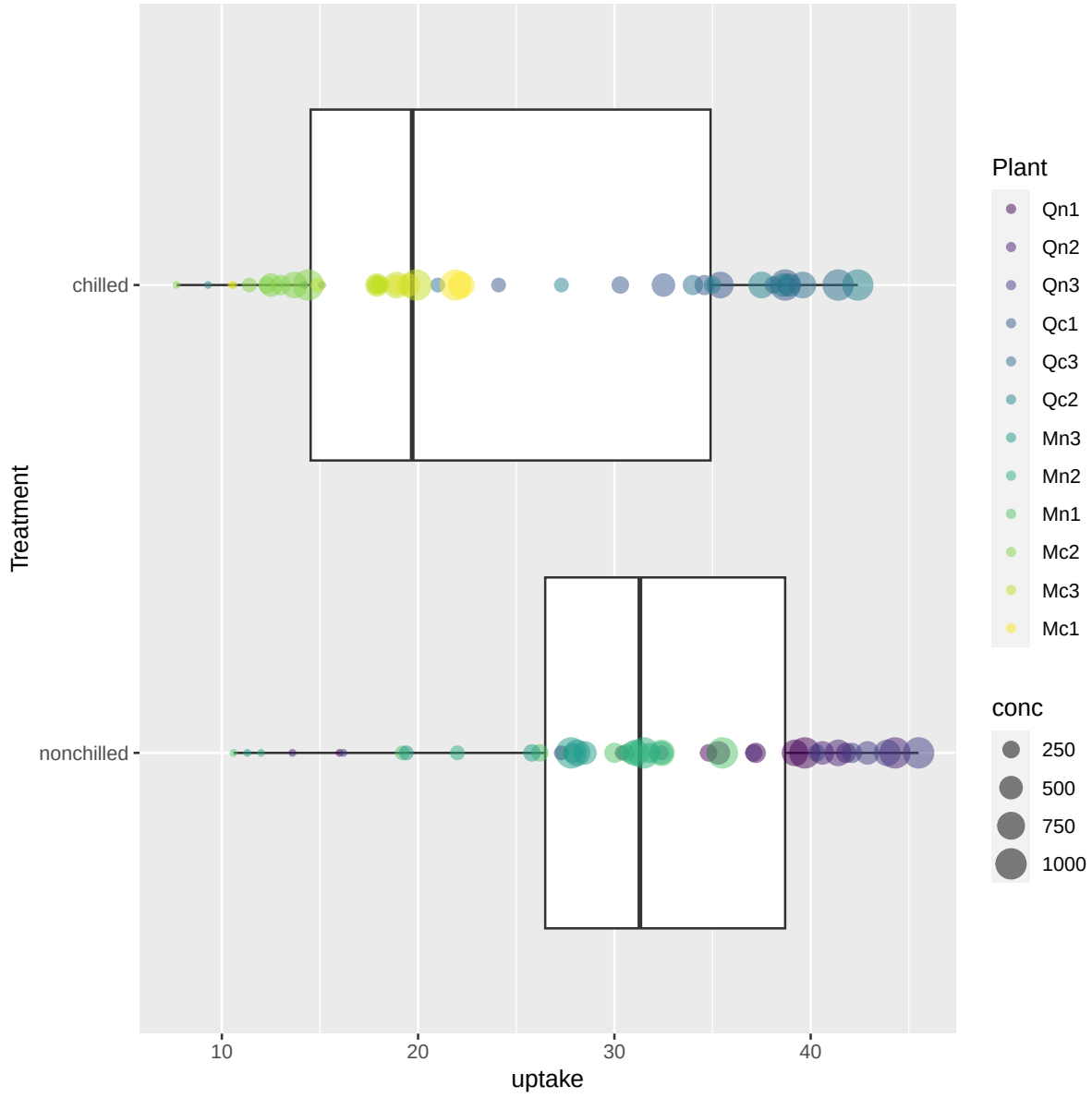
```
CO2 %>%
    ggplot(aes(x = Treatment, y = uptake)) + geom_boxplot() +
    geom_point(aes(size = conc, color = Plant), alpha = 0.5) +
    labs(title = "Chilled vs Non-chilled") + xlab("Treatment") +
    ylab("Uptake")
```



The argument *alpha* stands for opacity.

The aesthetics of *geom_point()* will not influence others. But the aesthetics of *ggplot()* will influence all others.

```
CO2 %>%
    ggplot(aes(x = Treatment, y = uptake)) + geom_boxplot() +
    geom_point(aes(size = conc, color = Plant), alpha = 0.5) +
    coord_flip()
```



Now we shall use **msleep** data set.

```
# ?msleep
```

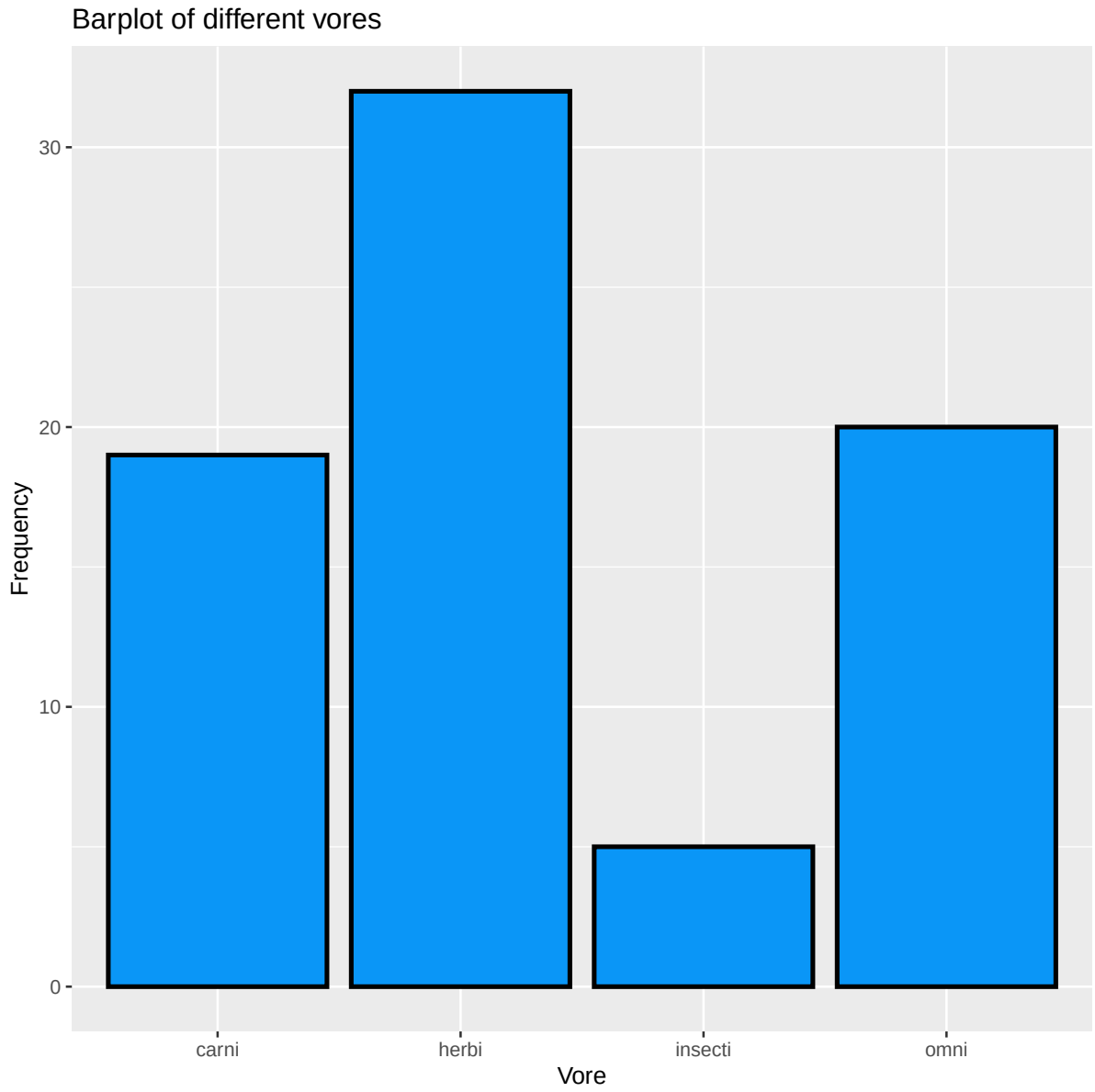
```
names(msleep)
```

```
##  [1] "name"         "genus"        "vore"         "order"        "conservation"
##  [6] "sleep_total"  "sleep_rem"    "sleep_cycle"  "awake"        "brainwt"
## [11] "bodywt"
```

```
view(msleep)
```

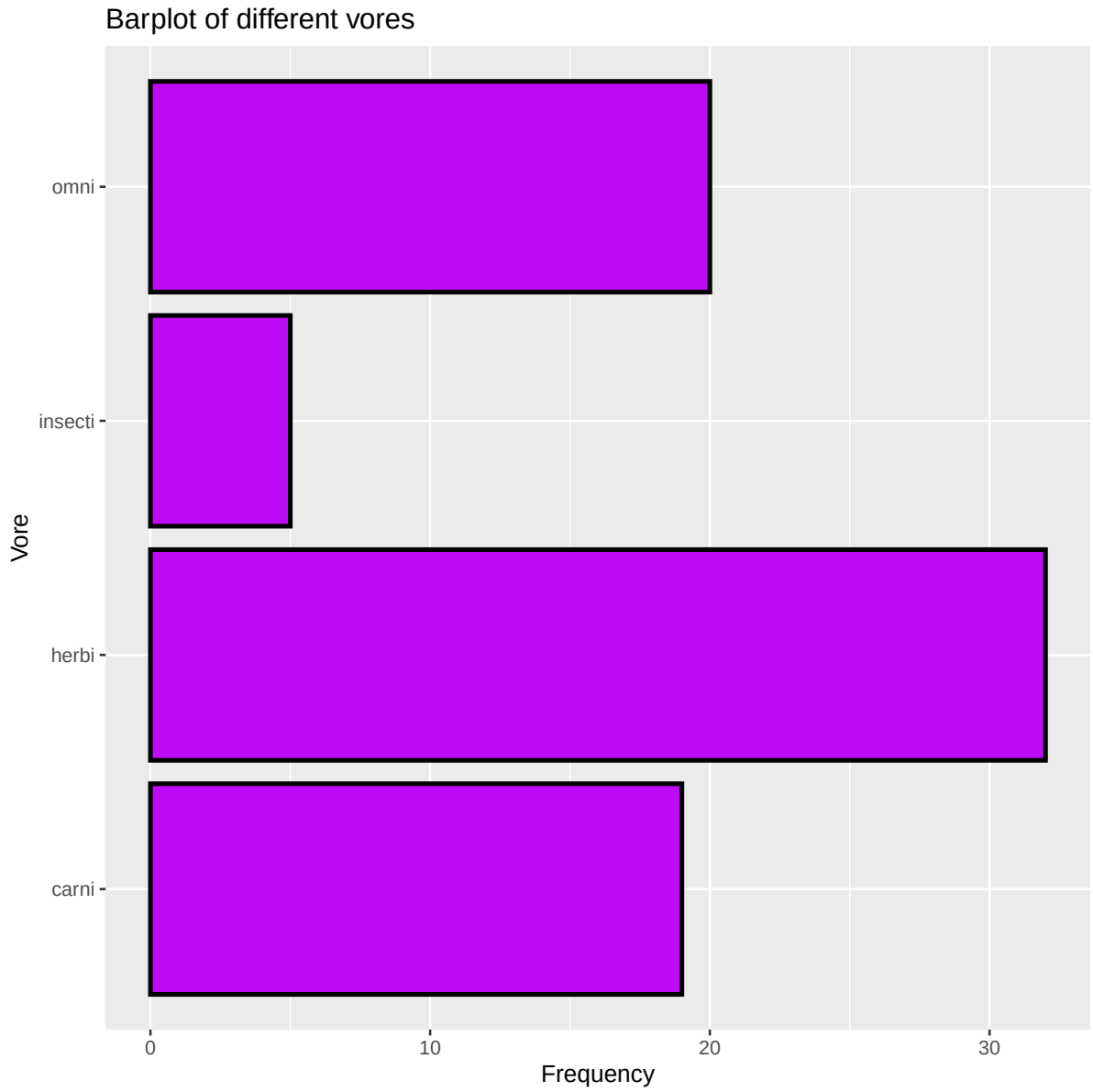
Barplot

```
msleep %>%
    drop_na(vore) %>%
    ggplot(aes(x = vore)) + geom_bar(fill = "#0A96F7",
    color = "black", linewidth = 1) + labs(x = "Vore",
    y = "Frequency", title = "Barplot of different vores")
```

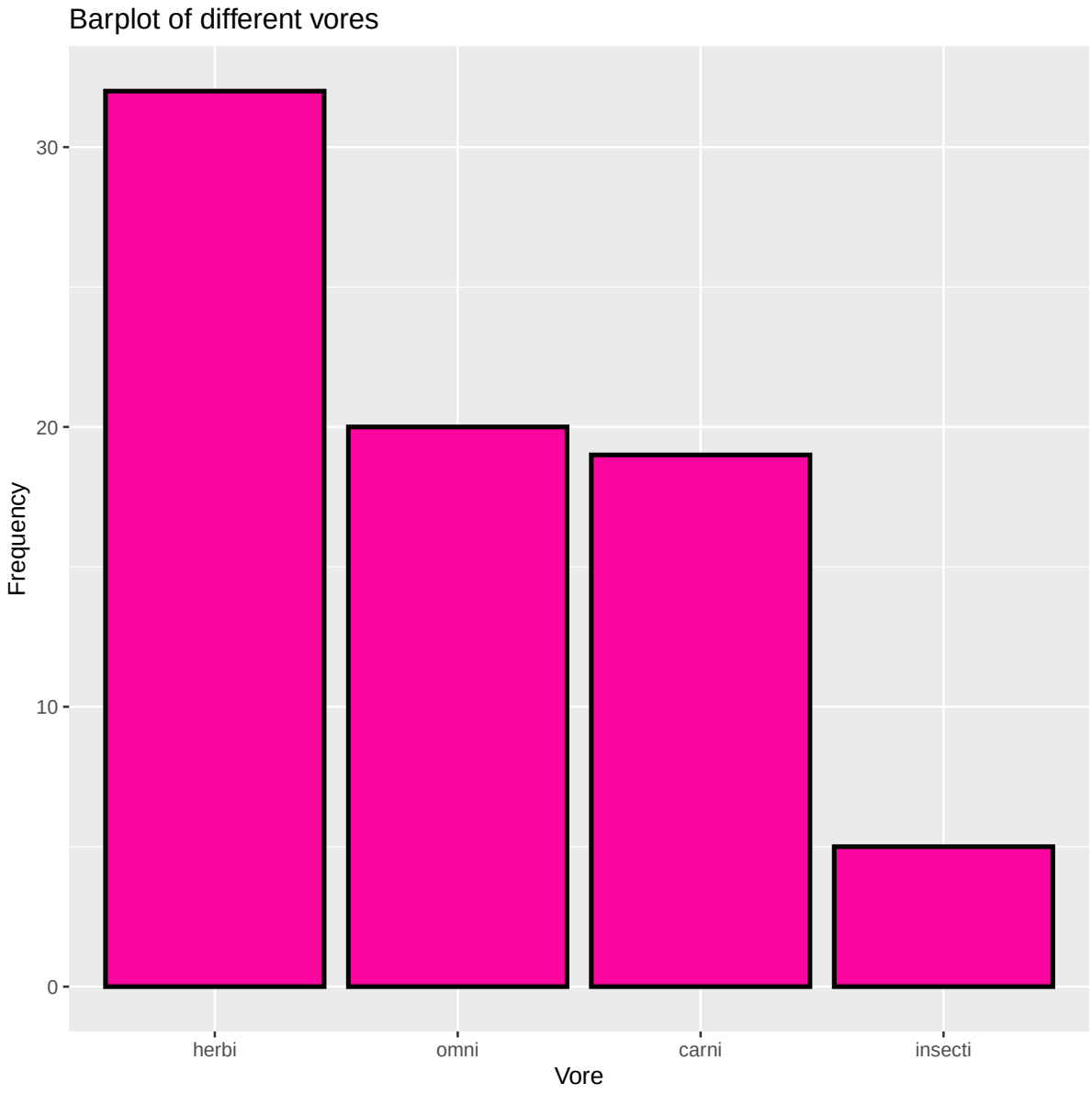


drop_na(vore) drops the *NA* values from the variable *vore* and pipes the filtered data set to ggplot.

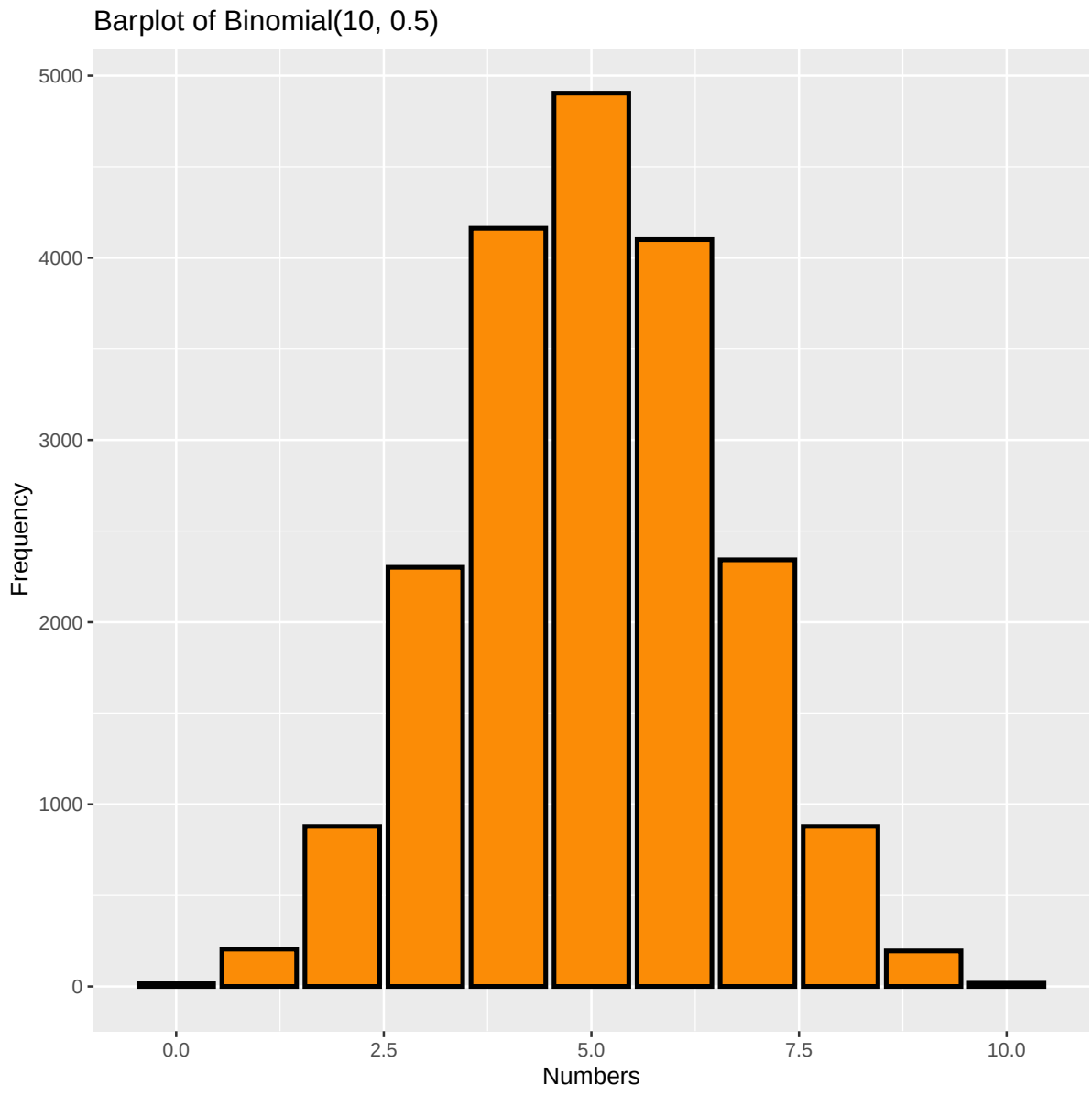
```r
msleep %>%
    drop_na(vore) %>%
    ggplot(aes(x = vore)) + geom_bar(fill = "#BE0AF7",
    color = "black", linewidth = 1) + coord_flip() +
    labs(x = "Vore", y = "Frequency", title = "Barplot of different vores")
```



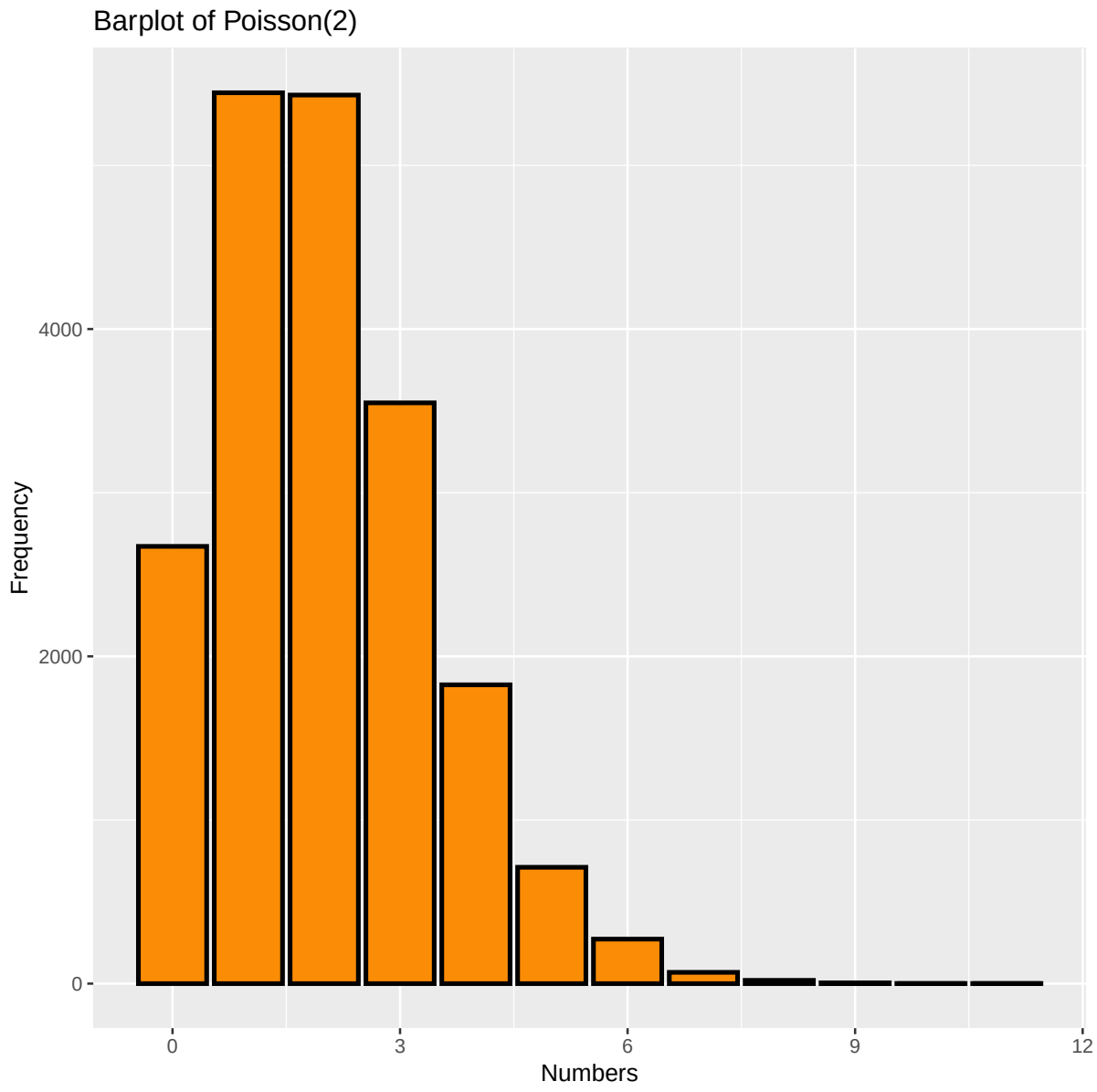
```r
msleep %>%
    drop_na(vore) %>%
    ggplot(aes(x = fct_infreq(vore))) + geom_bar(fill = "#FB069E",
    color = "black", linewidth = 1) + labs(x = "Vore",
    y = "Frequency", title = "Barplot of different vores")
```



```r
random_binomial_numbers <- rbinom(n = 20000, size = 10,
    prob = 0.5)

random_binomial_numbers <- data.frame(random_binomial_numbers)

ggplot(random_binomial_numbers, aes(x = random_binomial_numbers)) +
    geom_bar(fill = "#FB8C06", color = "black", linewidth = 1) +
    labs(x = "Numbers", y = "Frequency", title = "Barplot of Binomial(10, 0.5)")
```

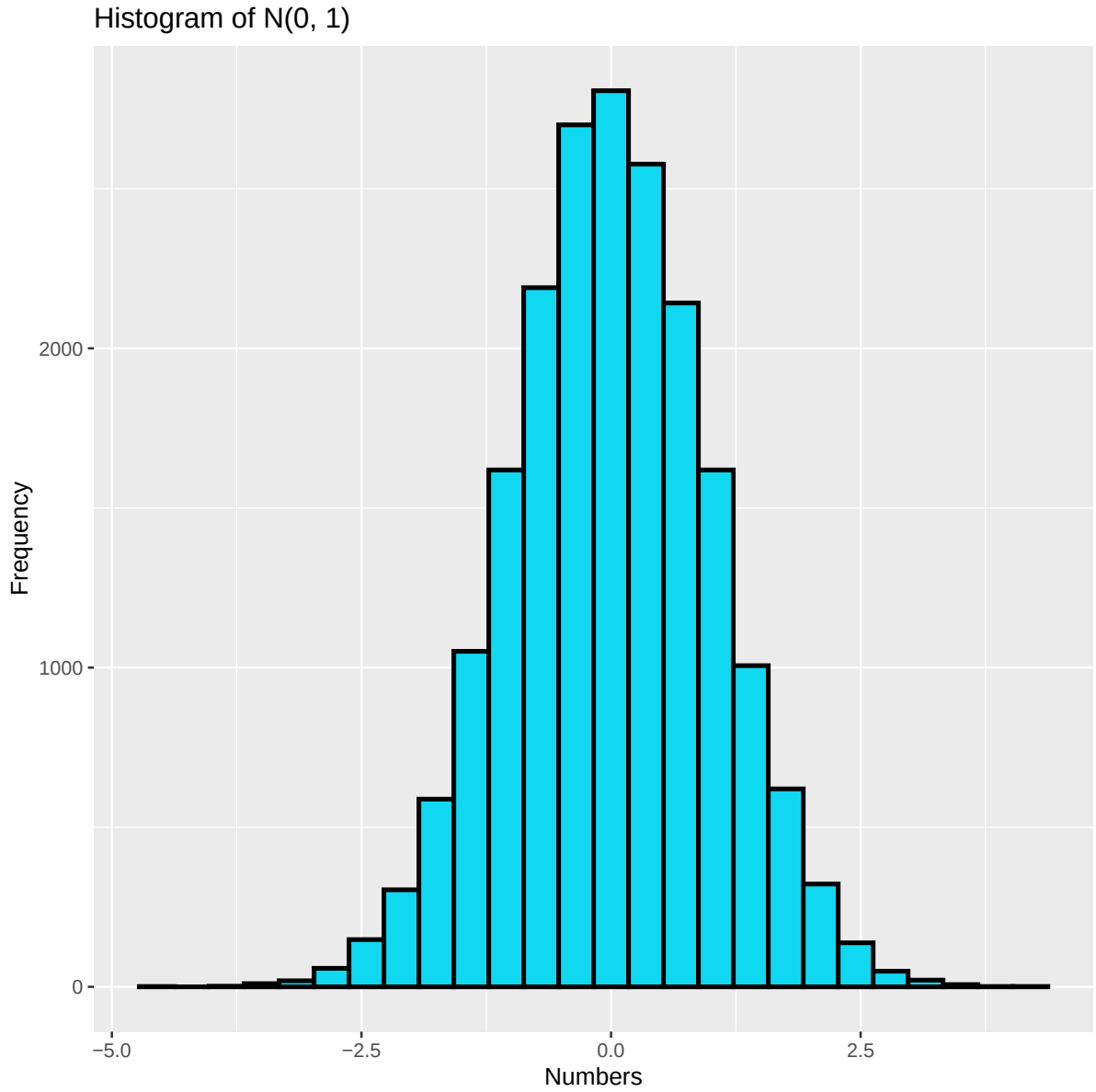


```r
random_poisson_numbers <- rpois(n = 20000, lambda = 2)

random_poisson_numbers <- data.frame(random_poisson_numbers)

ggplot(random_poisson_numbers, aes(x = random_poisson_numbers)) +
    geom_bar(fill = "#FB8C06", color = "black", linewidth = 1) +
    labs(x = "Numbers", y = "Frequency", title = "Barplot of Poisson(2)")
```



Histogram

```
random_normal_numbers <- rnorm(n = 20000, mean = 0,
    sd = 1)

random_normal_numbers <- data.frame(random_normal_numbers)

random_normal_numbers %>%
    ggplot(aes(x = random_normal_numbers)) + geom_histogram(binwidth = 0.35,
    fill = "#0FD8F0", color = "black", linewidth = 1) +
    labs(x = "Numbers", y = "Frequency", title = "Histogram of N(0, 1)")
```



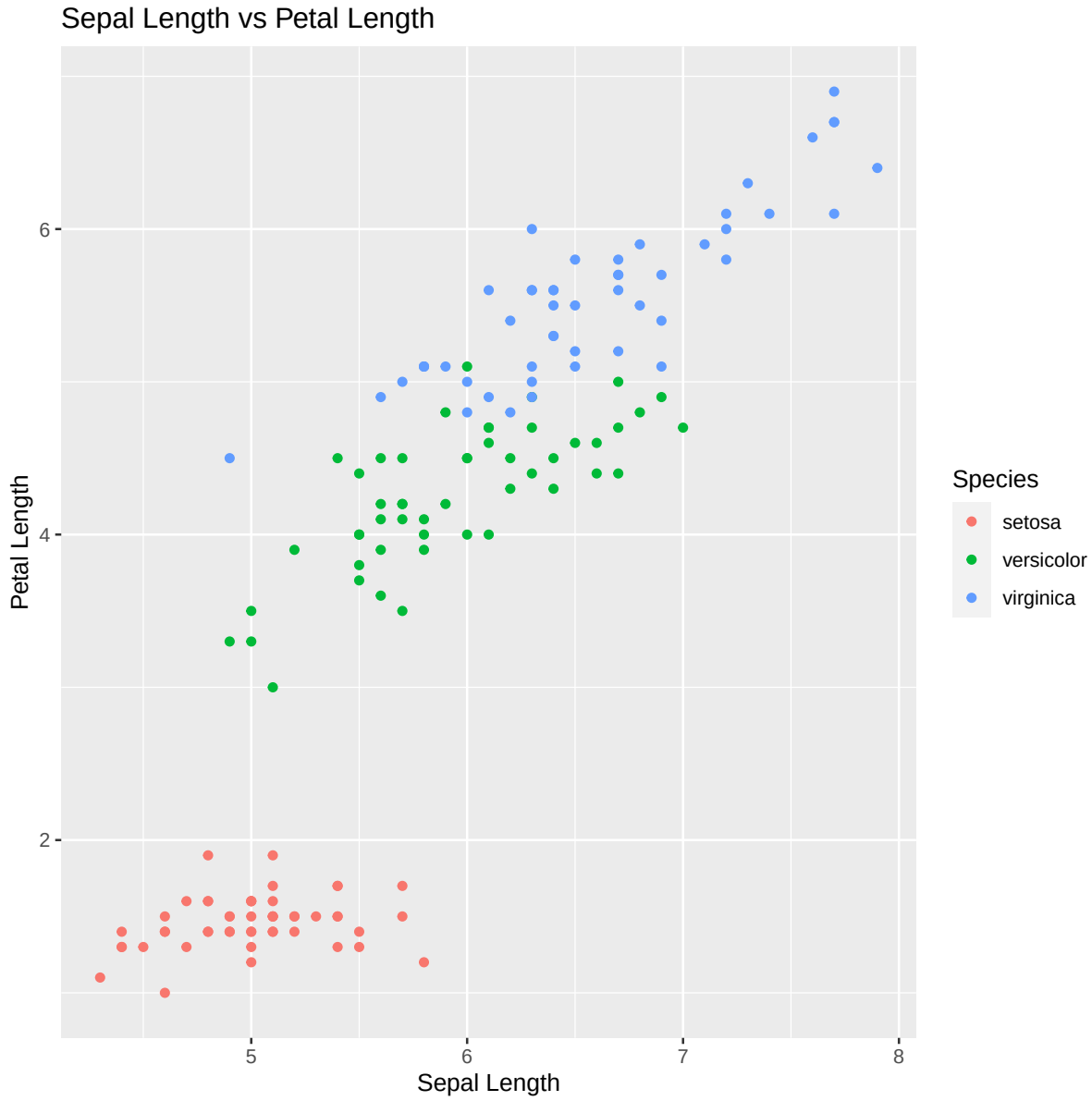
• Observe that, the parameter **binwidth** controls the number of bins in the histogram. The bigger the binwidth, the less the number of bins and vice versa.

Now we shall use **iris** data set.

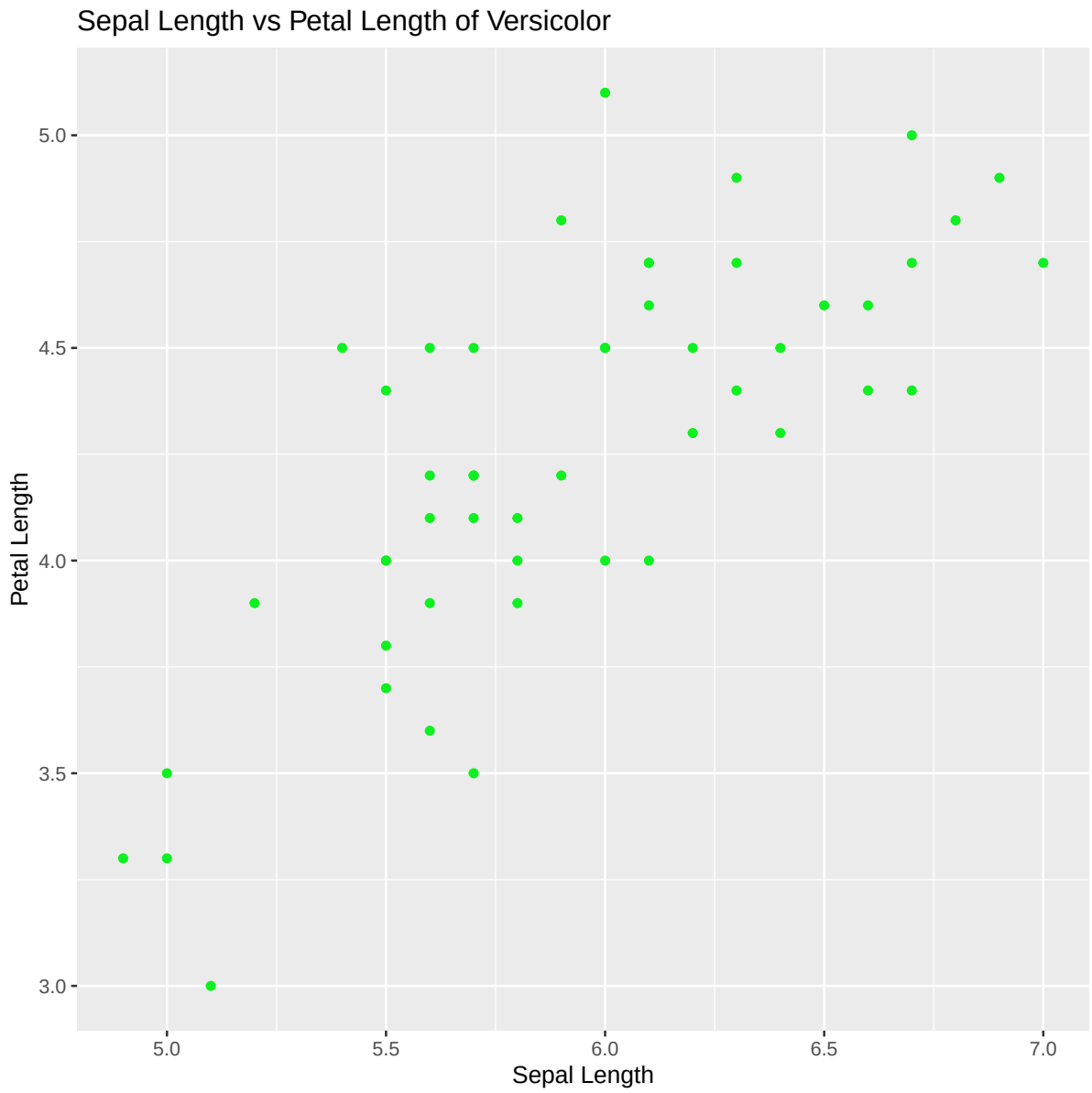
```
View(iris)
```

```
names(iris)

## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

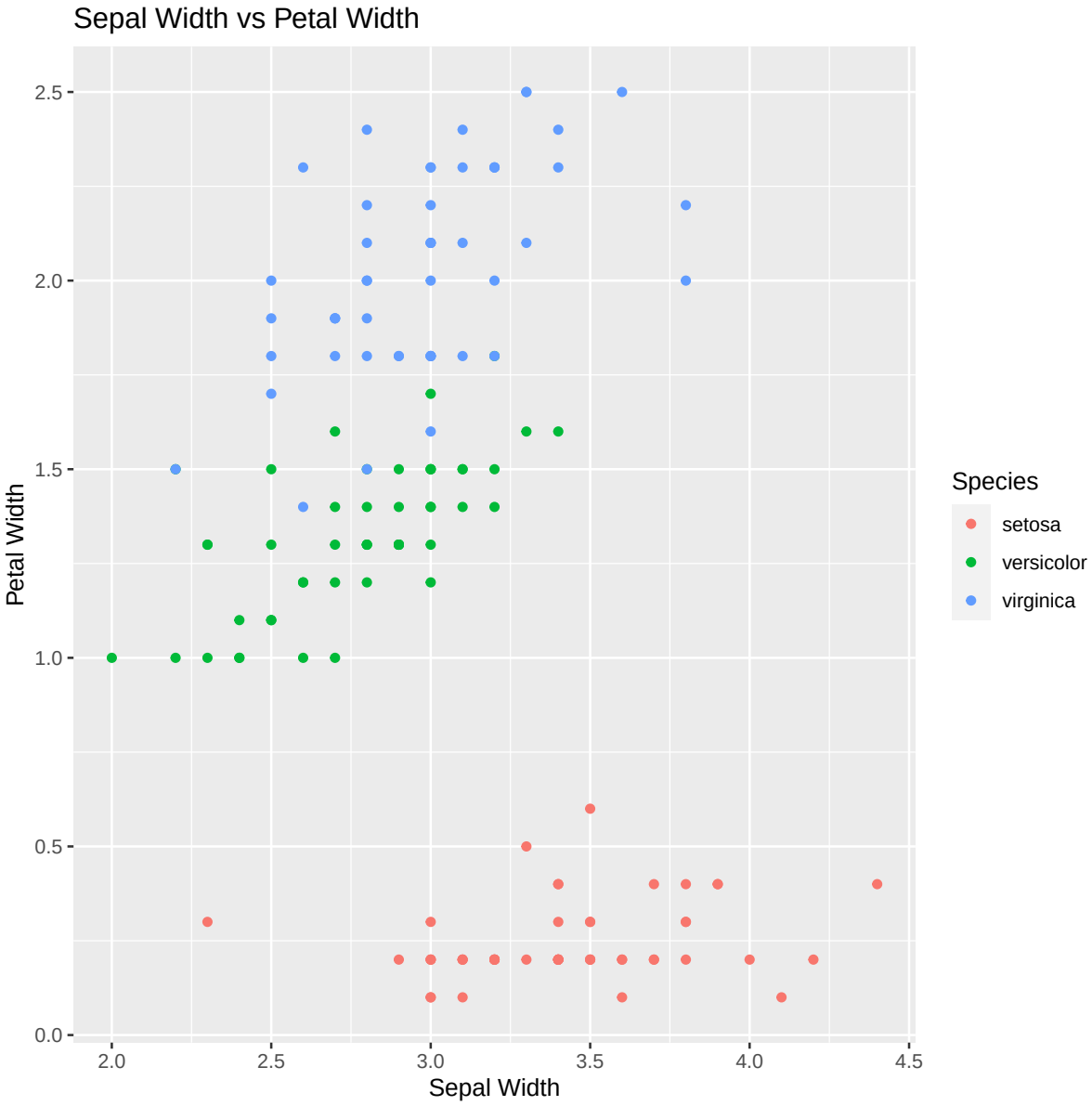
```
iris %>%
    ggplot(aes(x = Sepal.Length, y = Petal.Length,
        color = Species)) + geom_point() + labs(x = "Sepal Length",
    y = "Petal Length", title = "Sepal Length vs Petal Length")
```



```
iris %>%
    filter(Species == "versicolor") %>%
    ggplot(aes(x = Sepal.Length, y = Petal.Length)) +
    geom_point(col = "#0FF020") + labs(x = "Sepal Length",
    y = "Petal Length", title = "Sepal Length vs Petal Length of Versicolor")
```



```
iris %>%
    ggplot(aes(x = Sepal.Width, y = Petal.Width, color = Species)) +
    geom_point() + labs(x = "Sepal Width", y = "Petal Width",
    title = "Sepal Width vs Petal Width")
```



```
deepak_ntr_data <- read.csv("DEEPAKNTR-21-09-2023-to-02-10-2023.csv")
```

```
names(deepak_ntr_data)
```

```
##  [1] "Date"         "series"       "OPEN"         "HIGH"         "LOW"
##  [6] "PREV..CLOSE"  "ltp"          "close"        "vwap"         "X52W.H"
## [11] "X52W.L"       "VOLUME"       "VALUE"        "No.of.trades"
```

```
str(deepak_ntr_data)
```

```
## 'data.frame': 7 obs. of  14 variables:
##  $ Date         : chr  "29-09-2023" "28-09-2023" "27-09-2023" "26-09-2023" ...
##  $ series       : chr  "EQ" "EQ" "EQ" "EQ" ...
##  $ OPEN         : num  2101 2142 2115 2148 2127 ...
##  $ HIGH         : num  2140 2156 2150 2154 2157 ...
##  $ LOW          : num  2091 2082 2100 2100 2121 ...
##  $ PREV..CLOSE : num  2100 2142 2108 2138 2128 ...
##  $ ltp          : num  2119 2109 2140 2114 2138 ...
##  $ close        : num  2120 2100 2142 2108 2138 ...
##  $ vwap         : num  2119 2118 2131 2127 2144 ...
##  $ X52W.H       : num  2373 2373 2373 2373 2373 ...
##  $ X52W.L       : num  1730 1730 1730 1730 1730 1730 1730
##  $ VOLUME       : num  155386 230858 249896 263970 219732 ...
##  $ VALUE        : num  3.29e+08 4.89e+08 5.33e+08 5.61e+08 4.71e+08 ...
##  $ No.of.trades: num  12906 19540 20854 20503 21331 ...
```

```
sapply(deepak_ntr_data, class)
```

```
##         Date       series         OPEN         HIGH          LOW  PREV..CLOSE
##  "character"  "character"    "numeric"    "numeric"    "numeric"    "numeric"
##          ltp        close         vwap       X52W.H       X52W.L       VOLUME
##    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
##        VALUE No.of.trades
##    "numeric"    "numeric"
```

We see that the class of *Date* is *chr*. We shall change it.

```
deepak_ntr_data$Date <- as.POSIXct(deepak_ntr_data$Date)
```

We want to do a line graph of the closing price. The most recent observation is at first in the data set.

```
deepak_ntr_data %>%
    ggplot(aes(x = rev(Date), y = rev(close))) + geom_point() +
    geom_line() + geom_ribbon(aes(ymin = min(close) -
    10, ymax = rev(close)), fill = "#EE6573") + labs(x = "Date",
    y = "Closing Price", title = "Closing Price of D-NTR - 21-29 Sep")
```