UA Summer Workshop: 31 May 2016

Nicholas M. Caruso

Intro to R language

Here is some basic code to get accustomed to the language. You can easily perform calculations, use many of the built in functions for many mathematical operations. There are too many functions to go over today, but we will cover more as we move along.

```
1 + 4
```

```
## [1] 5
```

```
7/5
```

```
## [1] 1.4
```

```
log(10) # calculate the natural log of a number/vector
```

```
## [1] 2.302585
```

```
log10(5) # calculate the log base 10 of a number/vector
```

```
## [1] 0.69897
```

```
rnorm(n=10, mean=0, sd=1) # vector of n random numbers with a mean and standard deviation
```

```
##  [1]  0.30745780  1.77192240  1.94245165  0.09225553 -1.22128757
##  [6]  1.24972012  0.84454964 -0.12281177 -0.48888068  0.90097692
```

```
seq(from=0, to=10, by=1) # vector with min and max, with known increments or length
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10
```

```
rep(1, times=10) # repeat a number or vector
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1
```

```
rep(c(1,2), each=5); rep(c(1,2), times=5) #each and times specify repeats differently
```

```
##  [1] 1 1 1 1 1 2 2 2 2 2
```

```
##  [1] 1 2 1 2 1 2 1 2 1 2
```

```
x <- c(1,2,3); # store an object into a name
length(x) # length of object
```

```
## [1] 3
```

```
mean(x) # mean of object
```

```
## [1] 2
```

```
max(x) #maximum of object
```

```
## [1] 3
```

Dataframes

Dataframes are similar to excel spread sheets, they are arranged by row and column, here are some basic functions to inspect and manipulate dataframes (lots more on this later!)

```
iris #included in base R
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 3            4.7         3.2          1.3         0.2     setosa
## 4            4.6         3.1          1.5         0.2     setosa
## 5            5.0         3.6          1.4         0.2     setosa
## 6            5.4         3.9          1.7         0.4     setosa
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17           5.4         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7         0.3     setosa
## 20           5.1         3.8          1.5         0.3     setosa
## 21           5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
## 23           4.6         3.6          1.0         0.2     setosa
## 24           5.1         3.3          1.7         0.5     setosa
## 25           4.8         3.4          1.9         0.2     setosa
## 26           5.0         3.0          1.6         0.2     setosa
## 27           5.0         3.4          1.6         0.4     setosa
## 28           5.2         3.5          1.5         0.2     setosa
## 29           5.2         3.4          1.4         0.2     setosa
## 30           4.7         3.2          1.6         0.2     setosa
## 31           4.8         3.1          1.6         0.2     setosa
## 32           5.4         3.4          1.5         0.4     setosa
## 33           5.2         4.1          1.5         0.1     setosa
## 34           5.5         4.2          1.4         0.2     setosa
## 35           4.9         3.1          1.5         0.2     setosa
## 36           5.0         3.2          1.2         0.2     setosa
## 37           5.5         3.5          1.3         0.2     setosa
## 38           4.9         3.6          1.4         0.1     setosa
## 39           4.4         3.0          1.3         0.2     setosa
## 40           5.1         3.4          1.5         0.2     setosa
## 41           5.0         3.5          1.3         0.3     setosa
## 42           4.5         2.3          1.3         0.3     setosa
## 43           4.4         3.2          1.3         0.2     setosa
## 44           5.0         3.5          1.6         0.6     setosa
## 45           5.1         3.8          1.9         0.4     setosa
## 46           4.8         3.0          1.4         0.3     setosa
## 47           5.1         3.8          1.6         0.2     setosa
```

```
## 48          4.6         3.2          1.4         0.2     setosa
## 49          5.3         3.7          1.5         0.2     setosa
## 50          5.0         3.3          1.4         0.2     setosa
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
## 56          5.7         2.8          4.5         1.3 versicolor
## 57          6.3         3.3          4.7         1.6 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 59          6.6         2.9          4.6         1.3 versicolor
## 60          5.2         2.7          3.9         1.4 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 62          5.9         3.0          4.2         1.5 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
## 64          6.1         2.9          4.7         1.4 versicolor
## 65          5.6         2.9          3.6         1.3 versicolor
## 66          6.7         3.1          4.4         1.4 versicolor
## 67          5.6         3.0          4.5         1.5 versicolor
## 68          5.8         2.7          4.1         1.0 versicolor
## 69          6.2         2.2          4.5         1.5 versicolor
## 70          5.6         2.5          3.9         1.1 versicolor
## 71          5.9         3.2          4.8         1.8 versicolor
## 72          6.1         2.8          4.0         1.3 versicolor
## 73          6.3         2.5          4.9         1.5 versicolor
## 74          6.1         2.8          4.7         1.2 versicolor
## 75          6.4         2.9          4.3         1.3 versicolor
## 76          6.6         3.0          4.4         1.4 versicolor
## 77          6.8         2.8          4.8         1.4 versicolor
## 78          6.7         3.0          5.0         1.7 versicolor
## 79          6.0         2.9          4.5         1.5 versicolor
## 80          5.7         2.6          3.5         1.0 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 83          5.8         2.7          3.9         1.2 versicolor
## 84          6.0         2.7          5.1         1.6 versicolor
## 85          5.4         3.0          4.5         1.5 versicolor
## 86          6.0         3.4          4.5         1.6 versicolor
## 87          6.7         3.1          4.7         1.5 versicolor
## 88          6.3         2.3          4.4         1.3 versicolor
## 89          5.6         3.0          4.1         1.3 versicolor
## 90          5.5         2.5          4.0         1.3 versicolor
## 91          5.5         2.6          4.4         1.2 versicolor
## 92          6.1         3.0          4.6         1.4 versicolor
## 93          5.8         2.6          4.0         1.2 versicolor
## 94          5.0         2.3          3.3         1.0 versicolor
## 95          5.6         2.7          4.2         1.3 versicolor
## 96          5.7         3.0          4.2         1.2 versicolor
## 97          5.7         2.9          4.2         1.3 versicolor
## 98          6.2         2.9          4.3         1.3 versicolor
## 99          5.1         2.5          3.0         1.1 versicolor
## 100         5.7         2.8          4.1         1.3 versicolor
## 101         6.3         3.3          6.0         2.5  virginica
```

```
## 102          5.8         2.7          5.1         1.9  virginica
## 103          7.1         3.0          5.9         2.1  virginica
## 104          6.3         2.9          5.6         1.8  virginica
## 105          6.5         3.0          5.8         2.2  virginica
## 106          7.6         3.0          6.6         2.1  virginica
## 107          4.9         2.5          4.5         1.7  virginica
## 108          7.3         2.9          6.3         1.8  virginica
## 109          6.7         2.5          5.8         1.8  virginica
## 110          7.2         3.6          6.1         2.5  virginica
## 111          6.5         3.2          5.1         2.0  virginica
## 112          6.4         2.7          5.3         1.9  virginica
## 113          6.8         3.0          5.5         2.1  virginica
## 114          5.7         2.5          5.0         2.0  virginica
## 115          5.8         2.8          5.1         2.4  virginica
## 116          6.4         3.2          5.3         2.3  virginica
## 117          6.5         3.0          5.5         1.8  virginica
## 118          7.7         3.8          6.7         2.2  virginica
## 119          7.7         2.6          6.9         2.3  virginica
## 120          6.0         2.2          5.0         1.5  virginica
## 121          6.9         3.2          5.7         2.3  virginica
## 122          5.6         2.8          4.9         2.0  virginica
## 123          7.7         2.8          6.7         2.0  virginica
## 124          6.3         2.7          4.9         1.8  virginica
## 125          6.7         3.3          5.7         2.1  virginica
## 126          7.2         3.2          6.0         1.8  virginica
## 127          6.2         2.8          4.8         1.8  virginica
## 128          6.1         3.0          4.9         1.8  virginica
## 129          6.4         2.8          5.6         2.1  virginica
## 130          7.2         3.0          5.8         1.6  virginica
## 131          7.4         2.8          6.1         1.9  virginica
## 132          7.9         3.8          6.4         2.0  virginica
## 133          6.4         2.8          5.6         2.2  virginica
## 134          6.3         2.8          5.1         1.5  virginica
## 135          6.1         2.6          5.6         1.4  virginica
## 136          7.7         3.0          6.1         2.3  virginica
## 137          6.3         3.4          5.6         2.4  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 139          6.0         3.0          4.8         1.8  virginica
## 140          6.9         3.1          5.4         2.1  virginica
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

```
str(iris) #look at structure
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
```

```
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
dim(iris) #dimensions
```

```
## [1] 150   5
```

```
head(iris) #first 6 rows
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
tail(iris) #last 6 rows
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

```
colnames(iris) #column names
```

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

```
iris$Sepal.Length # Sepal.Length column
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
##  [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
##  [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
##  [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
##  [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
##  [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
## [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
## [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
## [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
iris['Sepal.Length'] #Sepal.Length column again
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
##  [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
##  [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
##  [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
##  [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
##  [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
## [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
## [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
## [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
iris[1,] #first row, all columns
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
```

```
iris[,1] #first column, all rows
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
##  [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
##  [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
##  [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
##  [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
##  [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
## [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
## [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
## [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
iris[1,1] #first row, first column
```

```
## [1] 5.1
```

dplyr: a package for manipulating dataframes

This overview will include functions for filtering, selecting, summarising, mutating, and joining datasets. First some new functions in *dplyr* that allow us to inspect dataframes.

```
# install.packages('dplyr') #install dplyr package, only need to do this once
library(dplyr) #use library function to load the package
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# convert dataframe to tbl, much easier to examine
iris.tbl <- tbl_df(iris)
iris; iris.tbl
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 3            4.7         3.2          1.3         0.2     setosa
## 4            4.6         3.1          1.5         0.2     setosa
## 5            5.0         3.6          1.4         0.2     setosa
## 6            5.4         3.9          1.7         0.4     setosa
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17           5.4         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7         0.3     setosa
## 20           5.1         3.8          1.5         0.3     setosa
## 21           5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
## 23           4.6         3.6          1.0         0.2     setosa
## 24           5.1         3.3          1.7         0.5     setosa
## 25           4.8         3.4          1.9         0.2     setosa
## 26           5.0         3.0          1.6         0.2     setosa
## 27           5.0         3.4          1.6         0.4     setosa
## 28           5.2         3.5          1.5         0.2     setosa
## 29           5.2         3.4          1.4         0.2     setosa
## 30           4.7         3.2          1.6         0.2     setosa
## 31           4.8         3.1          1.6         0.2     setosa
## 32           5.4         3.4          1.5         0.4     setosa
```

```
## 33          5.2         4.1          1.5         0.1     setosa
## 34          5.5         4.2          1.4         0.2     setosa
## 35          4.9         3.1          1.5         0.2     setosa
## 36          5.0         3.2          1.2         0.2     setosa
## 37          5.5         3.5          1.3         0.2     setosa
## 38          4.9         3.6          1.4         0.1     setosa
## 39          4.4         3.0          1.3         0.2     setosa
## 40          5.1         3.4          1.5         0.2     setosa
## 41          5.0         3.5          1.3         0.3     setosa
## 42          4.5         2.3          1.3         0.3     setosa
## 43          4.4         3.2          1.3         0.2     setosa
## 44          5.0         3.5          1.6         0.6     setosa
## 45          5.1         3.8          1.9         0.4     setosa
## 46          4.8         3.0          1.4         0.3     setosa
## 47          5.1         3.8          1.6         0.2     setosa
## 48          4.6         3.2          1.4         0.2     setosa
## 49          5.3         3.7          1.5         0.2     setosa
## 50          5.0         3.3          1.4         0.2     setosa
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
## 56          5.7         2.8          4.5         1.3 versicolor
## 57          6.3         3.3          4.7         1.6 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 59          6.6         2.9          4.6         1.3 versicolor
## 60          5.2         2.7          3.9         1.4 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 62          5.9         3.0          4.2         1.5 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
## 64          6.1         2.9          4.7         1.4 versicolor
## 65          5.6         2.9          3.6         1.3 versicolor
## 66          6.7         3.1          4.4         1.4 versicolor
## 67          5.6         3.0          4.5         1.5 versicolor
## 68          5.8         2.7          4.1         1.0 versicolor
## 69          6.2         2.2          4.5         1.5 versicolor
## 70          5.6         2.5          3.9         1.1 versicolor
## 71          5.9         3.2          4.8         1.8 versicolor
## 72          6.1         2.8          4.0         1.3 versicolor
## 73          6.3         2.5          4.9         1.5 versicolor
## 74          6.1         2.8          4.7         1.2 versicolor
## 75          6.4         2.9          4.3         1.3 versicolor
## 76          6.6         3.0          4.4         1.4 versicolor
## 77          6.8         2.8          4.8         1.4 versicolor
## 78          6.7         3.0          5.0         1.7 versicolor
## 79          6.0         2.9          4.5         1.5 versicolor
## 80          5.7         2.6          3.5         1.0 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 83          5.8         2.7          3.9         1.2 versicolor
## 84          6.0         2.7          5.1         1.6 versicolor
## 85          5.4         3.0          4.5         1.5 versicolor
## 86          6.0         3.4          4.5         1.6 versicolor
```

```
## 87          6.7         3.1          4.7         1.5 versicolor
## 88          6.3         2.3          4.4         1.3 versicolor
## 89          5.6         3.0          4.1         1.3 versicolor
## 90          5.5         2.5          4.0         1.3 versicolor
## 91          5.5         2.6          4.4         1.2 versicolor
## 92          6.1         3.0          4.6         1.4 versicolor
## 93          5.8         2.6          4.0         1.2 versicolor
## 94          5.0         2.3          3.3         1.0 versicolor
## 95          5.6         2.7          4.2         1.3 versicolor
## 96          5.7         3.0          4.2         1.2 versicolor
## 97          5.7         2.9          4.2         1.3 versicolor
## 98          6.2         2.9          4.3         1.3 versicolor
## 99          5.1         2.5          3.0         1.1 versicolor
## 100         5.7         2.8          4.1         1.3 versicolor
## 101         6.3         3.3          6.0         2.5  virginica
## 102         5.8         2.7          5.1         1.9  virginica
## 103         7.1         3.0          5.9         2.1  virginica
## 104         6.3         2.9          5.6         1.8  virginica
## 105         6.5         3.0          5.8         2.2  virginica
## 106         7.6         3.0          6.6         2.1  virginica
## 107         4.9         2.5          4.5         1.7  virginica
## 108         7.3         2.9          6.3         1.8  virginica
## 109         6.7         2.5          5.8         1.8  virginica
## 110         7.2         3.6          6.1         2.5  virginica
## 111         6.5         3.2          5.1         2.0  virginica
## 112         6.4         2.7          5.3         1.9  virginica
## 113         6.8         3.0          5.5         2.1  virginica
## 114         5.7         2.5          5.0         2.0  virginica
## 115         5.8         2.8          5.1         2.4  virginica
## 116         6.4         3.2          5.3         2.3  virginica
## 117         6.5         3.0          5.5         1.8  virginica
## 118         7.7         3.8          6.7         2.2  virginica
## 119         7.7         2.6          6.9         2.3  virginica
## 120         6.0         2.2          5.0         1.5  virginica
## 121         6.9         3.2          5.7         2.3  virginica
## 122         5.6         2.8          4.9         2.0  virginica
## 123         7.7         2.8          6.7         2.0  virginica
## 124         6.3         2.7          4.9         1.8  virginica
## 125         6.7         3.3          5.7         2.1  virginica
## 126         7.2         3.2          6.0         1.8  virginica
## 127         6.2         2.8          4.8         1.8  virginica
## 128         6.1         3.0          4.9         1.8  virginica
## 129         6.4         2.8          5.6         2.1  virginica
## 130         7.2         3.0          5.8         1.6  virginica
## 131         7.4         2.8          6.1         1.9  virginica
## 132         7.9         3.8          6.4         2.0  virginica
## 133         6.4         2.8          5.6         2.2  virginica
## 134         6.3         2.8          5.1         1.5  virginica
## 135         6.1         2.6          5.6         1.4  virginica
## 136         7.7         3.0          6.1         2.3  virginica
## 137         6.3         3.4          5.6         2.4  virginica
## 138         6.4         3.1          5.5         1.8  virginica
## 139         6.0         3.0          4.8         1.8  virginica
## 140         6.9         3.1          5.4         2.1  virginica
```

```
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

```
## Source: local data frame [150 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## ..          ...         ...          ...         ...     ...
```

```
glimpse(iris.tbl) # similar to str
```

```
## Observations: 150
## Variables: 5
## $ Sepal.Length (dbl) 5.1, 4.9, 4.7, 4.6, 5.0, 5.4, 4.6, 5.0, 4.4, 4.9,...
## $ Sepal.Width  (dbl) 3.5, 3.0, 3.2, 3.1, 3.6, 3.9, 3.4, 3.4, 2.9, 3.1,...
## $ Petal.Length (dbl) 1.4, 1.4, 1.3, 1.5, 1.4, 1.7, 1.4, 1.5, 1.4, 1.5,...
## $ Petal.Width  (dbl) 0.2, 0.2, 0.2, 0.2, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1,...
## $ Species      (fctr) setosa, setosa, setosa, setosa, setosa, setosa, ...
```

```
View(iris.tbl) # view the whole dataframe
arrange(iris.tbl, Sepal.Width) # arrange dataframe by column(s)
```

```
## Source: local data frame [150 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
##           (dbl)       (dbl)        (dbl)       (dbl)     (fctr)
## 1           5.0         2.0          3.5         1.0 versicolor
## 2           6.0         2.2          4.0         1.0 versicolor
## 3           6.2         2.2          4.5         1.5 versicolor
## 4           6.0         2.2          5.0         1.5  virginica
## 5           4.5         2.3          1.3         0.3     setosa
## 6           5.5         2.3          4.0         1.3 versicolor
## 7           6.3         2.3          4.4         1.3 versicolor
## 8           5.0         2.3          3.3         1.0 versicolor
## 9           4.9         2.4          3.3         1.0 versicolor
## 10          5.5         2.4          3.8         1.1 versicolor
## ..          ...         ...          ...         ...        ...
```

dplyr package makes use of "piping" %>% to pass functions to dataframes, creating more readable code. Plus there are lots of great functions in dplyr to help with data manipulation!

Filtering Datasets

Use == for exact matches, %in% for multiple matches and != for exclusions (does not equal). Can also use greater than/less than for numeric columns (>= for greater than and equal to). Use & to string multiple filtering arguments together

```
iris.tbl %>%
  filter(Species=='setosa')
```

```
## Source: local data frame [50 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## ..          ...         ...          ...         ...     ...
```

```
iris.tbl %>%
  filter(Species %in% c('setosa','versicolor'))
```

```
## Source: local data frame [100 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## ..          ...         ...          ...         ...     ...
```

```
iris.tbl %>%
  filter(Species != 'setosa')
```

```
## Source: local data frame [100 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
##           (dbl)       (dbl)        (dbl)       (dbl)     (fctr)
## 1           7.0         3.2          4.7         1.4 versicolor
```

```
## 2           6.4         3.2          4.5         1.5 versicolor
## 3           6.9         3.1          4.9         1.5 versicolor
## 4           5.5         2.3          4.0         1.3 versicolor
## 5           6.5         2.8          4.6         1.5 versicolor
## 6           5.7         2.8          4.5         1.3 versicolor
## 7           6.3         3.3          4.7         1.6 versicolor
## 8           4.9         2.4          3.3         1.0 versicolor
## 9           6.6         2.9          4.6         1.3 versicolor
## 10          5.2         2.7          3.9         1.4 versicolor
## ..          ...         ...          ...         ...        ...
```

```
iris.tbl %>%
  filter(!Species %in% c('versicolor','virginica'))
```

```
## Source: local data frame [50 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## ..          ...         ...          ...         ...     ...
```

```
iris.tbl %>%
  filter(Sepal.Width > 3 & Species=='setosa')
```

```
## Source: local data frame [42 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.7         3.2          1.3         0.2  setosa
## 3           4.6         3.1          1.5         0.2  setosa
## 4           5.0         3.6          1.4         0.2  setosa
## 5           5.4         3.9          1.7         0.4  setosa
## 6           4.6         3.4          1.4         0.3  setosa
## 7           5.0         3.4          1.5         0.2  setosa
## 8           4.9         3.1          1.5         0.1  setosa
## 9           5.4         3.7          1.5         0.2  setosa
## 10          4.8         3.4          1.6         0.2  setosa
## ..          ...         ...          ...         ...     ...
```

Select Columns in Dataframes

Can specify which columns to select or remove (denoted with '-'). Can have multiple arguments.

```
iris.tbl %>%
  select(Species)
```

```
## Source: local data frame [150 x 1]
##
##    Species
##     (fctr)
## 1   setosa
## 2   setosa
## 3   setosa
## 4   setosa
## 5   setosa
## 6   setosa
## 7   setosa
## 8   setosa
## 9   setosa
## 10  setosa
## ..     ...
```

```
iris.tbl %>%
  select(Sepal.Length, Species)
```

```
## Source: local data frame [150 x 2]
##
##    Sepal.Length Species
##           (dbl)  (fctr)
## 1           5.1  setosa
## 2           4.9  setosa
## 3           4.7  setosa
## 4           4.6  setosa
## 5           5.0  setosa
## 6           5.4  setosa
## 7           4.6  setosa
## 8           5.0  setosa
## 9           4.4  setosa
## 10          4.9  setosa
## ..          ...     ...
```

```
iris.tbl %>%
  select(-Species)
```

```
## Source: local data frame [150 x 4]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
##           (dbl)       (dbl)        (dbl)       (dbl)
## 1           5.1         3.5          1.4         0.2
## 2           4.9         3.0          1.4         0.2
## 3           4.7         3.2          1.3         0.2
## 4           4.6         3.1          1.5         0.2
## 5           5.0         3.6          1.4         0.2
## 6           5.4         3.9          1.7         0.4
## 7           4.6         3.4          1.4         0.3
## 8           5.0         3.4          1.5         0.2
## 9           4.4         2.9          1.4         0.2
## 10          4.9         3.1          1.5         0.1
## ..          ...         ...          ...         ...
```

Summarise Dataframes

The summarise functions are useful for making summary stats (e.g., mean, sd, etc) of your data. These functions take a vector of values and return a single value. Can be used in combination with group_by() function to examine different levels of a factor.

```
iris.tbl %>%
  summarise(mn.sepalWidth=mean(Sepal.Width),
            sd.sepalWidth=sd(Sepal.Width))
```

```
## Source: local data frame [1 x 2]
##
##   mn.sepalWidth sd.sepalWidth
##           (dbl)         (dbl)
## 1      3.057333     0.4358663
```

```
iris.tbl %>%
  group_by(Species) %>%
  summarise(mn.sepalWidth=mean(Sepal.Width),
            sd.sepalWidth=sd(Sepal.Width))
```

```
## Source: local data frame [3 x 3]
##
##      Species mn.sepalWidth sd.sepalWidth
##       (fctr)         (dbl)         (dbl)
## 1     setosa         3.428     0.3790644
## 2 versicolor         2.770     0.3137983
## 3  virginica         2.974     0.3224966
```

```
iris.summary <- iris.tbl %>%
  group_by(Species) %>%
  summarise_each(funs(mean, sd, n(), min, max, var, median))

View(iris.summary)
```

Mutate Functions

The mutate family of functions are useful for making new columns from your existing dataset (e.g., log-transforming a column). These functions take a vector of values and return a vector of values that is the same length. Can be applied over the whole column or used with group_by() similarly to summarise functions.

```
iris.tbl %>%
  mutate(log.sepalWidth=log(Sepal.Width))
```

```
## Source: local data frame [150 x 6]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
```

```
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## ..          ...         ...          ...         ...     ...
## Variables not shown: log.sepalWidth (dbl)
```

```
View(iris.tbl %>%
  group_by(Species) %>%
  mutate(mn.sepalWidth=mean(Sepal.Width),
         sd.sepalWidth=sd(Sepal.Width),
         sepalWidth.stnd=(Sepal.Width-mn.sepalWidth)/sd.sepalWidth))

# Some Useful mutate functions that I typically use
## Lag = previous value
## Lead = next value
## between = Logical, is the value between any two values
## cumsum = cumulative sum

iris.tbl %>%
  select(Sepal.Width) %>%
  mutate(lag(Sepal.Width),
         lead(Sepal.Width),
         between(Sepal.Width, 3, 4),
         cumsum(Sepal.Width))
```

```
## Source: local data frame [150 x 5]
##
##    Sepal.Width lag(Sepal.Width) lead(Sepal.Width)
##          (dbl)            (dbl)             (dbl)
## 1          3.5               NA               3.0
## 2          3.0              3.5               3.2
## 3          3.2              3.0               3.1
## 4          3.1              3.2               3.6
## 5          3.6              3.1               3.9
## 6          3.9              3.6               3.4
## 7          3.4              3.9               3.4
## 8          3.4              3.4               2.9
## 9          2.9              3.4               3.1
## 10         3.1              2.9               3.7
## ..         ...              ...               ...
## Variables not shown: between(Sepal.Width, 3, 4) (lgl), cumsum(Sepal.Width)
##   (dbl)
```

Join Functions

The join functions combines rows of dataframes based on identifiers (e.g., date, time, individual). We'll create our own dataframes to demonstrate these functions.

```
dat1 <- data_frame(x1=c(25,14,68,79,64,139,49,119,111),
                   x2=rep(letters[1:3], each=3),
                   x3=c(1,14,22,2,0,20,2,13,22))


dat2 <- data_frame(x2=rep(letters[c(1,2,4)], each=2),
```

```
                  y1=c(1:6),
                  y2=rep(c(22,2,0), times=2))

dat1; dat2
```

```
## Source: local data frame [9 x 3]
##
##      x1    x2    x3
##   (dbl) (chr) (dbl)
## 1    25     a     1
## 2    14     a    14
## 3    68     a    22
## 4    79     b     2
## 5    64     b     0
## 6   139     b    20
## 7    49     c     2
## 8   119     c    13
## 9   111     c    22
```

```
## Source: local data frame [6 x 3]
##
##      x2    y1    y2
##   (chr) (int) (dbl)
## 1     a     1    22
## 2     a     2     2
## 3     b     3     0
## 4     b     4    22
## 5     d     5     2
## 6     d     6     0
```

```
(new.dat <- left_join(x=dat1, y=dat2, by='x2')) #join from b to a
```

```
## Source: local data frame [15 x 5]
##
##       x1    x2    x3    y1    y2
##    (dbl) (chr) (dbl) (int) (dbl)
## 1     25     a     1     1    22
## 2     25     a     1     2     2
## 3     14     a    14     1    22
## 4     14     a    14     2     2
## 5     68     a    22     1    22
## 6     68     a    22     2     2
## 7     79     b     2     3     0
## 8     79     b     2     4    22
## 9     64     b     0     3     0
## 10    64     b     0     4    22
## 11   139     b    20     3     0
## 12   139     b    20     4    22
## 13    49     c     2    NA    NA
## 14   119     c    13    NA    NA
## 15   111     c    22    NA    NA
```

```
right_join(dat1, dat2, by='x2') # joing from a to b
```

```
## Source: local data frame [14 x 5]
##
```

```
##       x1    x2    x3    y1    y2
##    (dbl) (chr) (dbl) (int) (dbl)
## 1     25     a     1     1    22
## 2     14     a    14     1    22
## 3     68     a    22     1    22
## 4     25     a     1     2     2
## 5     14     a    14     2     2
## 6     68     a    22     2     2
## 7     79     b     2     3     0
## 8     64     b     0     3     0
## 9    139     b    20     3     0
## 10    79     b     2     4    22
## 11    64     b     0     4    22
## 12   139     b    20     4    22
## 13    NA     d    NA     5     2
## 14    NA     d    NA     6     0
```

```
full_join(dat1, dat2, by='x2') # keep all
```

```
## Source: local data frame [17 x 5]
##
##       x1    x2    x3    y1    y2
##    (dbl) (chr) (dbl) (int) (dbl)
## 1     25     a     1     1    22
## 2     25     a     1     2     2
## 3     14     a    14     1    22
## 4     14     a    14     2     2
## 5     68     a    22     1    22
## 6     68     a    22     2     2
## 7     79     b     2     3     0
## 8     79     b     2     4    22
## 9     64     b     0     3     0
## 10    64     b     0     4    22
## 11   139     b    20     3     0
## 12   139     b    20     4    22
## 13    49     c     2    NA    NA
## 14   119     c    13    NA    NA
## 15   111     c    22    NA    NA
## 16    NA     d    NA     5     2
## 17    NA     d    NA     6     0
```

```
inner_join(dat1, dat2, by='x2') #only keep matching in both
```

```
## Source: local data frame [12 x 5]
##
##       x1    x2    x3    y1    y2
##    (dbl) (chr) (dbl) (int) (dbl)
## 1     25     a     1     1    22
## 2     25     a     1     2     2
## 3     14     a    14     1    22
## 4     14     a    14     2     2
## 5     68     a    22     1    22
## 6     68     a    22     2     2
## 7     79     b     2     3     0
## 8     79     b     2     4    22
```

```
## 9     64     b     0     3     0
## 10    64     b     0     4    22
## 11   139     b    20     3     0
## 12   139     b    20     4    22
```

```
semi_join(dat1, dat2, by='x2') # keep rows in a that match b
```

```
## Source: local data frame [6 x 3]
##
##      x1    x2    x3
##   (dbl) (chr) (dbl)
## 1    25     a     1
## 2    14     a    14
## 3    68     a    22
## 4    79     b     2
## 5    64     b     0
## 6   139     b    20
```

```
anti_join(dat1, dat2, by='x2') # only keep a that don't match b
```

```
## Source: local data frame [3 x 3]
##
##      x1    x2    x3
##   (dbl) (chr) (dbl)
## 1    49     c     2
## 2   119     c    13
## 3   111     c    22
```

```
## Can join by multiple matching columns

dat1 <- dat1 %>%
  rename(y2=x3)

inner_join(dat1, dat2, by=c('x2','y2'))
```

```
## Source: local data frame [2 x 4]
##
##      x1    x2    y2    y1
##   (dbl) (chr) (dbl) (int)
## 1    68     a    22     1
## 2    64     b     0     3
```

tidyr: a package for changing the shape of datasets

This overview will include functions for changing the shape (layout) of datasets. The two functions I typically use are gather(), which gathers columns into rows (wide format to long format) and spread(), which spreads rows into columns (long format to wide).

```
# install.packages('tidyr')
# install.packages('lubridate')
library(tidyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
dat <- data_frame(ind=c(1,2,3), trial1=c(20,30,40), trial2=c(60,70,80),
                  date=dmy(c('26-05-2016','27-05-2016','28-05-2016')))

dat.g <- dat %>%
  gather(key=trial, value=response, trial1:trial2)

dat.g %>%
  spread(key=trial, value=response)
```

```
## Source: local data frame [3 x 4]
##
##     ind       date trial1 trial2
##   (dbl)     (date)  (dbl)  (dbl)
## 1     1 2016-05-26     20     60
## 2     2 2016-05-27     30     70
## 3     3 2016-05-28     40     80
```

```
dat %>%
  separate(date, c('year','month','day'), sep='-') %>%
  mutate_each_(funs(as.numeric), c('year','month','day')) %>%
  gather(trial, response, trial1:trial2)
```

```
## Source: local data frame [6 x 6]
##
##     ind  year month   day  trial response
##   (dbl) (dbl) (dbl) (dbl)  (chr)    (dbl)
## 1     1  2016     5    26 trial1       20
## 2     2  2016     5    27 trial1       30
## 3     3  2016     5    28 trial1       40
## 4     1  2016     5    26 trial2       60
## 5     2  2016     5    27 trial2       70
## 6     3  2016     5    28 trial2       80
```

```
## Or if we want Julian day

dat %>%
  mutate(julian.day=yday(date)) %>%
  gather(trial, response, trial1:trial2)
```

```
## Source: local data frame [6 x 5]
##
##     ind       date julian.day  trial response
##   (dbl)     (date)      (dbl)  (chr)    (dbl)
## 1     1 2016-05-26        147 trial1       20
## 2     2 2016-05-27        148 trial1       30
## 3     3 2016-05-28        149 trial1       40
## 4     1 2016-05-26        147 trial2       60
## 5     2 2016-05-27        148 trial2       70
## 6     3 2016-05-28        149 trial2       80
```

There are a lot of useful functions in the *lubridate* package for working with date objects.

ggplot2: a package for data visualization

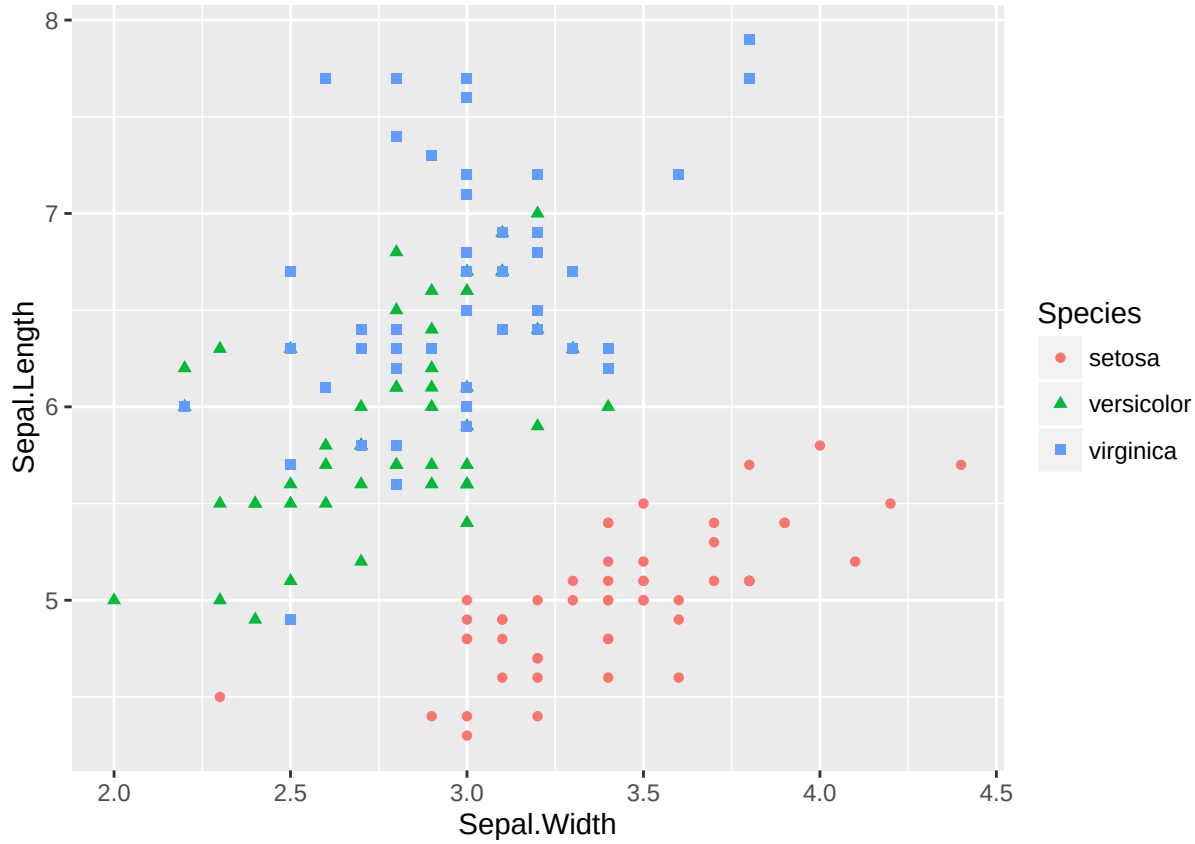
ggplot2 is a great alternative to the already flexible plotting included with base R. However, the advantage of ggplot is that it every graphic made is based on three components: the data, the coordinate system (e.g., x and y), and a set of visuals to display the data (geoms). First we build the graph with some data and the aesthetic properties (aes) of the graphic (e.g., x, y, groupings, shape of points, color, linetypes). Next we'll add some geoms to represent the data

ggplot2 basics

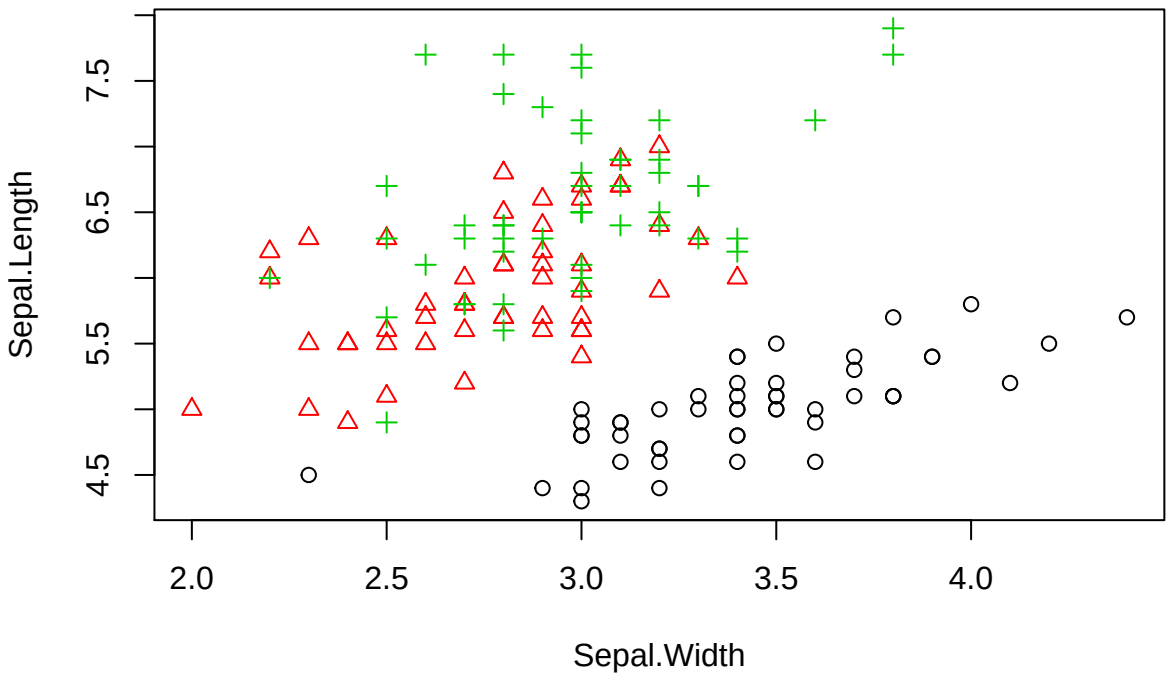
```
# install.packages('ggplot2')
library(ggplot2)

p <- ggplot(data=iris, aes(x=Sepal.Width, y=Sepal.Length))

p + geom_point()
```

We can include additional aesthetics to deliniate species in the graphic. For comparison here's how you would plot a similar graphic in base R.

```
ggplot(data=iris, aes(x=Sepal.Width, y=Sepal.Length, shape=Species, color=Species)) +
  geom_point()
```



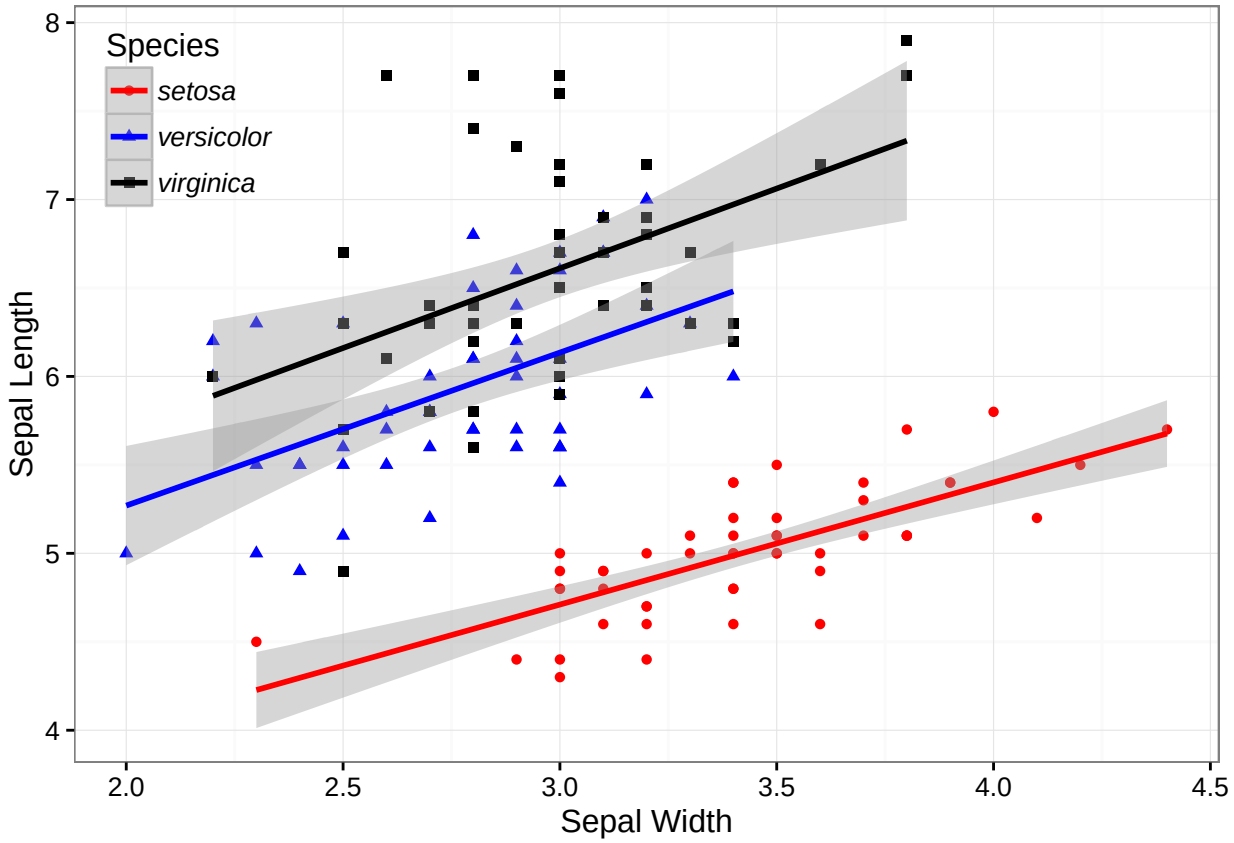

```
plot(Sepal.Length~Sepal.Width, data=iris, pch=as.numeric(iris$Species),
     col=as.numeric(iris$Species))
```



Its pretty obvious how much better the default ggplots look compared to base R. I'll clean up the plot a little bit more using ggplot to demonstrate what typically goes into making a more polished graphic. Multiple visual representations (geoms, themes, labels, etc) need to be separate by a '+'. For this graphic, we will change the x and y labels, apply a simple theme (there are a lot of ggthemes, or you can create your own), add linear model predictions (with standard errors) for each species, and move the legend to inside the plot to use the full plotting space and make the species names italics. Note that since we set up the aestics for species in the beginning of the plot, we don't have to do it again for the linear model predictions, but we could add aesthics within any geom (e.g., change line width, lwd) If we want to use one value for all levels, we would describe that aesthetic outside of the aes() function (e.g., alpha).

Scatterplot

```
ggplot(data=iris, aes(x=Sepal.Width, y=Sepal.Length, shape=Species, color=Species)) +
  geom_point() +
  labs(x='Sepal Width',
       y='Sepal Length') +
  geom_smooth(method='lm', formula=y~x, se=TRUE) +
  theme_bw() + # classic dark on light theme
  theme(legend.position=c(0.1, 0.85), # c(x, y) assume both are [0,1]
        legend.background=element_rect('transparent'),
        legend.text=element_text(face='italic')) +
  scale_color_manual(values=c('red','blue','black'))
```

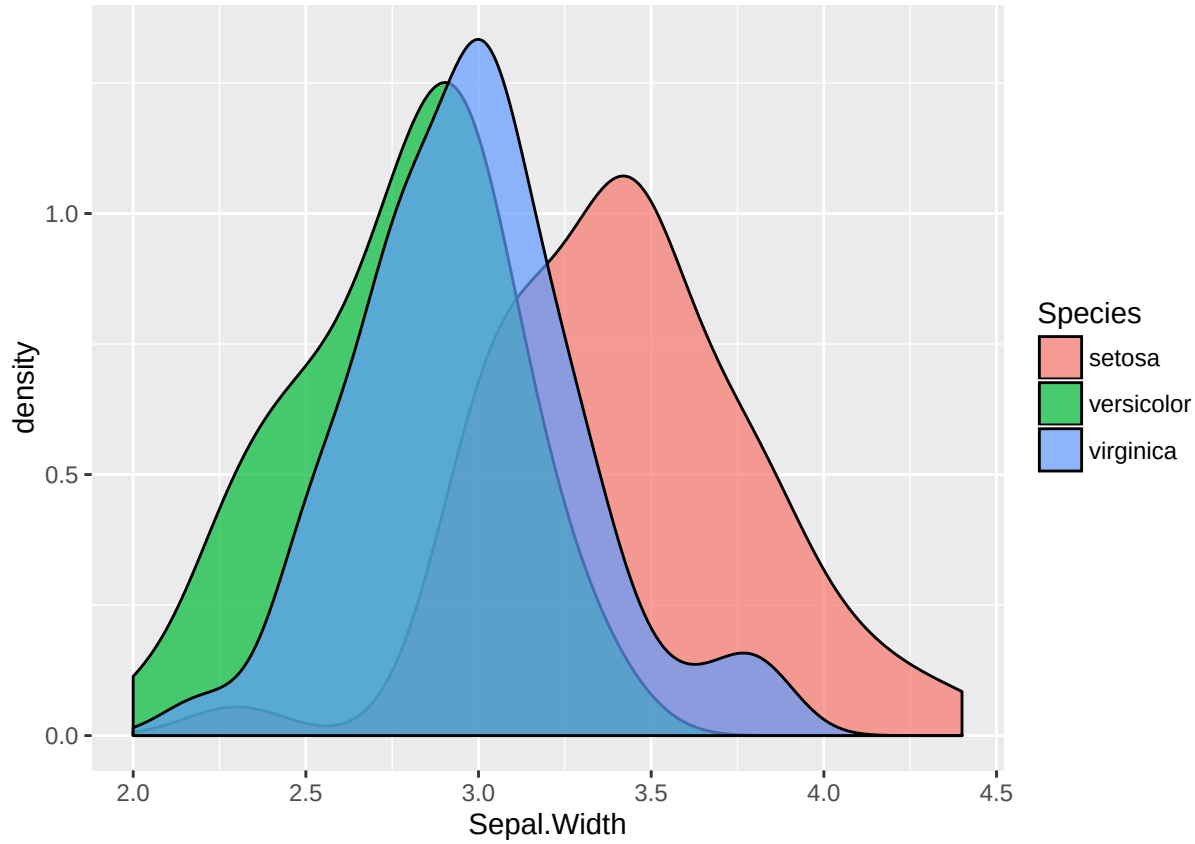


Typical geoms

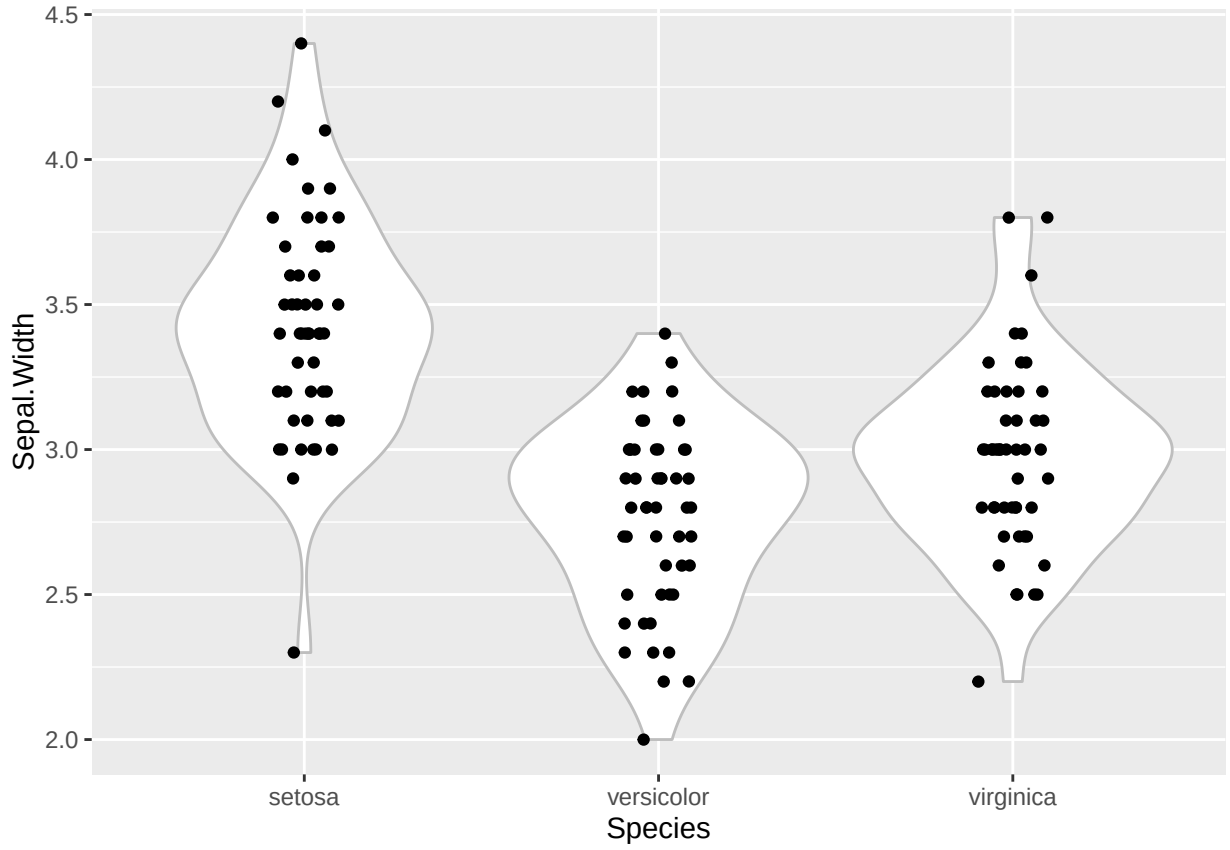
There are a lot more geoms to produce many different types of plots, here are the ones that I typically use
geom_density() distribution of variables
geom_histogram() histogram
geom_ribbon() typically used for showing confidence interval bands
geom_jitter() move points small increments if they overlap (e.g., binomial data)
geom_point() plot two continuous variables with a scatterplot
geom_smooth() add lm, glm, and loess smoothed fits to data
geom_bar() barplot
geom_boxplot() boxplot
geom_violin() violin plot
geom_desnity2d() 2 dimensional density contours
geom_errorbar() geom_errorbarh() error bars for vertical and horiztonal intervals respectively
geom_polygon() typically used to make background maps (use map_data() to retrieve reference maps)
geom_raster() plot rasters, must be transformed to a dataframe first

Examples

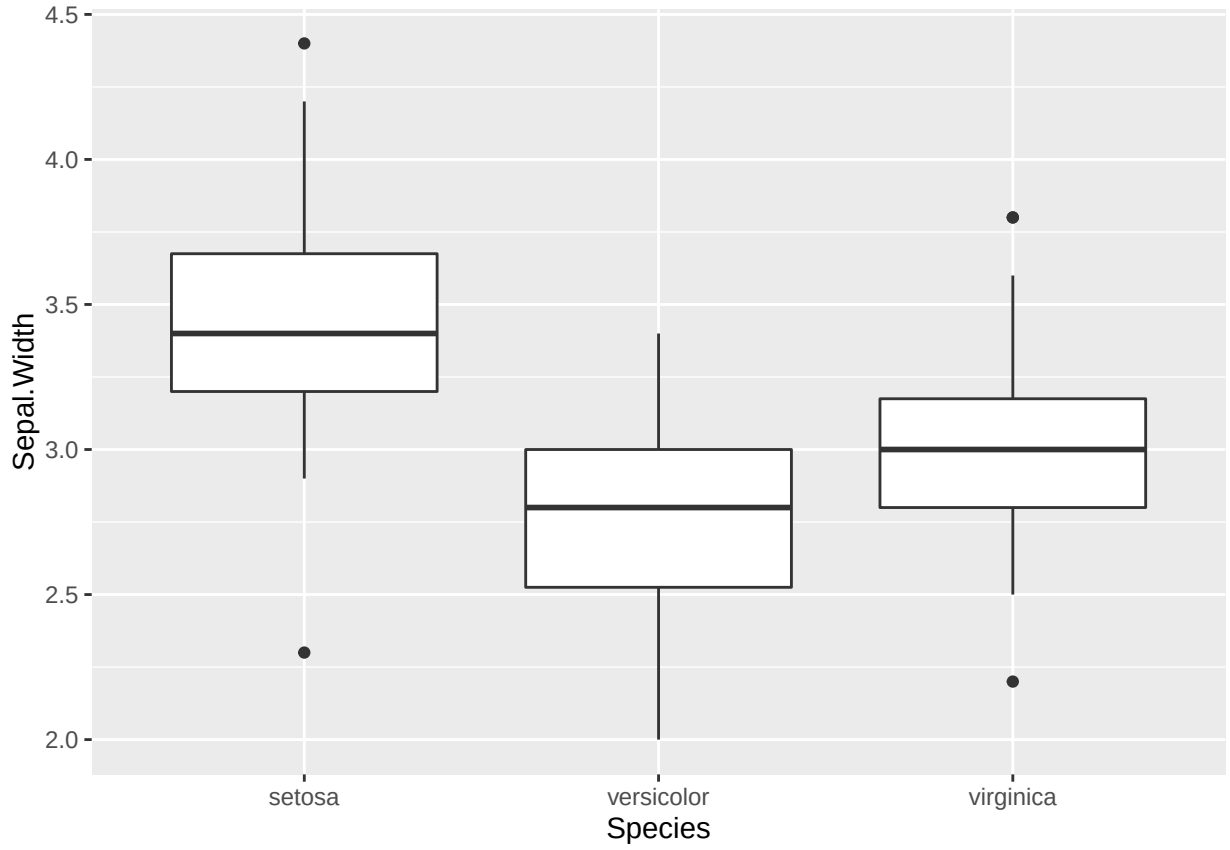
```
# Density plot
ggplot(iris, aes(x=Sepal.Width, fill=Species)) +
  geom_density(alpha=0.7)
```



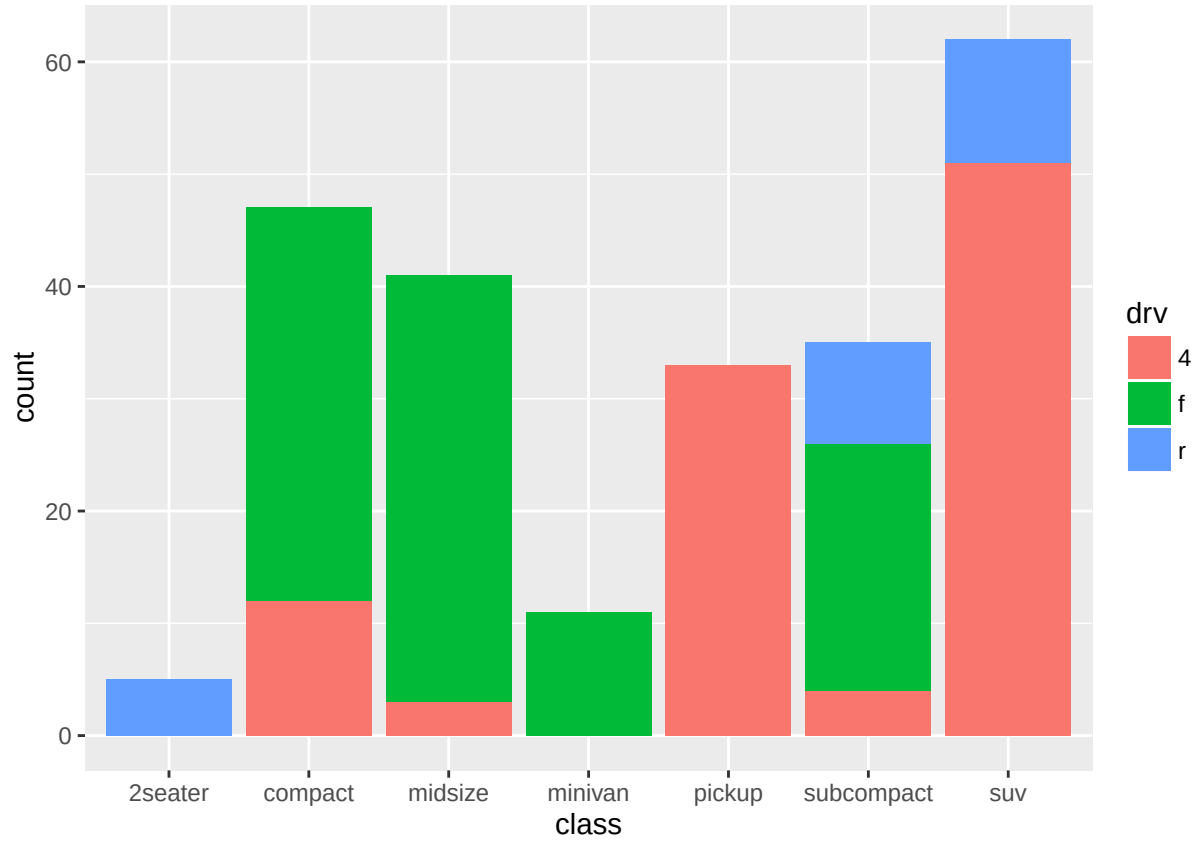

```
# Violin plot with jittered points
ggplot(iris, aes(x=Species, y=Sepal.Width)) +
  geom_violin(color='gray') +
  geom_jitter(width=0.25, height=0.001)
```



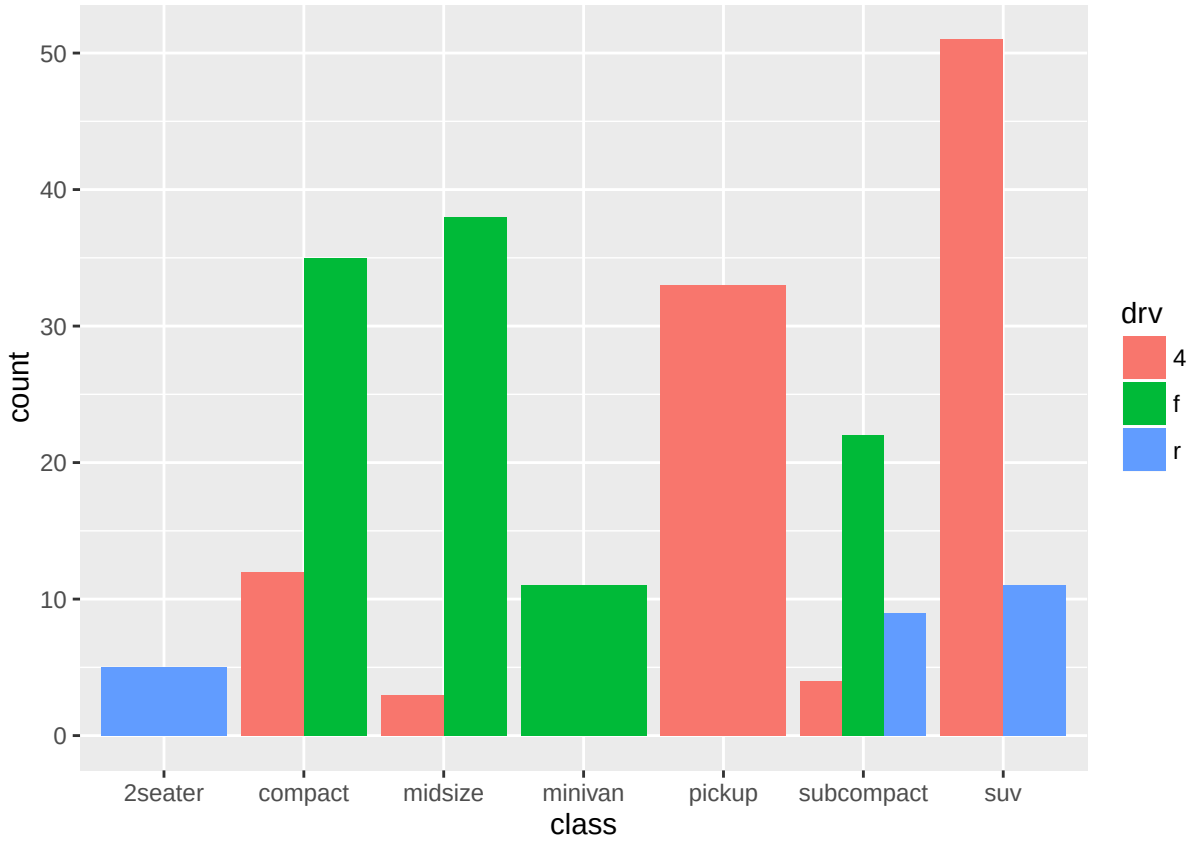

```
# Boxplot
ggplot(iris, aes(x=Species, y=Sepal.Width)) +
  geom_boxplot()
```



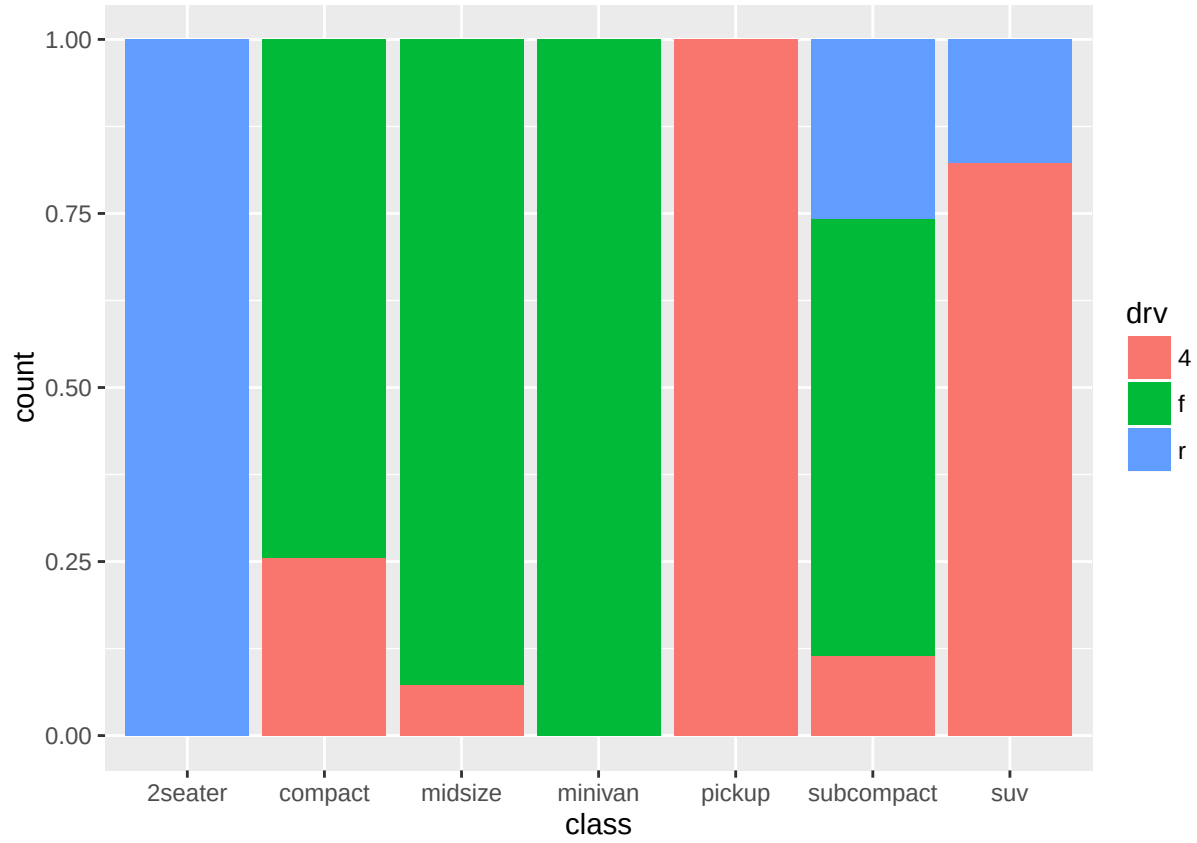

```
# Barplot - Stacked
ggplot(mpg, aes(x=class, fill=drv)) +
  geom_bar()
```



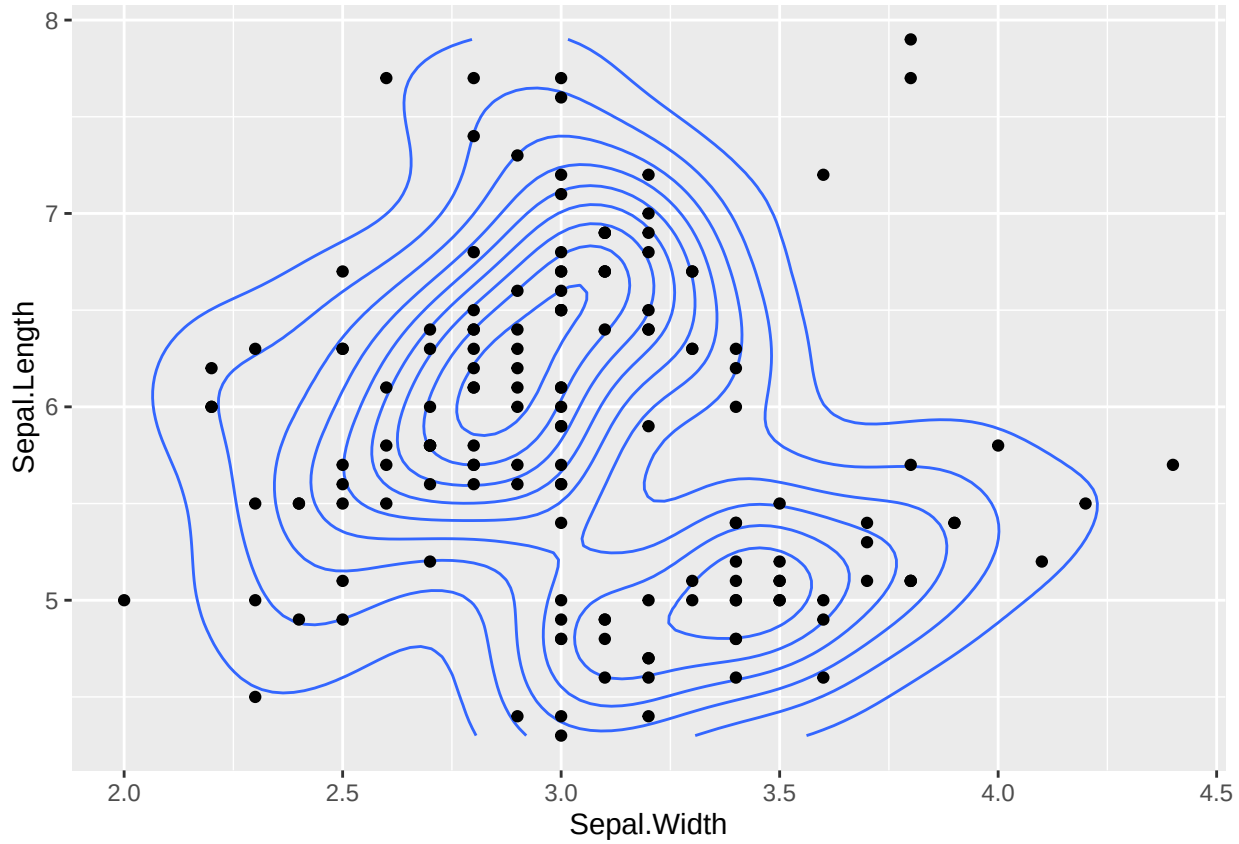

```
# Barplot, side by side
ggplot(mpg, aes(x=class, fill=drv)) +
  geom_bar(position='dodge')
```




```
# Barplot, filled
ggplot(mpg, aes(class, fill=drv)) +
  geom_bar(position='fill')
```




```
# 2d contour
ggplot(iris, aes(Sepal.Width, Sepal.Length)) +
  geom_density2d() +
  geom_point()
```




```
# maps
# install.packages('maps')
library(maps)
```

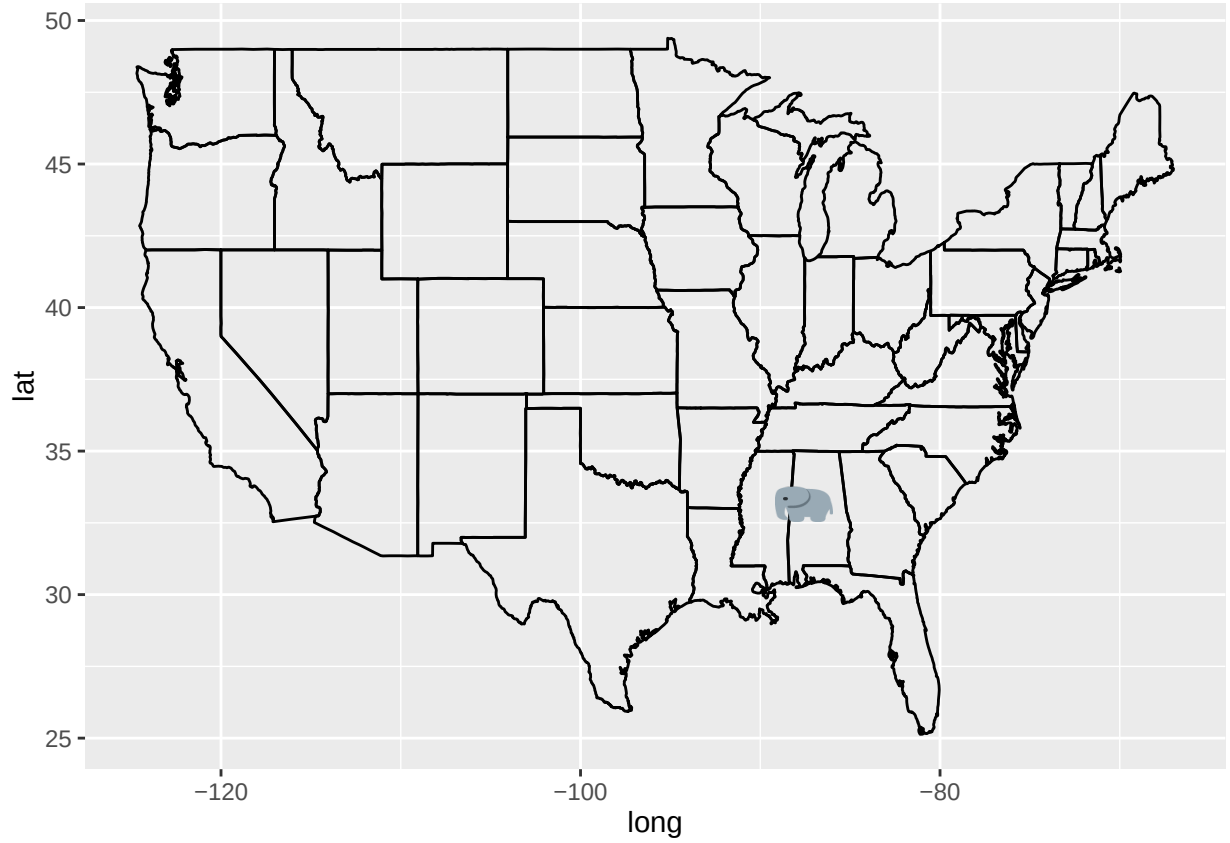
```
##
##  # maps v3.1: updated 'world': all lakes moved to separate new #
##  # 'lakes' database. Type '?world' or 'news(package="maps")'.  #
```

```
us.map <- map_data('state')

bama <- data_frame(long=-87.5692, lat=33.2098)
# install.packages('emoGG')
# devtools::install_github('dill/emoGG')
library(emoGG)

emoji_search("elephant")
```

```
##         emoji  code    keyword
## 786  elephant 1f418     animal
## 787  elephant 1f418     nature
## 788  elephant 1f418       nose
## 789  elephant 1f418   thailand
## 790  elephant 1f418     circus
## 924       ant 1f41c     animal
## 925       ant 1f41c     insect
## 926       ant 1f41c     nature
## 927       ant 1f41c        bug
```

```
## 2733        a 1f170 red-square
## 2734        a 1f170   alphabet
## 2735        a 1f170     letter
```

```
ggplot(bama)+
  geom_polygon(data=us.map, aes(x=long, y=lat, group=group), fill='transparent',
               color='black') +
  geom_emoji(emoji='1f418', aes(x=long, y=lat))
```




```
# raster
# install.packages('raster')
library(raster)
```

```
## Loading required package: sp
```

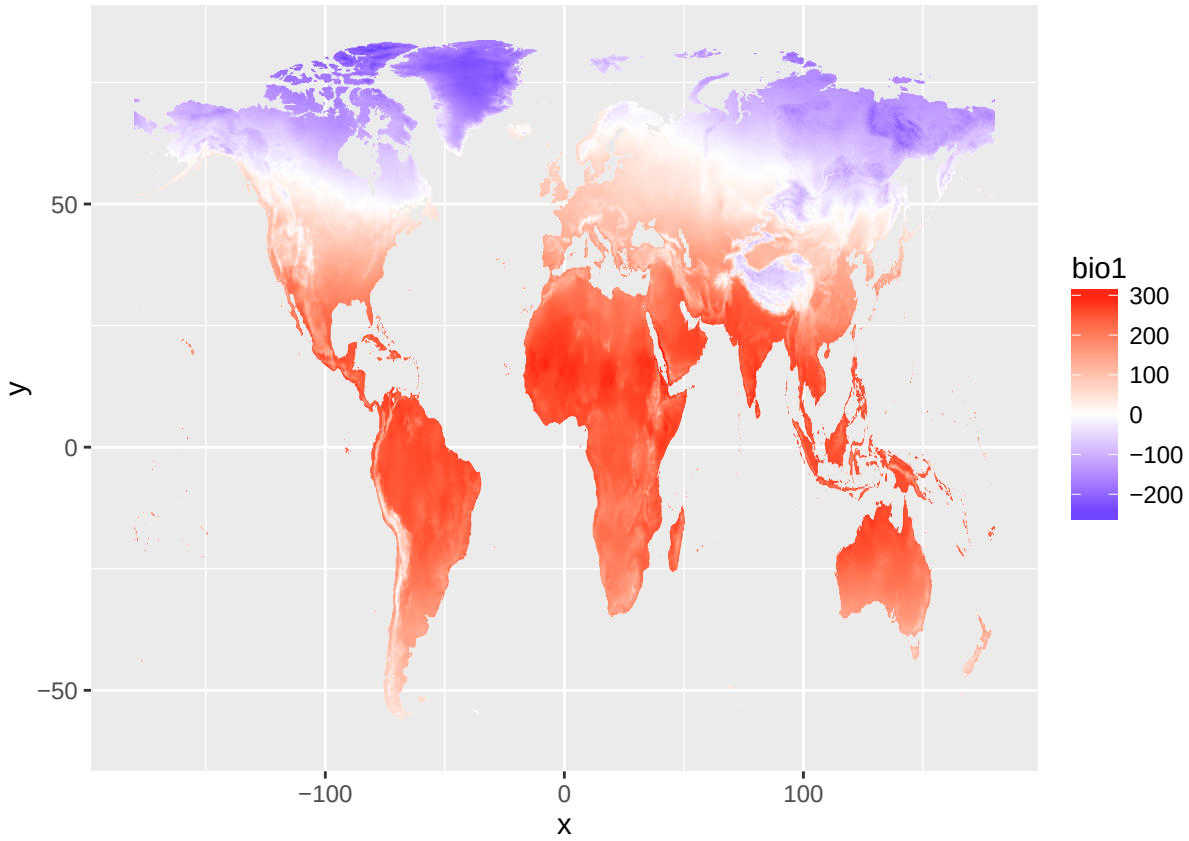
```
##
## Attaching package: 'raster'
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
bio.clim <- getData('worldclim', var='bio', res=10)

bio1.df <- as.data.frame(bio.clim[[1]], xy=TRUE)
```

```
bio1.df <- bio1.df %>%
  na.omit()

ggplot(bio1.df, aes(x=x, y=y)) +
  geom_raster(aes(fill=bio1)) +
  scale_fill_gradient2(low='blue', high='red')
```



Facets

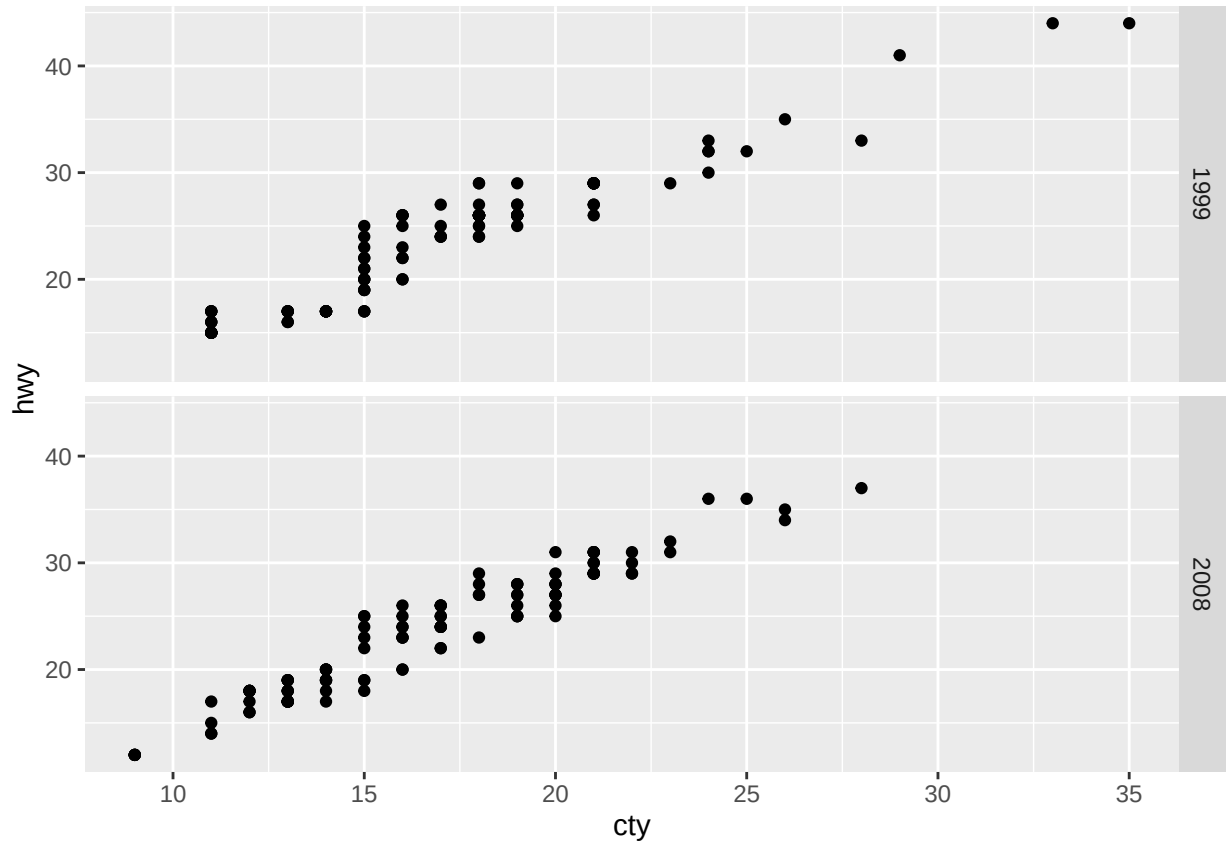
A nice feature of ggplots is facets, which divide a plot into subplots based on categorical (discrete) variables. There are two ways to facet plots facet_wrap() and facet_grid(). Wrap will only plot those subplots that have data, while grid will plot all combinations regardless of data, this is demonstrated below. Use the labeller argument to adjust how the facets are labelled.

```
mpg
```

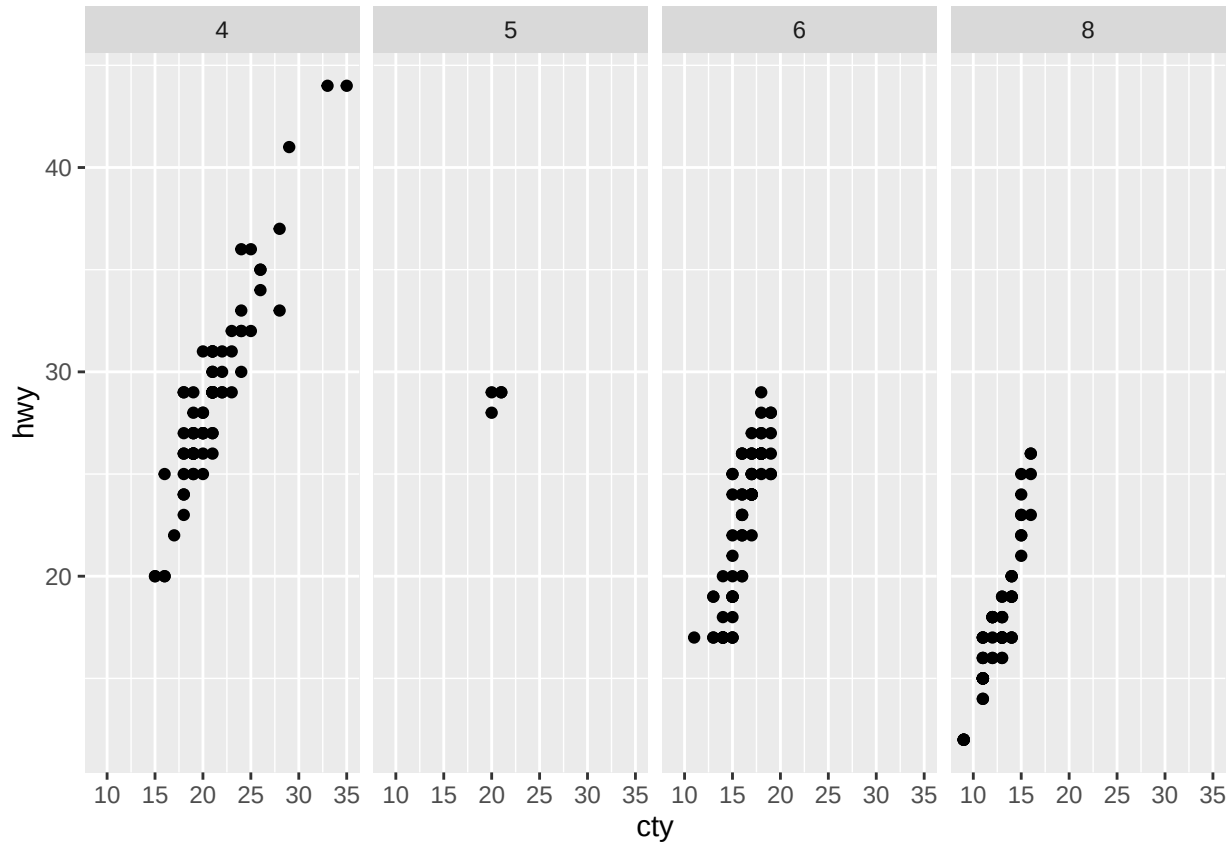
```
## Source: local data frame [234 x 11]
##
##    manufacturer      model displ  year   cyl      trans   drv   cty   hwy
##           (chr)      (chr) (dbl) (int) (int)      (chr) (chr) (int) (int)
## 1          audi         a4   1.8  1999     4   auto(l5)     f    18    29
## 2          audi         a4   1.8  1999     4 manual(m5)     f    21    29
## 3          audi         a4   2.0  2008     4 manual(m6)     f    20    31
## 4          audi         a4   2.0  2008     4   auto(av)     f    21    30
## 5          audi         a4   2.8  1999     6   auto(l5)     f    16    26
## 6          audi         a4   2.8  1999     6 manual(m5)     f    18    26
```

```
## 7          audi         a4   3.1  2008     6   auto(av)     f    18    27
## 8          audi a4 quattro   1.8  1999     4 manual(m5)     4    18    26
## 9          audi a4 quattro   1.8  1999     4   auto(l5)     4    16    25
## 10         audi a4 quattro   2.0  2008     4 manual(m6)     4    20    28
## ..          ...        ...   ...   ...   ...        ...   ...   ...   ...
## Variables not shown: fl (chr), class (chr)
```

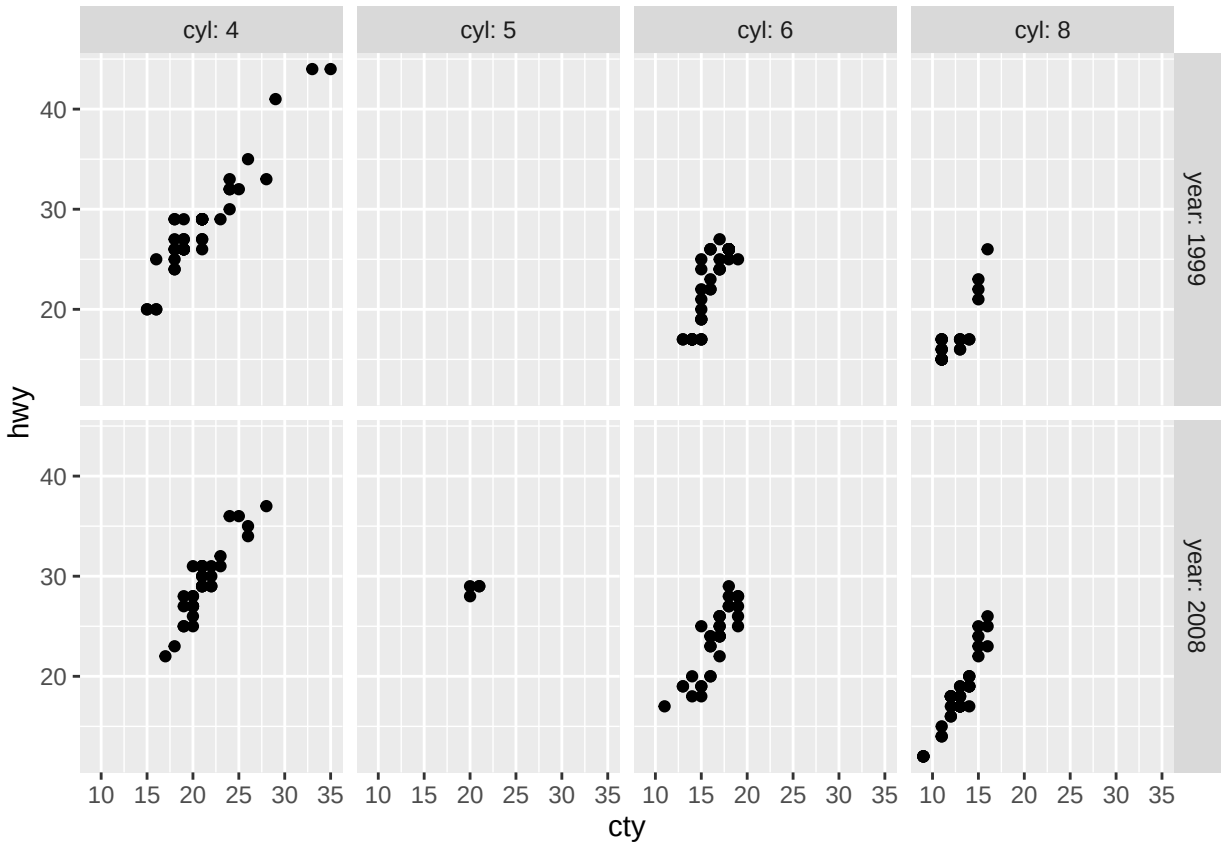
```
ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_point() +
  facet_grid(year~.)
```



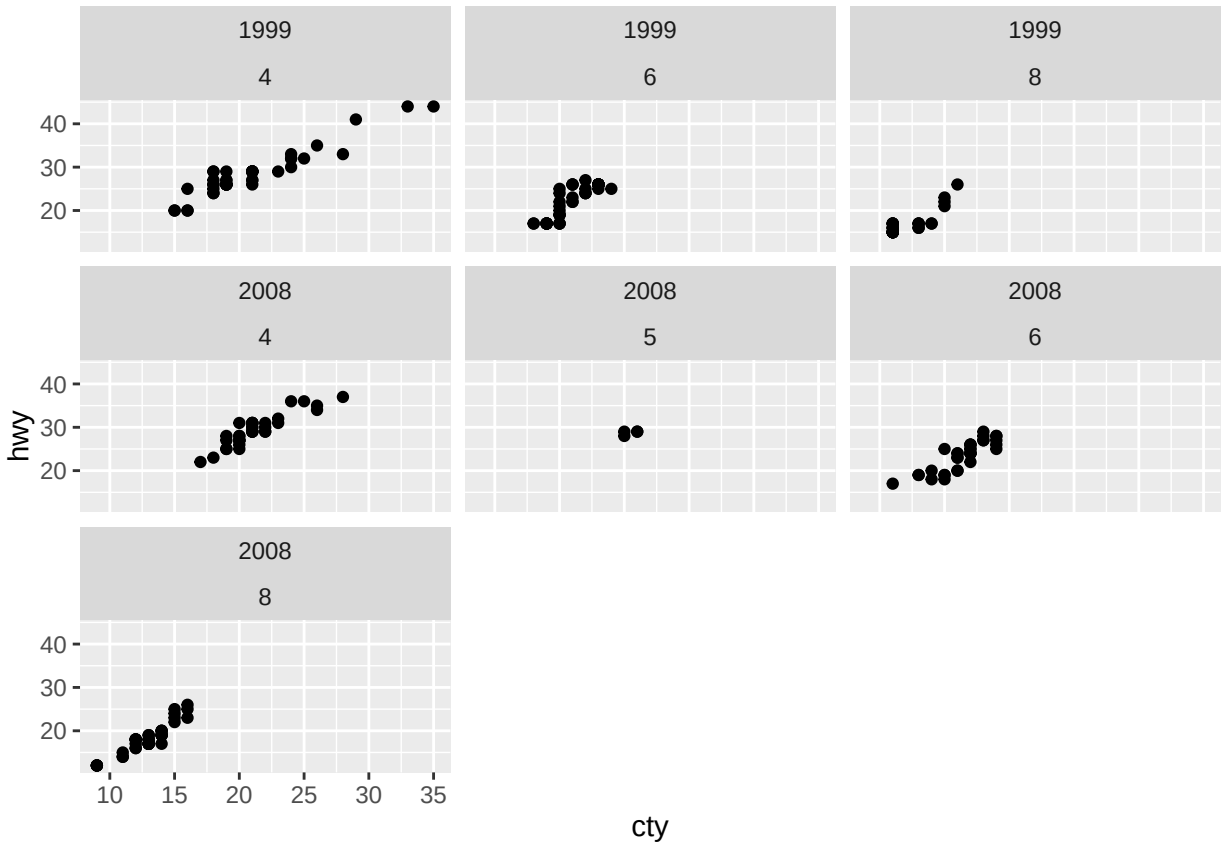

```
ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_point() +
  facet_grid(.~cyl)
```




```
ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_point() +
  facet_grid(year~cyl, labeller=label_both)
```




```
ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_point() +
  facet_wrap(year~cyl)
```



Multiple panels

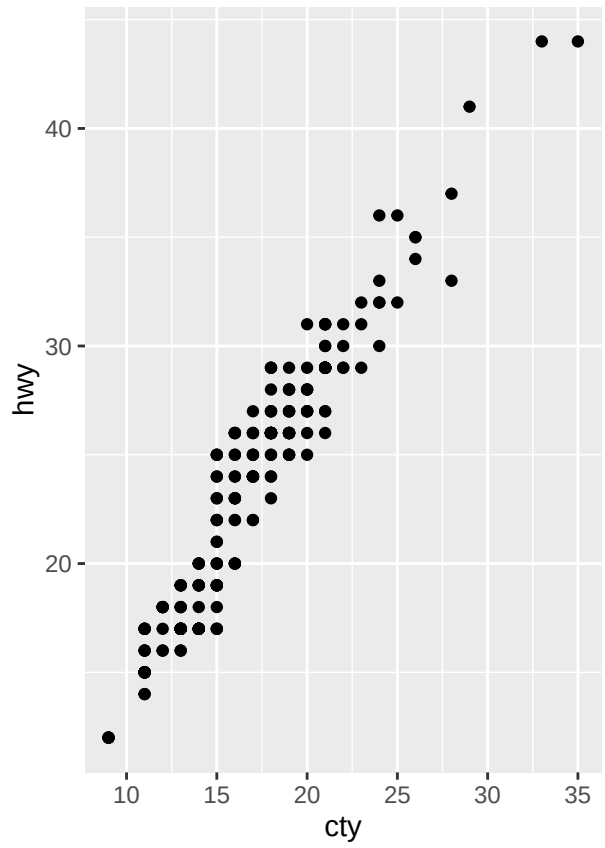
If you have multiple plots that you want to panel together in *ggplot2*, you will have to use the *gridExtra* package. The default is to give each plot equal sizes but you can adjust the layout to make them different sizes.

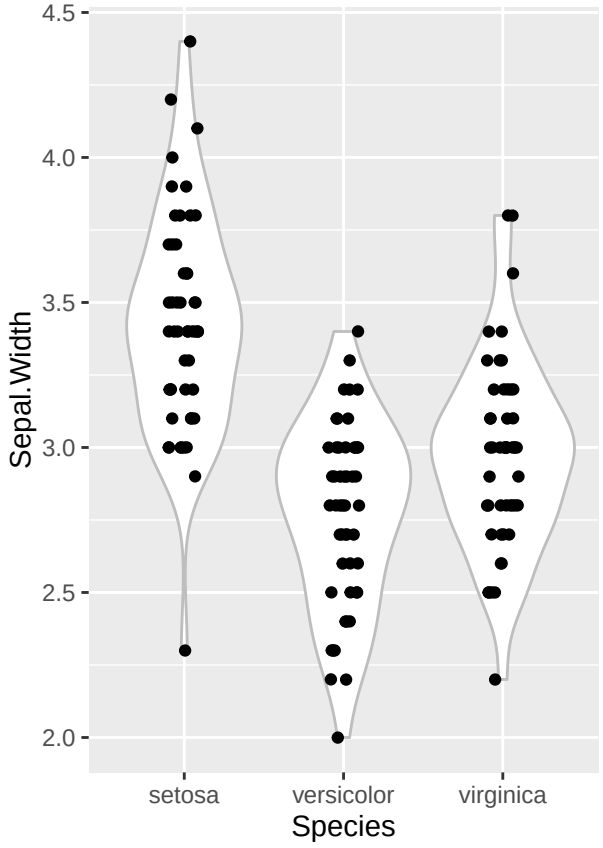
```
# install.packages('gridExtra')
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

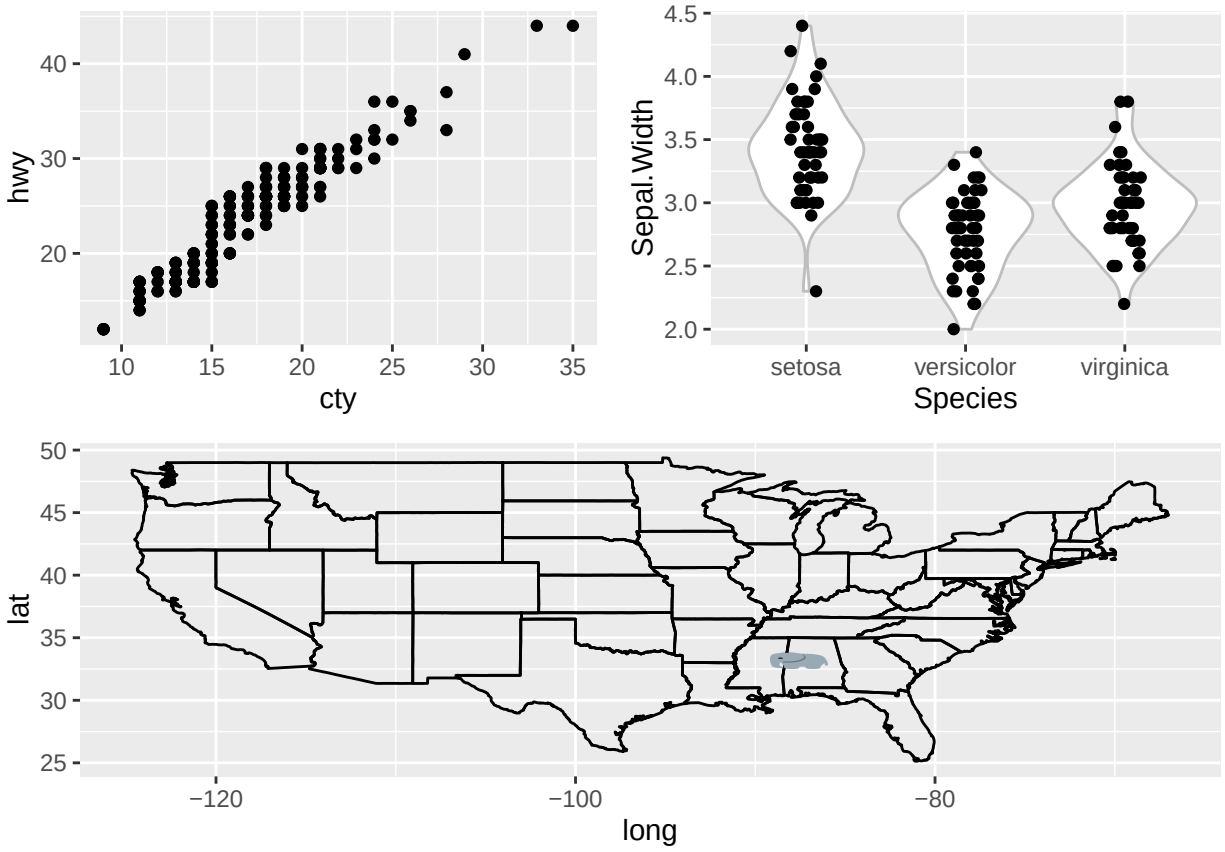
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
p1 <- ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_point()

p2 <- ggplot(iris, aes(x=Species, y=Sepal.Width)) +
  geom_violin(color='gray') +
  geom_jitter(width=0.25, height=0.001)

grid.arrange(p1, p2, ncol=2)
```






```
p3 <- ggplot(bama)+
  geom_polygon(data=us.map, aes(x=long, y=lat, group=group), fill='transparent',
               color='black') +
  geom_emoji(emoji='1f418', aes(x=long, y=lat))


grid.arrange(p1, p2, p3, layout_matrix = rbind(c(1,2),
                                               c(3,3)))
```



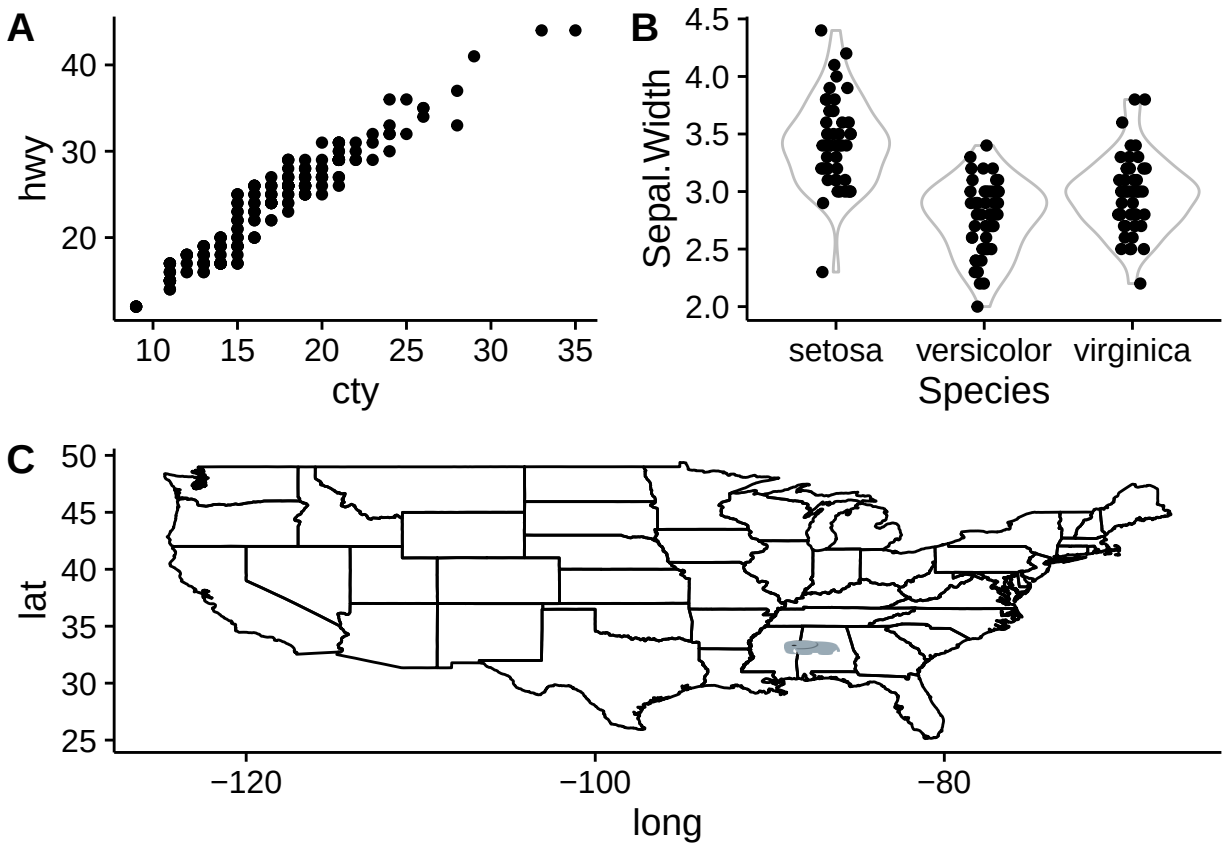
Alternatively you can use the *cowplot* package which automatically changes the theme (you can use any theme though) and includes a nice feature for adding plot identifiers (e.g., A, B, C, etc).

```
# install.packages('cowplot')
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

```
ggdraw() +
  draw_plot(plot=p1, x=0, y=0.5, width=0.5, height=0.5) +
  draw_plot(p2, 0.5, 0.5, 0.5, 0.5) +
  draw_plot(p3, 0, 0, 1, 0.5) +
  draw_plot_label(label=c('A','B','C'), x=c(0,0.5,0), y=c(1,1,0.5),
                            size=15)
```


A
hwy
cty
10
15
20
25
30
35
20
30
40
B
Sepal.Width
Species
setosa
versicolor
virginica
2.0
2.5
3.0
3.5
4.0
4.5
C
lat
long
25
30
35
40
45
50
−120
−100
−80