RSG III: Data Visualization

M. Kartje

March 7, 2019

Review of Data structures

I. Vectors

- Homogeneous
- 1-dimensional

EX

```
#a vector of numbers
vec_1<-c(1, 19, 34, 76)
vec_1
```

```
## [1]  1 19 34 76
```

```
#a vector of characters
vec_2<-c('cat', 'dog', 'rat', 'pet')
vec_2
```

```
## [1] "cat" "dog" "rat" "pet"
```

II. Lists

- Heterogeneous
- 1 - n-dimensional

EX

```
#a list o numbers
l_1<-list(1, 19, 34, 76)
l_1
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 19
##
## [[3]]
## [1] 34
##
## [[4]]
## [1] 76
```

```
#a list of characters
l_2<-list('cat', 'dog', 'rat', 'pet')
l_2
```

```
## [[1]]
## [1] "cat"
##
## [[2]]
## [1] "dog"
```

```
##
## [[3]]
## [1] "rat"
##
## [[4]]
## [1] "pet"
```

```
#a list of lists
l_3<-list(l_1, l_2)
l_3
```

```
## [[1]]
## [[1]][[1]]
## [1] 1
##
## [[1]][[2]]
## [1] 19
##
## [[1]][[3]]
## [1] 34
##
## [[1]][[4]]
## [1] 76
##
##
## [[2]]
## [[2]][[1]]
## [1] "cat"
##
## [[2]][[2]]
## [1] "dog"
##
## [[2]][[3]]
## [1] "rat"
##
## [[2]][[4]]
## [1] "pet"
```

```
#lsit of vectors
```

III. Dataframes

- List of vectors
- 2-dimensional (matrix-like)
- Homogeneous or heterogeneous?

EX

```
#The iris dataframe
head(iris)
```

```
##   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
```

```r
str(iris)
```

```
## '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 ...
```

```r
head(iris[,3])
```

```
## [1] 1.4 1.4 1.3 1.5 1.4 1.7
```

```r
head(iris$Petal.Length)
```

```
## [1] 1.4 1.4 1.3 1.5 1.4 1.7
```

```r
#The cars dataframe
head(cars)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```r
str(cars)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
##  $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

```r
head(cars[,2])
```

```
## [1]  2 10  4 22 16 10
```

```r
head(cars$dist)
```

```
## [1]  2 10  4 22 16 10
```

Dataframes make plotting easy!

Data Visualization

R has several basic functions for plotting data:

- `hist()`
- `plot()`
- `boxplot()`

These functions are built in – they come with R when you download it.

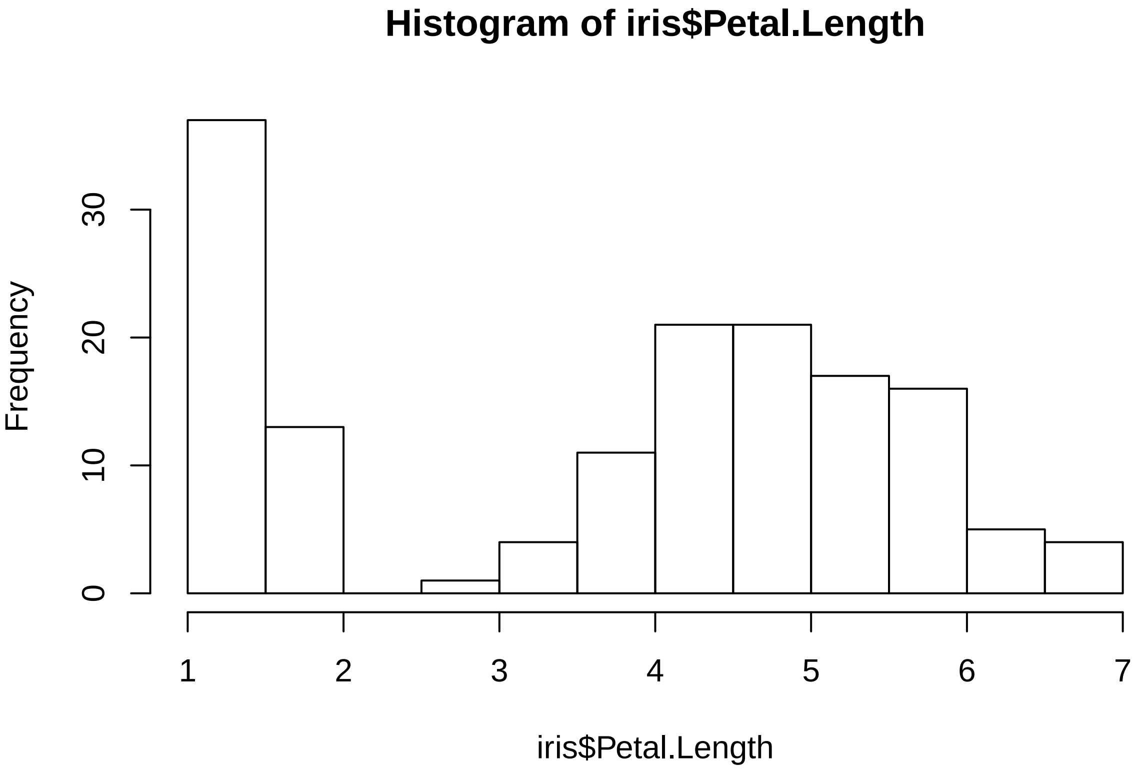
Various other plotting packages exist.

By far, the most popular is ggplot2. We'll get to this today if there's time.

Histograms

1) Look up how to use the `hist()` function using `?hist()`
2) Use `hist()` to examine the frequency distribution of petal lengths in thie `iris` dataset.

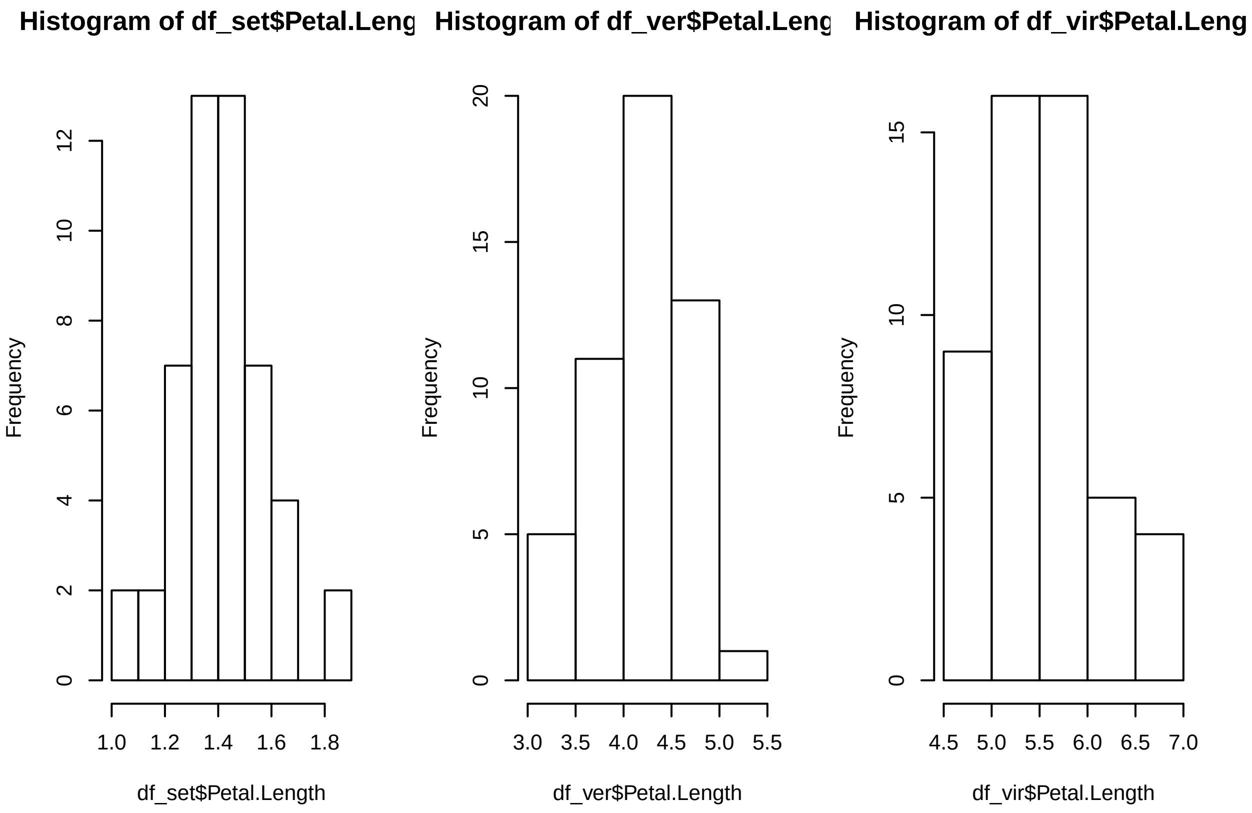
```
hist(iris$Petal.Length)
```



3) Subset the iris dataset to plot each species separately

```
df_set<-subset(iris, Species == 'setosa')
df_ver<-subset(iris, Species == 'versicolor')
df_vir<-subset(iris, Species == 'virginica')

#use par() to partition
par(mfrow = c(1, 3))
hist(df_set$Petal.Length)
hist(df_ver$Petal.Length)
hist(df_vir$Petal.Length)
```



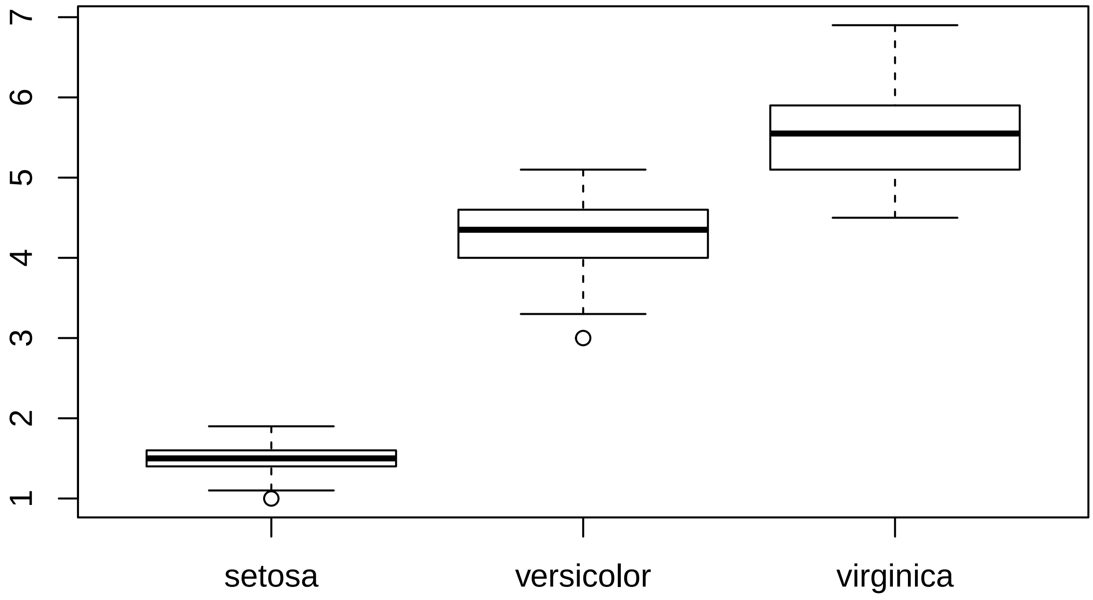
It looks like different species have different distributions of petal length… How to best visualize this?

Boxplots

Boxplots allow for comparison of data across different levels of a factor. Look at the help file for boxplots.

Using this info, construct a boxplot showing petal length vs. species

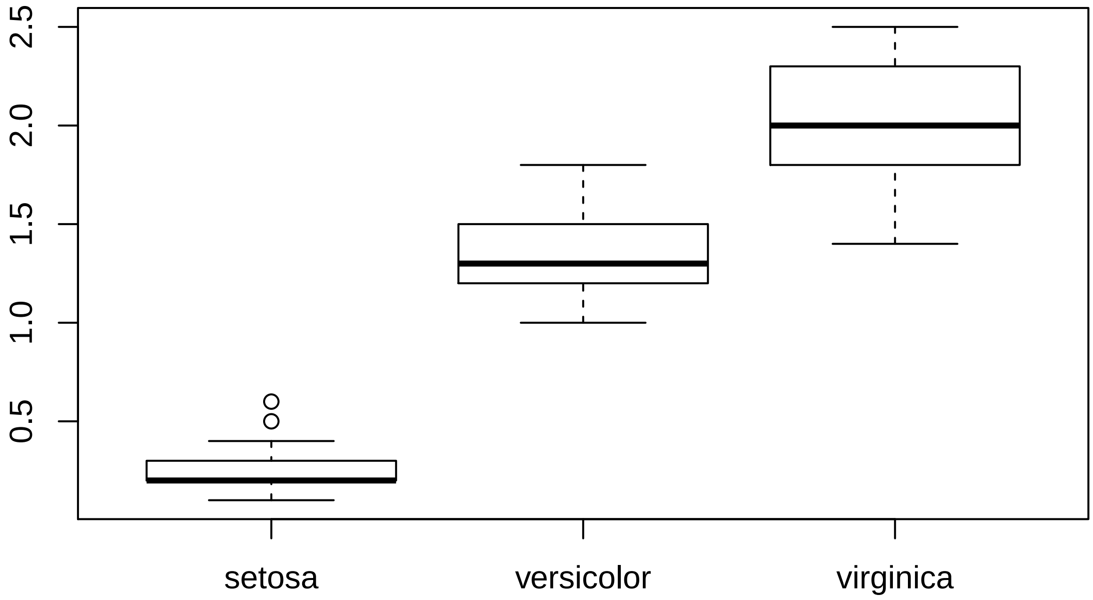
```
#boxplot of petal length vs. species
boxplot(iris$Petal.Length ~ iris$Species)
```



1st, 2nd and 3rd quartiles (25, 50, 75 quantiles)

Let's look at another variable – petal width

```
#boxplot of petal width
boxplot(iris$Petal.Width ~ iris$Species)
```



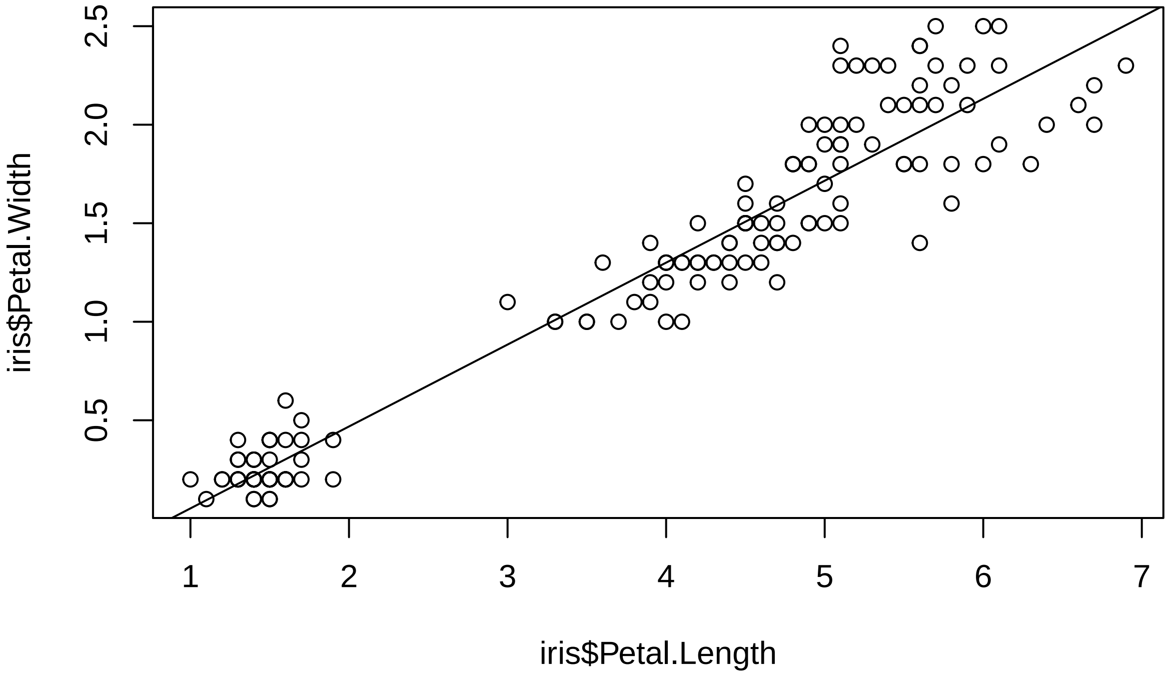
The pattern across species seems similar, are these variables correlated in some way?

Scatterplots

The simplest scatterplots can be constructed using `plot()`.

Examine the `plot()` help file to see how it's used. What do you notice about the way arguments can be supplied?

```
#scatter plot of petal width vs petal length
plot(iris$Petal.Length, iris$Petal.Width)
#add a regression line for fun
abline(lm(iris$Petal.Width ~ iris$Petal.Length))
```



ggplot2

install ggplot2 using ‘install.packages(‘ggplot2’)

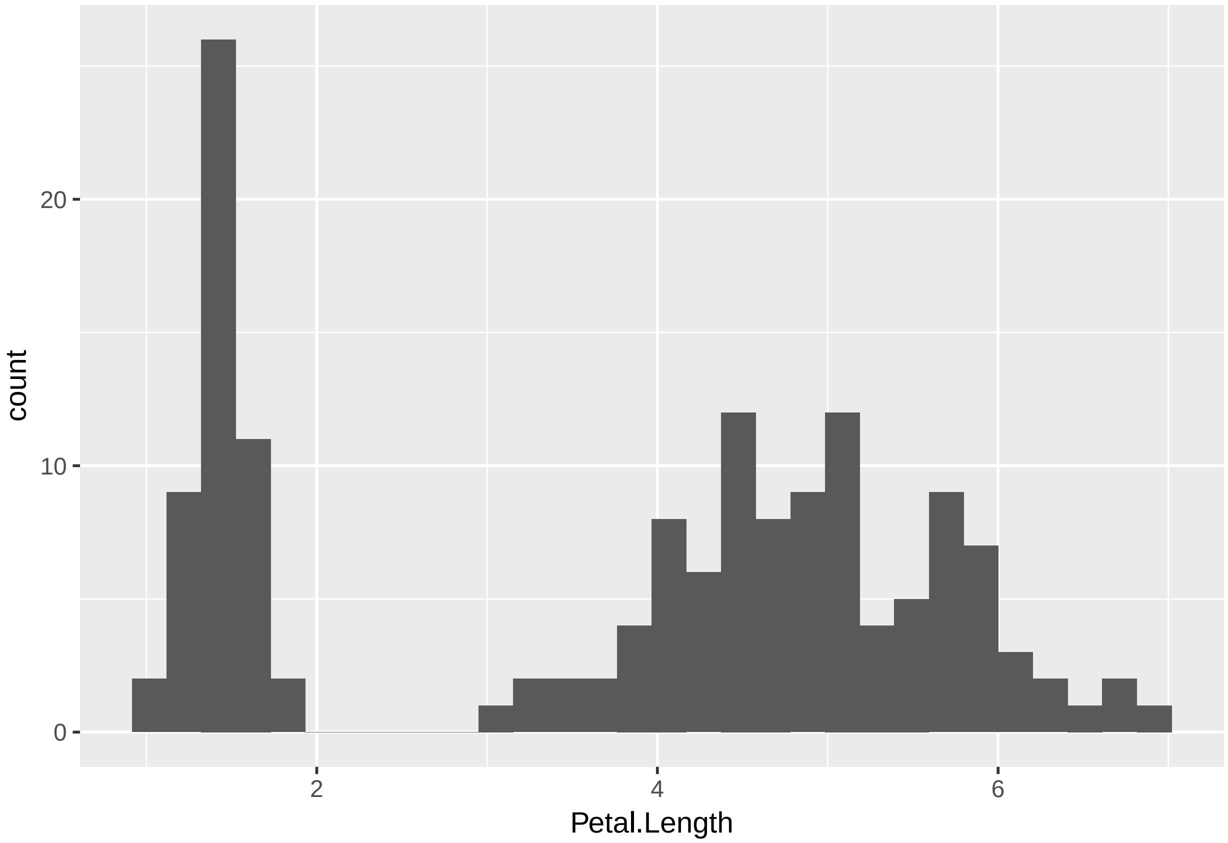
```
#load ggplot2
library(ggplot2)
```

We can recreate our plots from above using ggplot

Histograms

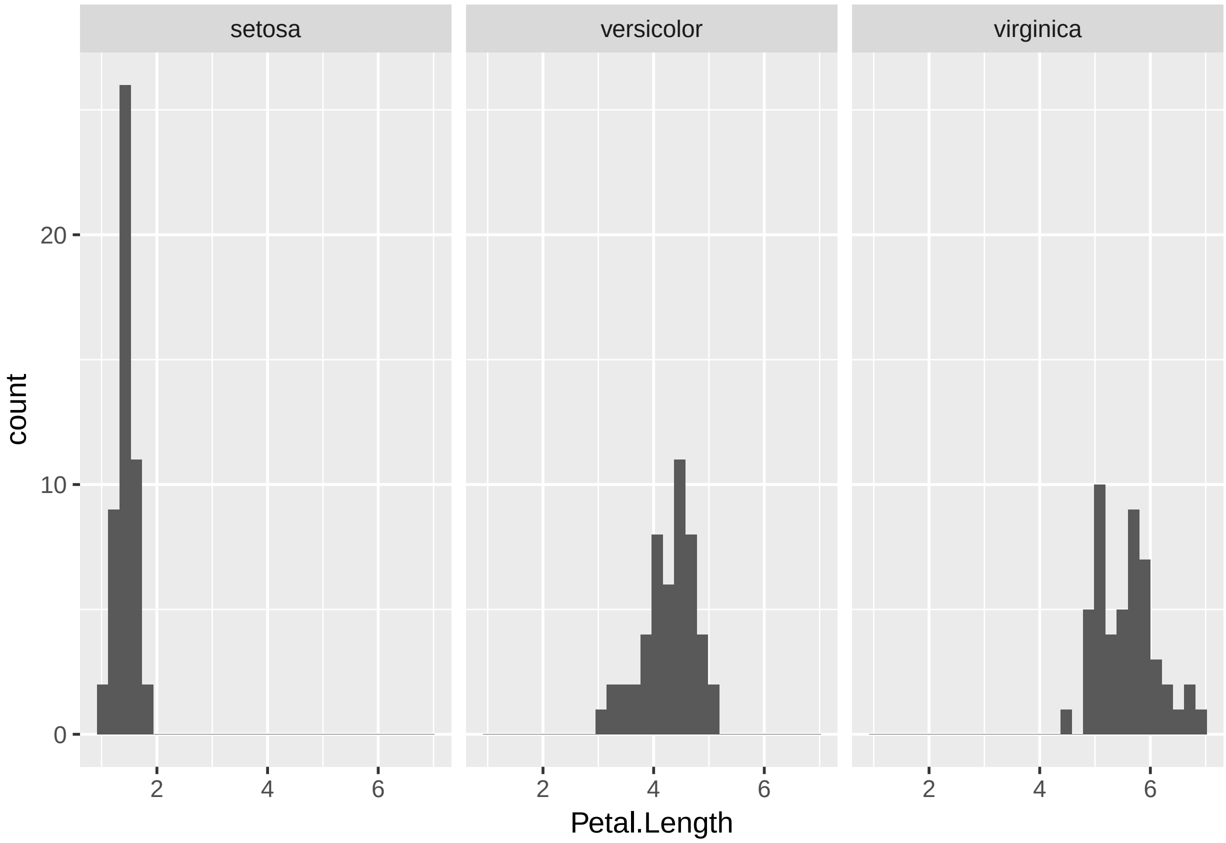
```
ggplot(iris) +
  geom_histogram(aes(x = Petal.Length))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



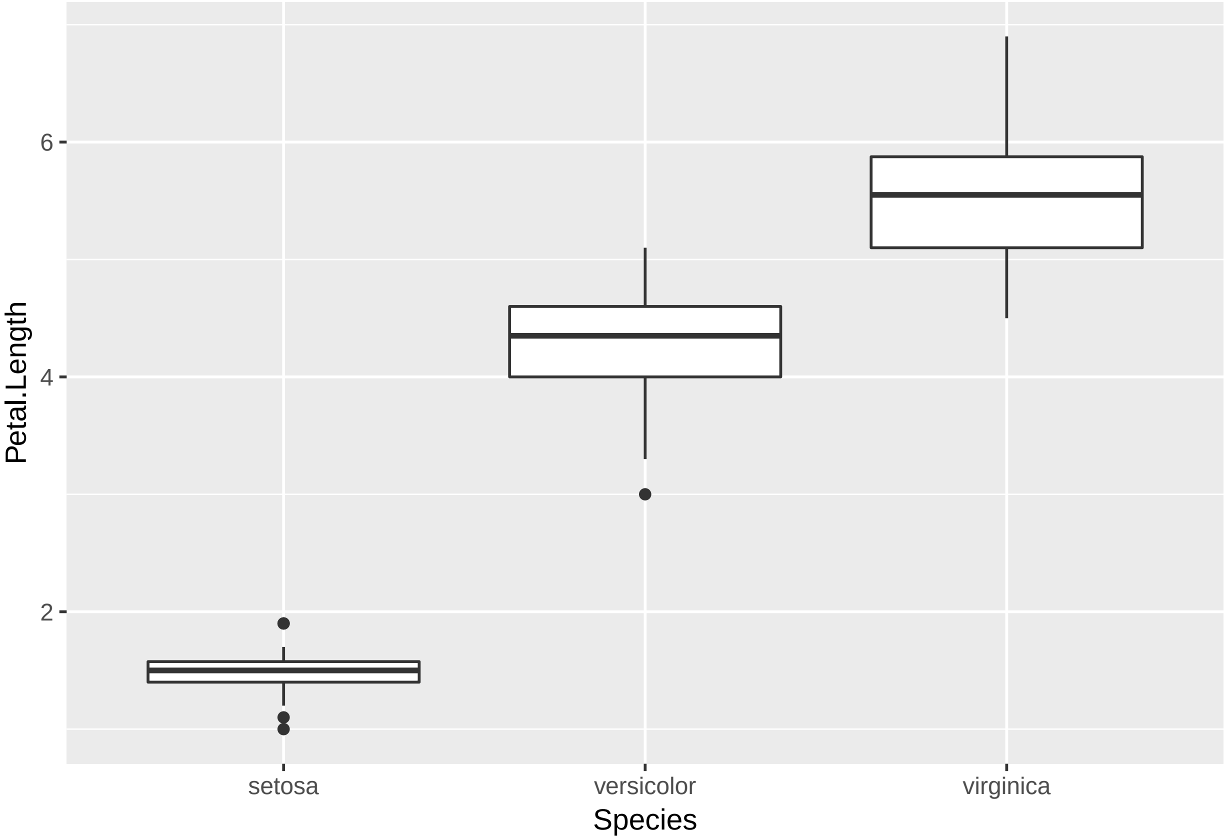
```r
#Divide histograms by species as before using facet_wrap()
ggplot(iris) +
  geom_histogram(aes(x = Petal.Length)) +
  facet_wrap(~Species)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



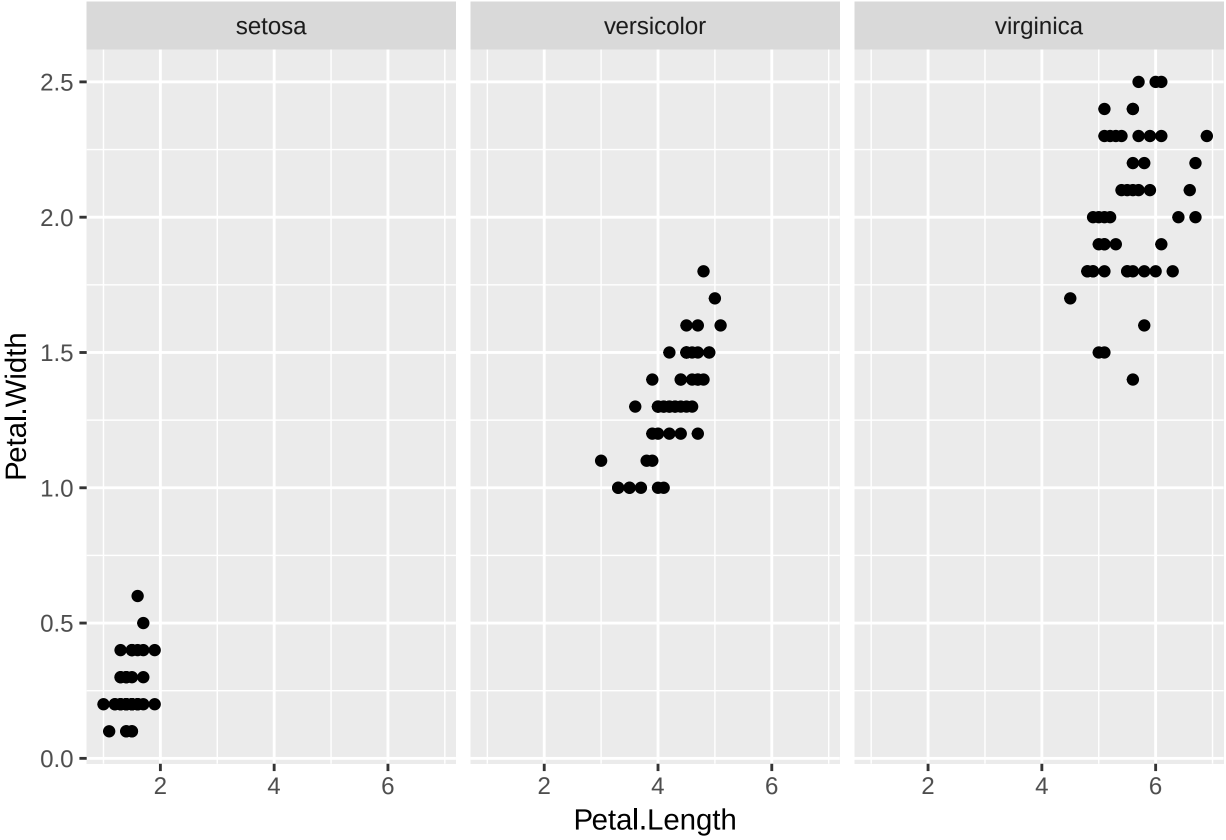
Boxlots

```
ggplot(iris) +
  geom_boxplot(aes(x = Species, y = Petal.Length))
```



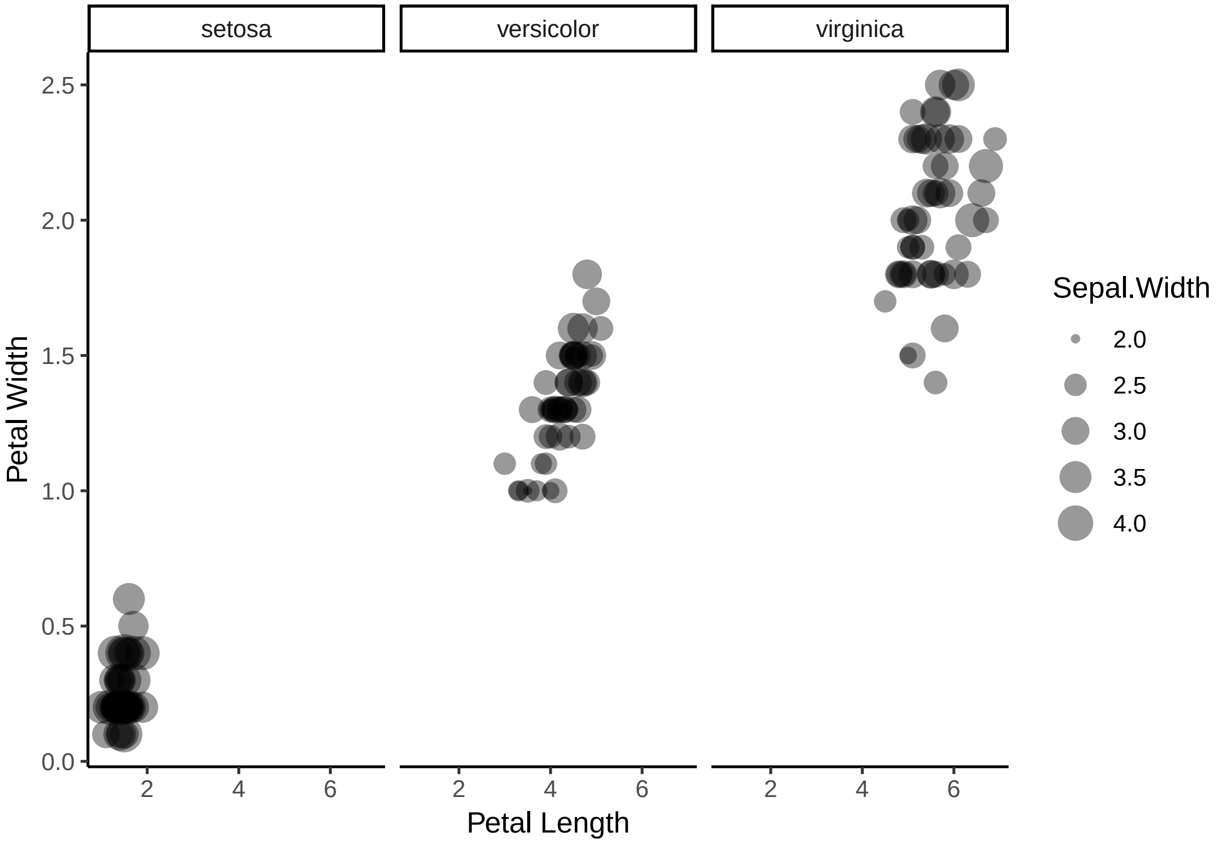
Scatterplots

```
ggplot(iris) +
  geom_point(aes(x = Petal.Length, y = Petal.Width)) +
  facet_wrap(~Species)
```



Using aes(), other variables can be mapped to facets of the plot, like the color, shape or size of points.

```r
ggplot(iris) +
  geom_point(alpha = 0.4, aes(x = Petal.Length, y = Petal.Width, size = Sepal.Width)) +
  facet_wrap(~Species) +
  xlab("Petal Length") + ylab("Petal Width") +
  #theme_ changes facets of the plot that don't map to data
  theme_classic()
```


setosa
versicolor
virginica
Petal Width
2.5
2.0
1.5
1.0
0.5
0.0
2
4
6
2
4
6
2
4
6
Petal Length
Sepal.Width
2.0
2.5
3.0
3.5
4.0