Lecture 13. Basic Data Visualization

R and Data Visualization

BIG2006, Hanyang University, Fall 2022

Data Visualization

Data visualization is an important part of a statistical analysis.

The visualization tools appropriate for a given data set are dependent upon the types of variables for which you have made observations.

Most commonly used data plots and examples will be discussed using both base R graphics and `ggplot2` functionality.

Barplots and Pie Charts

Barplots and pie charts are commonly used to visualize qualitative data by category frequency.

- `barplot`: draws either vertical or horizontal bars to visualize frequencies according to the relevant categories
- `pie`: an alternative option with appropriately sized "slices" representing the relative counts of each categories

Note: Pie charts are a bad way of displaying information. The eye is good at judging linear measures and bad at judging relative areas.

mtcar data: Barplot

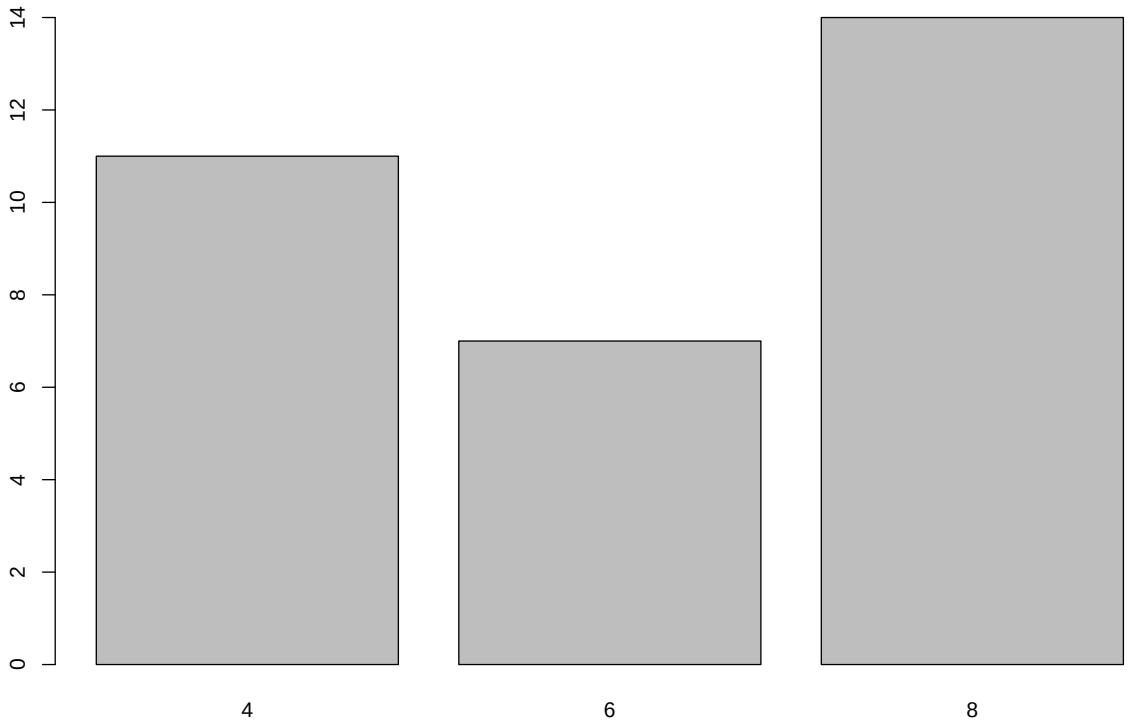
```
mtcars[1:5,]
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2

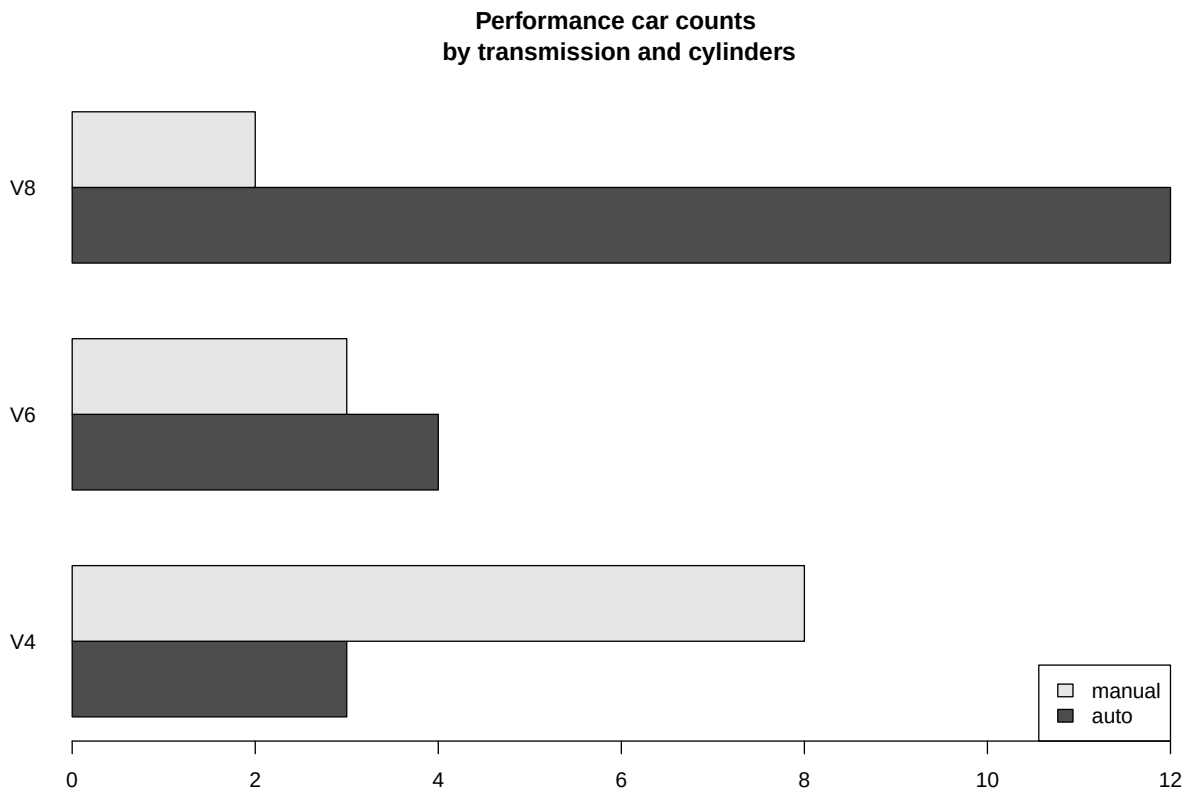
```
cyl.freq <- table(mtcars$cyl)
cyl.freq
```

```
##
##  4  6  8
## 11  7 14
```

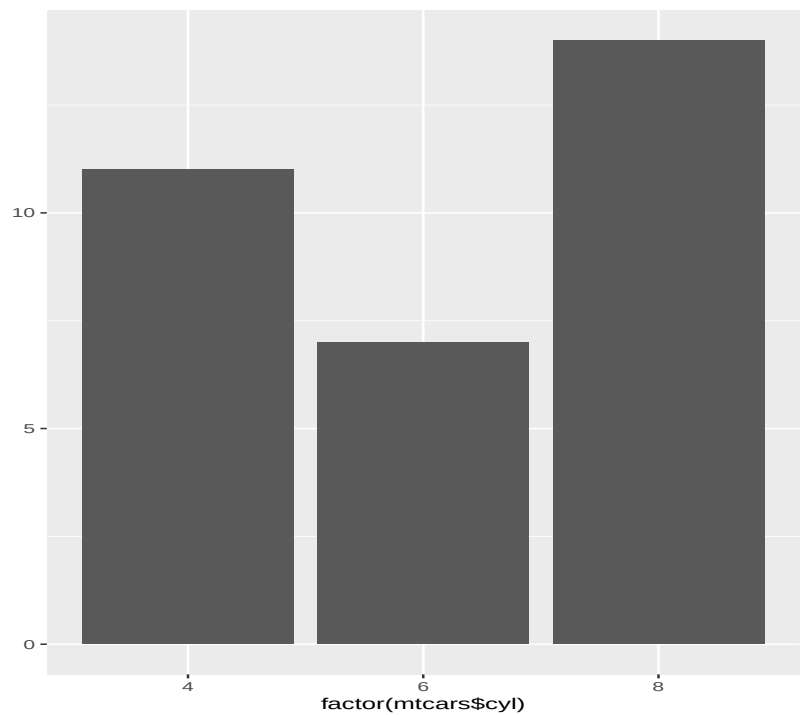
```r
# The number of cylinders in each engine (4, 6, and 8)
barplot(cyl.freq)
```

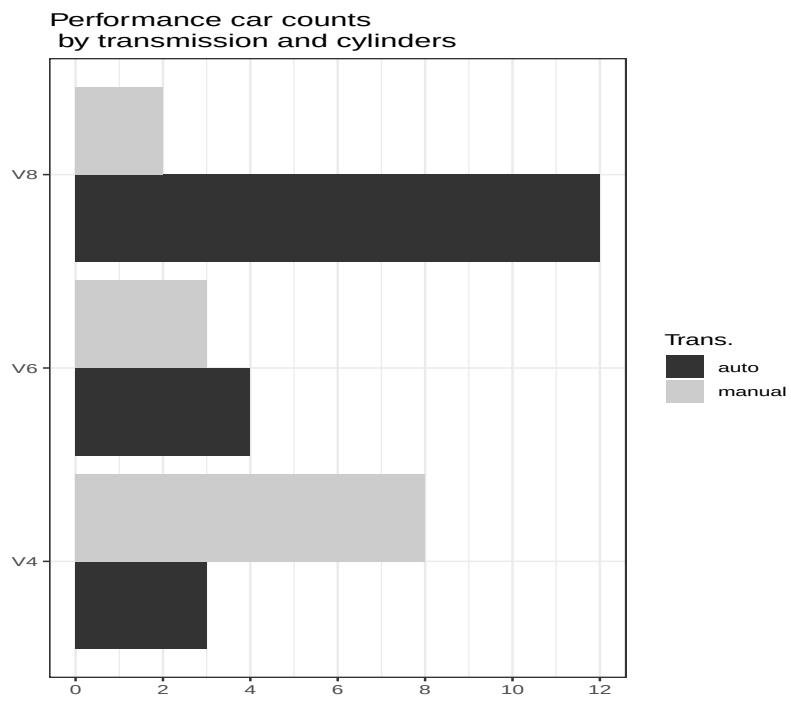


```r
# The number of cylinders and the transmission variable (auto/manual)
cyl.freq.matrix <- table(mtcars$am,mtcars$cyl)
barplot(cyl.freq.matrix,beside=TRUE,horiz=TRUE,las=1,
  main="Performance car counts\n by transmission and cylinders",
  names.arg=c("V4","V6","V8"),legend.text=c("auto","manual"),
  args.legend=list(x="bottomright"))
```



```r
a <- qplot(factor(mtcars$cyl),geom="bar")
b <- qplot(factor(mtcars$cyl),geom="blank",fill=factor(mtcars$am),xlab="",
ylab="",main="Performance car counts\n by transmission and cylinders") +
 geom_bar(position="dodge") + scale_x_discrete(labels=c("V4","V6","V8")) +
 scale_y_continuous(breaks=seq(0,12,2)) + theme_bw() + coord_flip() +
 scale_fill_grey(name="Trans.",labels=c("auto","manual"))
grid.arrange(a,b, nrow=1, ncol=2) # gridExtra package
```



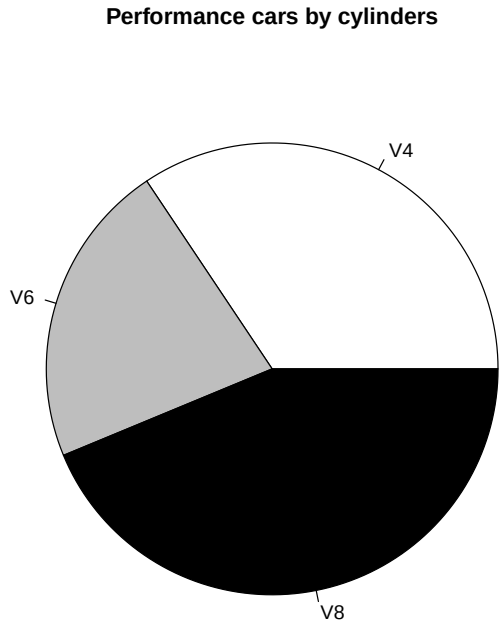


Note a number of new additions to the basic `qplot` setup:

- `scale_x_discrete`: specifies labels for each category
- `scale_y_continuous`: controls the axis labels for the frequencies
- `theme_bw`: changes the visual theme of the image (black-and-white color theme)
- `coord_flip`: flips the axes and provides horizontal bars rather than the default vertical style
- `scale_fill_grey`: forces the color to be grayscale and alters the labels of the automatically generated legend to match at the same time

mtcar data: Pie chart

```
pie(table(mtcars$cyl),labels=c("V4","V6","V8"),
    col=c("white","gray","black"),
    main="Performance cars by cylinders")
```



Histograms

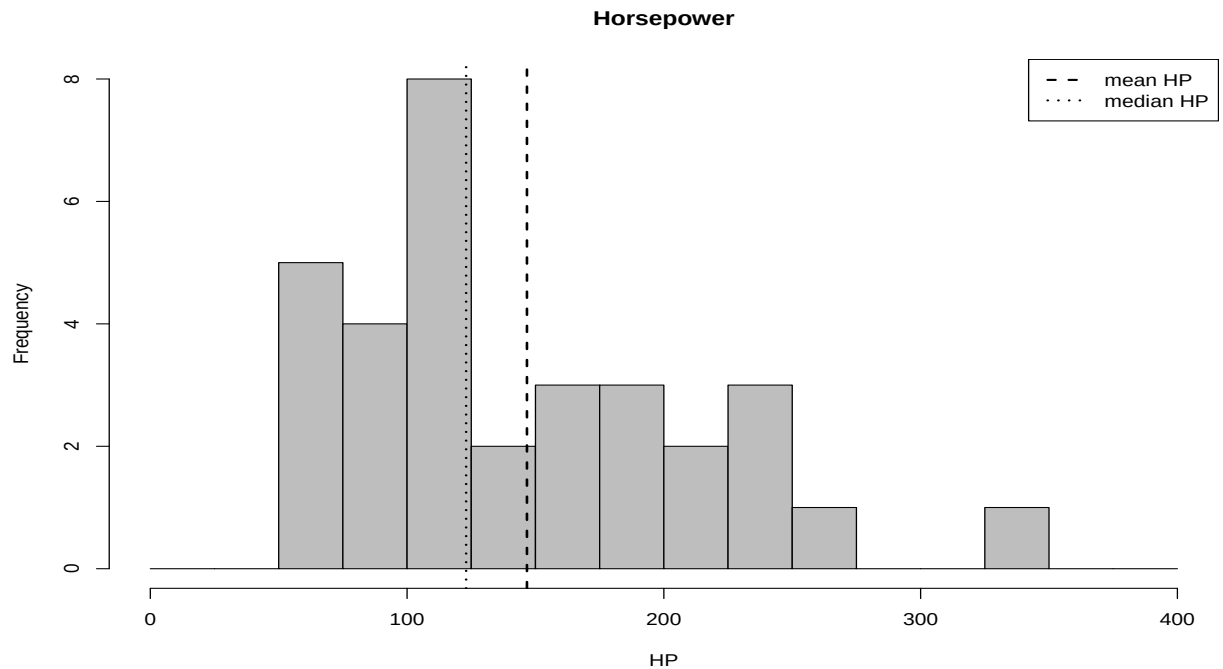
To visualize the distribution of continuous measurements, you can use a $histogram$.

A histogram also measures frequencies, but in targeting a numeric-continuous variable.

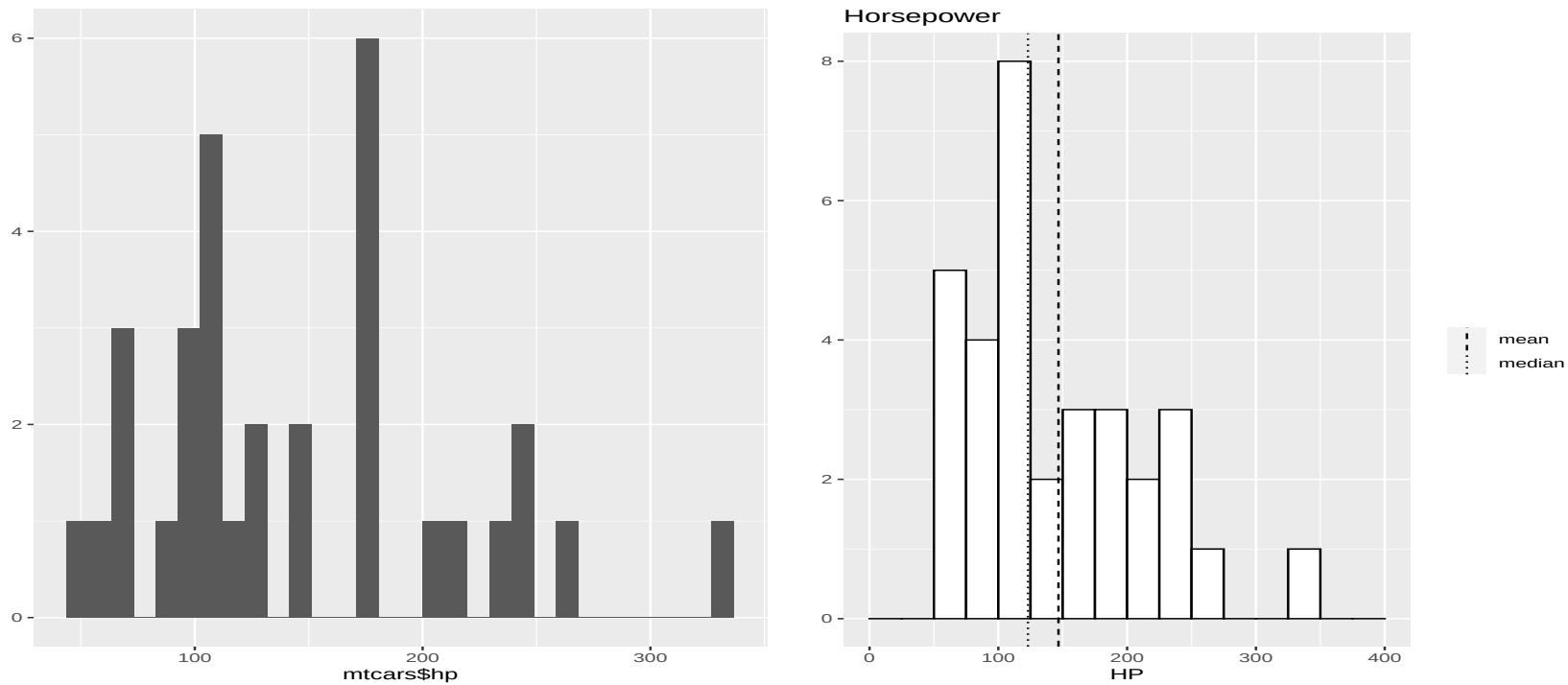
It is first necessary to "bin" the observed data, meaning to define intervals and then count the number of observations that fall within each one. The size of this interval is known as the $binwidth$.

```
mtcars$hp
 [1] 110 110  93 110 175 105 245  62  95 123 123 180 180 180 205 215 230
[18]  66  52  65  97 150 150 245 175  66  91 113 264 175 335 109
```

```
hist(mtcars$hp,breaks=seq(0,400,25),col="gray",main="Horsepower",xlab="HP")
abline(v=c(mean(mtcars$hp),median(mtcars$hp)),lty=c(2,3),lwd=2)
legend("topright",legend=c("mean HP","median HP"),lty=c(2,3),lwd=2)
```



```r
a <- qplot(mtcars$hp)
b <- qplot(mtcars$hp,geom="blank",main="Horsepower",xlab="HP") +
 geom_histogram(color="black",fill="white",breaks=seq(0,400,25),
                closed="right") +
 geom_vline(mapping=aes(xintercept=c(mean(mtcars$hp),median(mtcars$hp)),
            linetype=factor(c("mean","median"))),show.legend=TRUE) +
 scale_linetype_manual(values=c(2,3)) + labs(linetype="")
grid.arrange(a,b, nrow=1, ncol=2) # gridExtra package
```



Box-and-Whisker Plots

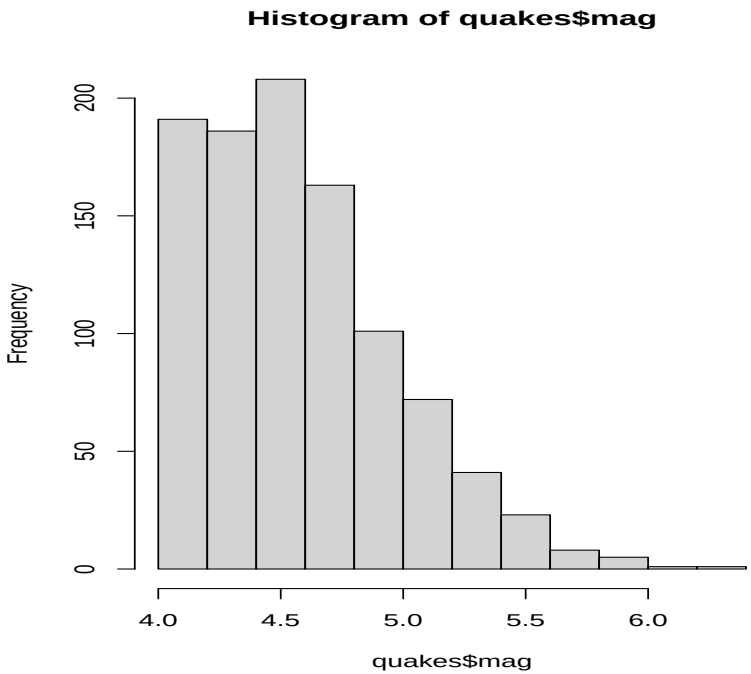
A visual representation of the five-number summary (min, Q1, median, Q3, max)

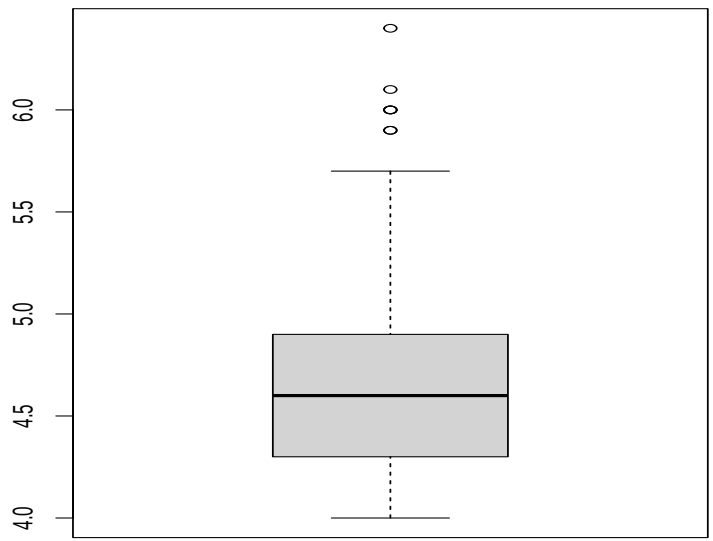
A boxplot shows important features of the distribution, such as overall centrality, spread, and skewness. However, it is not possible to see local features, such as multiple significant peaks in the distribution.

By default, `boxplot` defines an *outlier* as an observation that lies more than 1.5 times the IQR below and lower quartile or above the upper quartile.

quakes data: Stand-Alone Boxplots

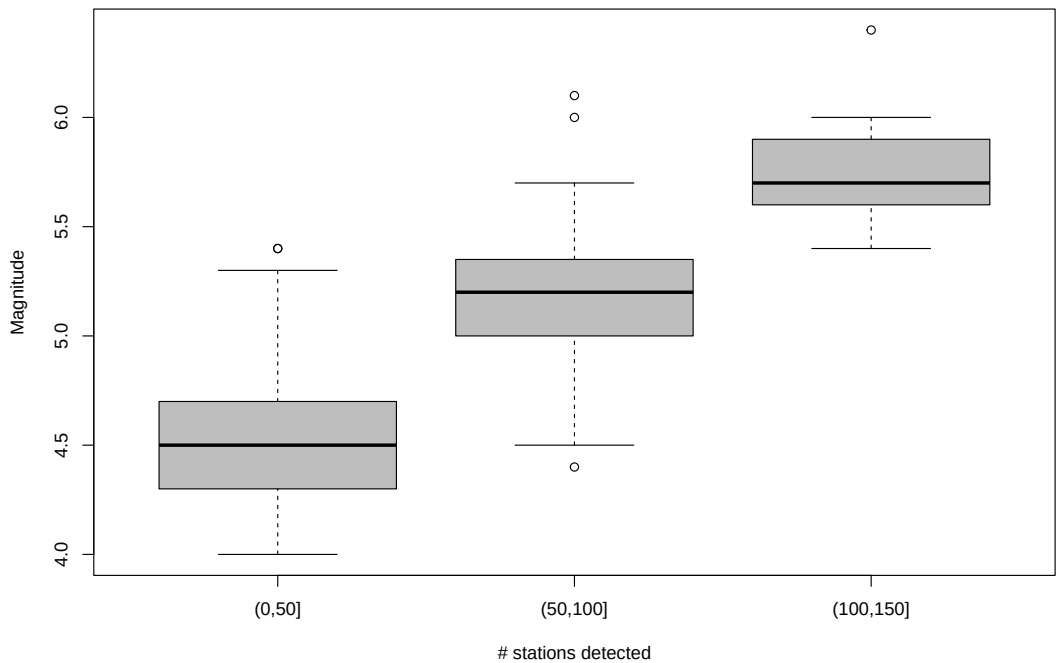
```
par(mfrow=c(1,2))
hist(quakes$mag)
boxplot(quakes$mag)
```



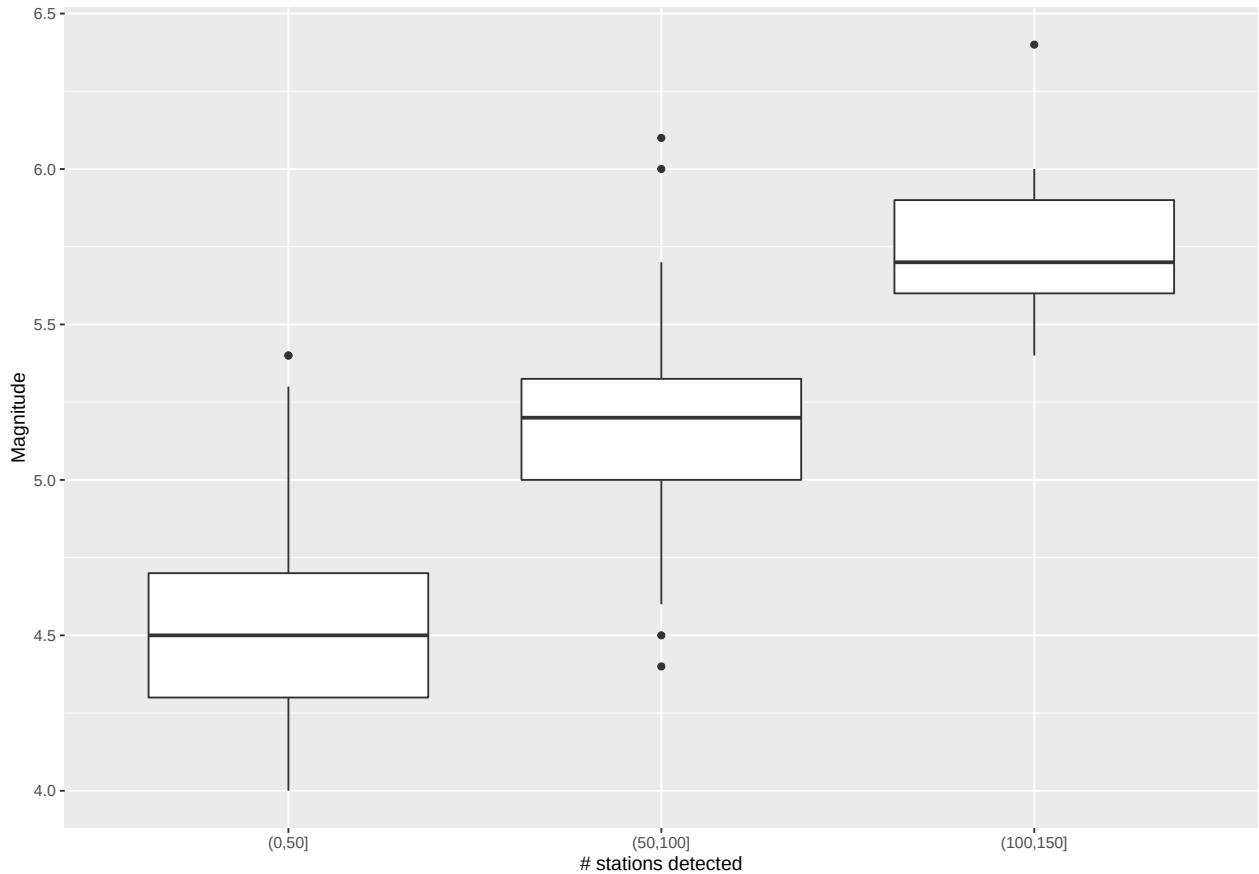


quakes data: Side-by-Side Boxplots

```
stations.fac <- cut(quakes$stations,breaks=c(0,50,100,150))
boxplot(quakes$mag~stations.fac,xlab="# stations detected",
              ylab="Magnitude",col="gray")
```



```
qplot(stations.fac,quakes$mag,geom="boxplot",
          xlab="# stations detected",ylab="Magnitude")
```



Scatterplots

A $scatterplot$ is most frequently used to identify a relationship between the observed values of two different numeric-continuous variables, displayed as x-y coordinate plots.

When there are more continuous variables of interest, we often generate a two-variable scatterplot for each pair of variables and show them together in a structured way; $scatterplotmatrix$.

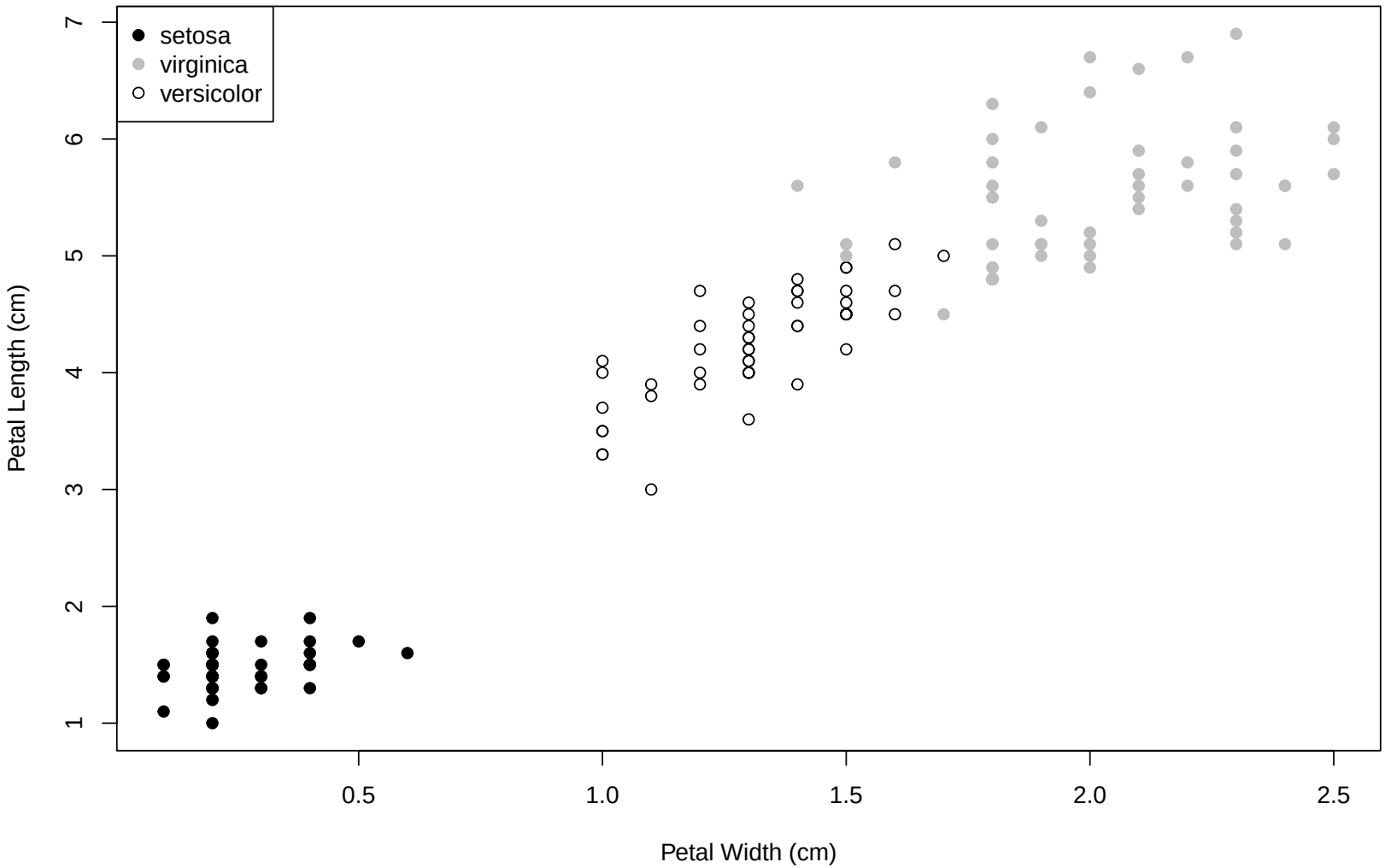
iris data

- Collected in the mid 1930s.
- The data frame of 150 rows and 5 columns consists of petal and sepal measurements for three species of perennial iris flowers, *Iris setosa*, *Iris virginica*, and *Iris versicolor*.

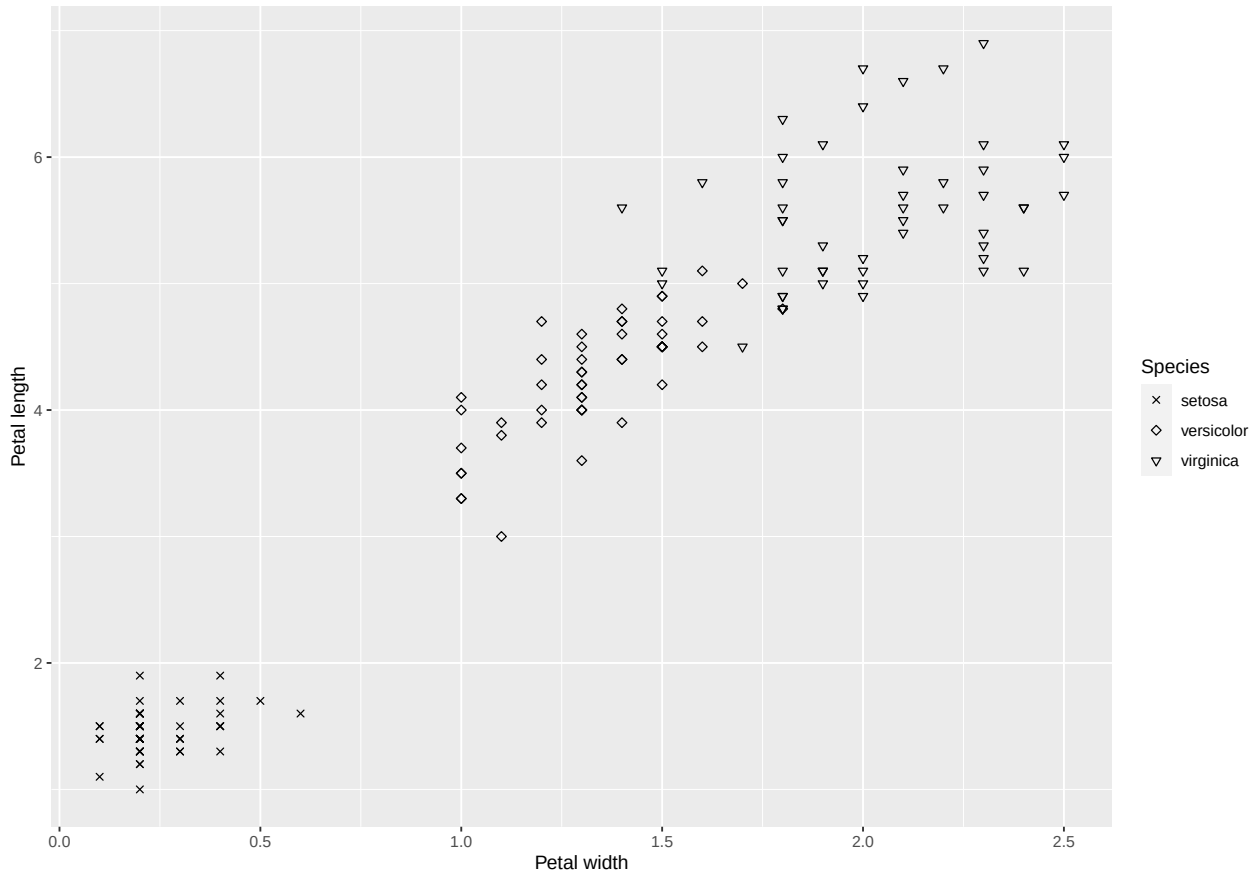
```
iris[1:5,]
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
```

```r
# Method 1: generate empty plot, add points, and alter their style
plot(iris[,4],iris[,3],type="n",xlab="Petal Width",ylab="Petal Length")
points(iris[iris$Species=="setosa",4],
       iris[iris$Species=="setosa",3],pch=19,col="black")
points(iris[iris$Species=="virginica",4],
       iris[iris$Species=="virginica",3],pch=19,col="gray")
points(iris[iris$Species=="versicolor",4],
       iris[iris$Species=="versicolor",3],pch=1,col="black")
legend("topleft",legend=c("setosa","virginica","versicolor"),
       col=c("black","gray","black"),pch=c(19,19,1))
```

```r
# Method 2: set up vectors that specify the desired point style first
iris_pch <- rep(19,nrow(iris)); iris_pch[iris$Species=="versicolor"] <- 1
iris_col <- rep("black",nrow(iris))
iris_col[iris$Species=="virginica"] <- "gray"
plot(iris[,4],iris[,3],col=iris_col,pch=iris_pch,
xlab="Petal Width (cm)",ylab="Petal Length (cm)")
legend("topleft",legend=c("setosa","virginica","versicolor"),
       col=c("black","gray","black"),pch=c(19,19,1))
```

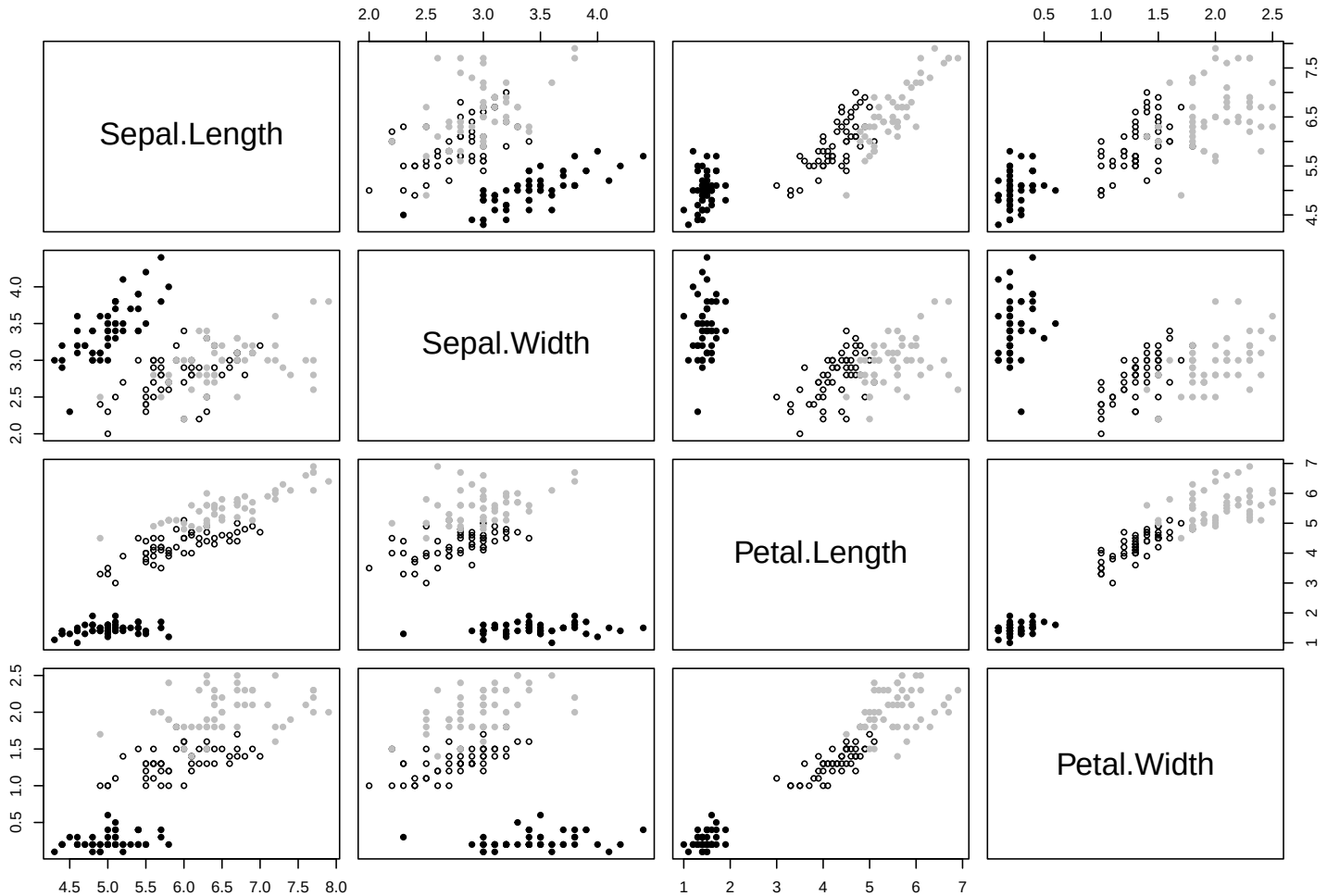

setosa
virginica
versicolor
Petal Length (cm)
Petal Width (cm)
1
2
3
4
5
6
7
0.5
1.0
1.5
2.0
2.5

```
qplot(iris[,4],iris[,3],xlab="Petal width",ylab="Petal length",
      shape=iris$Species) +
  scale_shape_manual(values=4:6) + labs(shape="Species")
```

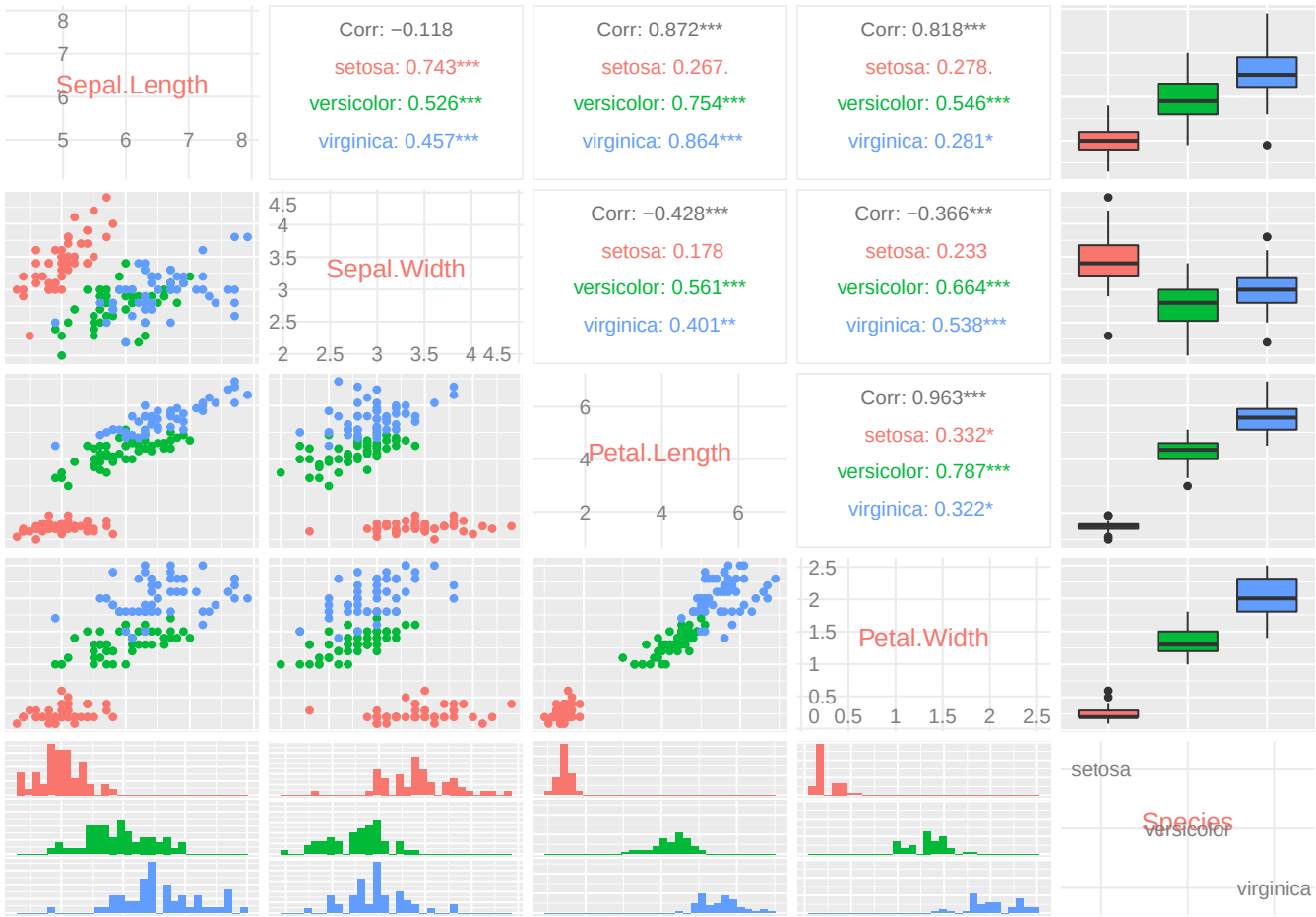



```
# scale_shape_manual: alter ggplot2 point characters
```

```
pairs(iris[,1:4],pch=iris_pch,col=iris_col,cex=0.75)
```



```r
# GGally package
ggpairs(iris,mapping=aes(col=Species),axisLabels="internal")
```



Reference

- Davies, T. M. The Book of R. No Starch Press. Chapter 14.