Basic Plotting in R

Xuemao Zhang
Department of Mathematics
East Stroudsburg University

June 24, 2019

Outline

- Preview
- Bar charts
- Colors in R
- Pie charts
- Density plots: histograms and kernel density plots
- Box plots
- Scatterplots and Scatter plot matrix
- Line charts
- Saving a plot

Install the following package if you don't have it.

```
install.packages("grDevices");
```

R graphics preview

- R has strong graphic capabilities. `plot()` is a generic function for plotting of R objects.
- There are many plot functions which are specific to some tasks.
- Titles, legends and annotations.
  - `main` gives the main title, `sub` the subtitle.
  - `legend()`. The position can be “bottomleft”, “bottomright”, “topleft”, “topright” or exact coordinates.
  - `xlab` specifies the X-axis label; `ylab` specifies the Y-axis label.
  - `xlim` specifies the range of the X-axis; `ylim` specifies the range of the Y-axis.
  - `mtext()` puts some texts in the margin. The margin can be at the bottom (1), the left (2), the top (3) or the right (4).
  - `text()`puts Text in the graph.
  - We can add mathematical symbols using `expression()`.
  - The type of a plot can be : `n`(none), `p`(points), `l`(lines) etc.
  - For more information, type `?title` and `?text` in R console.

Bar chart

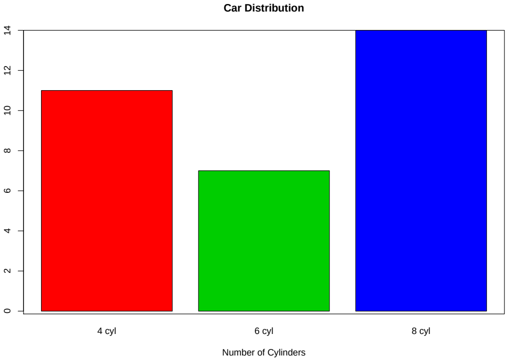
- Bar chart is for categorical data.
- `barplot()` specifies the height of each bar and (optionally) a vector of labels for each bar.

```
counts= table(mtcars$cyl);
counts;  #get the count of 4, 6 & 8 cylinder cars
```

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

Bar chart

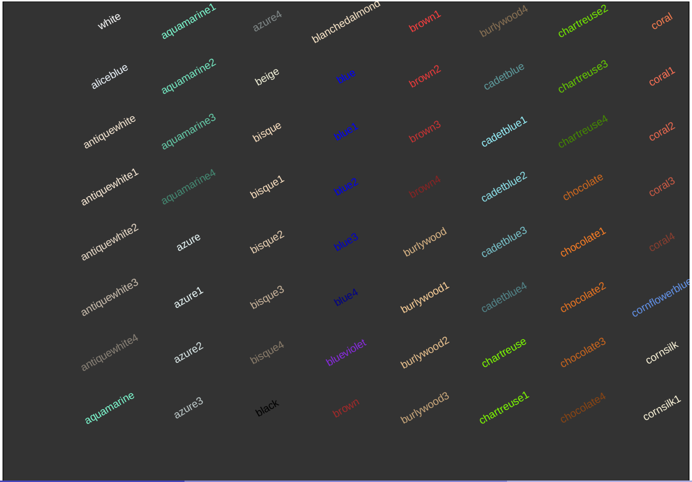
```r
barplot(counts,main="Car Distribution", xlab="Number of Cylinders",
        col=c(2,3,4),names.arg=c("4 cyl", "6 cyl", "8 cyl"));
# col is used to specified colors
box(); #draw a box around the plot
```



Colors in R

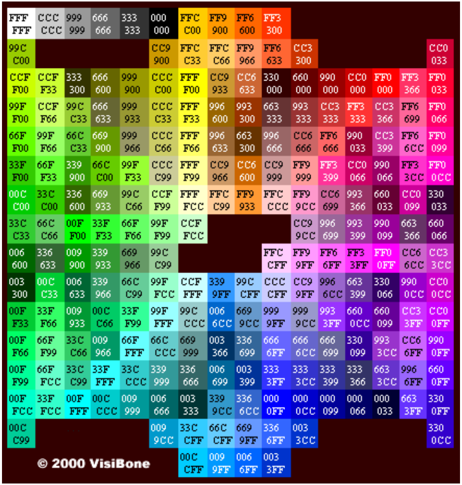
- In R, colors can be specified either by name (e.g col = "red"), integers or hexadecimal colors (#rrggbb). The following shows the first sixty-four color names.

```
## Loading required package: grid
```



Colors in R

- Colors can be specified using hexadecimal color code, #rrggbb, where rr, gg, and bb refer to color intensity in the red, green, and blue channels, respectively. For more information, see http://stat545.com/block018_colors.html.



FFF FFF	CCC CCC	999 999	666 666	333 333	000 000	FFC C00	FF9 900	FF6 600	FF3 300						
99C C00					CC9 900	FFC C33	FFC C66	FF9 966	FF6 633	CC3 300					CC0 033
CCF F00	CCF F33	333 300	666 600	999 900	CCC C00	FFF F00	CC9 933	CC6 633	330 000	660 000	990 000	CC0 000	FF0 000	FF3 366	FF0 033
99F F00	CCF F66	99C C33	666 633	999 933	CCC C33	FFF F33	996 600	993 300	663 333	993 333	CC3 333	FF3 333	CC3 366	FF6 699	FF0 066
66F F00	99F F66	66C C33	669 900	999 966	CCC C66	FFF F66	996 633	663 300	996 666	CC6 666	FF6 666	990 033	CC3 399	FF6 6CC	FF0 099
33F F00	66F F33	339 900	66C C00	99F F33	CCC C99	FFF F99	CC9 966	CC6 600	CC9 999	FF9 999	FF3 399	CC0 066	990 066	FF3 3CC	FF0 0CC
00C C00	33C C00	336 600	669 933	99C C66	CCF F99	FFF FCC	FFC C99	FF9 933	FFC CCC	FF9 9CC	CC6 699	993 366	660 033	CC0 099	330 033
33C C33	66C C66	00F F00	33F F33	66F F66	99F F99	CCF FCC				CC9 9CC	996 699	993 399	990 099	663 366	660 066
006 600	336 633	009 900	339 933	669 966	99C C99				FFC CFF	FF9 9FF	FF6 6FF	FF3 3FF	FF0 0FF	CC6 6CC	CC3 3CC
003 300	00C C33	006 633	339 966	66C C99	99F FCC	CCF FFF	339 9FF	99C CFF	CCC CFF	CC9 9FF	996 6CC	663 399	330 066	990 0CC	CC0 0CC
00F F33	33F F66	009 933	00C C66	33F F99	99F FFF	99C CCC	006 6CC	669 9CC	999 9FF	999 9CC	993 3FF	660 0CC	660 099	CC3 3FF	CC0 0FF
00F F66	66F F99	33C C66	009 966	66F FFF	66C CCC	669 999	003 366	336 699	666 6FF	666 6CC	666 699	330 099	993 3CC	CC6 6FF	990 0FF
00F F99	66F FCC	33C C99	33F FFF	33C CCC	339 999	336 666	006 699	003 399	333 3FF	333 3CC	333 399	333 366	663 3CC	996 6FF	660 0FF
00F FCC	33F FCC	00F FFF	00C CCC	009 999	006 666	003 333	339 9CC	336 6CC	000 0FF	000 0CC	000 099	000 066	000 033	663 3FF	330 0FF
00C C99					009 9CC	33C CFF	66C CFF	669 9FF	336 6FF	003 3CC					330 0CC
						00C CFF	009 9FF	006 6FF	003 3FF						

© 2000 VisiBone

Colors in R

- In R you can call colors by their numbers. The `palette()` function within the `grDevices` library allows a table of colors to be referenced by a numeric index. The default color palette is

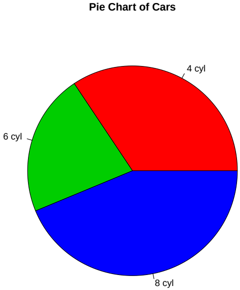
```
1 = "black"
2 = "red"
3 = "green"
4 = "blue"
5 = "cyan"
6 = "magenta"
7 = "yellow"
8 = "gray"
```

- To set these colors as parameters, simply use the index.

Pie charts

- Pie charts are for categorical data.
- A pie chart presents each category as a slice of a circle so that each slice has a size that is proportional to the whole in each category.
- Pie charts are not recommended since people are able to judge height more accurately than area.

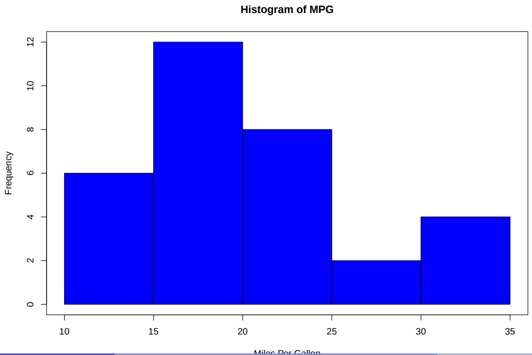
```
counts= table(mtcars$cyl); #create a frequency table
pie(counts,main="Pie Chart of Cars",
        col=c(2,3,4), labels=c("4 cyl", "6 cyl", "8 cyl"));
```



Histogram

- Histogram is for numerical data.
- A histogram shows a partition of a data set and the number of observations in each class.

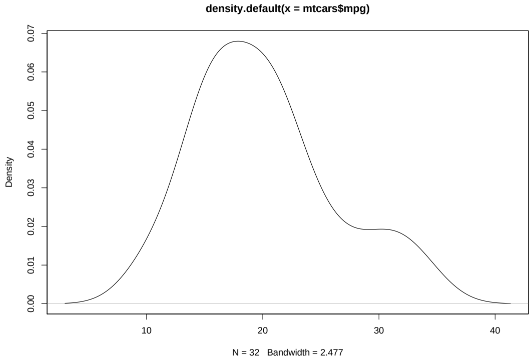
```
hist(mtcars$mpg,  col="blue", xlab="Miles Per Gallon",#breaks = 10,
    main="Histogram of MPG");
#we can use breaks argument to determine the number of bins
box();
```



Kernel density plot

- In general, we assume that a data set is from a larger population.
- Kernel density is an **estimate** of the distribution of the variable.

```
d = density(mtcars$mpg); # returns the density data
plot(d); # plots the results
box();
```

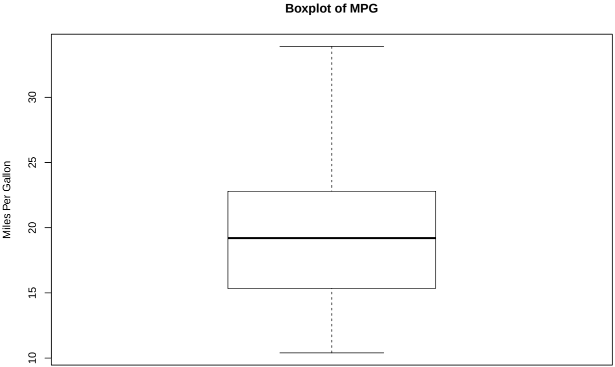


Box plot

- A box plot (or box and whisker plot) shows the five number summary: Min, Q_1,Median, Q_3, Max and outliers
  - Min is the minimum value in the data.
  - 25% observations in the sorted data are less than Q_1.
  - Mdeian is the absolute center.
  - 75% observations in the sorted data are less than Q_3.
  - $Q_3 - Q_1$ is called IQR (Interquartile Range).
  - Max is the maximum value in the data.
  - Outliers are data points far away from other data values.
  - Any values less than $Q1 - 1.5 \times IQR$ or greater than $Q3 + 1.5 \times IQR$ are defined as outliers.
  - The whiskers extend only as far as the minimum data value that is not an outlier and the maximum data value that is not an outlier.
- Boxplots can be created for individual variables or for variables by group using the function `boxplot`.

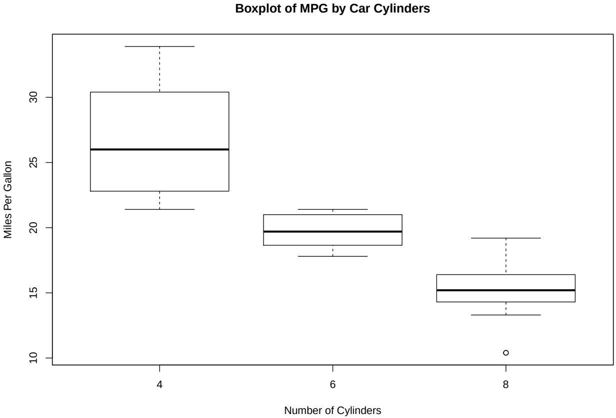
Box plot

```
boxplot(mtcars$mpg, main="Boxplot of MPG",
        ylab="Miles Per Gallon"); # Boxplot of MPG
box();
```



Box plot

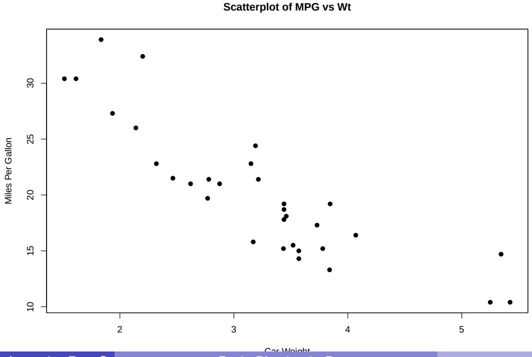
```
boxplot(mpg~cyl,data=mtcars,main="Boxplot of MPG by Car Cylinders",
    xlab="Number of Cylinders", ylab="Miles Per Gallon");
box();
```



Scatter plot

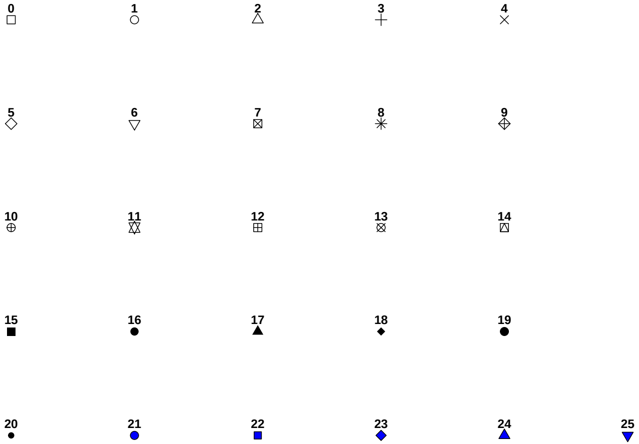
- A scatter plot is used to show the relationship between two variables.
- Each data point with two measurements is plotted on the Cartesian (x,y) plane.

```
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
   xlab="Car Weight", ylab="Miles Per Gallon", pch=19);
#pch is used to specify symbols to use;
#type ?pch for more information
box();
```



Scatter plot

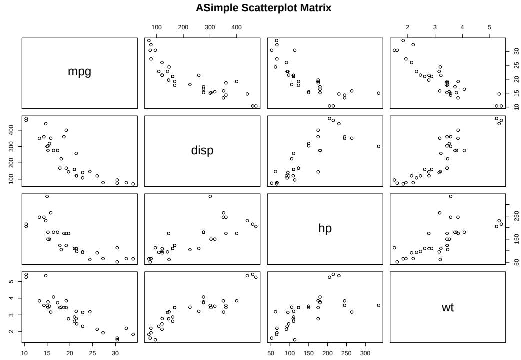
- Points shapes available in R:



Scatter plot matrix

- To see the relationship between any two variables, it is useful to look at the scatter plot matrix.
- pairs() function creates beautiful scatter plot matrix.

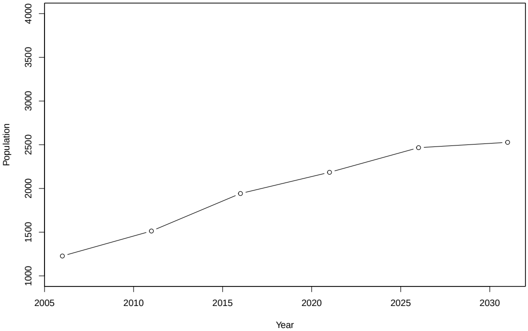
```
pairs(~mpg+disp+hp+wt,data=mtcars,
   main="ASimple Scatterplot Matrix");
```



Line chart

- Line chart is just a scatter plot by specifying specify type = "b" for points joined line or type = "l" for line.

```
y = c(1227.3, 1513.1, 1942.1, 2184.7, 2466.6, 2527.6);
x = c(2006, 2011, 2016, 2021, 2026, 2031);
plot(x,y, type="b", xlab="Year", ylab="Population",
     xlim=c(2006, 2031), ylim=c(1000, 4000));
box();
```



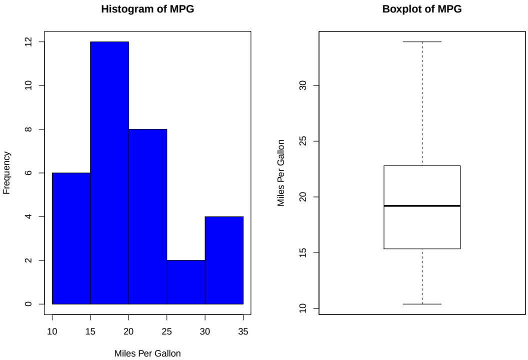
par() function

- par() defines the default settings for plots such as fonts, colors, axes, titles.
- The format is `par(optionname=value, optionname=value, ...)`
- If you set parameter values using par(), the changes will be in effect for the rest of the session or until you change them again.
- type ?par in R console to see more information.
- Especially, par() can be used to put multiple graphs in a single plot. The syntax to set the plotting area into a $m \times n$ matrix is

par(mfrow=c(m,n))

par() function

```
par(mfrow=c(1,2));
hist(mtcars$mpg,  col="blue", xlab="Miles Per Gallon",#breaks = 10,
    main="Histogram of MPG");
box();
boxplot(mtcars$mpg, main="Boxplot of MPG",
        ylab="Miles Per Gallon"); # Boxplot of MPG
box();
```



Saving a plot

- All graphs we plot in R programming are displayed on the screen by default.
- The graphs can be saved manually using the R menu “File->Save as. . . ”. But a lot of times it is not convenient.
- We can save plots as a file on disk with the help of built-in functions.
- We need to call the function `dev.off()` after all the plotting, to save the file and return control to the screen.

Saving a plot

- To save a plot as jpeg image we need the `jpeg()` function.

```r
#setwd("F:/DataCamp/Day1"); #set/change the working directory
jpeg(file="scatterplot1.jpeg")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19);
box();
dev.off();
```

```
## pdf
##   2
```

Saving a plot

- To save a plot as png image we need the `png()` function.

```
png(file="scatterplot2.png")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19);
box();
dev.off();
```

```
## pdf
##   2
```

Saving a plot

- To save a plot as bmp image we need the bmp() function.

```
bmp(file="scatterplot3.bmp")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19);
box();
dev.off();
```

```
## pdf
##   2
```

Saving a plot

- To save a plot as tiff format we need the `tiff()` function.

```
tiff(file="scatterplot4.tiff")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19);
box();
dev.off();
```

```
## pdf
##   2
```

Saving a plot

- We can save our plots as **vector image** in pdf or postscript formats.
- The beauty of vector image is that it is easily resizable. Zooming on the image will not compromise its quality.
- To save a plot as pdf format we need the `pdf()` function.

```
pdf(file="scatterplot5.pdf")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19);
box();
dev.off();
```

```
## pdf
##   2
```

Saving a plot

- To save a plot as ps(postscript) or eps(encapsulated postscript) format we need the `postscript()` function.

```
postscript(file="scatterplot6.eps")
plot(mtcars$wt, mtcars$mpg, main="Scatterplot of MPG vs Wt",
    xlab="Car Weight", ylab="Miles Per Gallon", pch=19);
box();
dev.off();
```

```
## pdf
##   2
```

Questions?

?