

蘇州大學

生物信息学系  
Department of Bioinformatics

---

## ***Lab 3 Basic plot***

Xiaoqin Yang  
[yangxiaoqin@suda.edu.cn](mailto:yangxiaoqin@suda.edu.cn)

# Tools we use to generate statistics graph?

---

- Excel



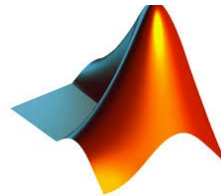
- SAS



- SigmaPlot



- Matlab



- R



- ....

# Why we use R?

---

- It's free.
- more flexible.
- It could be integrated into analyzing pipeline.
- Set almost everything based on your personal needs.

# Creating a Scatter Plot

---

- **Problem**

- You have paired observations:  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ .  
You want to create a scatter plot of the pairs

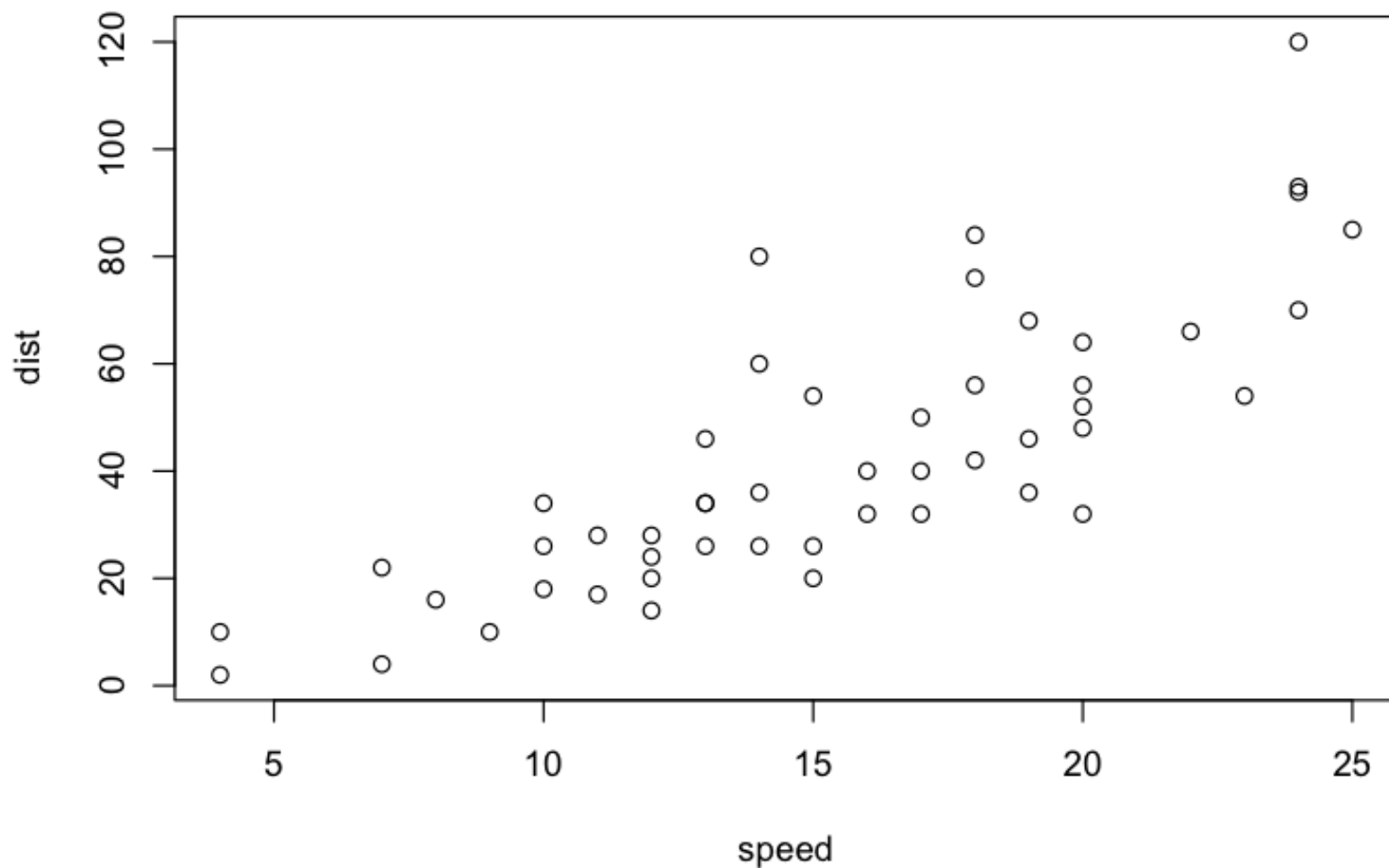
- **Solution**

- If your data are held in two parallel vectors, `x` and `y`, then use them as arguments of **plot**:  
    `> plot(x, y)`
- If your data is held in a (two-column) data frame, plot the data frame:  
    `> plot(dfrm)`

# Creating a Scatter Plot

---

```
> plot(cars)
```



# Adding a Title and Labels

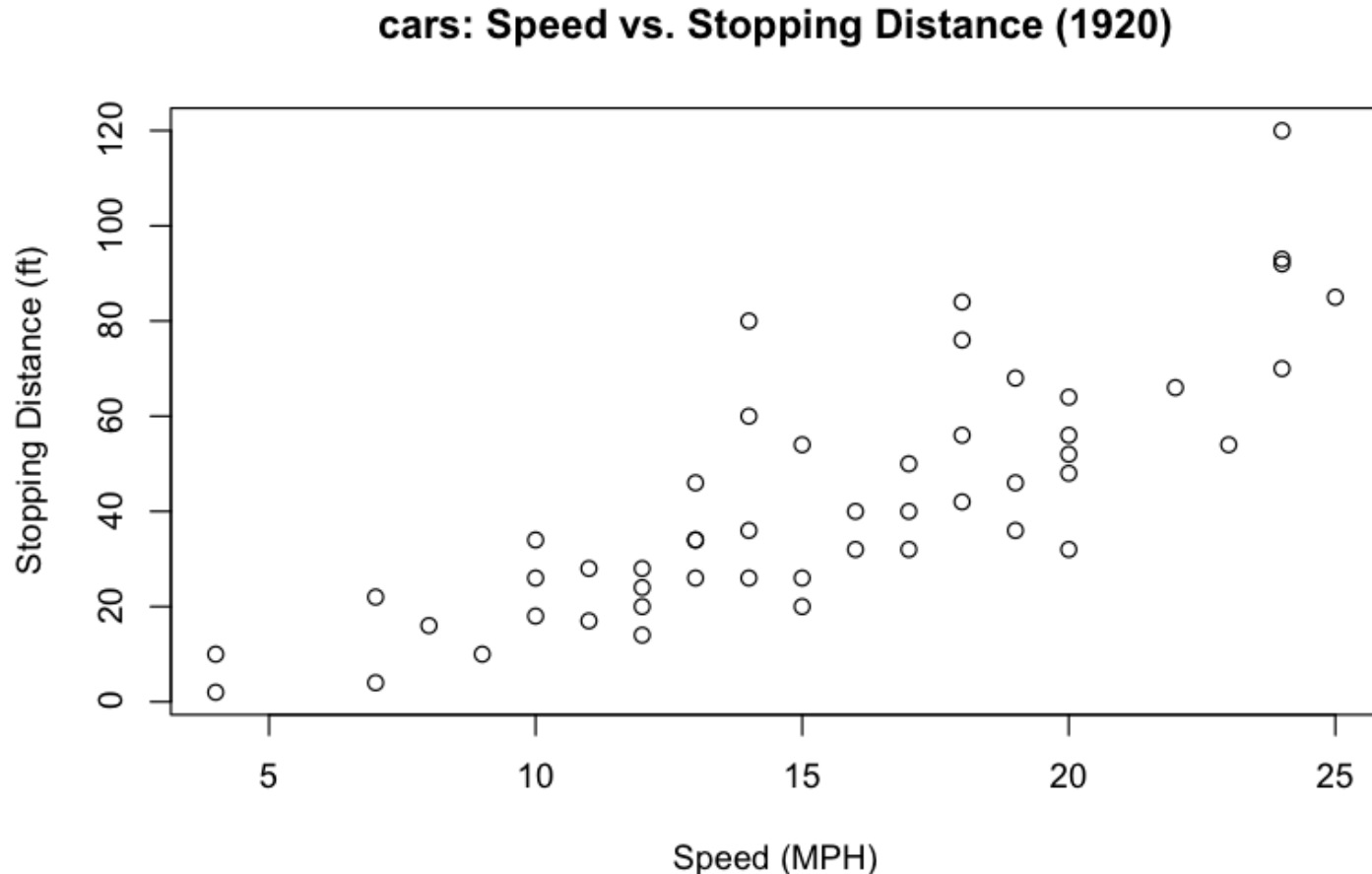
---

- **Problem**
  - You want to add a title to your plot or add labels for the axes.
- **Solution**
  - When calling **plot**:
    - Use the **main** argument for a title;
    - Use the **xlab** argument for an x-axis label;
    - Use the **ylab** argument for a y-axis label.

# Adding a Title and Labels

---

```
> plot(cars, main="cars: Speed vs. Stopping Distance (1920)",  
+      xlab="Speed (MPH)", ylab="Stopping Distance (ft)")
```



# Adding a Grid

---

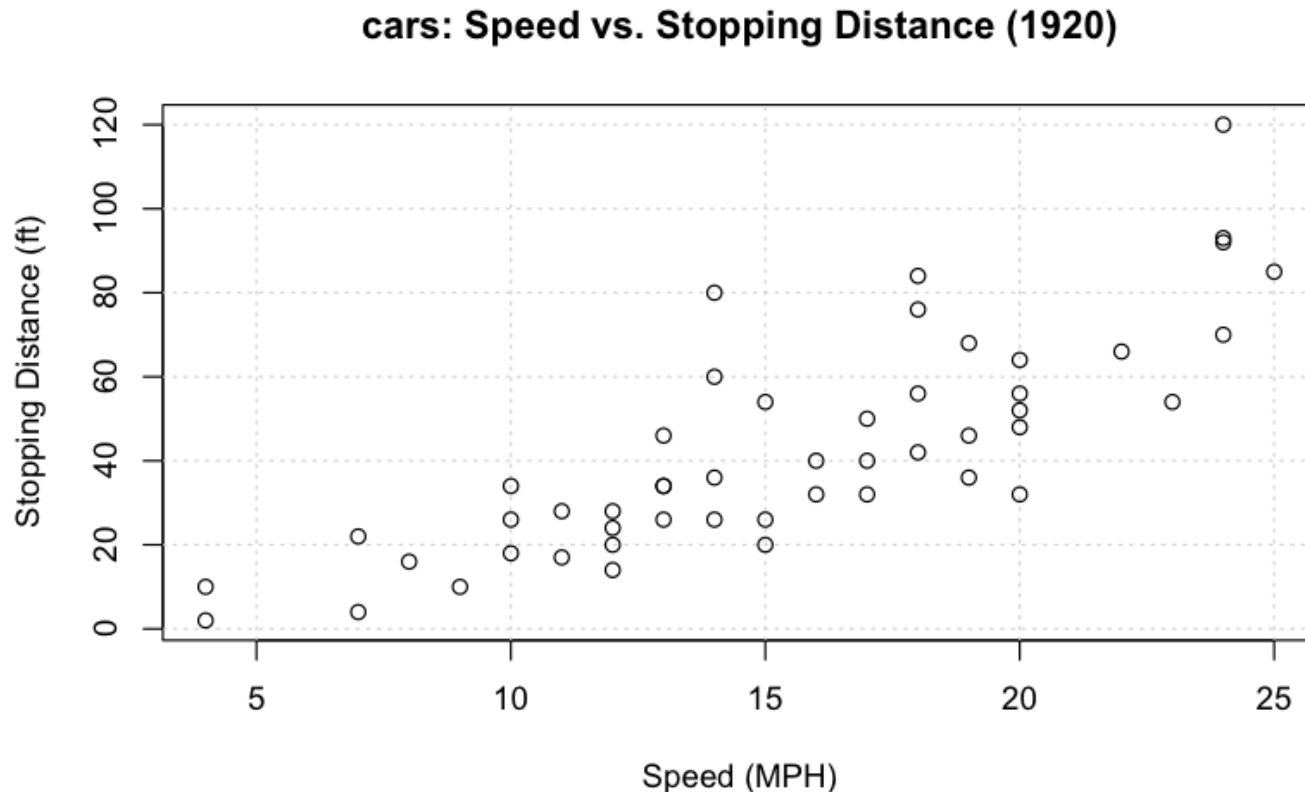
- **Problem**
  - You want to add a grid to your graphic.
- **Solution**
  - Call **plot** with **type="n"** to initialize the graphics frame without displaying the data.
  - Call the **grid** function to draw the grid.
  - Call low-level graphics functions, such as **points** and **lines**, to draw the graphics overlaid on the grid.



# Adding a Grid

---

```
> plot(cars, main="cars: Speed vs. Stopping Distance (1920)",  
+ xlab="Speed (MPH)", ylab="Stopping Distance (ft)", type="n")  
> grid()  
> points(cars)
```



# Creating a Scatter Plot of Multiple Groups

---

- **Problem**

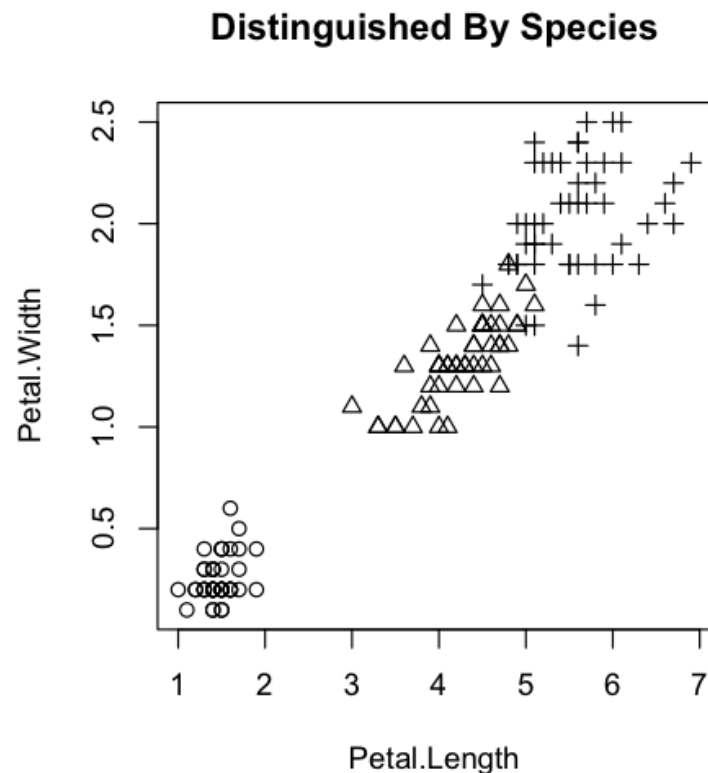
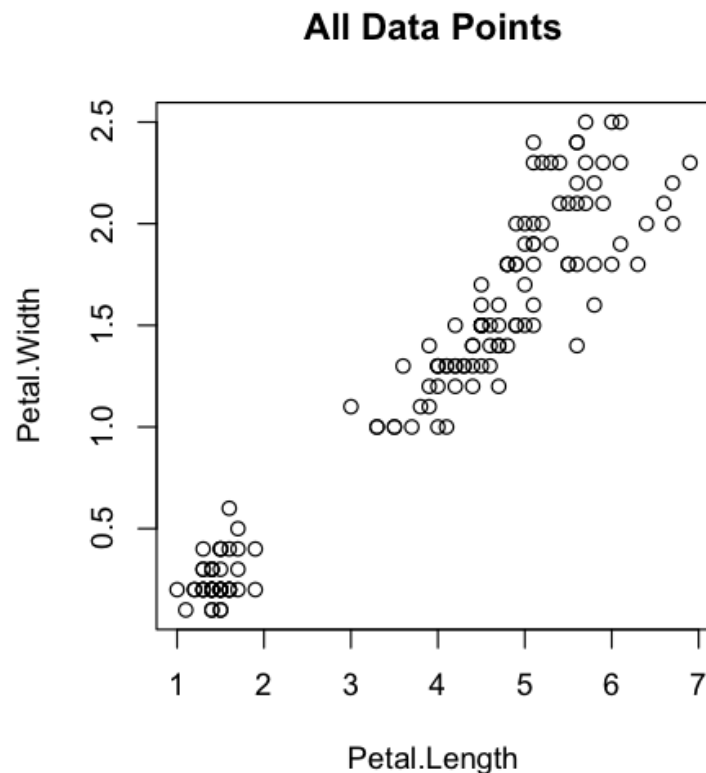
- You have paired observations in two vectors,  $x$  and  $y$ , and a parallel factor  $f$  that indicates their groups. You want to create a scatter plot of  $x$  and  $y$  that distinguishes among the groups.

- **Solution**

- Use the **pch** argument of **plot**. It will plot each point with a different plotting character, according to its group:  
    `> plot(x, y, pch=as.integer(f))`

# Creating a Scatter Plot of Multiple Groups

```
> par(mfrow=c(1,2))  
> with(iris, plot(Petal.Length, Petal.Width, main="All Data Points"))  
> with(iris, plot(Petal.Length, Petal.Width, pch=as.integer(Species),  
+ main="Distinguished By Species"))
```



# Adding a Legend

---

- **Problem**

- You want your plot to include a legend, the little box that decodes the graphic for the viewer.

- **Solution**

- After calling **plot**, call the **legend** function:

- *Legend for points*

- `legend(x, y, labels, pch=c(pointtype1, pointtype2, ...))`

- *Legend for lines according to line type*

- `legend(x, y, labels, lty=c(linetype1, linetype2, ...))`

- *Legend for lines according to line width*

- `legend(x, y, labels, lwd=c(width1, width2, ...))`

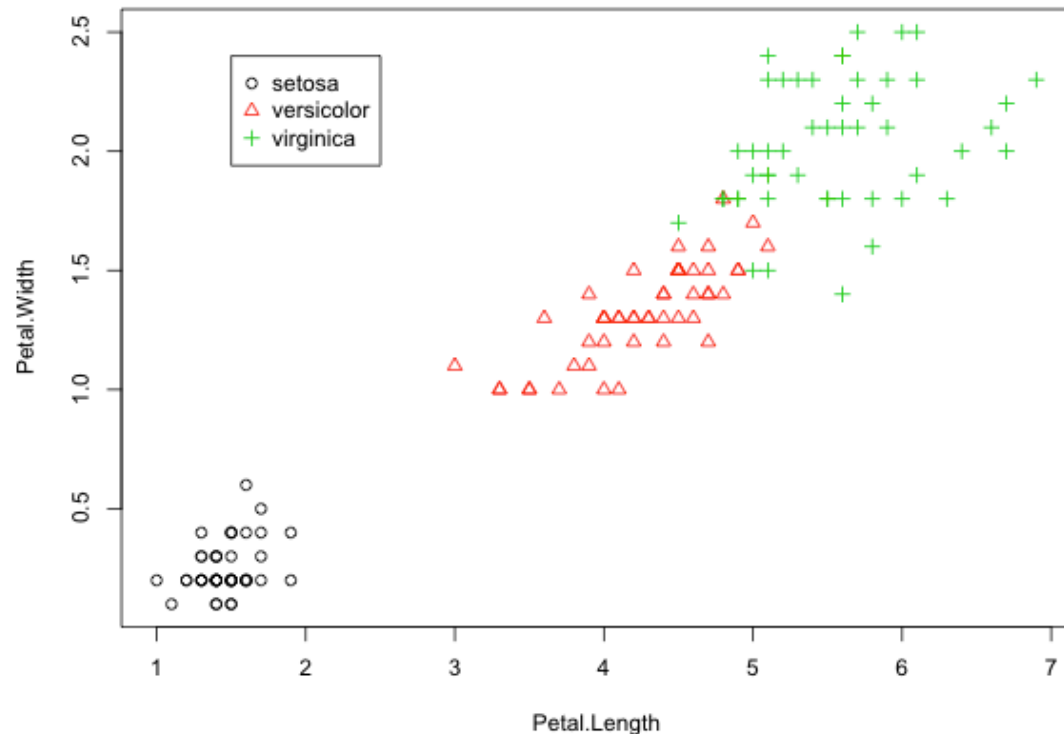
- *Legend for colors*

- `legend(x, y, labels, col=c(color1, color2, ...))`

# Adding a Legend

---

```
> f <- factor(iris$Species)
> with(iris, plot(Petal.Length, Petal.Width, pch=as.integer(f),
+   col=as.integer(f)))
> legend(1.5, 2.4, as.character(levels(f)), pch=1:length(levels(f)),
+   col=1:length(levels(f)))
```



# Plotting the Regression Line of a Scatter Plot

---

- **Problem**

- You are plotting pairs of data points, and you want to add a line that illustrates their linear regression.

- **Solution**

- Create a model object, plot the (x, y) pairs, and then plot the model object using the **abline** function

```
> m <- lm(y ~ x)
```

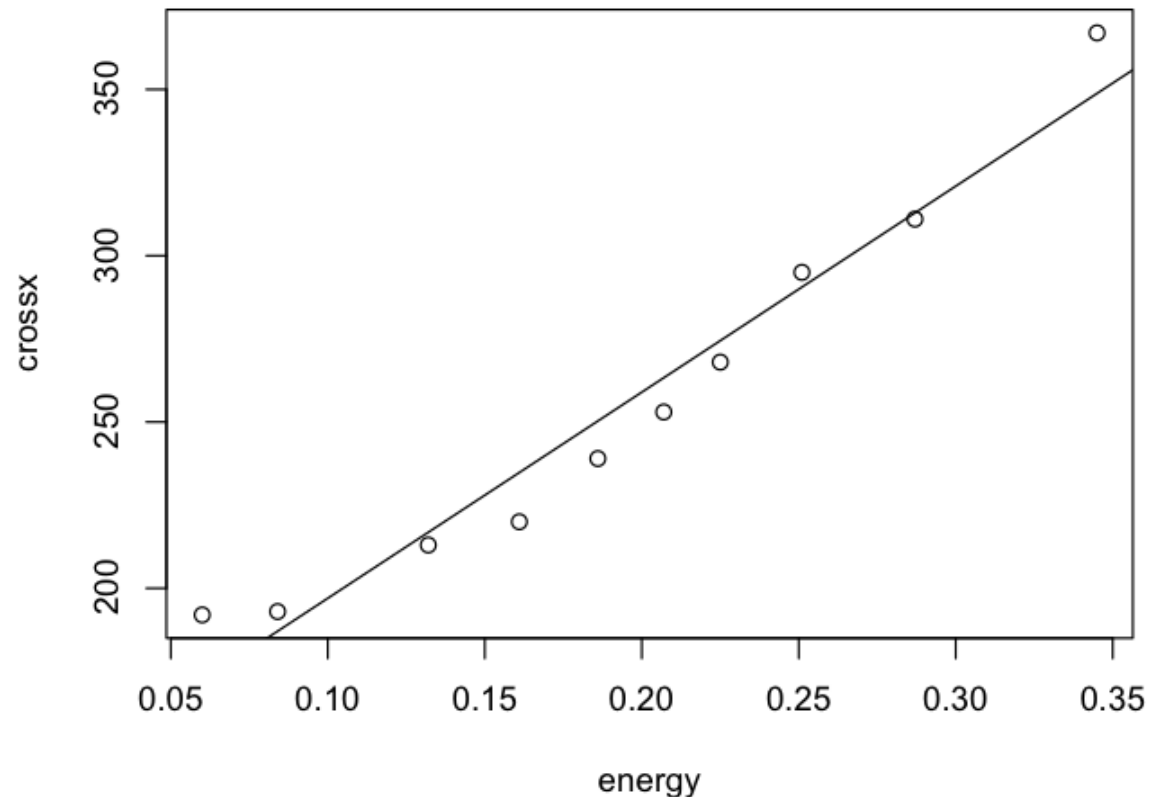
```
> plot(y ~ x)
```

```
> abline(m)
```

# Plotting the Regression Line of a Scatter Plot

---

```
> library(faraway)
> data(strongx)
> m <- lm(crossx ~ energy, data=strongx)
> plot(crossx ~ energy, data=strongx)
> abline(m)
```



# Plotting All Variables Against All Other Variables

---

- **Problem**

- Your dataset contains multiple numeric variables. You want to see scatter plots for all pairs of variables.

- **Solution**

- Place your data in a data frame and then plot the data frame. R will create one scatter plot for every pair of columns:

```
> plot(dfrm)
```

```
> head(iris)
```

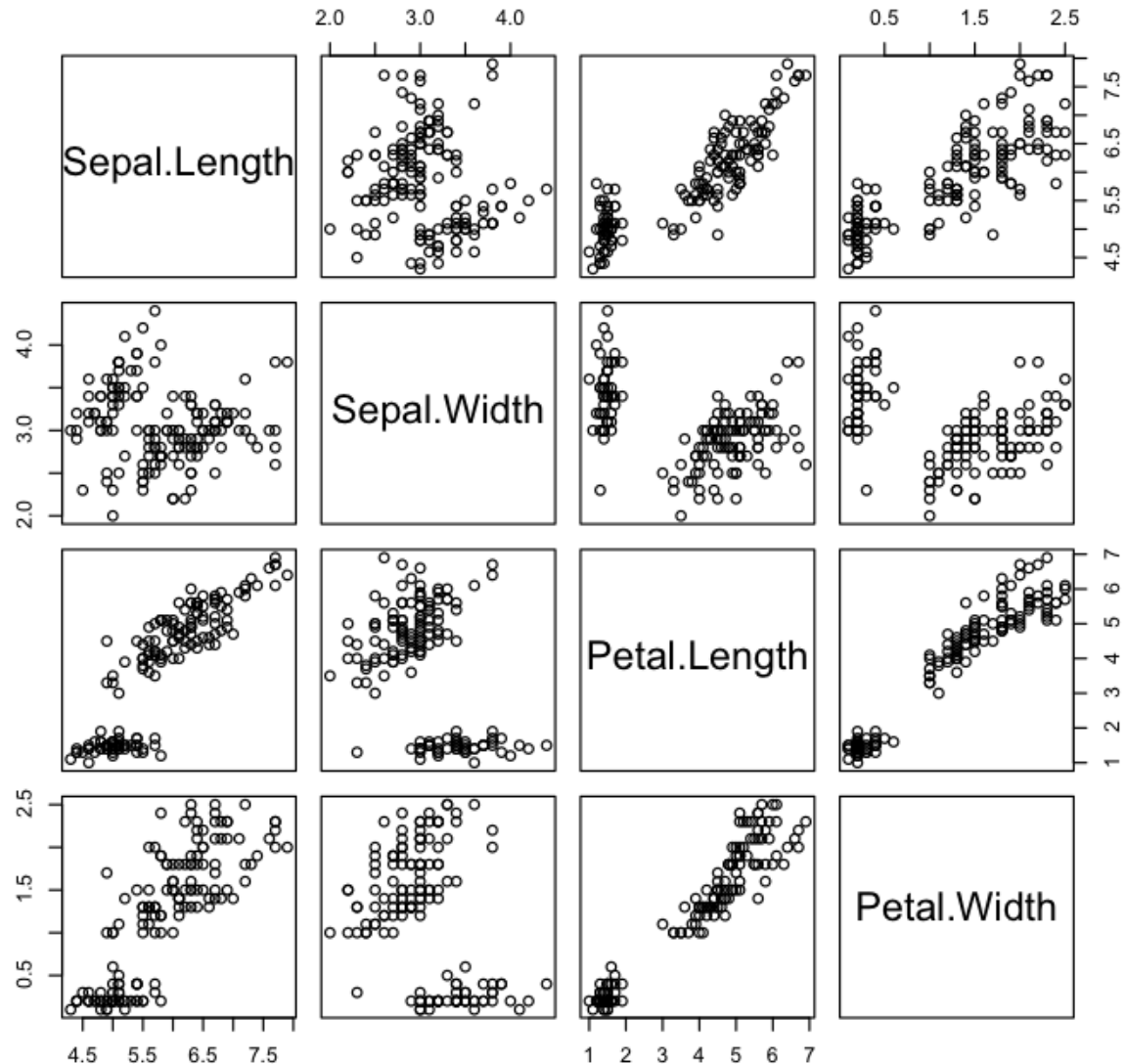
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa

.....



# Plotting All Variables Against All Other Variables

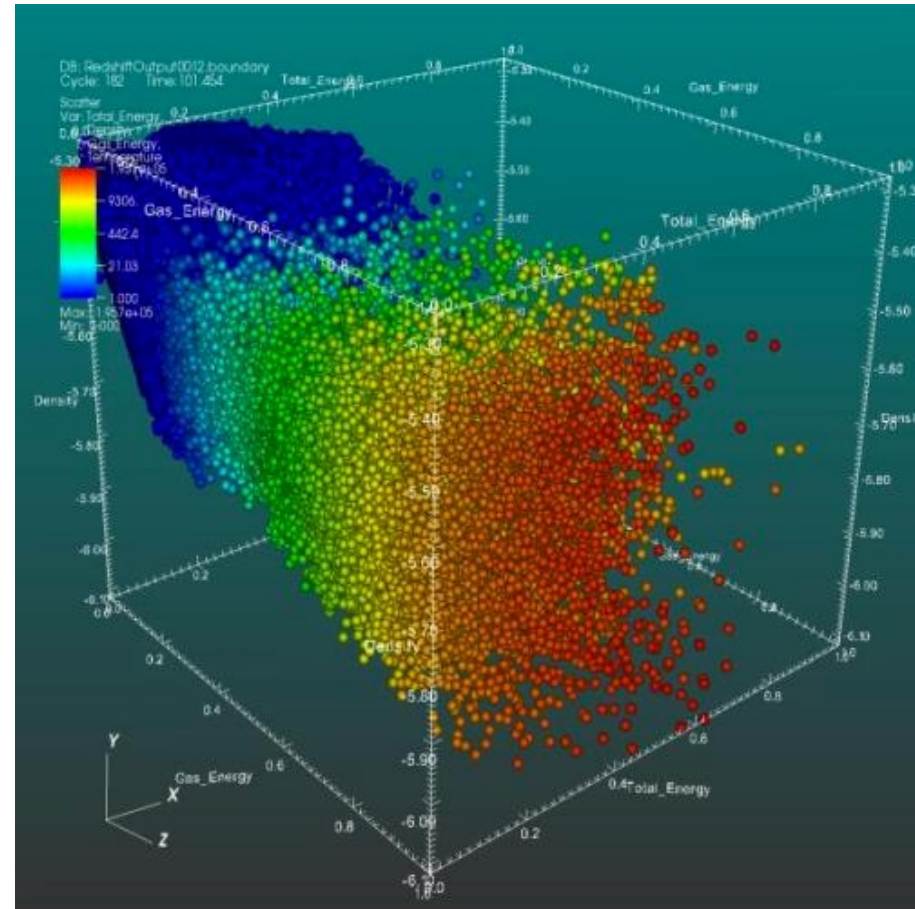
```
> plot(iris[,1:4])
```



# An example for scatterplot

---

- For a set of data variables (dimensions)  $X_1, X_2, \dots, X_k$ , the scatter plot matrix shows all the pairwise scatter plots of the variables on a single view with multiple scatterplots in a matrix format.



# Objective of this test

---

- Learn how to make the scatterplot.
- Learn how to use the *grid()* function.
- Know the details of the *plot()* function.

## The example of *plot()*

---

```
>plot(dfrm1[,2],dfrm2[,2],  
      xlab="2015-temp(° C aka centigrade )",  
      ylab="2016-temp(° C aka centigrade )",  
      xlim=range(0,33), ylim=range(0,33), type="n",  
      main="XXX",cex=0.5)
```

# The example

---

Everyday temp. in HongKong in 2015 and 2016

