

Chương 2

Hiển thị dữ liệu trong R

2.1 Các nguyên lý cơ bản

R có sẵn các chức năng đơn giản để tạo ra nhiều loại biểu đồ. Ví dụ:

Kiểu plot	Hàm
Scatter plot	<code>plot()</code>
Scatter plot matrix	<code>pairs()</code>
Box plot	<code>boxplot()</code>
Strip chart	<code>stripchart()</code>
Histogram plot	<code>hist()</code>
Density plot	<code>density()</code>
Bar plot	<code>barplot()</code>
Line plot	<code>plot()</code> và <code>line()</code>
Pie charts	<code>pie()</code>
Dot charts	<code>dotchart()</code>
Chèn text vào hình	<code>text()</code>

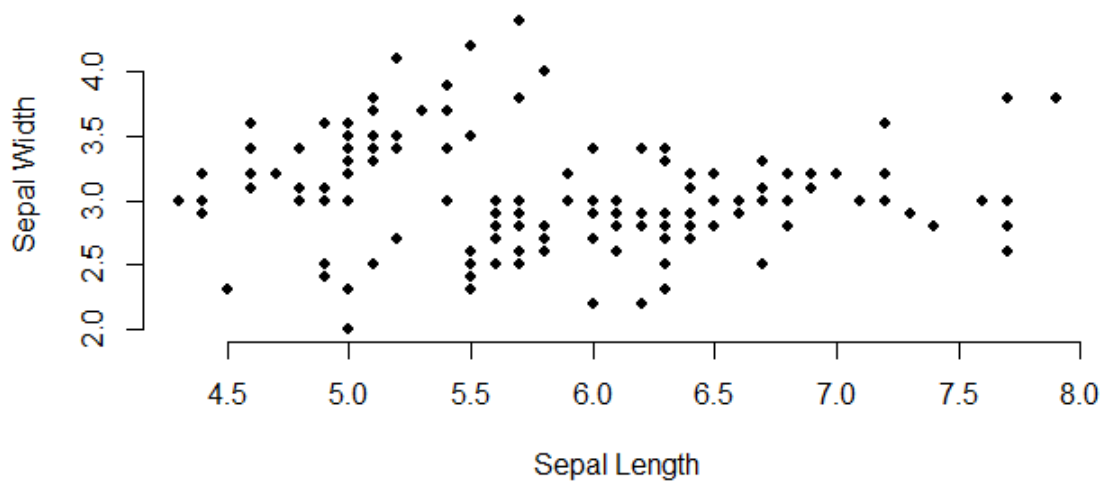
Trong hầu hết các trường hợp, ta có thể dùng các đối số sau để tùy chỉnh:

- `pch`: thay đổi hình dạng điểm
- `cex`: thay đổi kích thước điểm
- `col`: thay đổi màu của điểm
- `frame`: giá trị logic.
- `main`, `xlab`, `ylab`: Tên tiêu đề và dán nhãn các trục x, y.

2.1.1 Scatterplot

```
data("iris")
# Print the first n = 3 rows
head(iris, n = 3)
# Tao mot scatter lot
plot( x = iris$Sepal.Length, y = iris$Sepal.Width, pch = 19, ...
      cex = 0.8, frame = FALSE, xlab = "Sepal Length",ylab = ...
      "Sepal Width" )
```

Ta thu được kết quả sau:



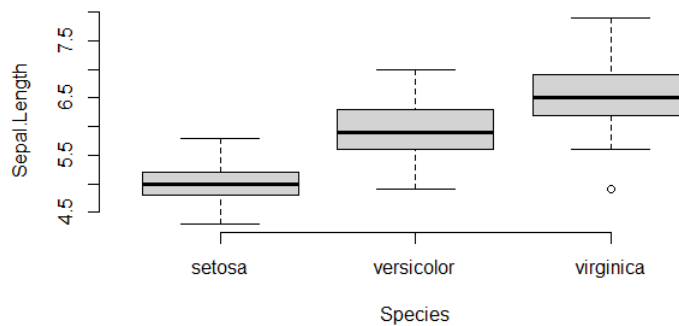
Hình 9: Scatter plot

2.1.2 Boxplot

```
data("iris")
# Print the first n = 3 rows
head(iris, n = 3)
# Tao mot box plot
```

```
boxplot(Sepal.Length ~ Species, data = iris, ylab = ...  
        "Sepal.Length", frame = FALSE, col = "lightgray")
```

Ta thu được kết quả sau:



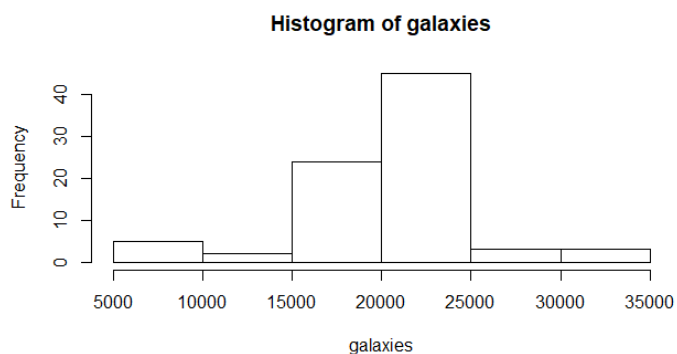
Hình 10: Box plot

2.1.3 Histogram

Để tạo một histogram từ tập dữ liệu galaxies trong packages MASS. Ta sử dụng các lệnh dưới đây:

```
> library(MASS)  
> hist(galaxies)
```

Ta thu được kết quả:

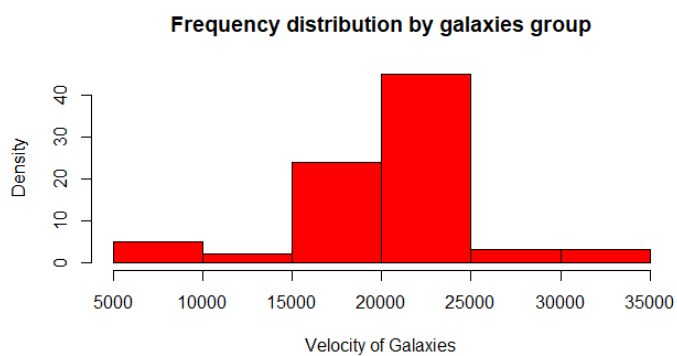


Hình 11: Histogram

Để thêm tên biểu đồ (main); tên cột x,y (xlab, ylab); thêm màu (col):

```
hist(galaxies, main= "Frequency distribution by galaxies ...
      group", xlab= "Velocity of Galaxies", ylab= "Density", ...
      col="red")
```

Ta thu được kết quả:

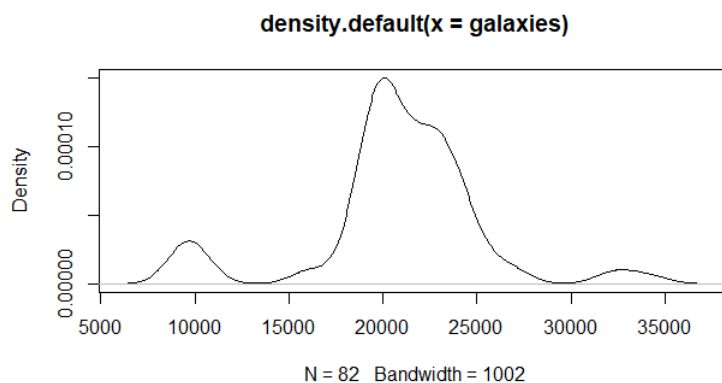


Hình 12: Histogram với màu sắc, tên cột và tên biểu đồ

Biểu đồ đường density:

```
plot(density(galaxies), add=TRUE)
```

Ta thu được kết quả:



Hình 13: Đường density

2.2 Ba hệ thống cơ bản

2.2.1 Lattice graphics

Đồ họa Lattice cung cấp một hệ thống hiển thị cải thiện hơn đồ họa R cơ bản. Chúng ta có thể cài đặt gói lattice bằng lệnh `install.packages("lattice")`. Cụ thể được cho như bảng dưới đây:

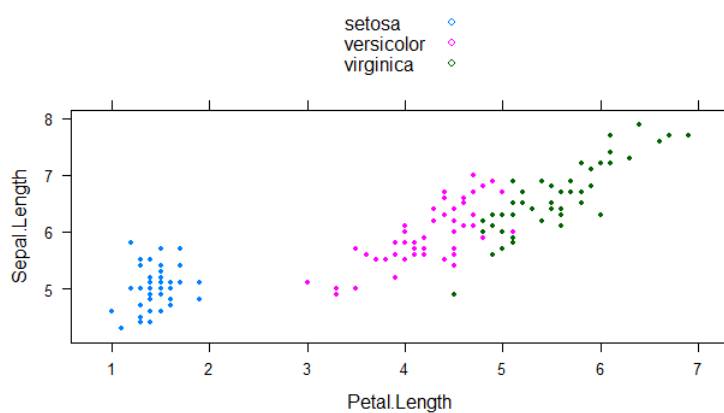
Kiểu plot	Hàm lattice
Scatter plot	<code>xyplot()</code>
Scatter plot matrix	<code>splom()</code>
3D scatter plot	<code>cloud()</code>
Strip plot (1D scatter plots)	<code>stripplot()</code>
Box plot	<code>bwplot()</code>
Dot plot	<code>dotplot()</code>
Bar chart	<code>barchart()</code>
Histogram	<code>histogram()</code>
Density plot	<code>densityplot()</code>
Theoretical quantile plot	<code>qqmath()</code>
Two-sample quantile plot	<code>qq()</code>
3D contour plot of surfaces	<code>qontourplot()</code>

Ví dụ:

```
> library(lattice)
```

```
> xyplot( Sepal.Length ~ Petal.Length, group = Species, data ...
          = iris, auto.key = TRUE, pch = 19, cex = 0.5 )
```

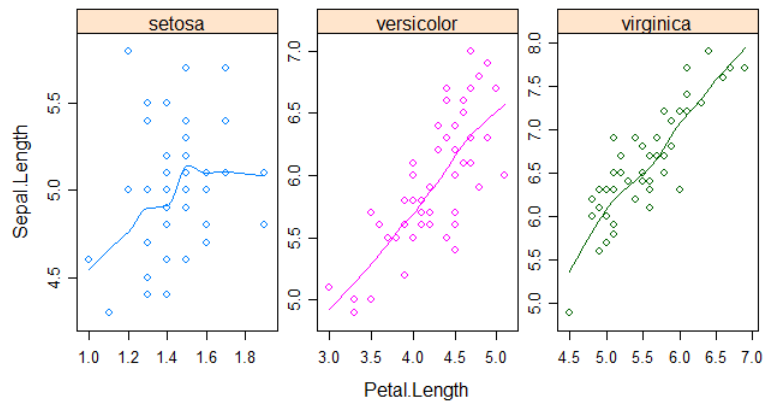
ta thu được kết quả sau:



Hình 14: Vẽ hình với thư viện lattice

```
xyplot(
  Sepal.Length ~ Petal.Length | Species,
  layout = c(3, 1),          # panel with ncol = 3 and nrow ...
    = 1
  group = Species, data = iris,
  type = c("p", "smooth"),   # Show points and smoothed line
  scales = "free"            # Make panels axis scales ...
    independent
)
```

ta được:



Hình 15: Vẽ hình với thư viện lattice

2.2.2 qqplot

Một cú pháp đơn giản của qqplot là:

```
library(ggplot2)
qqplot(x, y=NULL, data, geom="auto", xlim = c(NA, NA), ylim ...
      =c(NA, NA))
```

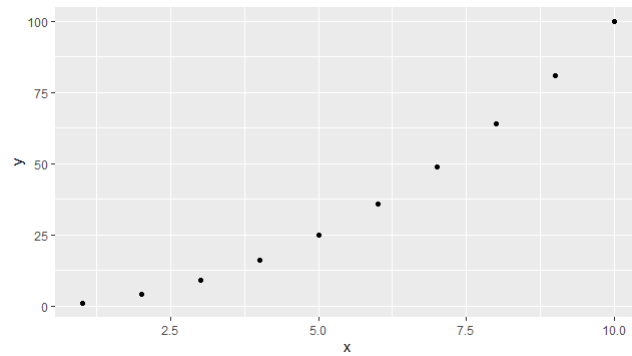
Trong đó:

- x : giá trị x.
- y : giá trị y (tùy chọn).
- data : data frame sử dụng để hiển thị.
- geom : vecto kí tự chỉ định vẽ. Mặc định là "điểm" nếu x và y được chỉ định và "histogram" nếu chỉ x được chỉ định.
- xlim, ylim: giới hạn trục x và trục y.

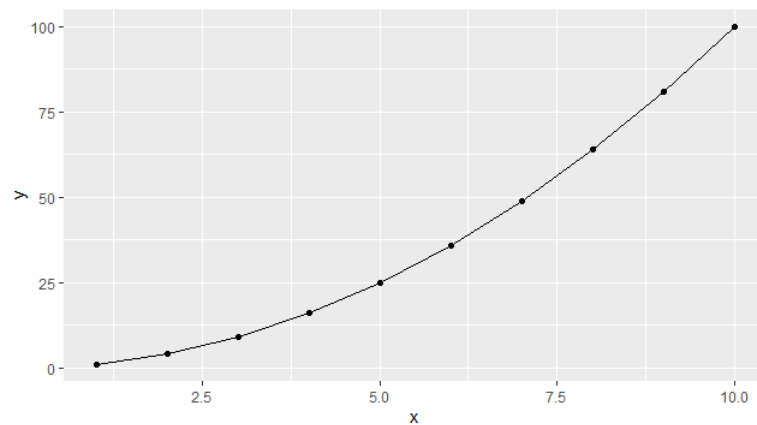
Ví dụ:

```
library(ggplot2)
# data là vector
x <- 1:10; y = x*x
# plot cơ bản
qqplot(x,y)
# thêm line
qqplot(x, y, geom=c("point", "line"))
```

Ta thu được kết quả dưới đây:



Hình 16: Vẽ hình cơ bản với qplot

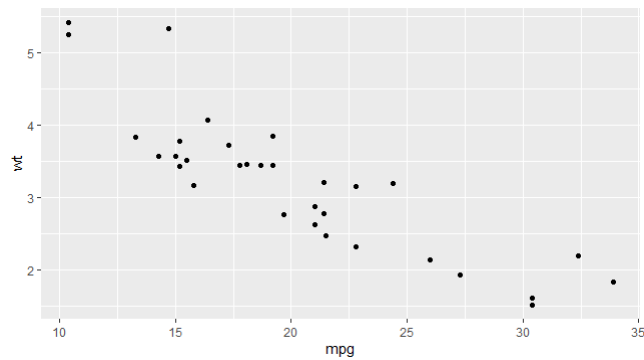


Hình 17: Thêm lines với qplot

Ngoài ra ta có thể sử dụng dữ liệu mtcars trong R.

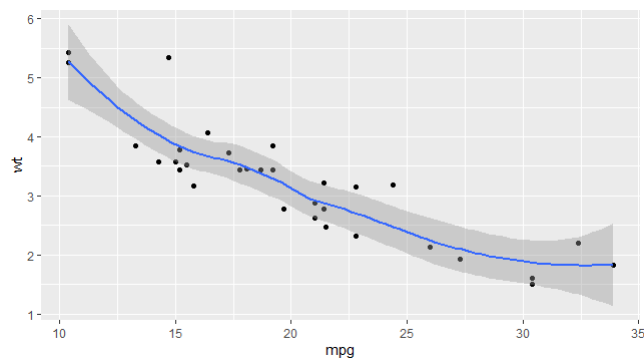
```
library(ggplot2)
> data(mtcars)
# plot cơ bản
> qplot(mpg, wt, data=mtcars)
```

ta được kết quả:



Hình 18: Vẽ hình với qplot

```
# Smoothing
qplot(mpg, wt, data = mtcars, geom = c("point", "smooth"))
```



Hình 19: Smoothing với qplot

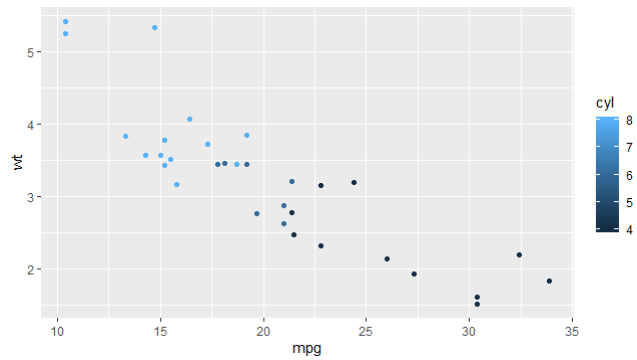
```
# Thay doi mau sac
qplot(mpg, wt, data = mtcars, colour = cyl)

# Thay doi mau sac theo nhom
df <- mtcars
df[, 'cyl'] <- as.factor(df[, 'cyl'])
qplot(mpg, wt, data = df, colour = cyl)

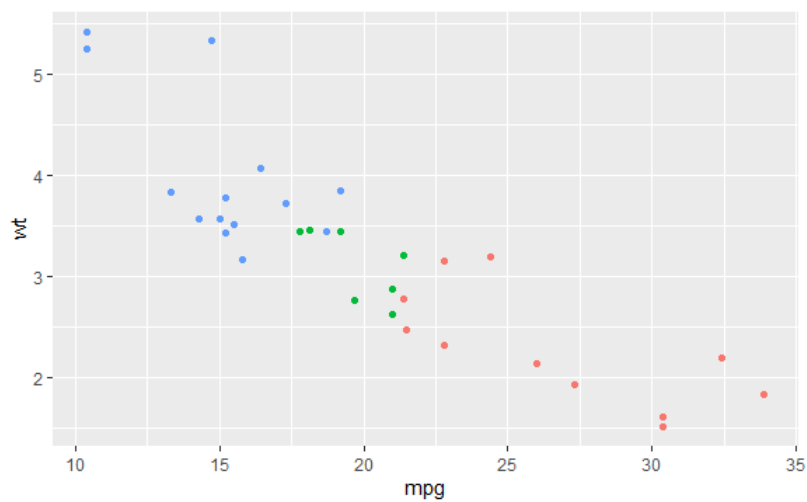
# Them lines
qplot(mpg, wt, data = df, colour = cyl,
```

```
geom=c("point", "line"))
```

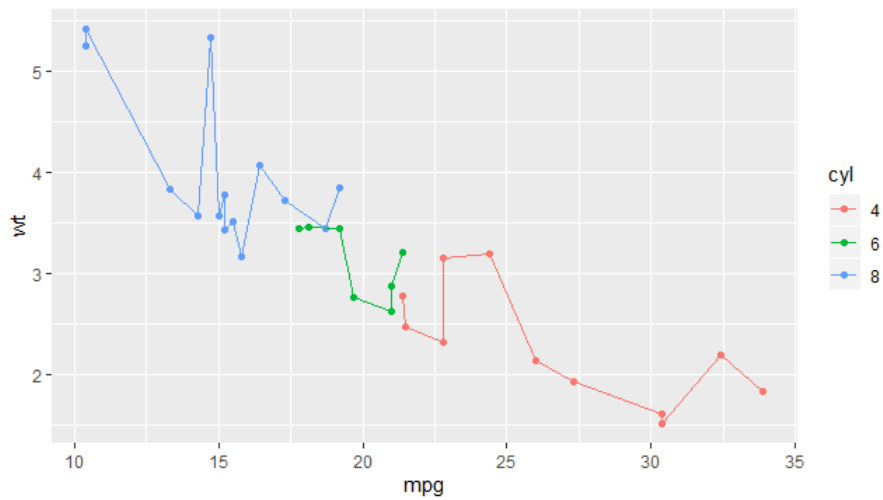
Ta thu được kết quả lần lượt như sau:



Hình 20: Thay đổi màu sắc với qplot



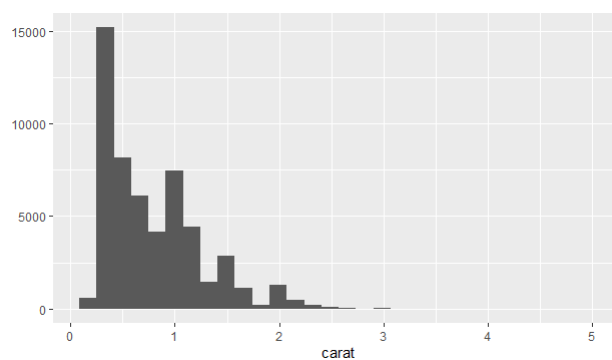
Hình 21: Thay đổi màu theo nhóm với qplot



Hình 22: Thêm lines với qplot

Ta cũng có thể vẽ biểu đồ histogram (a) và smooth (b) bằng qplot của tập dữ liệu diamonds với biến carat và price trong gói packages ggpubr.

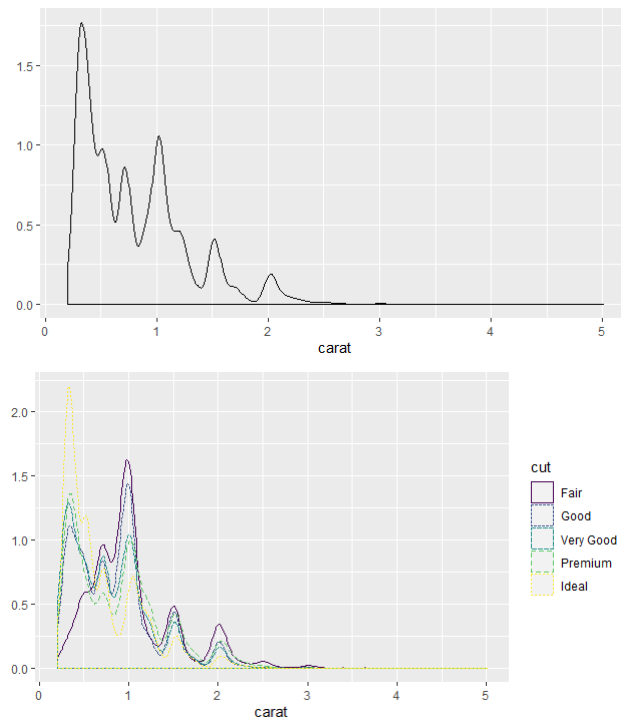
```
library(ggpubr)
> qplot(carat, data = diamonds, geom = "histogram", ...
      bandwidth = 100)
```



Hình 23: Vẽ hình với qplot

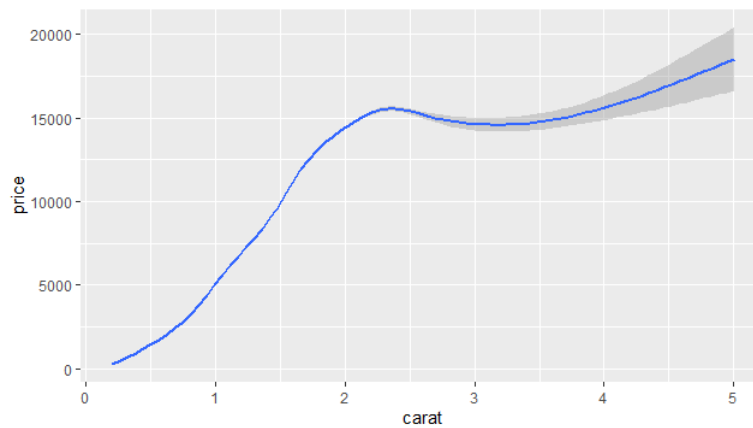
```
density plot
>qplot(carat, data=diamonds, geom='density')
```

```
>qplot(carat, data=diamonds, geom='density',color ...  
      =cut,linetype=cut)
```



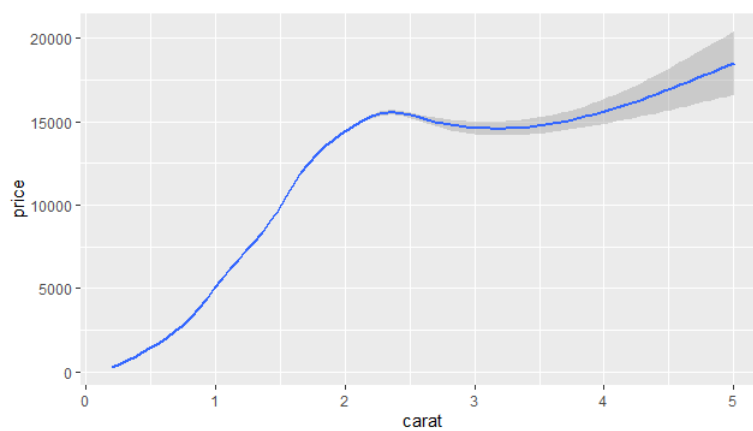
Hình 24: Vẽ density với qplot

```
# smooth  
qplot(carat, price, data=diamonds, geom='smooth')
```



Hình 25: Smoothing với qplot

```
# smooth
qplot(carat, price, data=diamonds, geom='smooth')
```



Hình 26: Smoothing với qplot

2.2.3 ggplot

`ggplot()` dùng để khởi tạo một đối tượng `ggplot` ban đầu, hiển thị tùy theo bộ dữ liệu đầu vào được tùy chỉnh bởi các tham số.

```
ggplot(data = NULL, mapping = aes(), ...,
environment = parent.frame())
```

2.3 Sử dụng ggplot2

Ggplot2 là một gói hỗ trợ hiển thị rất mạnh và linh hoạt trong R được thực hiện bởi Hadley Wickham. Ggplot2 dựa trên khái niệm "ngữ pháp của biểu đồ". Cấu trúc của ggplot2 được chia thành các phần chính. Plot = data + Aesthetics + Geometry.

- Data: một data frame.
- Aesthetics: chỉ biến x và biến y. Nó cũng có thể dùng để điều chỉnh màu sắc, kích thước, và hình dạng của các điểm...
- Geometry: tương ứng với các dạng đồ thị (histogram, boxplot, line plot,...).

Kiểu plot	Hàm ggplot2
Initialize a ggplot	<code>ggplot()</code>
Scatter plot	<code>geom_point()</code>
Strip chart	<code>geom_jitter()</code>
Dot plot	<code>geom_dotplot()</code>
Line plot	<code>geom_line()</code>
Bar Chart	<code>geom_bar()</code>
Histogram	<code>geom_histogram()</code>
Density plot	<code>geom_density()</code>
QQ plot	<code>stat_qq()</code>
Box plot	<code>geom_boxplot()</code>
Title and axis labels	<code>labs()</code>

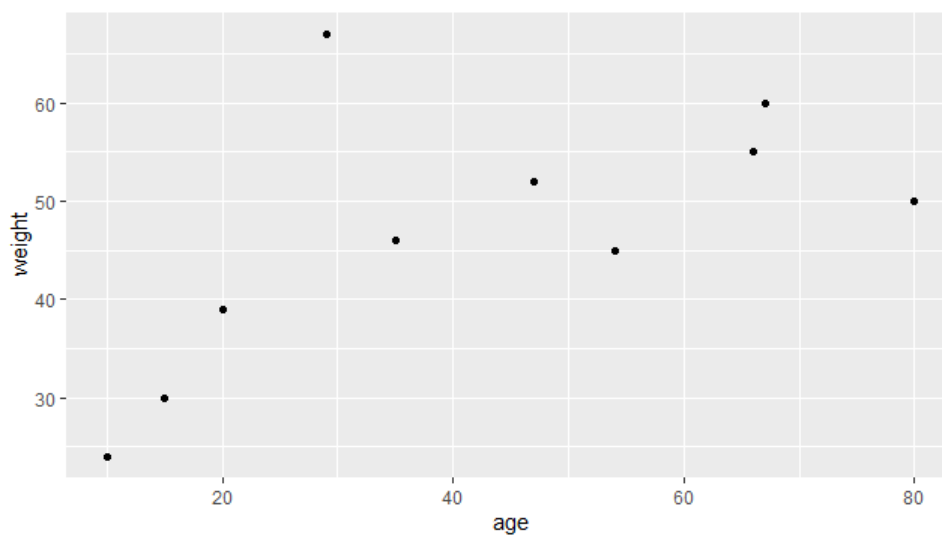
Chúng ta có thể hiển thị dữ liệu thành nhiều bảng theo từng biến trên cùng một hình bằng hàm `facet_wrap()`.

Ví dụ 1: Cho tập dữ liệu `data1` như sau:

Age	Weight	Gender
10	24	male
20	39	female
35	46	male
67	60	female
54	45	female
80	50	male
15	30	female
47	52	male
66	55	female
29	67	female

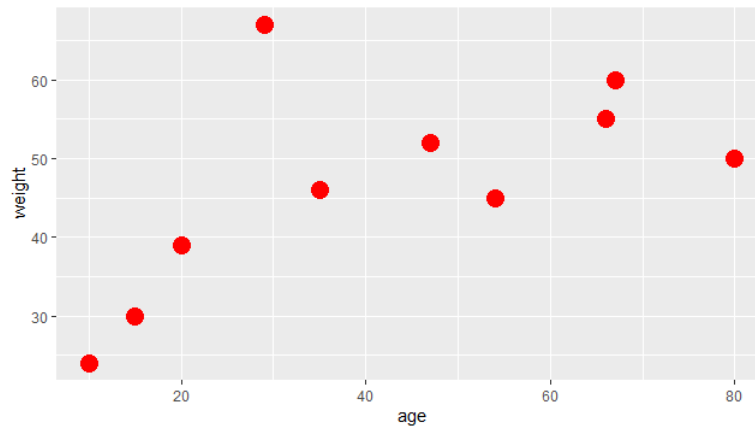
Ta sử dụng các dòng lệnh sau:

```
> age <- c(10, 20, 35, 67, 54, 80, 15, 47, 66, 29)
> weight <- c(24, 39, 46, 60, 45, 50, 30, 52, 55, 67)
> gender <- c("male", "female", "male", "female", "female", ...
  "male", "female", "male", "female", "female" )
> data1 <- data.frame(age, weight, gender)
> library(ggplot2)
# hiển thị cơ bản
> ggplot(data1, aes(age, weight)) + geom_point()
```



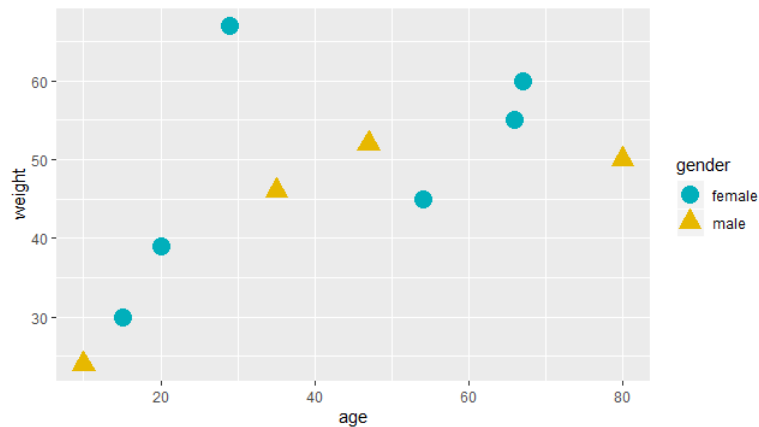
Hình 27: Vẽ hình cơ bản với ggplot 2

```
#Thay đổi kích thước điểm
> ggplot(data1, aes(age, weight)) + geom_point(size = 5, ...
  color="red")
```



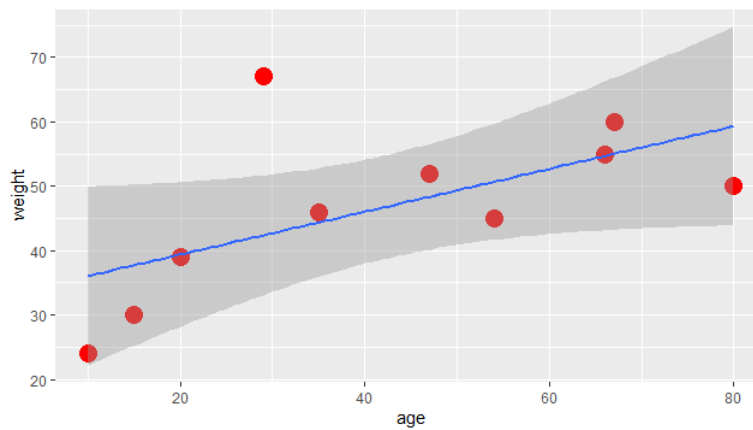
Hình 28: Thay đổi kích thước điểm với ggplot 2

```
#hien thi nhom diem theo mau
> ggplot(data1, aes(age, weight)) + geom_point(size = 5, ...
  aes(color = gender, shape = gender)) + ...
  scale_color_manual(values = c("#00AFBB", "#E7B800"))
```



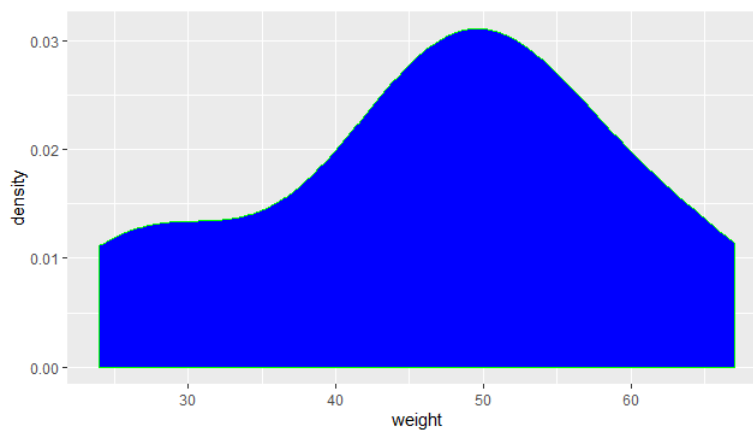
Hình 29: Hiển thị nhóm điểm theo màu với ggplot 2

```
# them smooth
>ggplot(data1, aes(age, weight)) + geom_point(size = 5, ...
  color="red") + geom_smooth(method = "lm")
```



Hình 30: Smoothing với ggplot 2

```
# Them density
> ggplot(data1, aes(weight)) + geom_density(fill = "blue", ...
  color = "green")
```

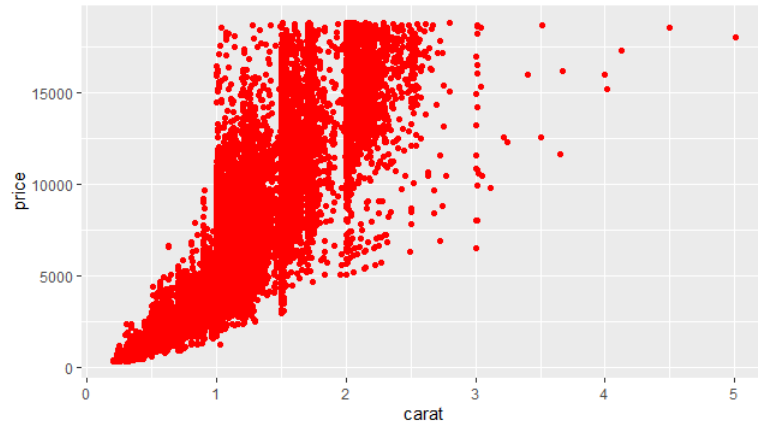


Hình 31: Density với ggplot 2

Ví dụ 2: Vẽ biểu đồ chấm điểm trong tập dữ liệu diamonds trong gói ggplot2.

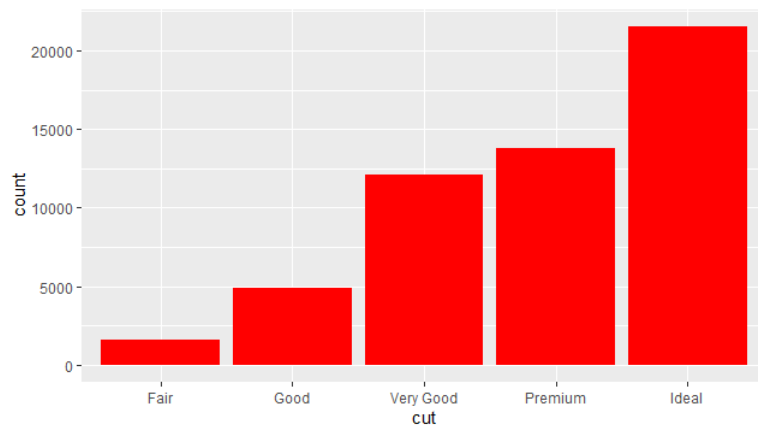
```
> library(ggplot2)
> str(diamonds)
> View(diamonds)
```

```
> ggplot(diamonds, aes(carat, price)) + geom_point(color = ...  
  "red")
```



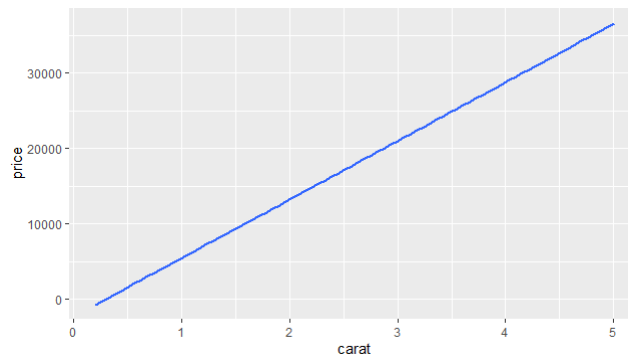
Hình 32: Vẽ biểu đồ chấm điểm với ggplot 2

```
> ggplot(diamonds, aes(cut)) + geom_bar(fill = "red")
```



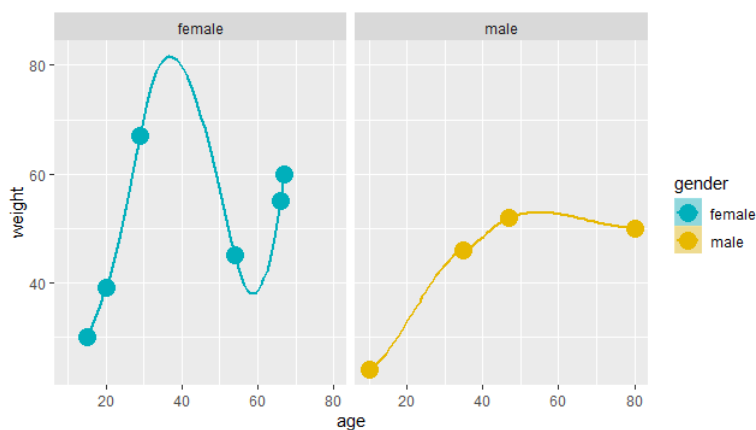
Hình 33: Vẽ biểu đồ cột với ggplot 2

```
ggplot(diamonds, aes(carat, price)) + geom_smooth(method = ...  
  "lm")
```



Hình 34: Smoothing với ggplot 2

```
# facet
>ggplot(data1, aes(age, weight))+ ...
  geom_point(size=5,aes(color = gender))+
  geom_smooth(aes(color = gender, fill = gender))+ ...
  facet_wrap(~gender, ncol = 2, nrow = 1)+ ...
  scale_color_manual(values = c("#00AFBB", "#E7B800"))+ ...
  scale_fill_manual(values = c("#00AFBB", "#E7B800"))
```

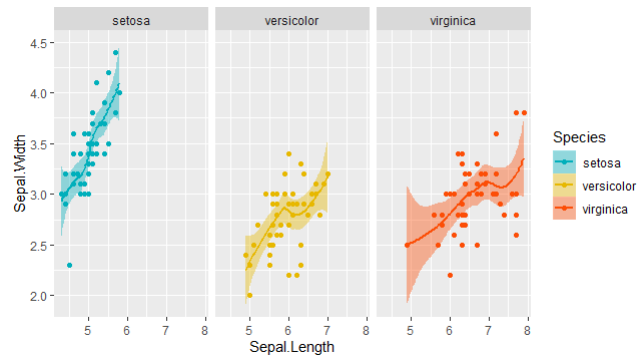


Hình 35: Facet với ggplot 2

Ví dụ 3: Mô tả facet trên bộ dữ liệu iris trên R.

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

```
geom_smooth(aes(color = Species, fill = Species))+
facet_wrap(~Species, ncol = 3, nrow = 1)+
scale_color_manual(values = c("#00AFBB", "#E7B800", ...
                              "#FC4E07"))+ scale_fill_manual(values = c("#00AFBB", ...
                              "#E7B800", "#FC4E07"))
```



Hình 36: Facet với ggplot 2 trên bộ dữ liệu iris