

R 語言與資料視覺化

丘祐瑋
David Chiu

環境資訊頁面

- 所有課程補充資料、投影片皆位於
 - ▣ https://github.com/ywchiu/cdc_course

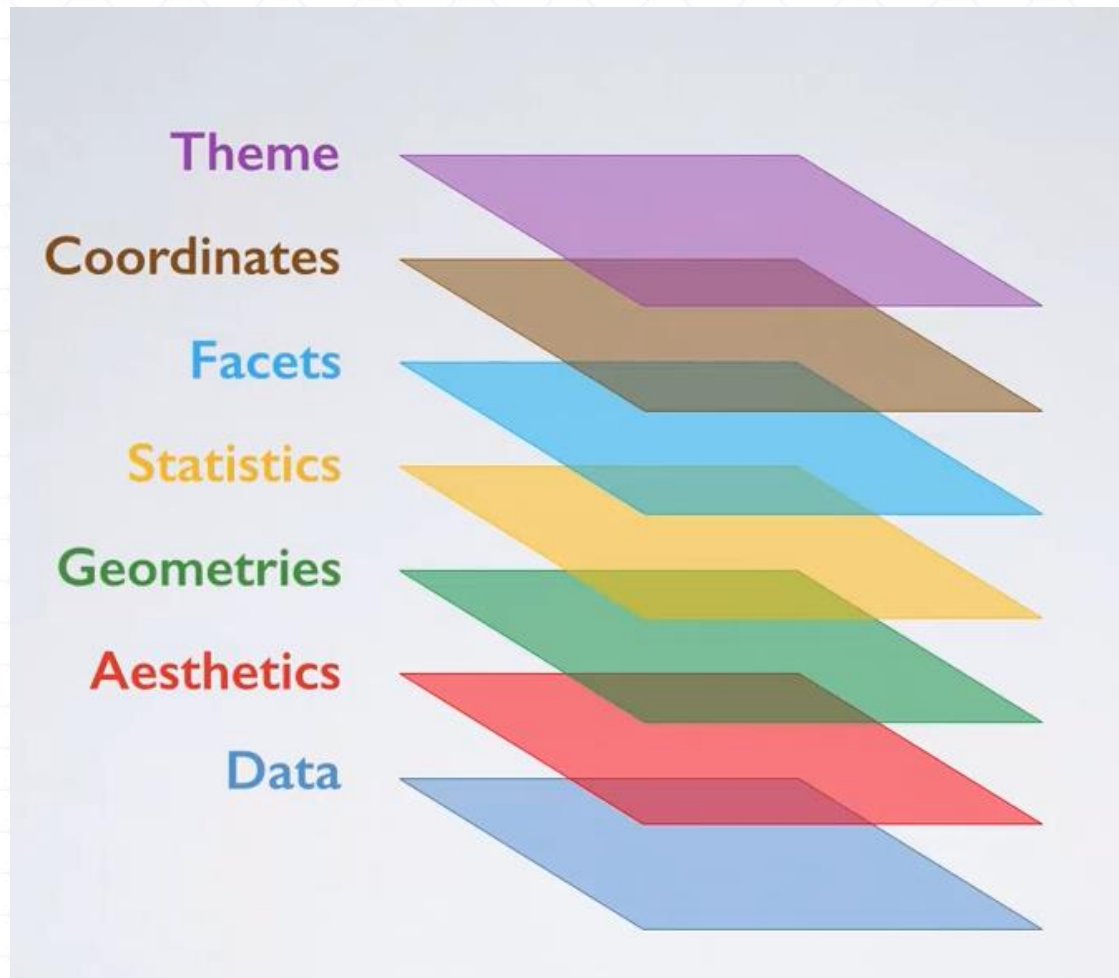
GGPlot2

繪圖語法 (Grammar of Graphics)

- Wilkinson 用 Grammar of Graphics 解釋統計圖形 (statistical graphic) 的概念
- Wickham 則是以該語法為基礎發展出 ggplot 繪圖系統



GGPlot繪圖架構



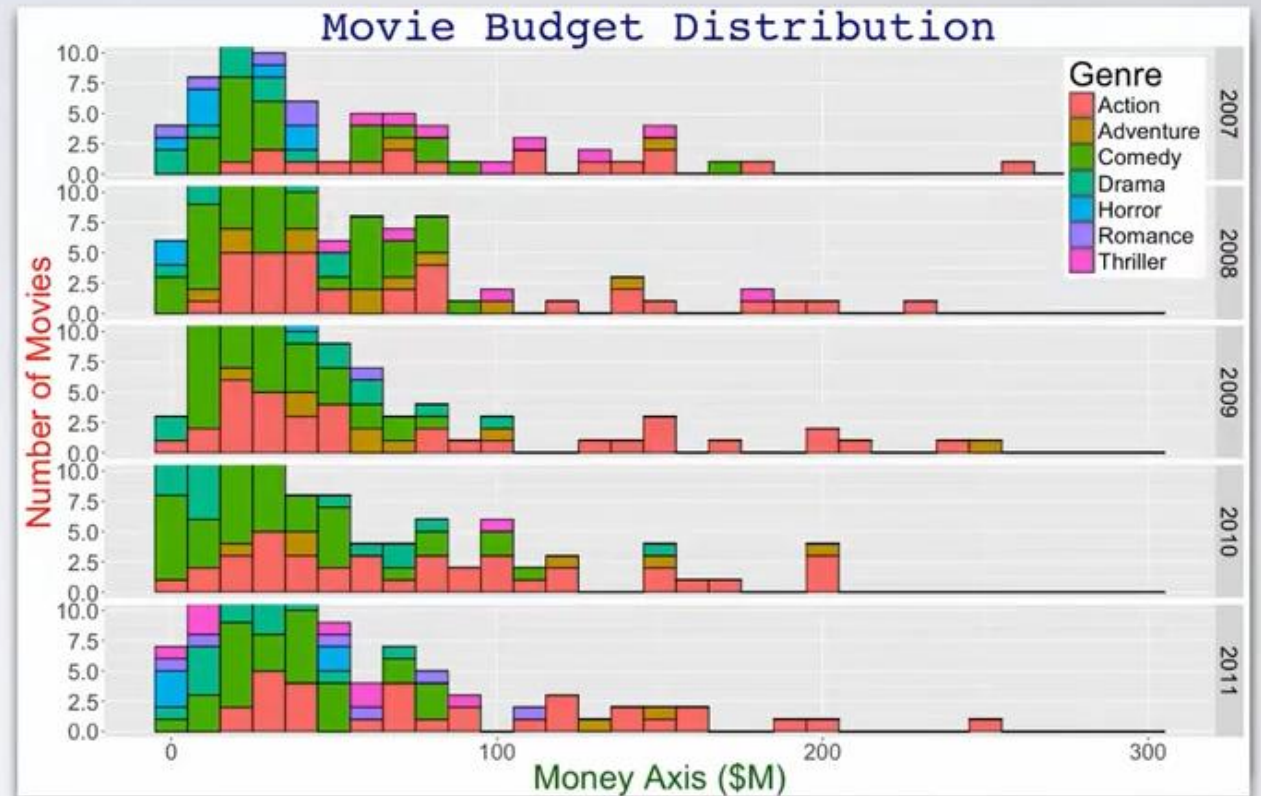
- 主題 (Theme)
 - ▣ 控制資料以外的繪圖組件
- 座標系統 (Coordinates)
 - ▣ 指定繪圖時所使用的座標系統
- 繪圖面 (Facets)
 - ▣ 指定如何將資料分散在多張子圖形中繪製
- 統計轉換 (Statistics)
 - ▣ 指定如何以將資料轉換為各種統計量。
- 幾何圖案 (Geometries)
 - ▣ 繪製資料用的幾何圖形
- 美學對應 (Aesthetic)
 - ▣ 指定原始資料與圖形之間的對應關係
- 資料來源 (Data)
 - ▣ 原始資料來源

繪圖範例



Aesthetics: x, Fill

Movie Name	Budget	Genre
ABC	\$5 M	Action
DEF	\$50 M	Action
GHI	\$12 M	Comedy



安裝GGPlot2

■ 安裝ggplot2

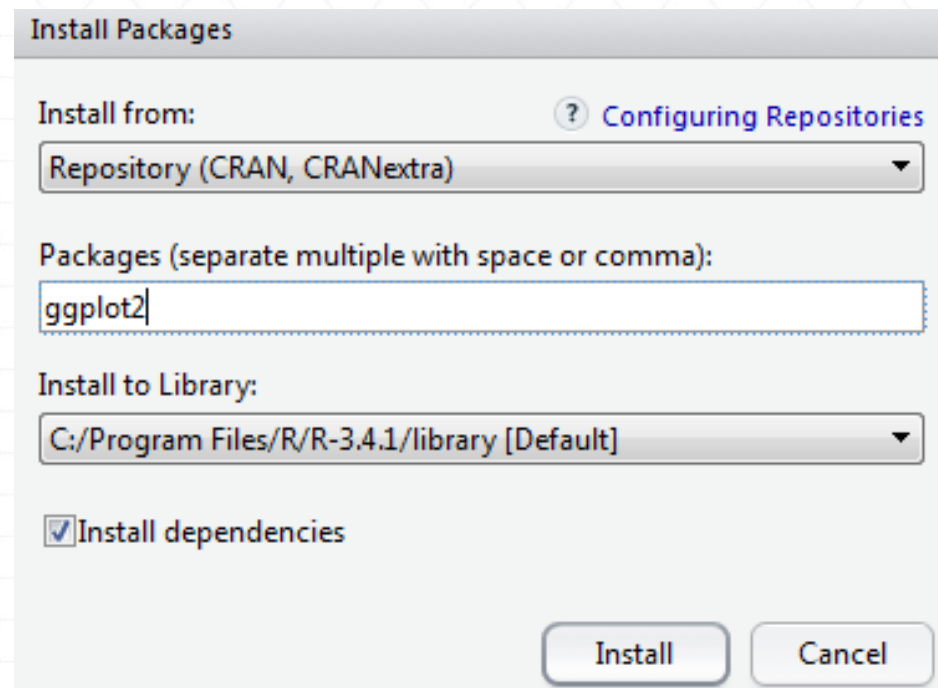
□ `install.packages("ggplot2")`

■ 使用ggplot2

□ `library(ggplot2)`

■ 觀看說明頁

□ `help(package='ggplot2')`



資料來源 (Data)

- 從疫情中心的開放資料網站蒐集到「地區年齡性別統計表-麻疹」(2003年起各地區、各年齡層、性別之病例數統計表(疾病名稱：麻疹，日期種類：發病日，病例種類：確定病例，感染來源：本土、境外移入))

■ 讀取資料

```
library(readr)
```

```
measles <- read_csv("https://raw.githubusercontent.com/ywchiu/cdc_course/master/data/measles.csv")
```

```
head(measles)
```

確定病名 <chr>	發病年份 <int>	發病月份 <int>	縣市 <chr>	鄉鎮 <chr>	性別 <chr>	是否為境外移入 <chr>	年齡層 <chr>	確定病例數 <int>
麻疹	2009	4	高雄市	小港區	M	否	20-24	2
麻疹	2009	5	基隆市	中山區	M	否	30-34	1
麻疹	2011	6	新北市	蘆洲區	F	是	25-29	1
麻疹	2014	1	高雄市	三民區	F	是	0	1
麻疹	2017	3	台北市	松山區	F	是	0	1
麻疹	2018	4	桃園市	蘆竹區	F	否	30-34	1

日期資料轉換

■ Date

- 為簡單版的時間資料格式

■ POSIXct, POSIXlt

- POSIXct: 以整數形式存放時間資料 (1970/01/01 至今的秒數)
- POSIXlt: 以年、月、週、日等有效訊息存放資料

日期函式

```
x <- as.Date("2018-05-08")
```

```
class(x)
```

```
unclass(x)
```

```
y <- as.Date("1970-01-01")
```

```
unclass(y)
```


時間處理 (1)

■ 印出系統時間

```
x <- Sys.time()
```

```
x
```

■ 時間轉換

```
p <- as.POSIXlt(x)
```

```
unclass(p)
```

```
p$sec
```

```
p1 <- as.POSIXct(x)
```

```
unclass(p1)
```

時間處理 (2)

■ 利用strptime 轉換時間

```
ds <- c("May 8, 2018 12:00")  
x <- strftime(ds, "%B %d, %Y %H:%M")  
x
```

■ 時間運算

```
x <- strftime(ds, "%B %d, %Y %H:%M")  
x1 <- as.POSIXlt(as.Date('2018-05-08'))  
x - x1
```


使用Lubridate 轉換時間資料

■ 安裝lubridate

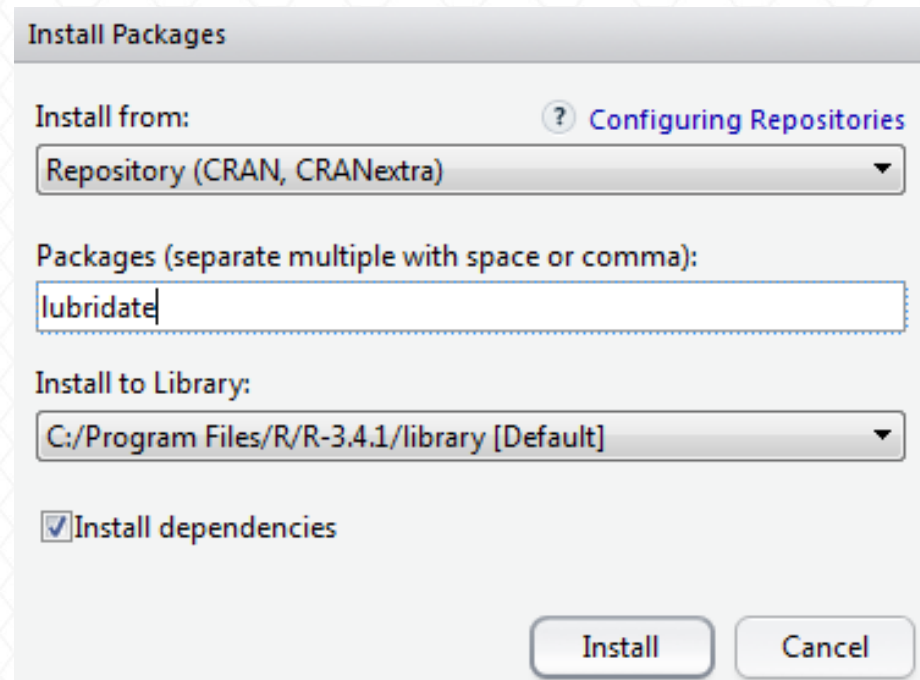
□ `install.packages("lubridate")`

■ 使用lubridate

□ `library(lubridate)`

■ 觀看說明頁

□ `help(package='lubridate')`



使用Lubridate 轉換時間資料

```
library(lubridate)  
ymd("20180605")  
ymd("2018/06/05")  
dmy("05/06/2011")  
ymd_hms('2018-05-06 08:00:00')
```


取得年月等資訊

```
bday <- dmy("15/10/1988")  
month(bday)
```

```
wday(bday, label = TRUE)
```

```
year(bday) <- 2018  
wday(bday, label = TRUE)
```

轉換時間資料

```
measles$發病時間 <- as.Date(with(measles, paste(發病年份, 發病月份, '01', sep="-")), "%Y-%m-%d")
```

With 語法可指名發病年份與發病月份資料來自measles 資料集

確定病名 <chr>	發病年份 <int>	發病月份 <int>	縣市 <chr>	鄉鎮 <chr>	性別 <chr>	是否為境外移 <chr>	年齡層 <chr>	確定病例數 <int>	發病時間 <date>
麻疹	2009	4	高...	小...	M	否	20-24	2	2009-04-01
麻疹	2009	5	基...	中...	M	否	30-34	1	2009-05-01
麻疹	2011	6	新...	蘆...	F	是	25-29	1	2011-06-01
麻疹	2014	1	高...	三...	F	是	0	1	2014-01-01
麻疹	2017	3	台...	松...	F	是	0	1	2017-03-01
麻疹	2018	4	桃...	蘆...	F	否	30-34	1	2018-04-01

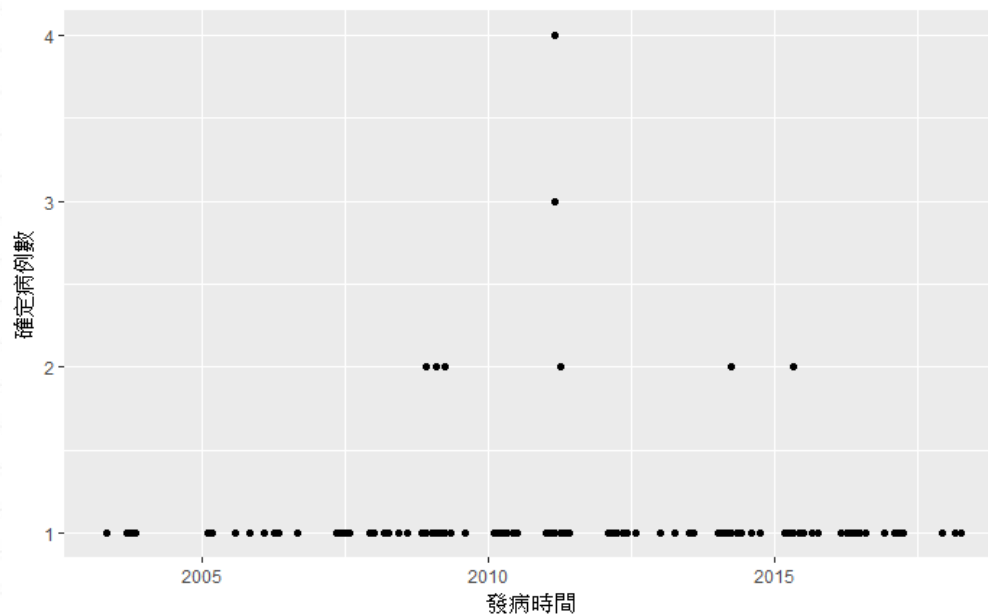
美學對應 (Aesthetic)

繪製散佈圖

```
library(ggplot2)
```

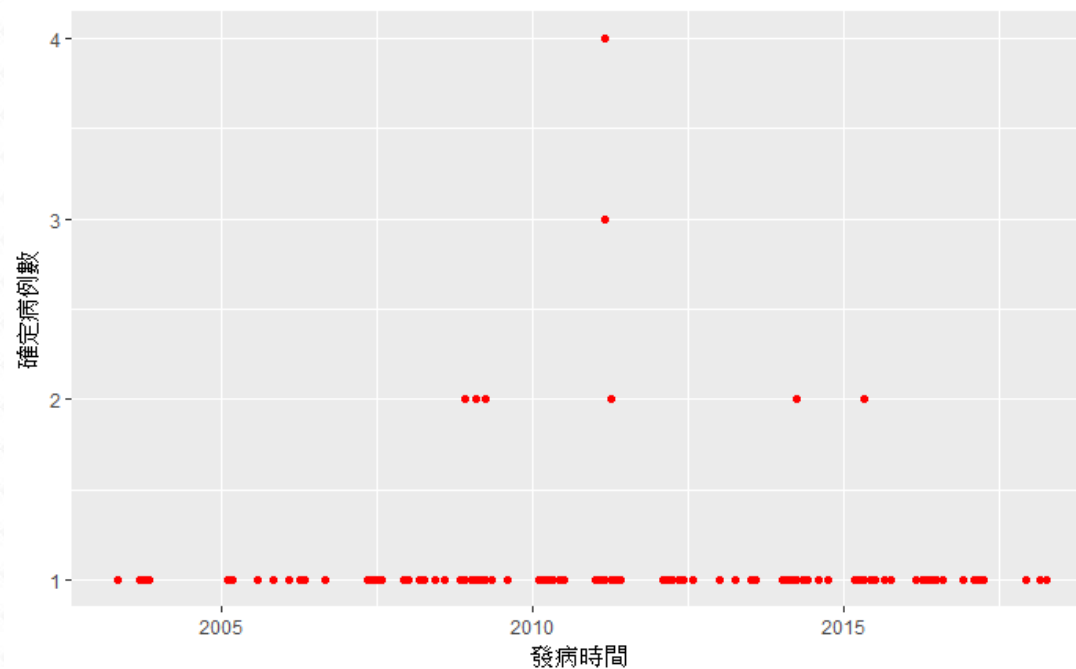
```
p1 <- ggplot(measles, aes(x = 發病時間, y = 確定病例數))
```

```
p1 + geom_point()
```



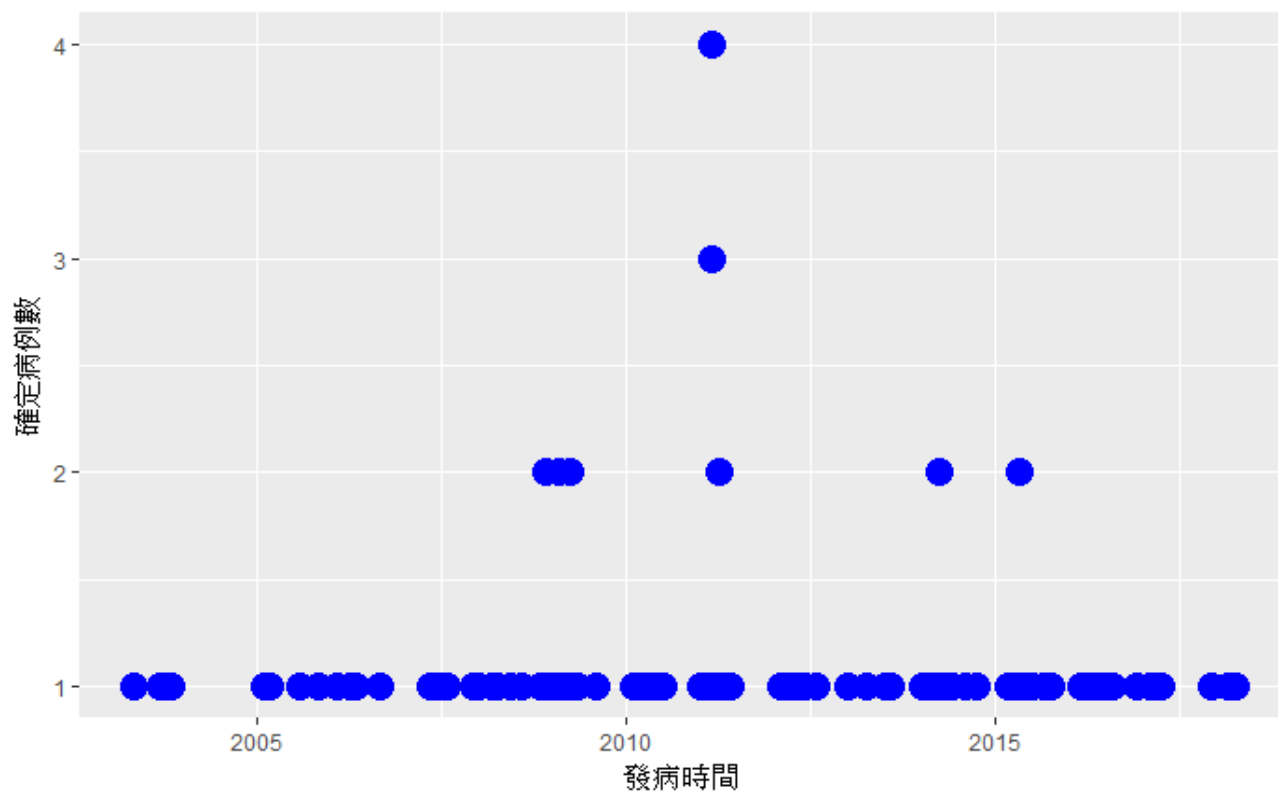
修改顏色

```
p1 <- ggplot(measles, aes(x = 發病時間, y = 確定病例數))  
p1 + geom_point(color='red')
```



修改點的大小、顏色、形狀

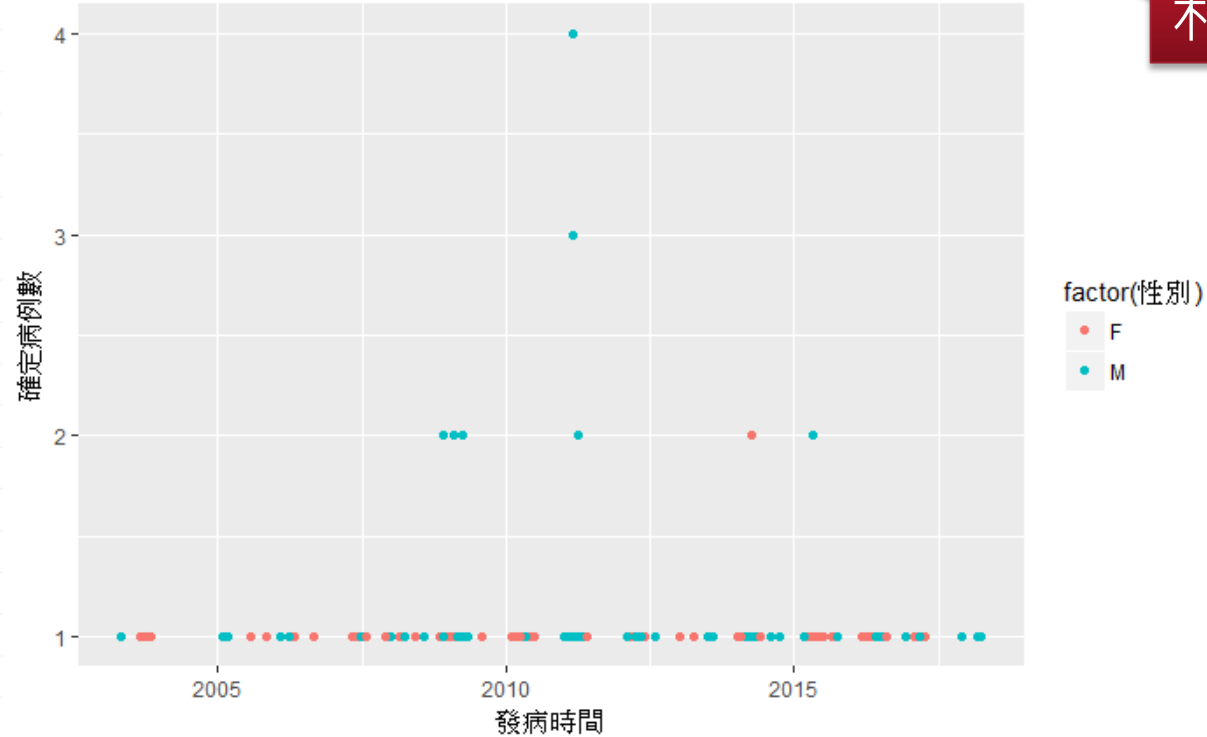
p1 + geom_point(color='blue', size = 5, shape = 19)



根據類別 (factor) 修改

```
p1 + geom_point(aes(color=factor(性別)))
```

利用Factor 指定類別



指定各類別的顏色

p1 +

```
geom_point(aes(color=factor(性別))) +
```

```
scale_color_manual(values = c("orange", "purple"))
```

指定類別顏色

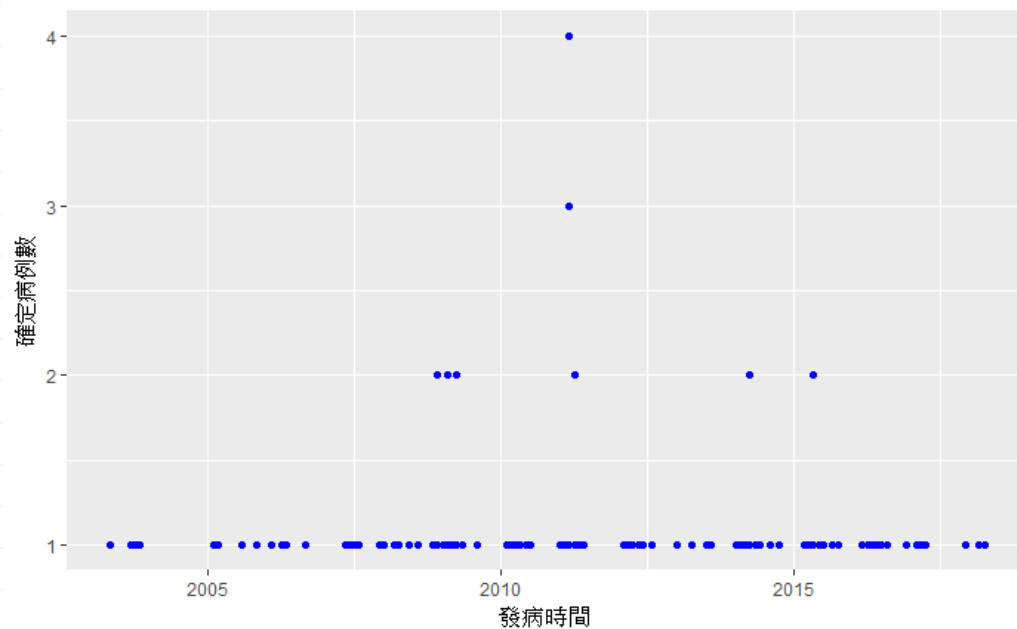


覆載原本的點樣式

p1 +

```
geom_point(color='red' ) +
```

```
geom_point(color='blue' )
```



修改原本點的顏色

修改顏色與性別

```
source('multiplot.R')
```

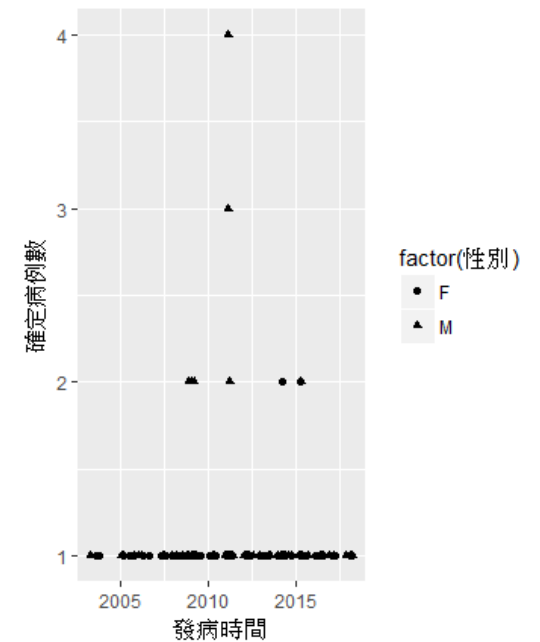
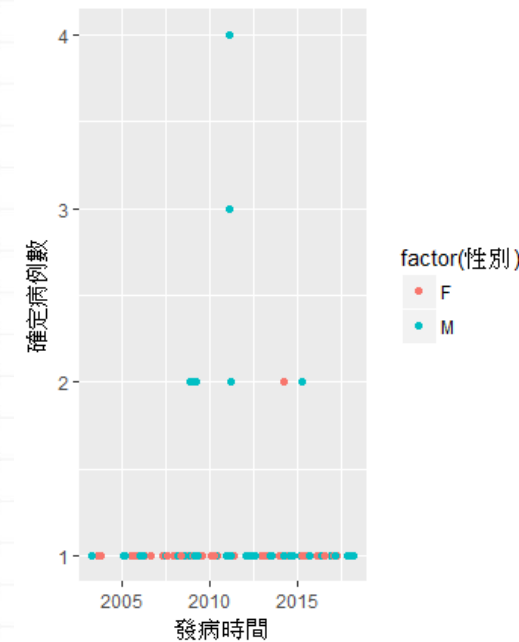
```
p2 <- p1 + geom_point(aes(color=factor(性別)))
```

```
p3 <- p1 + geom_point(aes(shape=factor(性別)))
```

```
multiplot(p2, p3, cols=2)
```

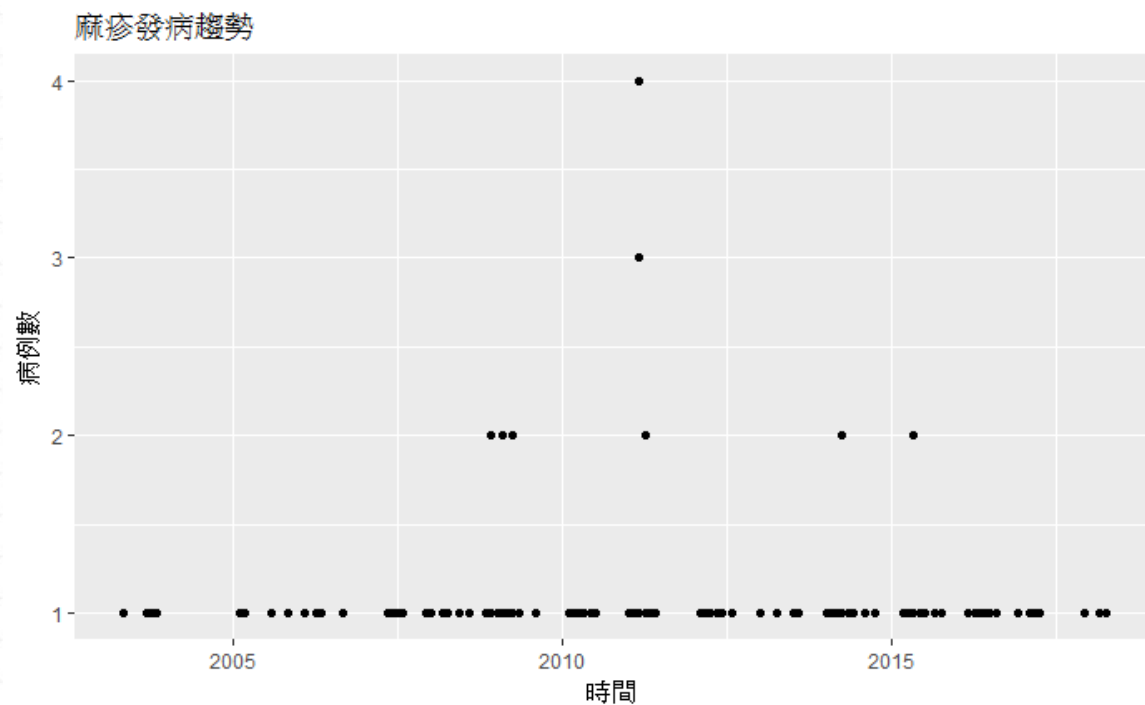
multiplot.R 可載自

https://raw.githubusercontent.com/ywchiu/cdc_course/master/script/multiplot.R



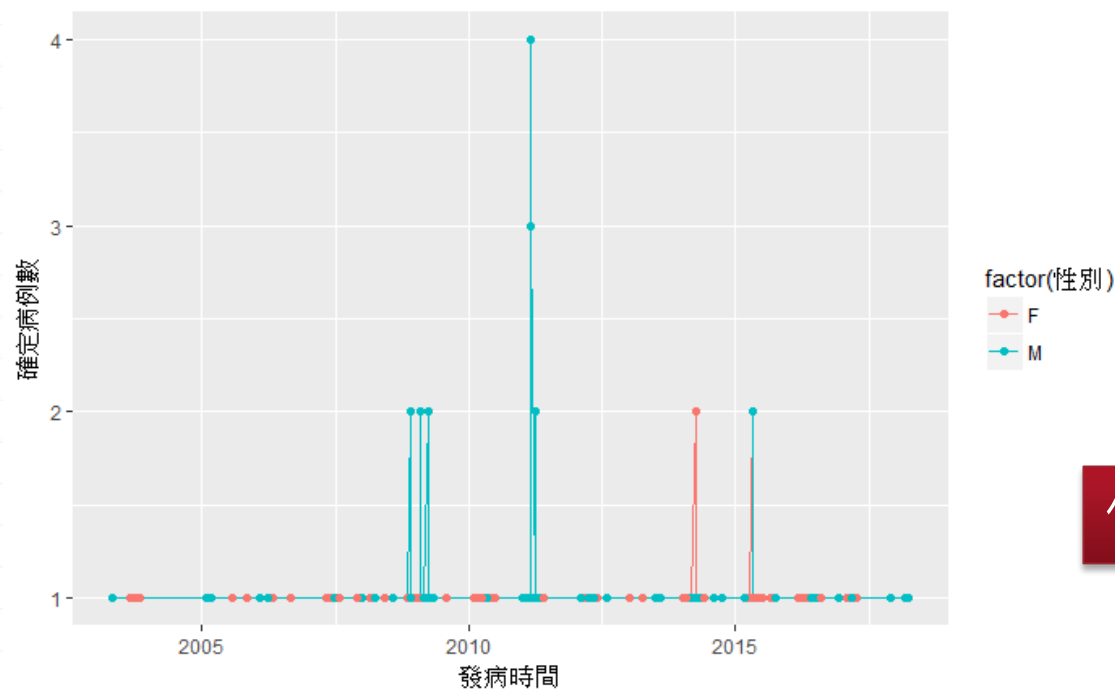
修改標籤名稱與標題名稱

```
p1 +  
  geom_point() +  
  xlab("時間") +  
  ylab("病例數") +  
  ggtitle("麻疹發病趨勢")
```



圖層疊加

```
p1 + geom_point(aes(color=factor(性別))) + geom_line(aes(color=factor(性別)))  
p1 + geom_line(aes(color=factor(性別))) + geom_point(aes(color=factor(性別)))
```

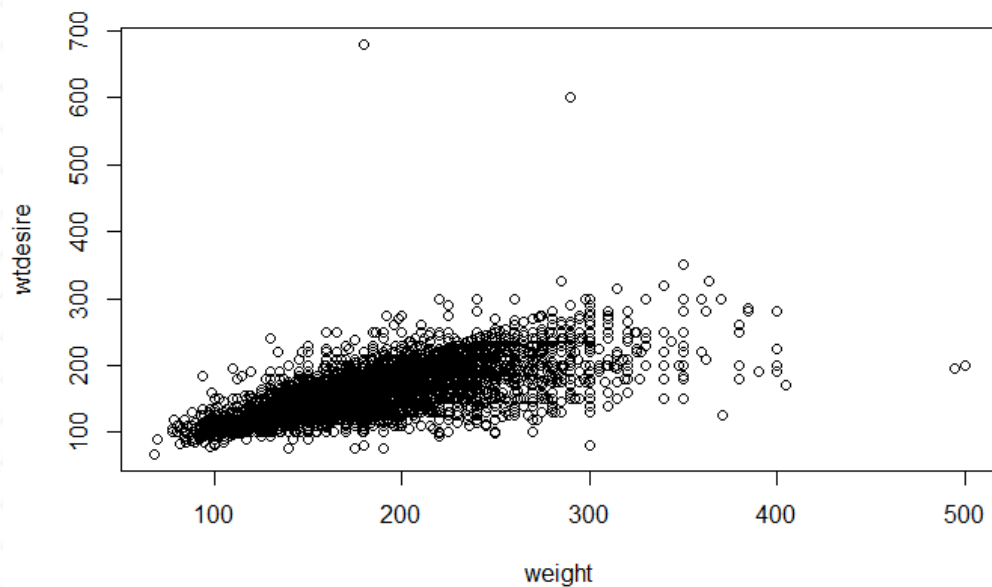


使用geom_line 增添直線

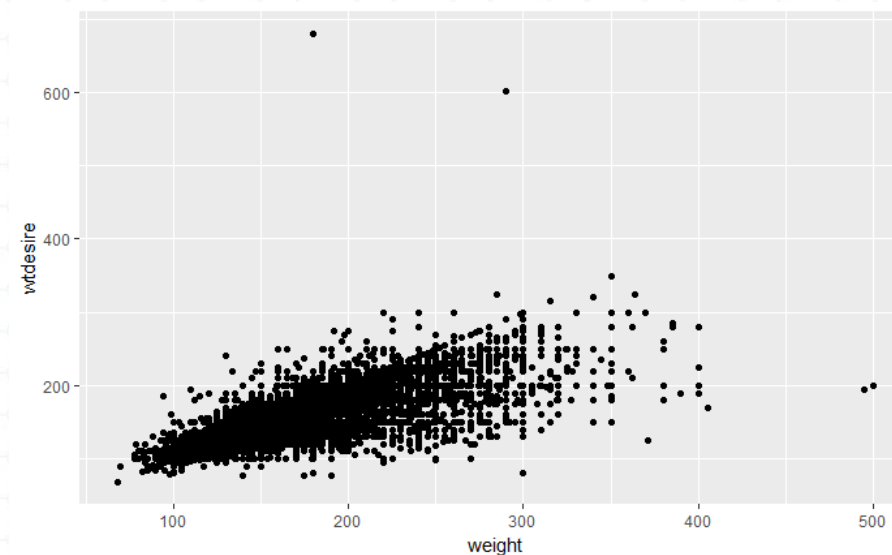
幾何圖案 (Geometries)

散佈圖 (Scatter Plot)

```
load('cdc.RData')  
plot(wtdesired ~ weight, data = cdc)
```

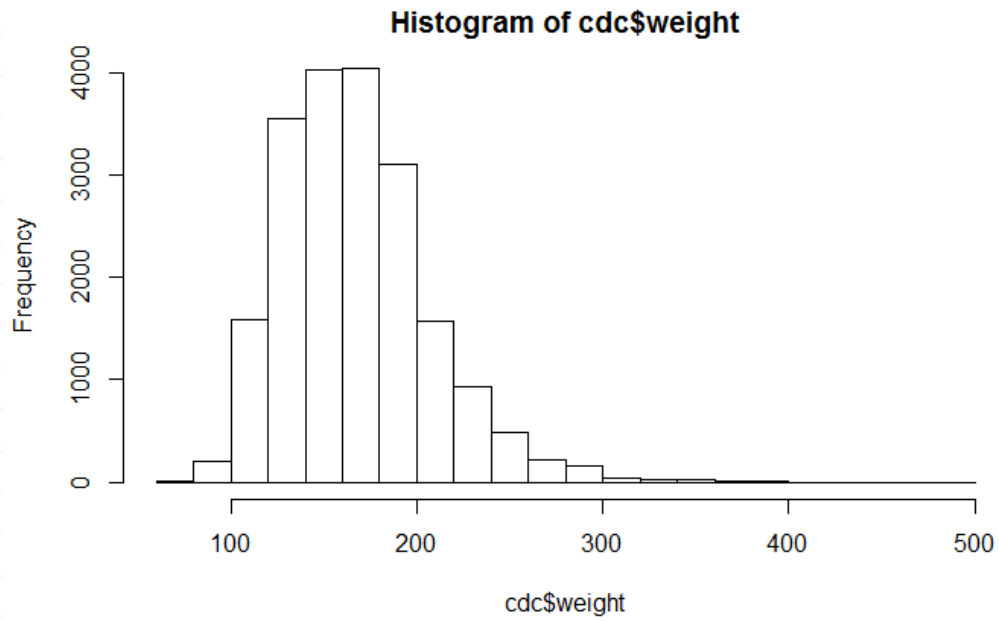


```
load('cdc.RData')  
ggplot(data=cdc, aes(x=weight, y =  
wtdesired)) + geom_point()
```

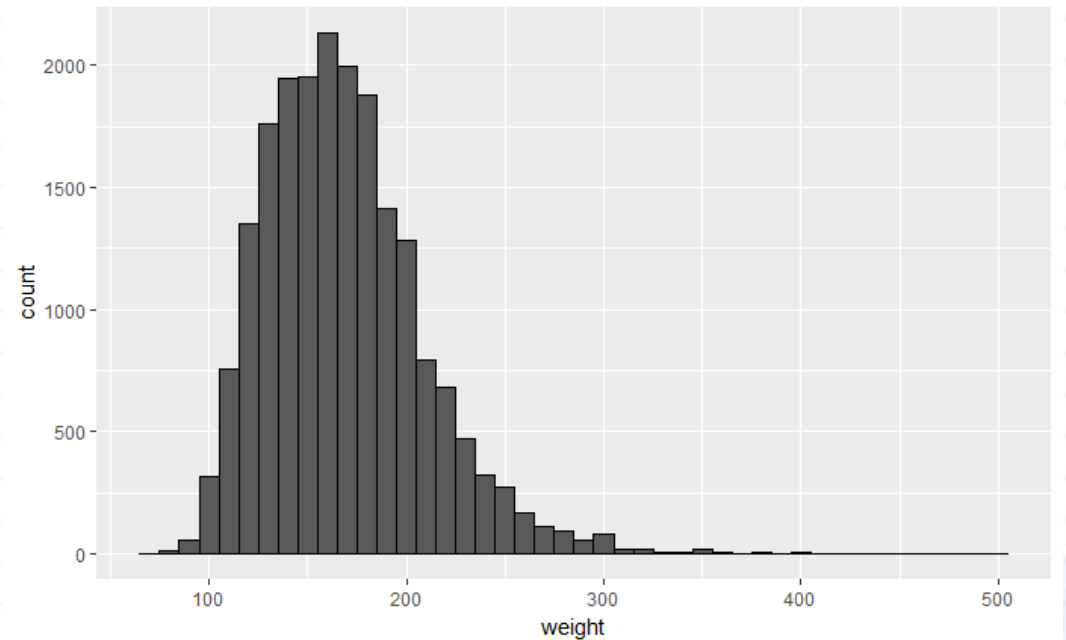


直方圖 (Histogram)

```
hist(cdc$weight)
```

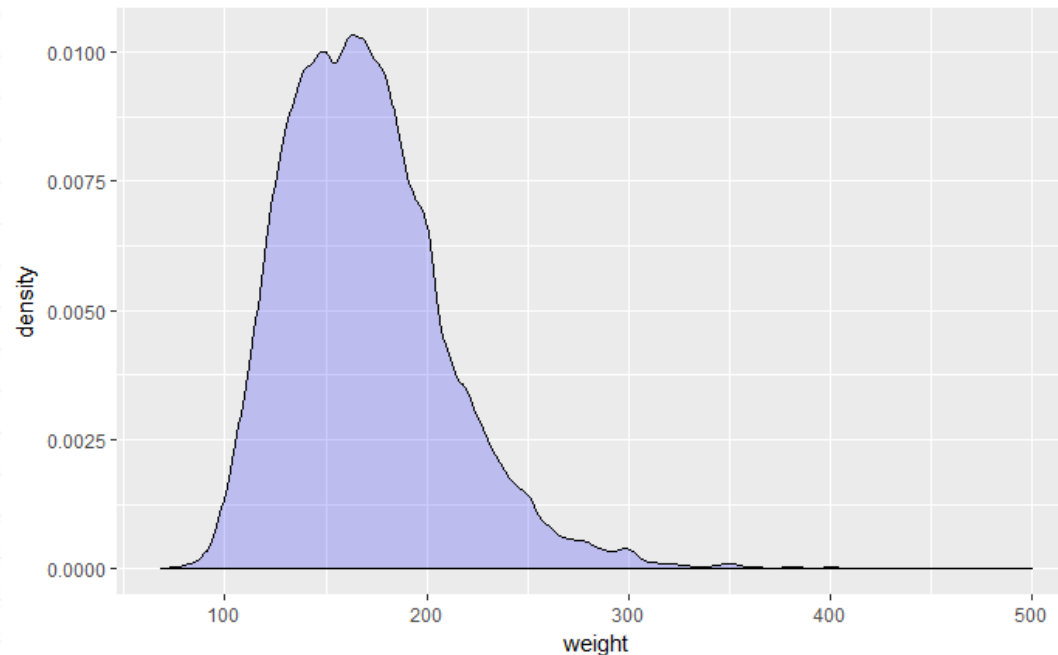


```
histogram <- ggplot(data=cdc, aes(x=weight))  
histogram + geom_histogram(binwidth=10,  
color="black")
```



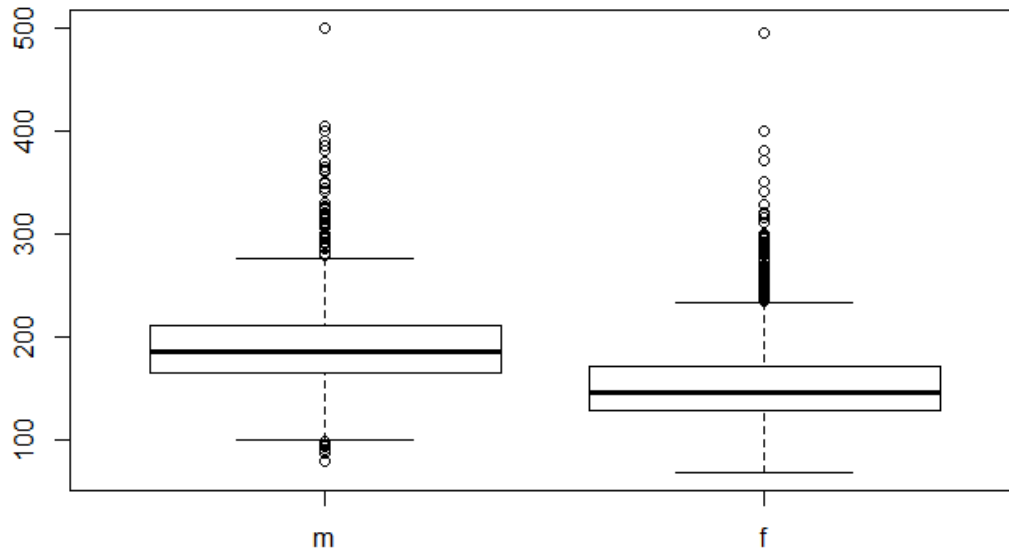
密度圖 (Density Plot)

```
density <- ggplot(data=cdc, aes(x=weight))  
density + geom_density(stat="density", alpha=0.2, fill='blue')
```



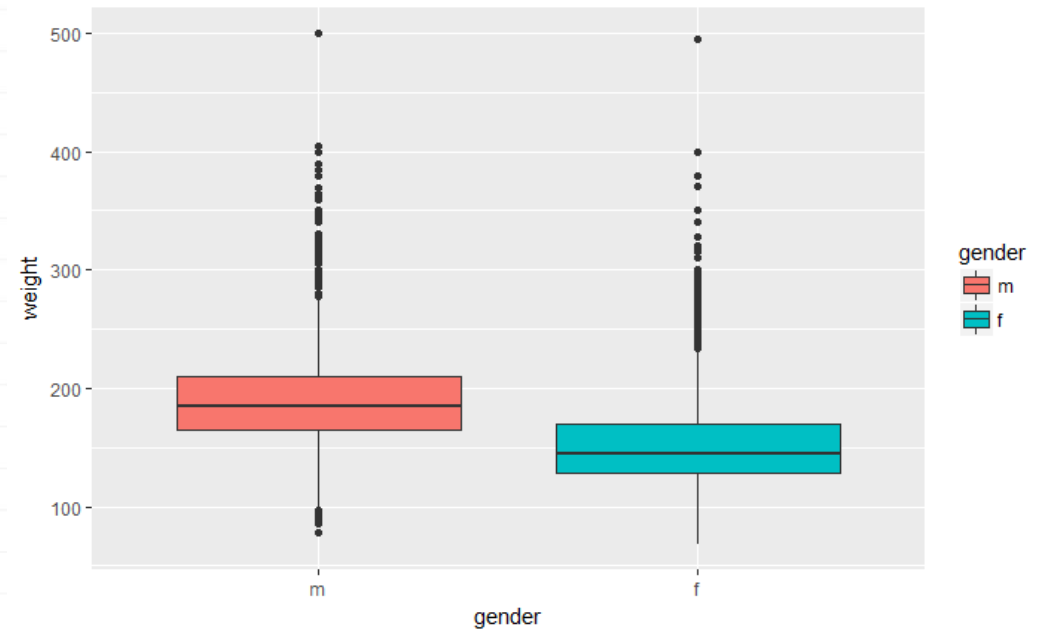
箱型圖 (Box Plot)

```
boxplot(cdc$weight~ cdc$gender)
```



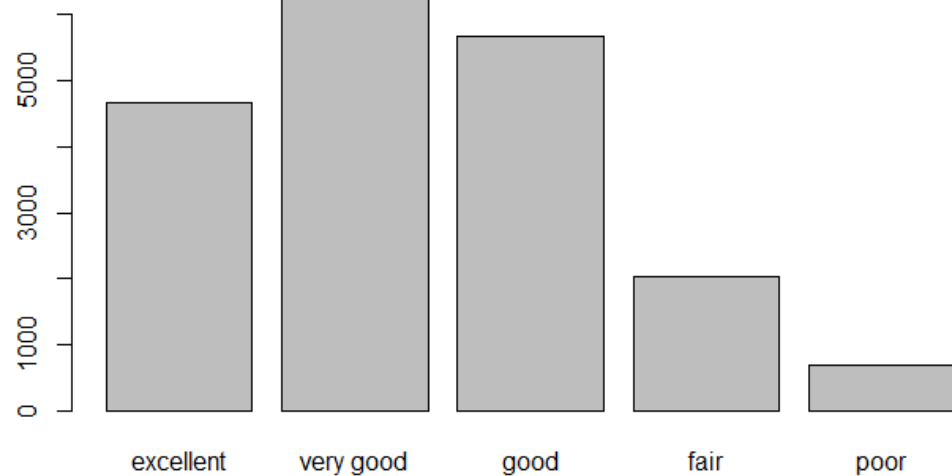
```
box <- ggplot(data=cdc,  
aes(x=gender, y=weight))
```

```
box + geom_boxplot(aes(fill=gender ))
```

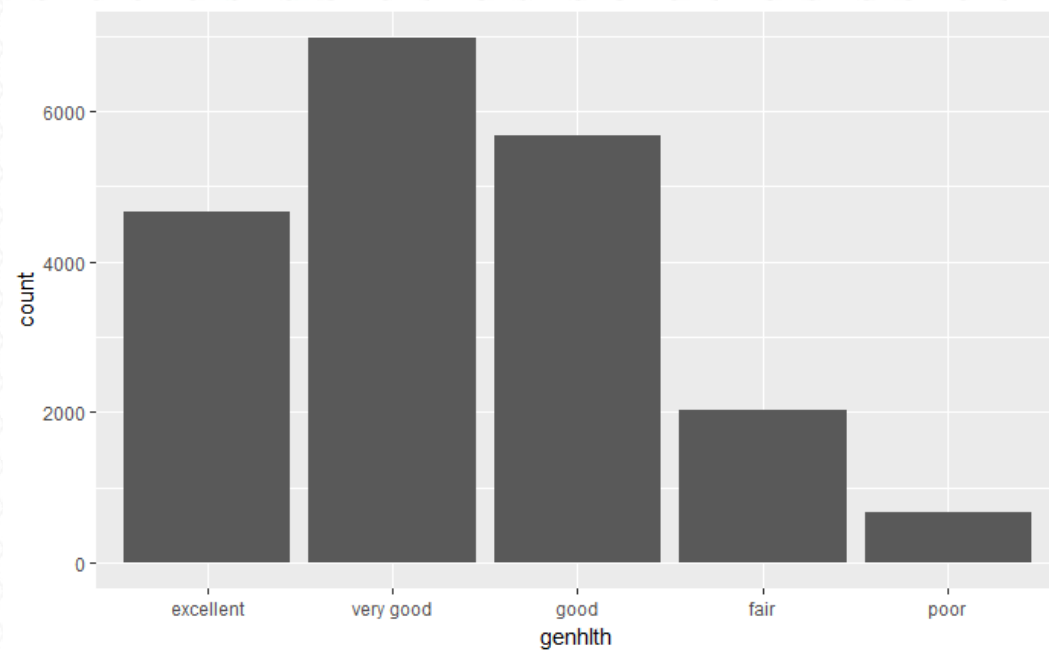


長條圖 (Bar Plot)

```
barplot(table(cdc$genhlth))
```

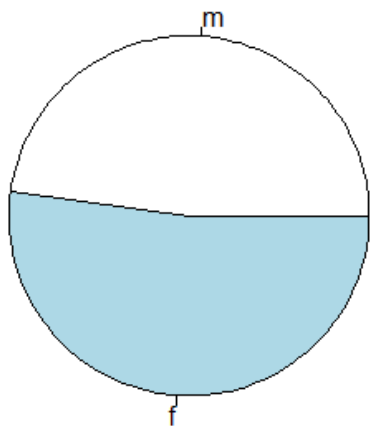


```
bar <- ggplot(data=cdc, aes(x=genhlth))  
bar + geom_bar()
```

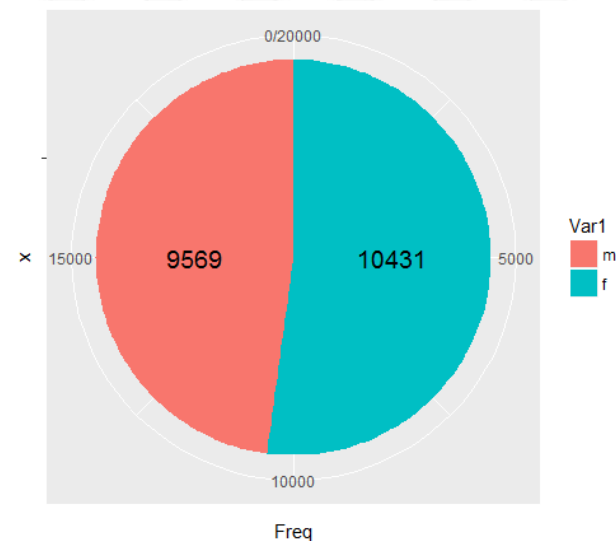


圓餅圖 (Pie Chart)

```
pie(table(cdc$gender))
```



```
pie <- ggplot(cdc_sex, aes(x="", y=Freq,  
  fill=Var1 )) + geom_bar(width=1, stat =  
  "identity") + geom_text(aes(label=Freq, y  
    =c(15000, 5000) ), size=5)  
pie + coord_polar(theta="y", start = 0)
```

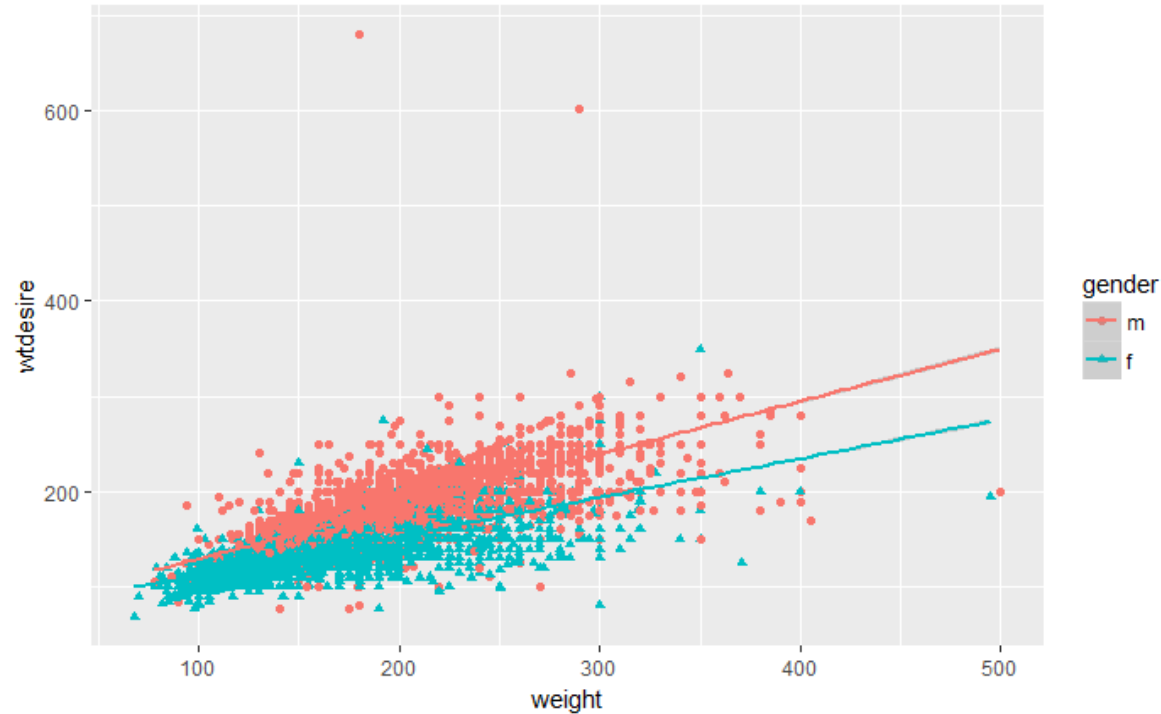


統計轉換 (Statistics)

增添迴歸線 (Smooth)

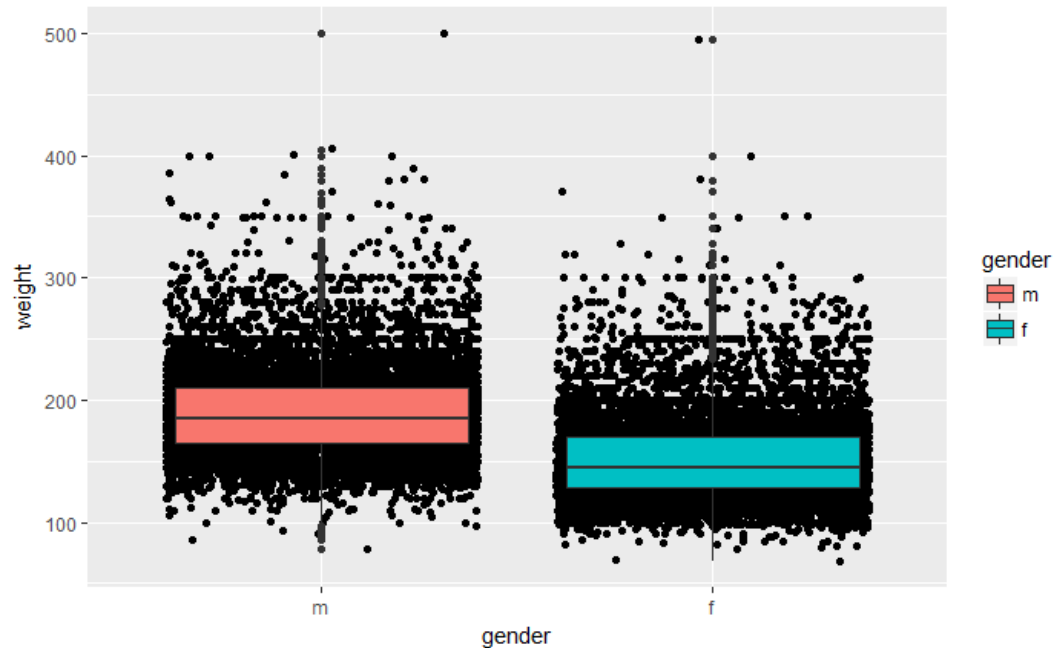
```
smooth <- ggplot(data=cdc, aes(x=weight, y=wtdesired, color=gender)) +  
  geom_point(aes(shape=gender), size=1.5)  
smooth + geom_smooth(method="lm")
```

可替換方法為
method="loess"



增添資料點到箱型圖上

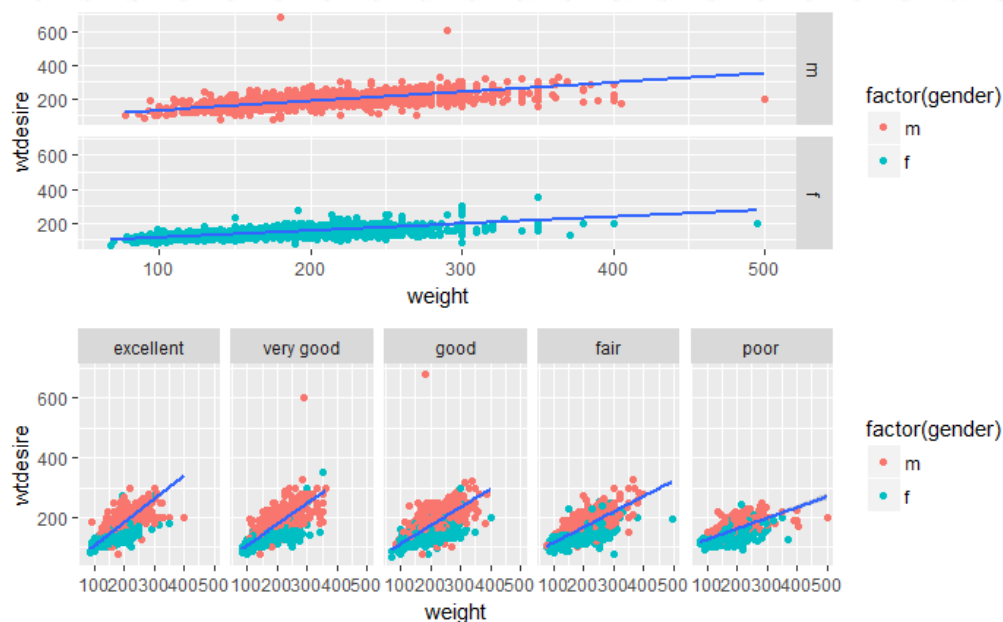
```
box <- ggplot(data=cdc, aes(x=gender, y=weight))  
box + geom_jitter() + geom_boxplot(aes(fill=gender ))
```



繪圖面 (Facets)

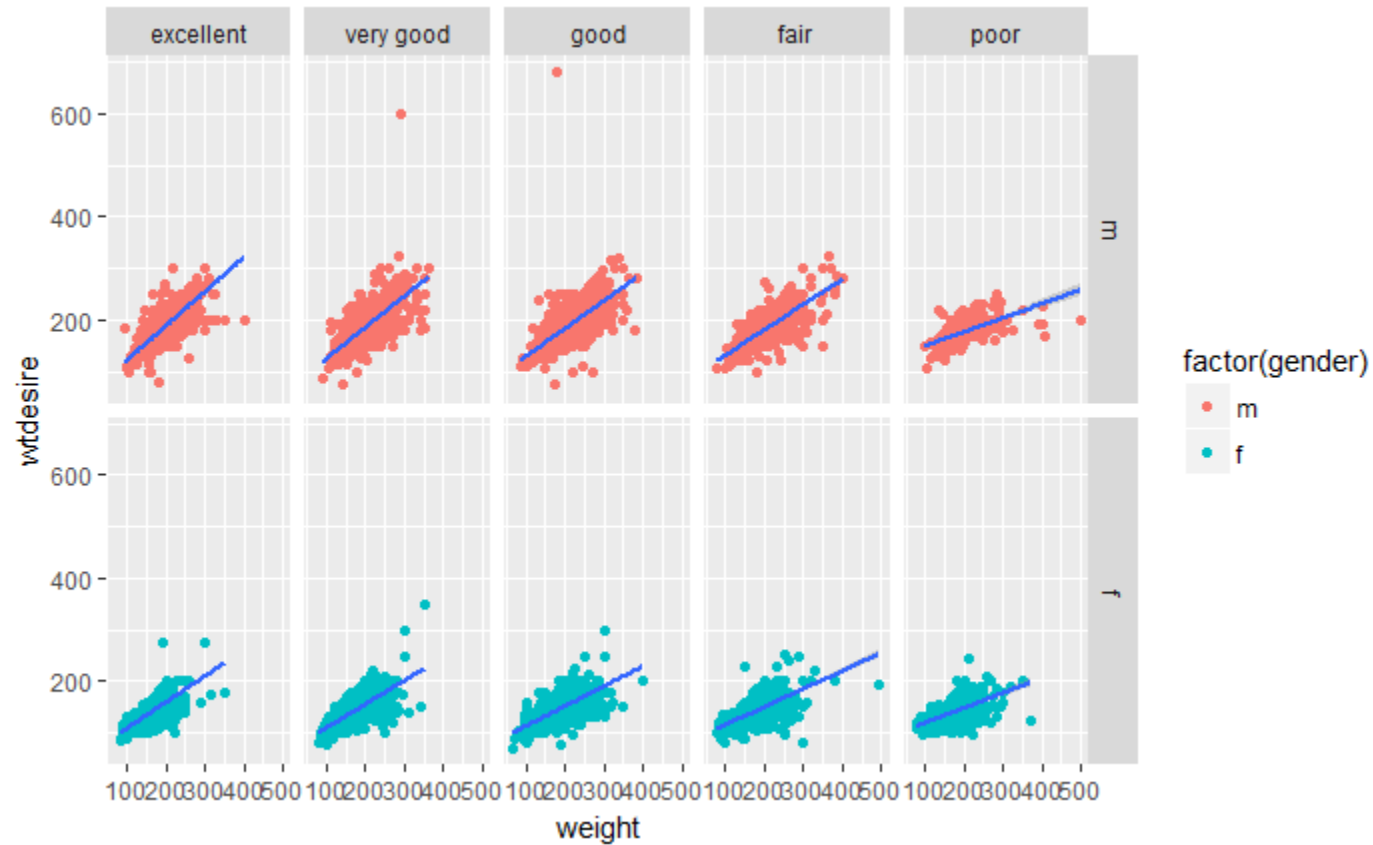
產生多張子圖

```
w <- ggplot(data=cdc, aes(x=weight, y = wtdesired)) + geom_point(aes(color=factor(gender))) +  
  geom_smooth(method = 'lm')  
w1 <- w + facet_grid(gender~.)  
w2 <- w + facet_grid(.~genhlth)  
multiplot(w1, w2, cols = 1)
```



產生多張子圖 (2)

w + facet_grid(gender~genhlth)



座標系統 (Coordinates)

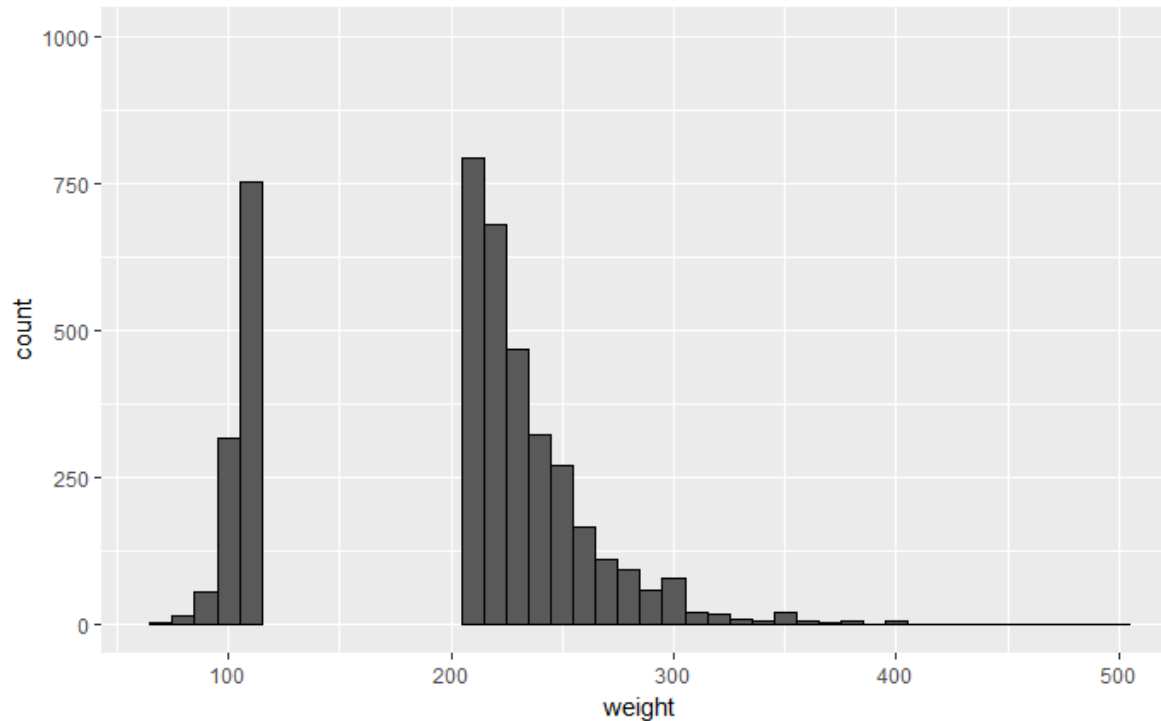
局部放大

```
w <- ggplot(data=cdc, aes(x=weight, y = wtdesired)) +  
  geom_point(aes(color=factor(gender))) + geom_smooth(method = 'lm')  
w + xlim(100,200) + ylim(100,200)
```



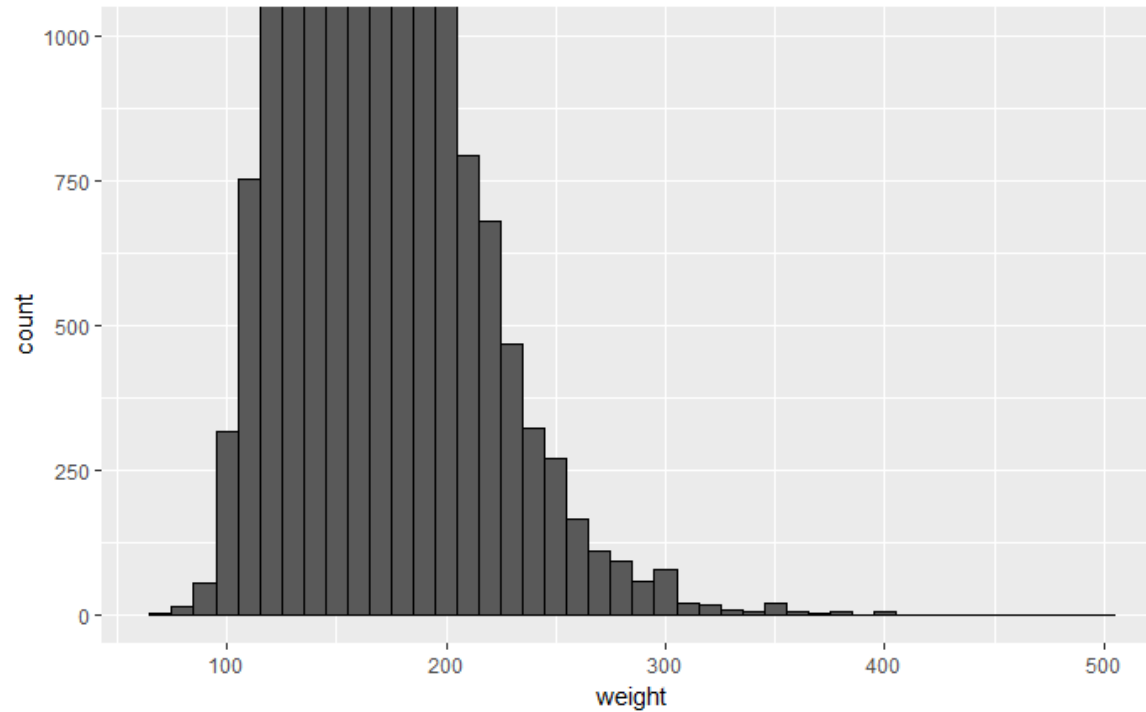
並非調整ylim 就可以調整所有圖表

```
histogram <- ggplot(data=cdc, aes(x=weight))  
histogram + geom_histogram(binwidth=10, color="black") + ylim(0,1000)
```



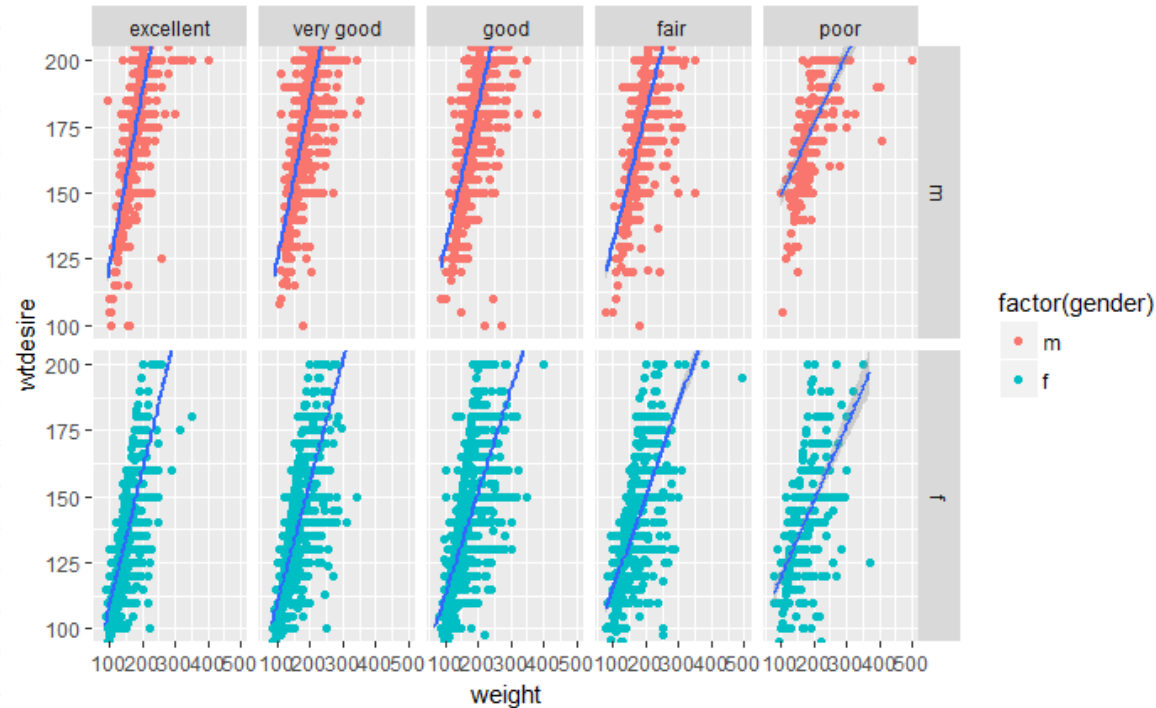
調整座標

```
histogram + geom_histogram(binwidth=10, color="black") +  
coord_cartesian( ylim = c(0,1000) )
```



調整座標 (2)

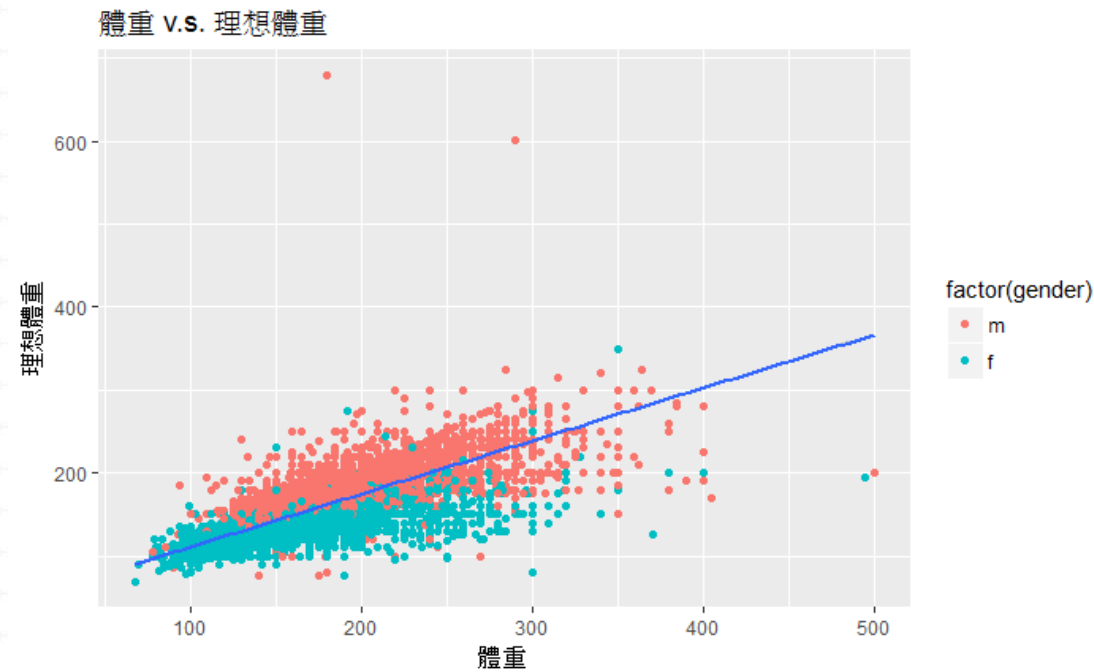
```
w <- ggplot(data=cdc, aes(x=weight, y = wtdesired)) + geom_point(aes(color=factor(gender))) +  
geom_smooth(method = 'lm')  
w + facet_grid(gender~genhlth) + coord_cartesian( ylim = c(100,200) )
```



主題 (Theme)

增加圖表標籤敘述

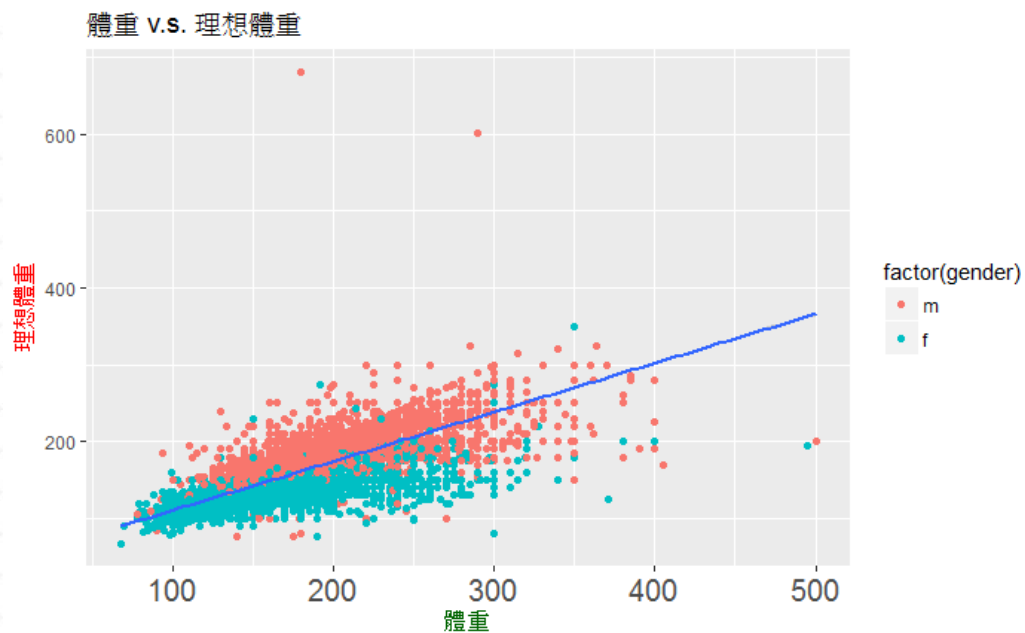
```
w <- ggplot(data=cdc, aes(x=weight, y = wtdesired)) +  
  geom_point(aes(color=factor(gender))) + geom_smooth(method = 'lm')  
w + xlab('體重') + ylab('理想體重') + ggtitle('體重 v.s. 理想體重')
```



修改標籤樣式

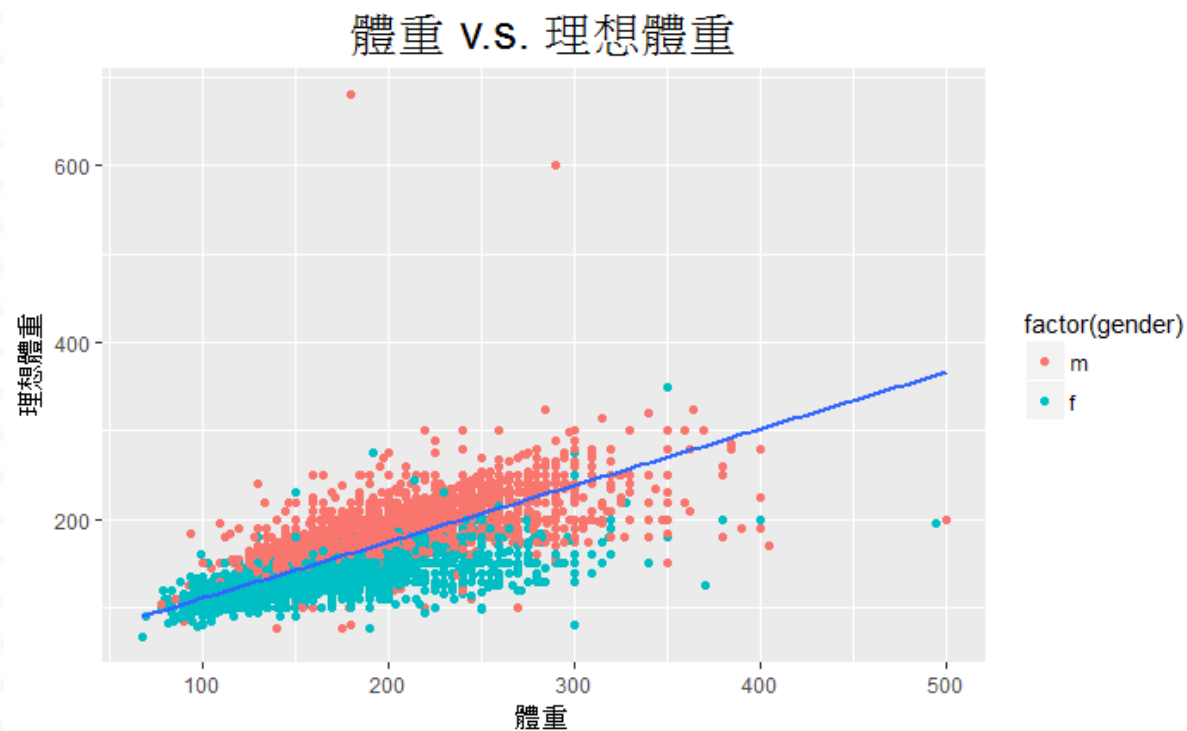
```
w + xlab('體重') + ylab('理想體重') + ggtitle('體重 v.s. 理想體重') +  
  theme(axis.title.x = element_text(color = 'DarkGreen', size = 10),  
        axis.title.y = element_text(color = 'Red', size = 10),  
        axis.text.x = element_text(size = 15),  
        axis.text.y = element_text(size = 15))
```

善用?theme 觀看選項



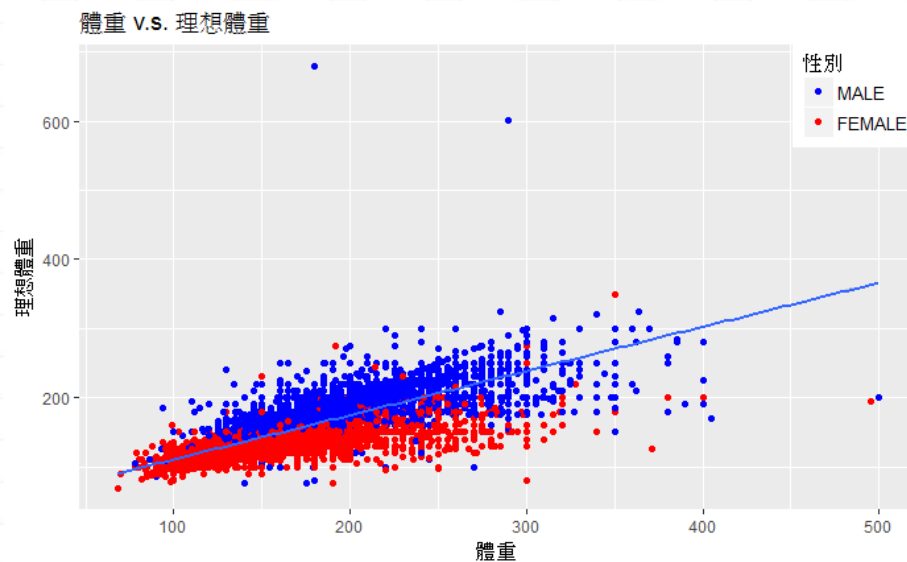
調整標題

```
w + xlab('體重') + ylab('理想體重') + ggtitle('體重 v.s. 理想體重') +  
  theme(plot.title = element_text(size = 20, hjust = 0.5))
```



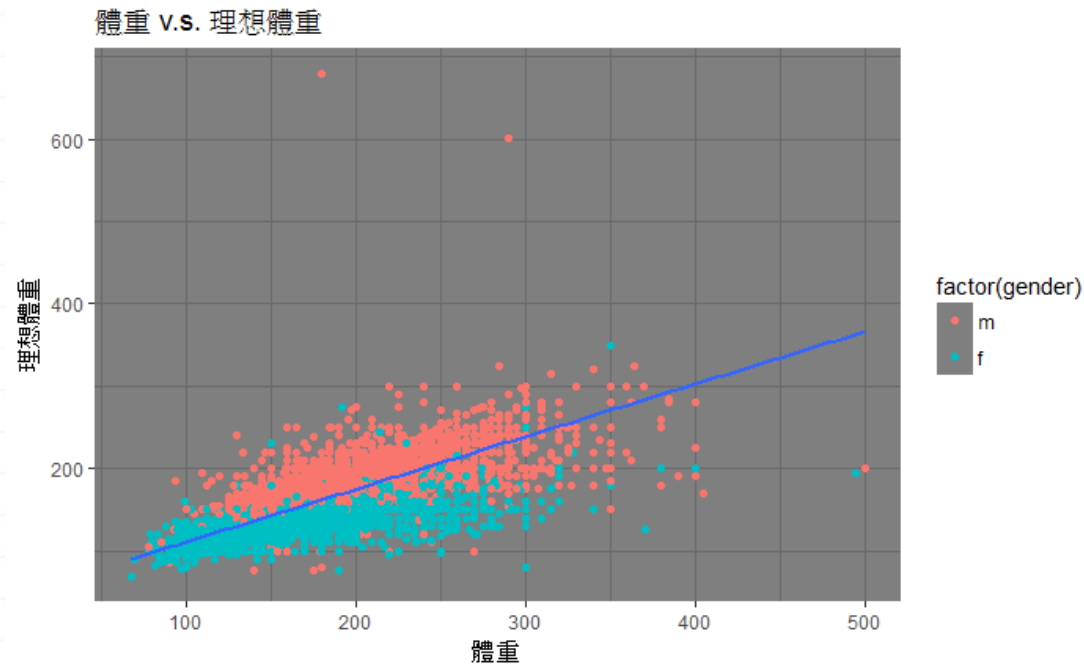
修改圖標

```
w + xlab('體重') + ylab('理想體重') + ggtitle('體重 v.s. 理想體重') +  
  scale_color_manual(name='性別', labels = c("MALE", "FEMALE"), values = c("blue", "red")) +  
  theme(legend.text = element_text(size = 10),  
        legend.title = element_text(size = 10),  
        legend.position = c(1,1),  
        legend.justification = c(1,1))
```



修改圖背景樣式

```
w <- ggplot(data=cdc, aes(x=weight, y = wtdesired)) + geom_point(aes(color=factor(gender))) +  
geom_smooth(method = 'lm')  
w1 <- w + xlab('體重') + ylab('理想體重') + ggtitle('體重 v.s. 理想體重')  
w1 + theme_dark()
```



匯出圖片

使用ggsave 存儲圖片

```
w <- ggplot(data=cdc, aes(x=weight, y = wtdesired)) + geom_point(aes(color=factor(gender))) +  
geom_smooth(method = 'lm')  
w1 <- w + xlab('體重') + ylab('理想體重') + ggtitle('體重 v.s. 理想體重')  
w1 + theme_dark()  
ggsave("plot1.png")  
ggsave(w1, file="plot2.png")  
ggsave(w1, file="plot3.png", width=6, height=4)
```




ggmap

安裝ggmap

■ 安裝ggmap

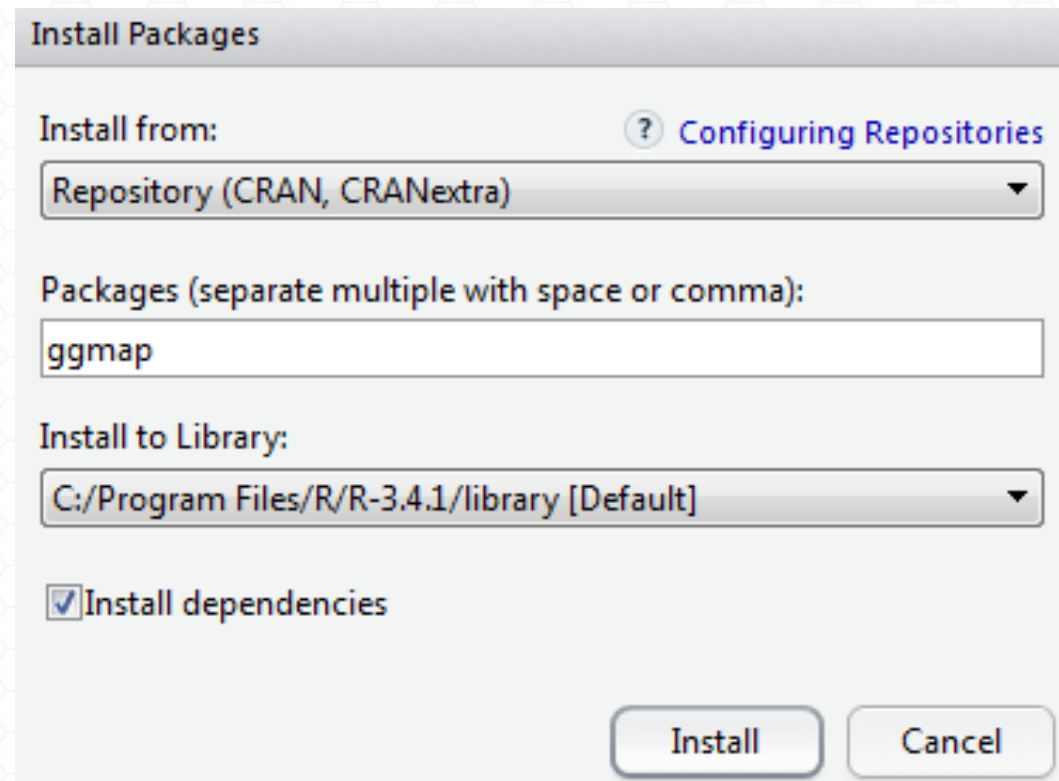
□ `install.packages("ggmap")`

■ 使用ggmap

□ `library(ggmap)`

■ 觀看說明頁

□ `help(package='ggmap')`

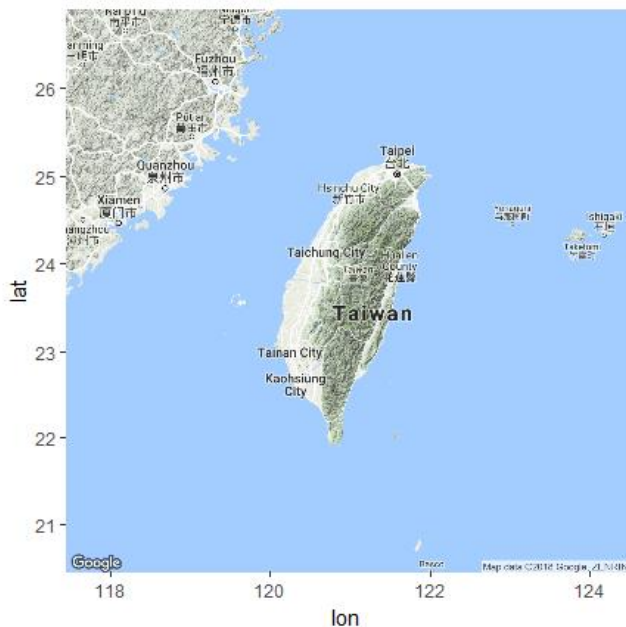


顯示台灣地圖

```
library(ggmap)
```

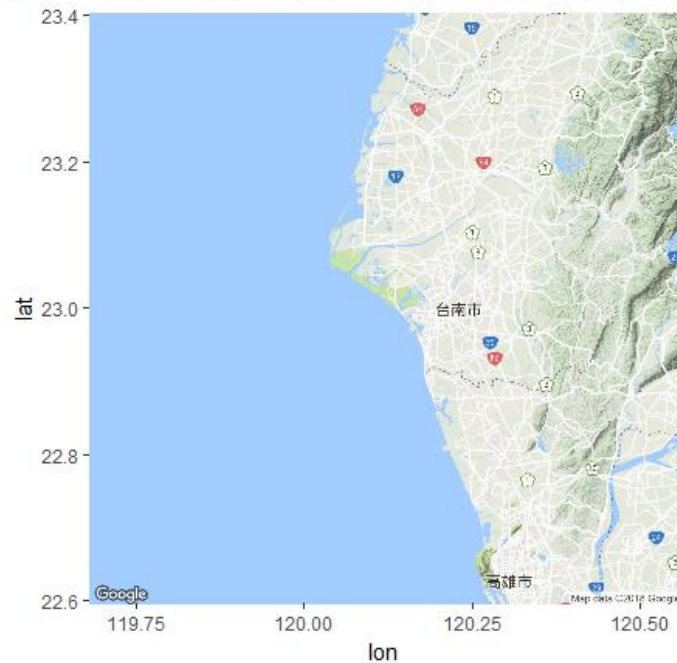
```
map <- get_map(location = 'Taiwan', zoom = 7)
```

```
ggmap(map)
```



顯示特定經緯度地圖

```
map <- get_map(location = c(lon = 120.12, lat = 23.00),  
  zoom = 10, language = "zh-TW")  
ggmap(map)
```



蒐集資料

- 從疫情中心的開放資料網站蒐集到登革熱1998年起每日確定病例統計數字

```
library(readr)
```

```
Dengue <- read_csv("Dengue_Daily")
```

```
head(Dengue)
```

將資料點繪製於地圖

```
map <- get_map(location = c(lon = 120.246100, lat = 23.121198),  
  zoom = 10, language = "zh-TW")  
ggmap(map, darken = c(0.5, "white")) +  
  geom_point(aes(x = 最小統計區中心點X, y = 最小統計區中心點Y),  
    color = "red", data = Dengue_Daily)
```

其他繪圖套件

高級繪圖套件介紹

- plotly、googleVis
- 其他：
 - Ggplot2
 - Ggvis
 - 地圖：RgoogleMaps
 - 社會網路: igraph
 - 熱繪圖：heatmap
 - 類別資料視覺化：vcd
 - 多變量條件式繪圖：lattice
 - 股票：quantmod::chartSeries

plotly

- 可使用Python,R,Matlab,Javascript撰寫的開源繪圖套件，快速、方便
- 產生HTML檔

Donut Chart

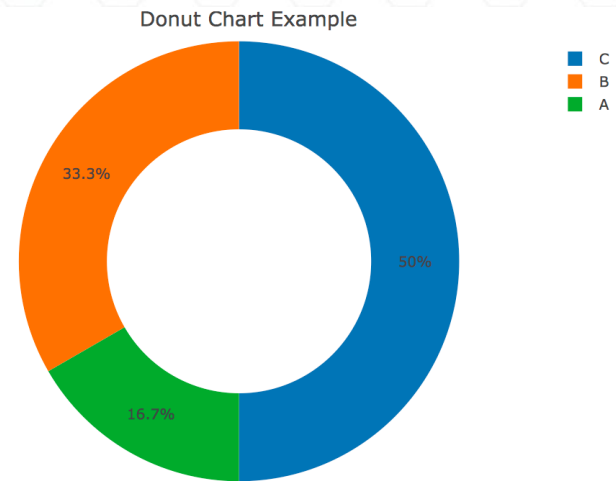
■ 函數：plot_ly (type = "pie")

```
library(plotly)
```

```
ds <- data.frame(labels = c("A", "B", "C"),  
                  values = c(10, 20, 30))
```

```
plot_ly(ds, labels = ds$labels, values = ds$values, type = "pie", hole=0.6) %>%
```

```
layout(title = "Donut Chart Example")
```



Area Chart

■ 函數：plot_ly ()

```
library(plotly)
```

```
month<- c(1,2,3,4,5)
```

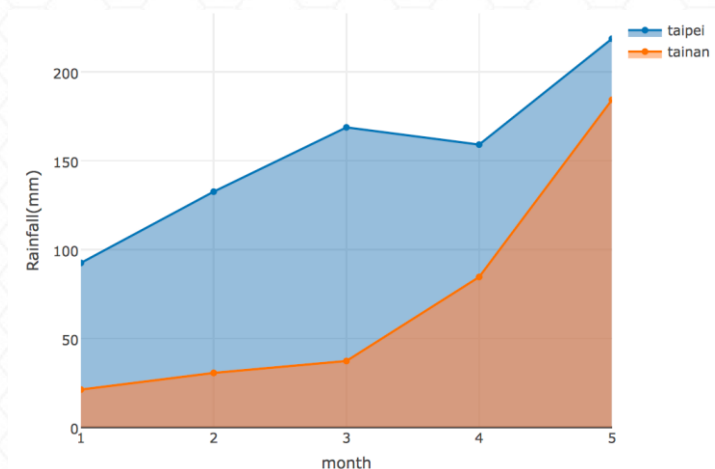
```
taipei <- c(92.5,132.6,168.8,159.1,218.7)
```

```
tainan <- c(21.2, 30.6, 37.3, 84.6, 184.3)
```

```
plot_ly(x = month, y = taipei, fill = "tozeroy", name="taipei",type='scatter', mode= 'markers') %>%
```

```
add_trace(x = month, y = tainan, fill = "tozeroy",name="tainan") %>%
```

```
layout(yaxis = list(title = 'rainfall'))
```

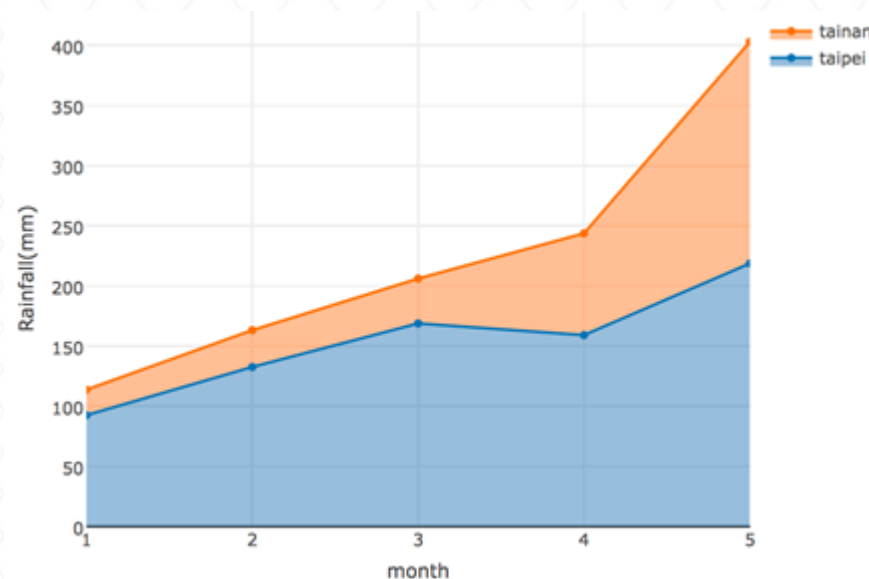


Area Chart (Stacked)

■ 函數：plot_ly ()

■ 範例：

```
library(plotly)
month<- c(1,2,3,4,5)
taipei<- c(92.5,132.6,168.8,159.1,218.7)
tainan <- c(21.2, 30.6, 37.3, 84.6, 184.3)
total <- taipei + tainan
y <- list(title="Rainfall")
plot_ly(x = month, y = taipei, fill = "tozeroy", name="taipei",type='scatter', mode= 'markers') %>%add_trace(x = month, y
= total, fill = "tonexty", name="tainan") %>%
layout(yaxis = y)
```



Bubble Chart

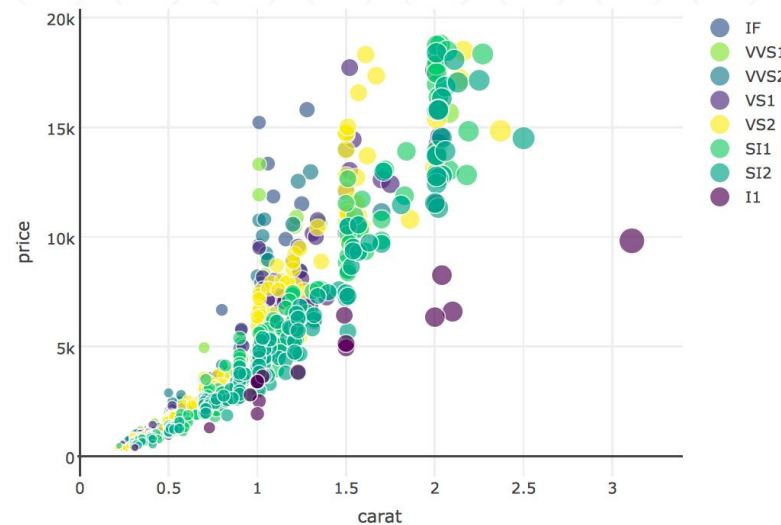
■ 函數：plot_ly (mode="markers")

```
data("diamonds")
```

```
diamonds
```

```
d <- diamonds[sample(nrow(diamonds), 1000), ]
```

```
plot_ly(d, x = d$carat, y = d$price, color = d$clarity, type='scatter', mode= 'markers', size = d$carat, text= paste("Clarity", d$clarity))
```



Multiple Plots

■ 函數：subplot(margin,nrows,...)

data("economics")

```
p <- subplot(
```

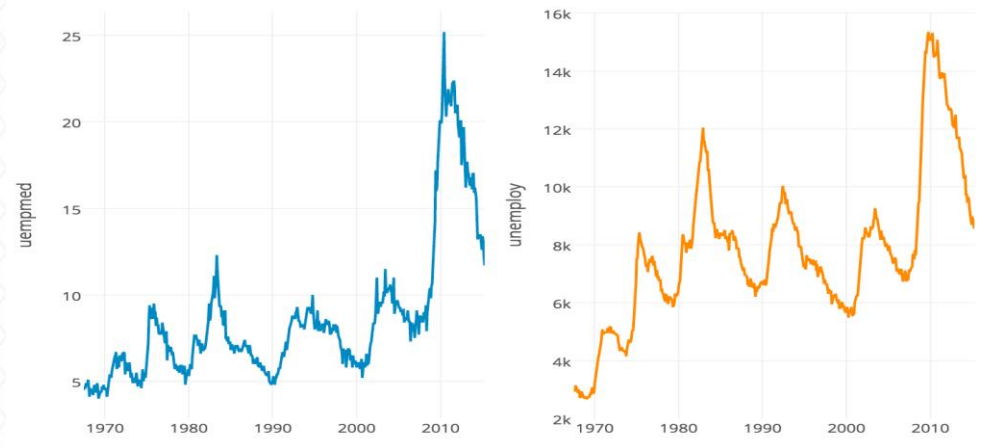
```
plot_ly(economics, x = economics$date, y = economics$unemploy, type='scatter', mode = 'lines'),
```

```
plot_ly(economics, x = economics$date, y = economics$uempmed, type='scatter', mode = 'lines'),
```

```
margin=0.05
```

```
)
```

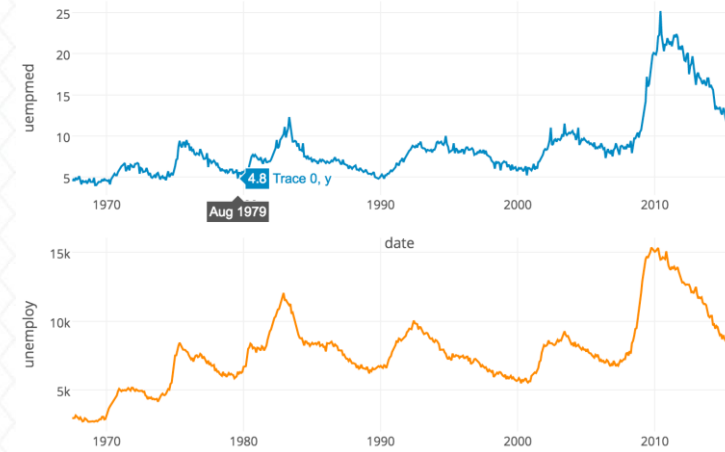
```
p %>% layout(showlegend=FALSE)
```



Multiple Plots

■ 函數：subplot(margin,nrows,...)

```
p <- subplot(  
  plot_ly(economics, x = economics$date, y = economics$unemploy, type='scatter', mode = 'lines'),  
  plot_ly(economics, x = economics$date, y = economics$uempmed, type='scatter', mode = 'lines'),  
  margin=0.05,  
  nrows = 2  
)  
p %>% layout(showlegend=FALSE)
```



The background features a light gray hexagonal grid pattern. Overlaid on this is a series of concentric, semi-transparent circles in shades of light blue and white. The circles have a slightly irregular, hand-drawn appearance. A dark blue horizontal line runs across the top of the image, and a darker blue, textured horizontal band runs across the bottom.

THANK YOU