

R PROGRAMMING

Part 4



ผู้ช่วยศาสตราจารย์ ดร. อัจฉmann รัตน์เลิศนุสรณ์
สาขาสถิติประยุกต์ คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

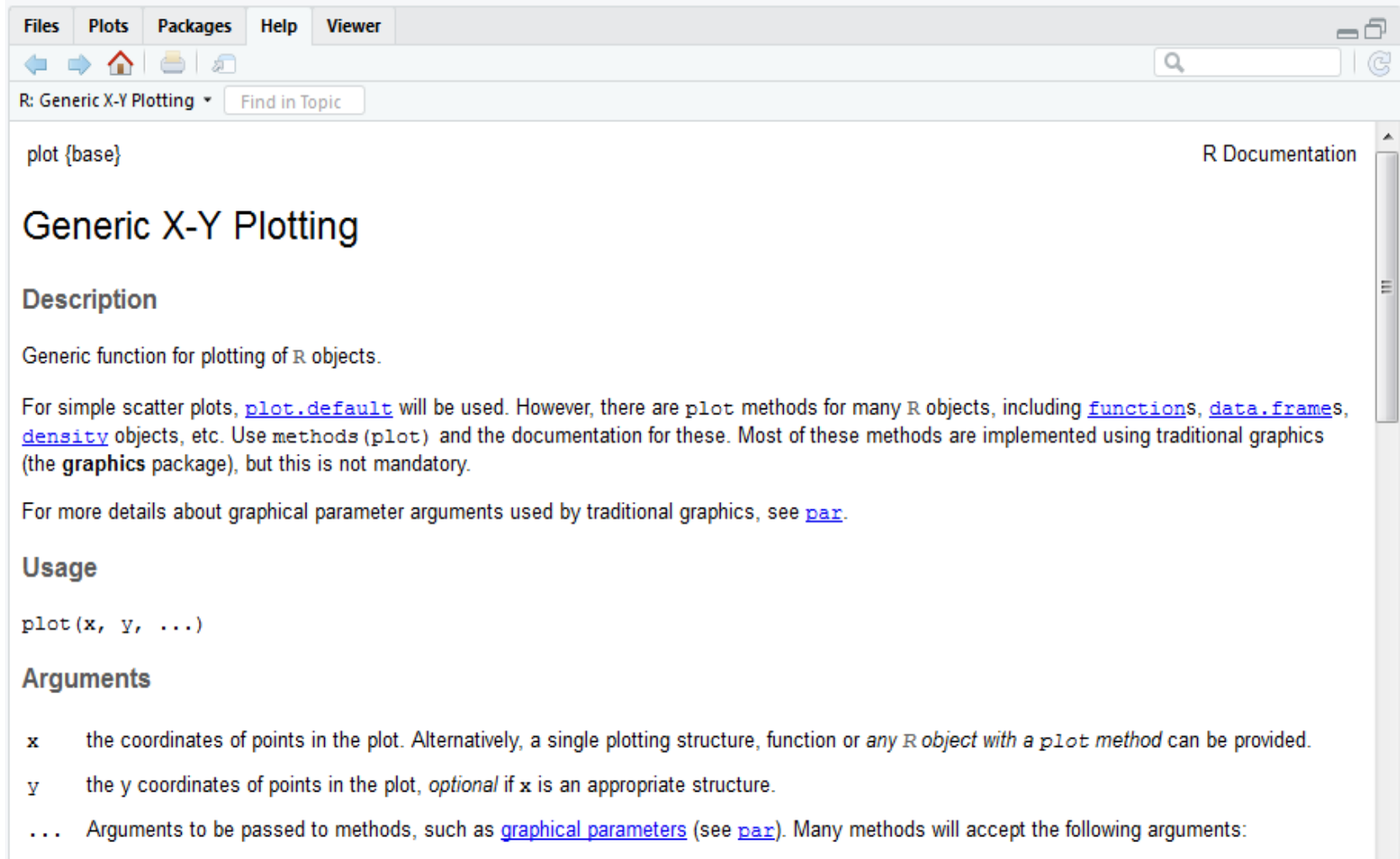
สารบัญ

Contents

- การสร้างกราฟด้วยคำสั่ง plot
- การใช้ฟังก์ชัน lines
- แบบฝึกหัด การสร้างกราฟการแจกแจงปกติมาตรฐาน
- เฉลย

การสร้างกราฟด้วยคำสั่ง plot

- ฟังก์ชัน `plot()` มีรูปแบบดังนี้



The screenshot shows the R Documentation interface for the `plot` function. The top navigation bar includes tabs for Files, Plots, Packages, Help, and Viewer. Below the navigation bar is a search bar and a "Find in Topic" button. The main content area is titled "Generic X-Y Plotting" and includes a "Description" section, a "Usage" section, and an "Arguments" section. The "Description" section explains that `plot` is a generic function for plotting R objects, with `plot.default` used for simple scatter plots. The "Usage" section shows the function signature `plot(x, y, ...)`. The "Arguments" section lists the parameters: `x` (coordinates of points), `y` (y coordinates), and `...` (arguments to be passed to methods).

Files Plots Packages Help Viewer

R: Generic X-Y Plotting Find in Topic

plot {base} R Documentation

Generic X-Y Plotting

Description

Generic function for plotting of R objects.

For simple scatter plots, [plot.default](#) will be used. However, there are `plot` methods for many R objects, including [functions](#), [data.frames](#), [density](#) objects, etc. Use `methods(plot)` and the documentation for these. Most of these methods are implemented using traditional graphics (the **graphics** package), but this is not mandatory.

For more details about graphical parameter arguments used by traditional graphics, see [par](#).

Usage

```
plot(x, y, ...)
```

Arguments

x the coordinates of points in the plot. Alternatively, a single plotting structure, function or *any R object with a `plot` method* can be provided.

y the y coordinates of points in the plot, *optional* if **x** is an appropriate structure.

... Arguments to be passed to methods, such as [graphical parameters](#) (see [par](#)). Many methods will accept the following arguments:

การสร้างกราฟด้วยคำสั่ง plot

... Arguments to be passed to methods, such as [graphical parameters](#) (see [par](#)). Many methods will accept the following arguments:

`type`

what type of plot should be drawn. Possible types are

- "p" for points,
- "l" for lines,
- "b" for both,
- "c" for the lines part alone of "b",
- "o" for both 'overplotted',
- "h" for 'histogram' like (or 'high-density') vertical lines,
- "s" for stair steps,
- "S" for other steps, see 'Details' below,
- "n" for no plotting.

All other `types` give a warning or an error; using, e.g., `type = "punkte"` being equivalent to `type = "p"` for S compatibility. Note that some methods, e.g. [plot.factor](#), do not accept this.

`main`

an overall title for the plot: see [title](#).

`sub`

a sub title for the plot: see [title](#).

`xlab`

a title for the x axis: see [title](#).

`ylab`

a title for the y axis: see [title](#).

การสร้างกราฟด้วยคำสั่ง plot

- ฟังก์ชัน plot() มีรูปแบบดังนี้

```
plot(x, y = NULL, type = "p", xlim = NULL, ylim = NULL,  
     log = "", main = NULL, sub = NULL, xlab = NULL, ylab = NULL,  
     ann = par("ann"), axes = TRUE, frame.plot = axes,  
     panel.first = NULL, panel.last = NULL, asp = NA,  
     xgap.axis = NA, ygap.axis = NA,  
     ...)
```

โดยที่

`x, y` เป็นเวกเตอร์ข้อมูลในแกน `x` และแกน `y` ของรูปกราฟตามลำดับ

`type` เป็นชนิดของรูปกราฟ ค่า default กำหนดเป็น จุด(point: `p`)สามารถกำหนดใหม่
เป็นเส้น (line: `l`) หรือทั้งจุดและเส้น (both: `b`) หรือ อื่นๆ

`main` เป็นชื่อรูปกราฟ

`xlab, ylab` เป็นข้อความที่จะให้ปรากฏบนแกน `x` และแกน `y` ตามลำดับ

การเพิ่มเส้นเข้าไปที่คำสั่ง plot

lines {graphics}

R Documentation

Add Connected Line Segments to a Plot

Description

A generic function taking coordinates given in various ways and joining the corresponding points with line segments.

Usage

```
lines(x, ...)
```

```
## Default S3 method:
```

```
lines(x, y = NULL, type = "l", ...)
```

Arguments

x, y coordinate vectors of points to join.

type character indicating the type of plotting; actually any of the types as in [plot.default](#).

... Further graphical parameters (see [par](#)) may also be supplied as arguments, particularly, line type, lty, line width, lwd, color, col and for type = "b", pch. Also the line characteristics lend, ljoin and lmitre.

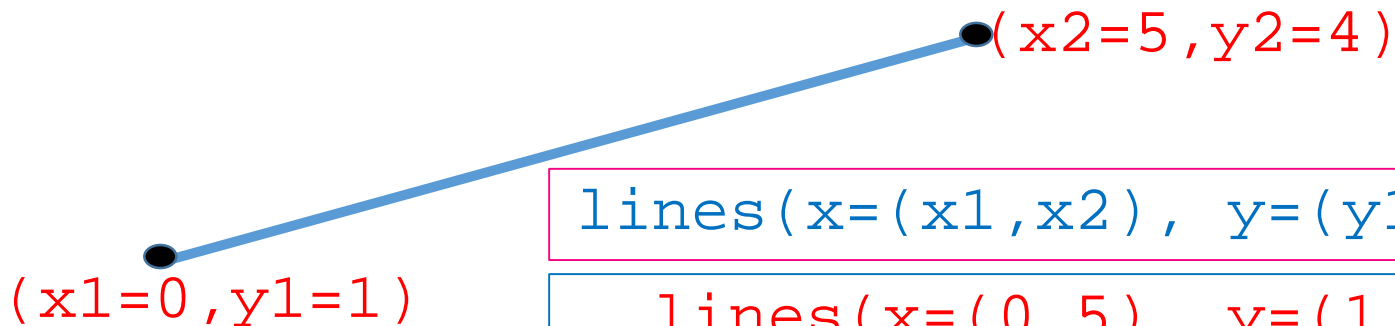
การเพิ่มเส้นเข้าไปที่คำสั่ง plot

- ฟังก์ชัน `lines()` มีรูปแบบดังนี้

```
lines(x, y = NULL, type = "l", ...)
```

โดยที่

`x, y` coordinate vectors of points to join.



```
lines(x=(x1,x2), y=(y1,y2))
```

```
lines(x=(0,5), y=(1,4))
```

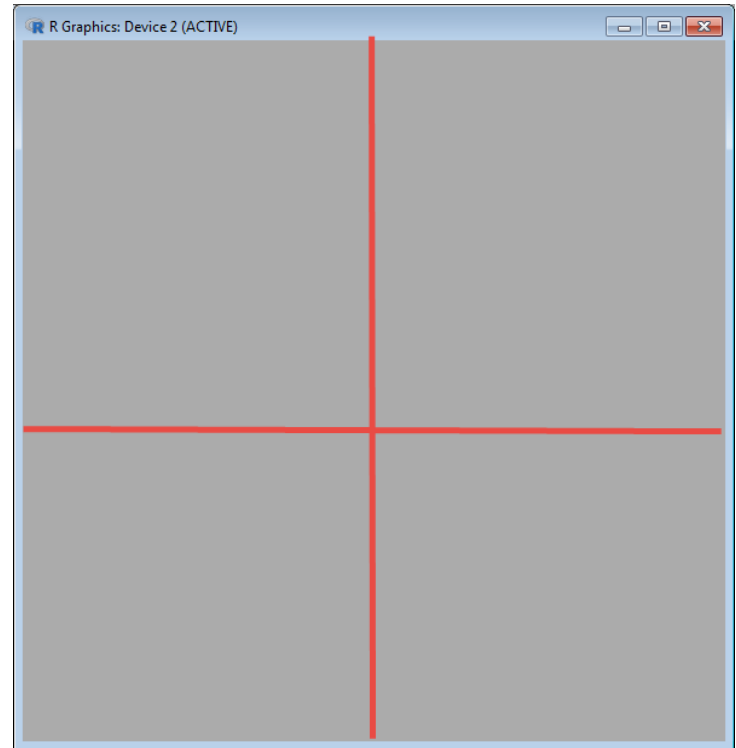
การสร้างกราฟด้วยคำสั่ง plot

- ฟังก์ชัน `par()` ใช้สำหรับกำหนดค่าพารามิเตอร์ของรูปภาพ

ตัวอย่างการใช้ฟังก์ชัน

```
> par(mfrow=c(2,2))
```

```
# กำหนดให้แสดงรูปภาพจำนวน 4 รูป
```

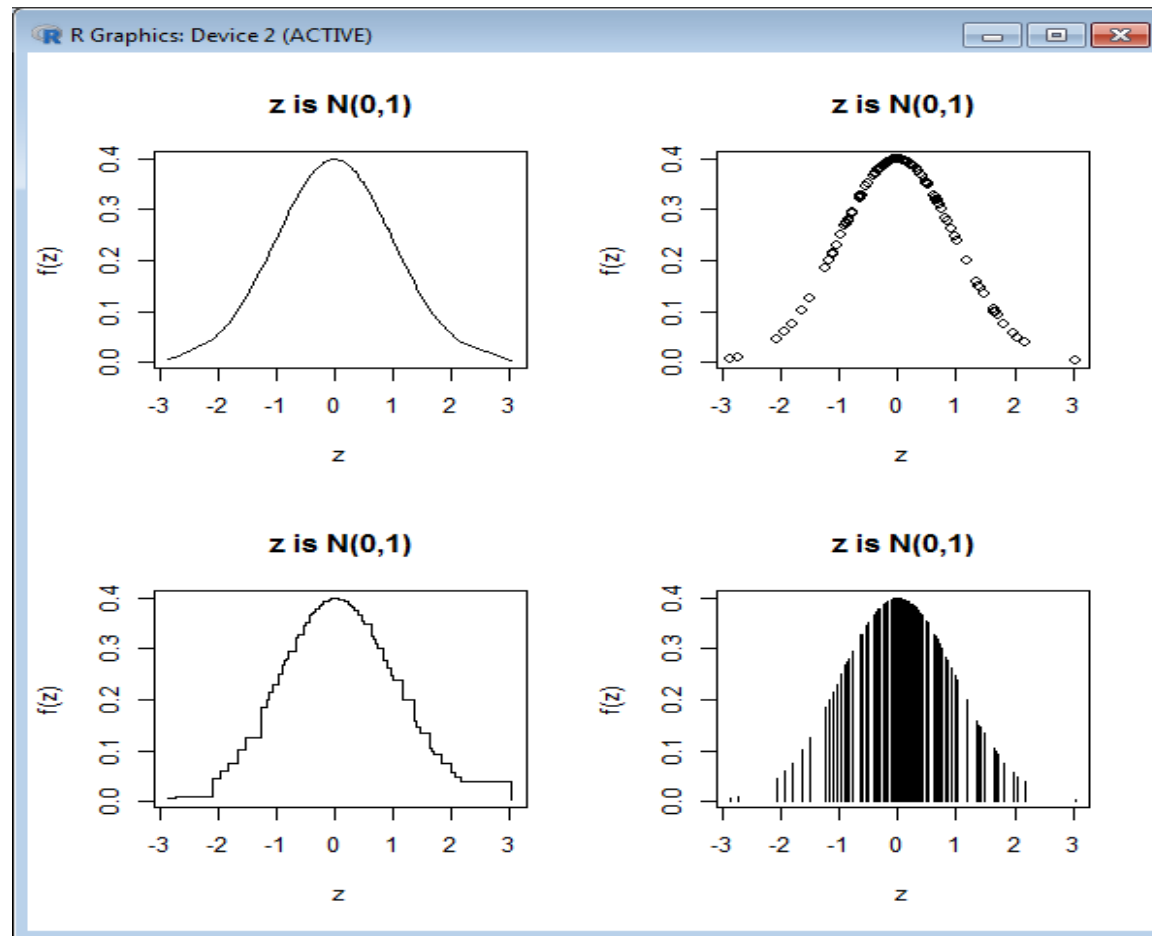


การสร้างกราฟด้วยคำสั่ง plot

- ตัวอย่างการใช้ฟังก์ชัน `par()` และ `plot()`

```
> par(mfrow=c(2,2))  
> plot(x.sort,dnorm(x.sort),type='l',xlab='z',  
ylab='f(z)',main='z is N(0,1)')  
> plot(x.sort,dnorm(x.sort),type='p',xlab='z',  
ylab='f(z)',main='z is N(0,1)')  
> plot(x.sort,dnorm(x.sort),type='s',xlab='z',  
ylab='f(z)',main='z is N(0,1)')  
> plot(x.sort,dnorm(x.sort),type='h',xlab='z',  
ylab='f(z)',main='z is N(0,1)')
```

การสร้างกราฟด้วยคำสั่ง plot

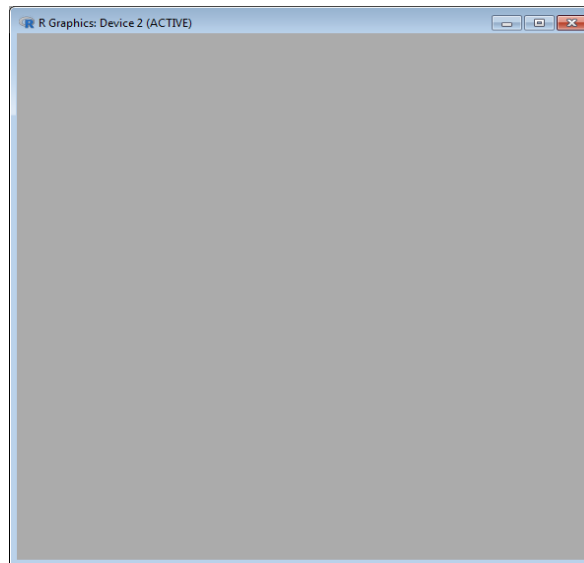


การสร้างกราฟด้วยคำสั่ง plot

- เมื่อต้องการใช้ค่าพารามิเตอร์ของรูปภาพที่โปรแกรม R กำหนดให้คือวาดรูปภาพได้ 1 รูป ให้ใช้คำสั่งดังนี้

```
> par( )
```

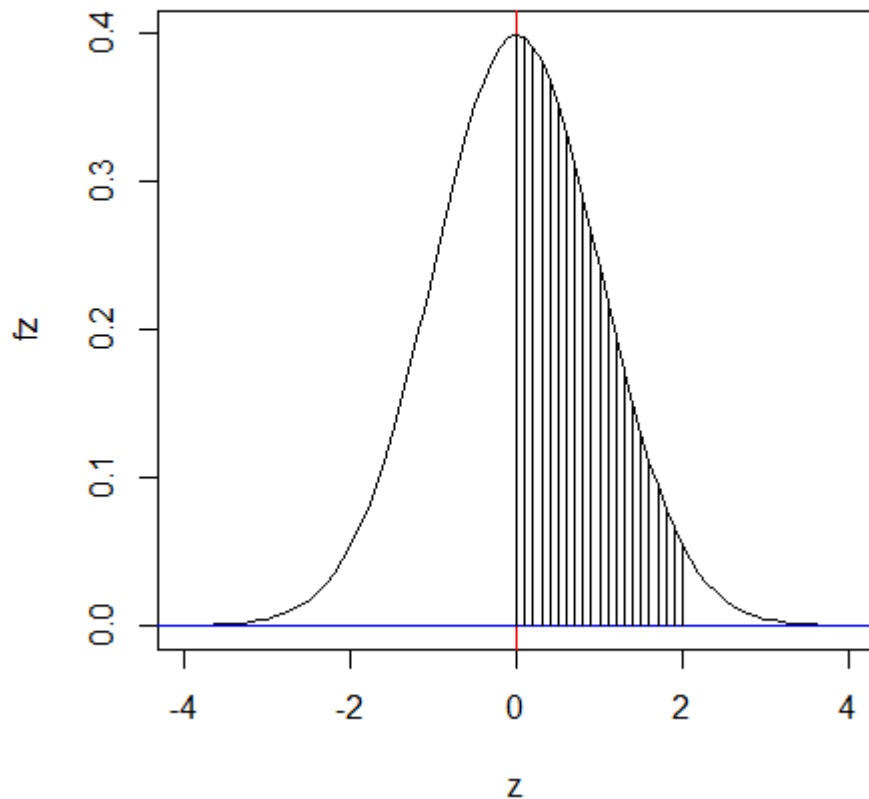
```
# ให้แสดงรูปภาพ 1 รูปตามที่โปรแกรม R กำหนดมาให้
```



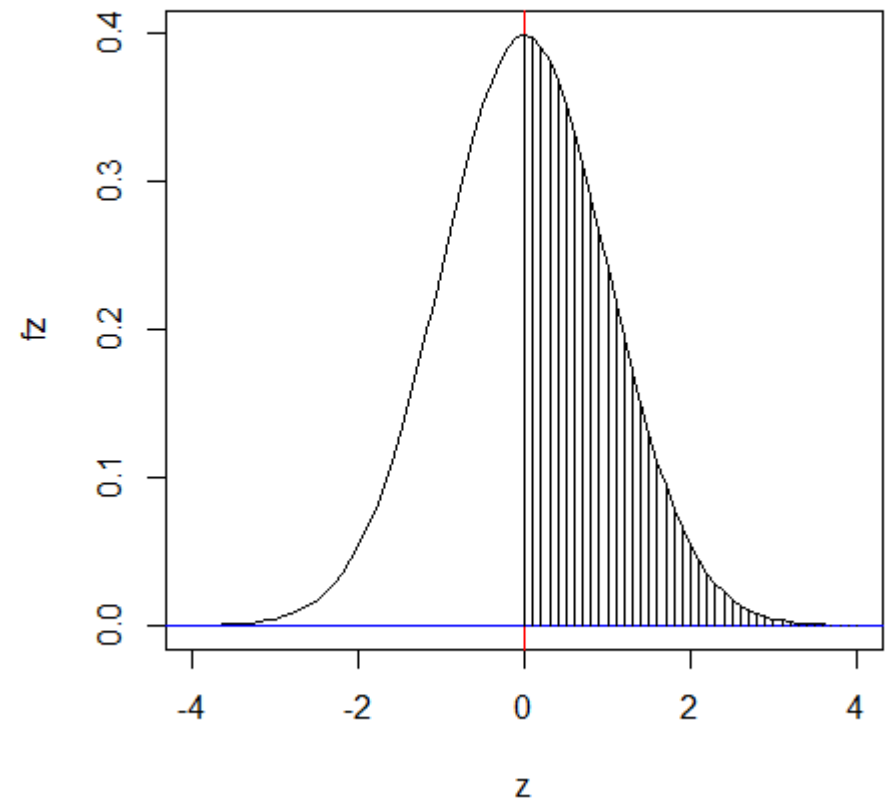
แบบฝึกหัด

จงสร้างกราฟแสดงการแจกแจงแบบปกติมาตรฐาน ดังรูป

$Z=N(0,1)$



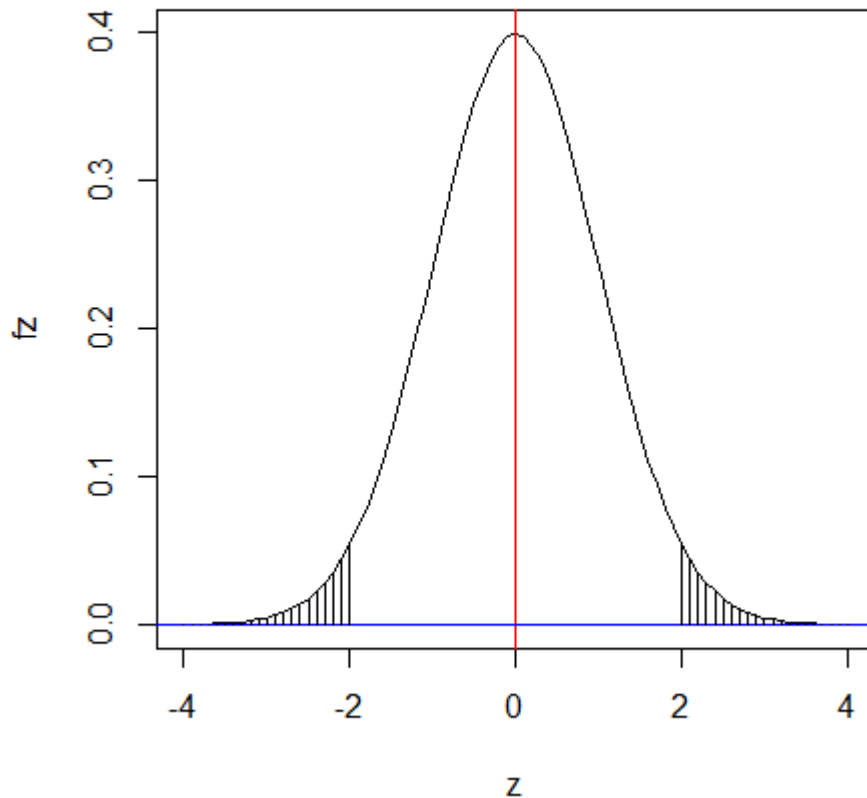
$Z=N(0,1)$



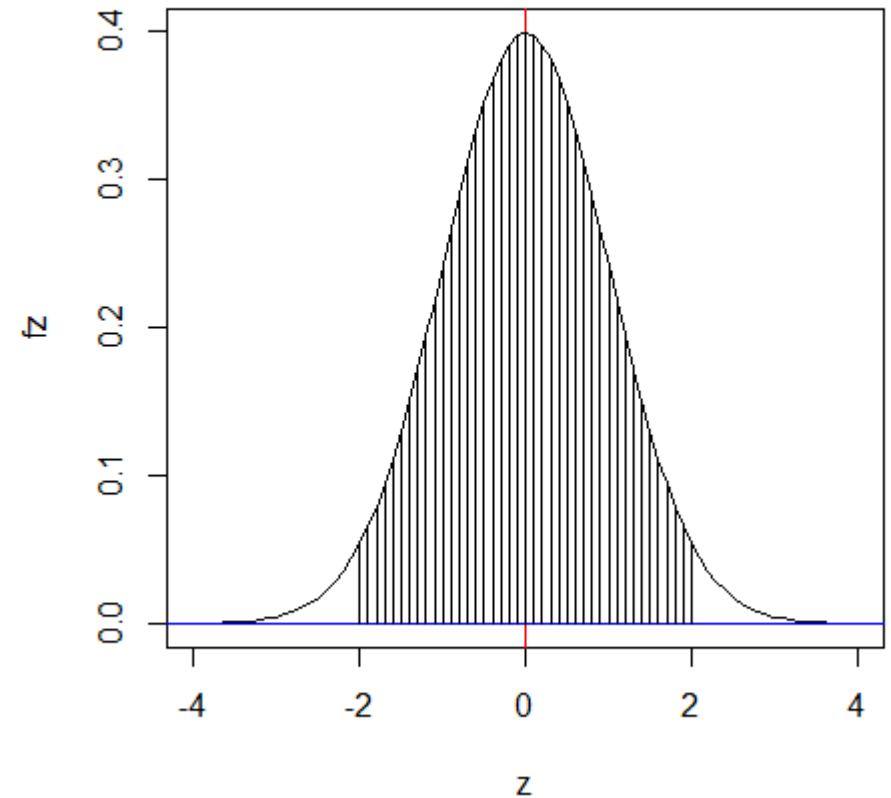
แบบฝึกหัด

จงสร้างกราฟแสดงการแจกแจงแบบปกติมาตรฐาน ดังรูป

$Z=N(0,1)$

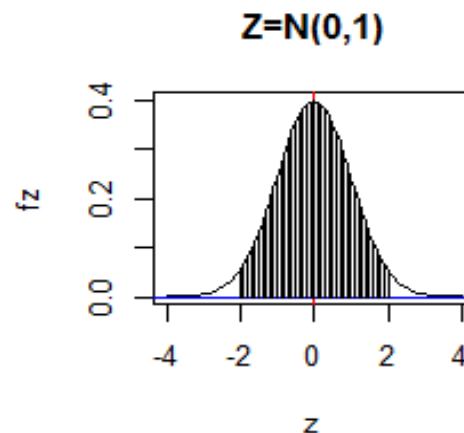
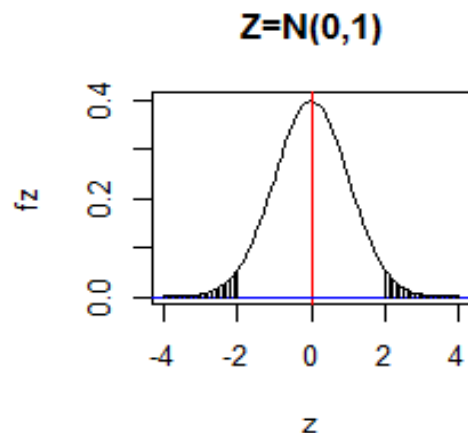
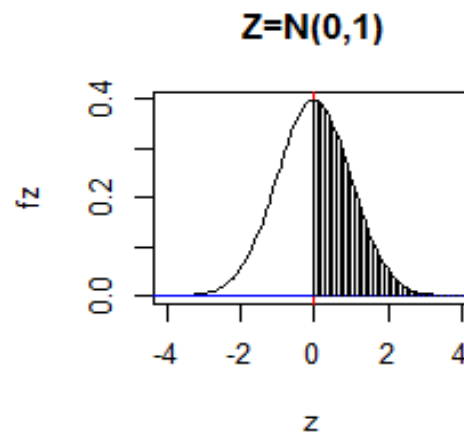
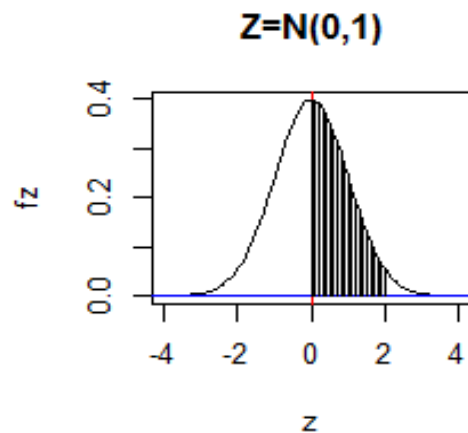


$Z=N(0,1)$



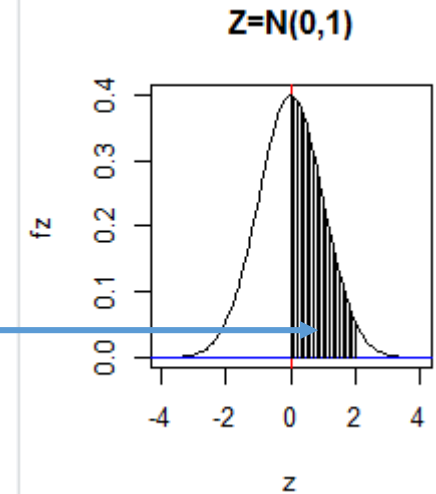
แบบฝึกหัด

จงสร้างกราฟแสดงการแจกแจงแบบปกติมาตรฐาน ดังรูป



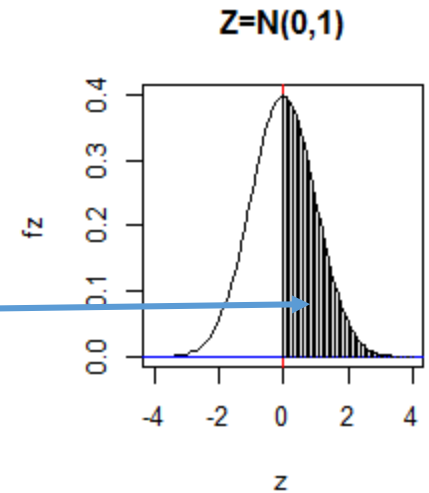
เฉลย

```
#####  
# plot.r  
# date: 28/12/2020  
#####  
?plot  
#### z01_plot.png ####  
z<-seq(-4,4,by=0.1)  
fz<-dnorm(z)  
plot(z,fz,type="l",main="Z=N(0,1)" )  
abline(v=0.00,col="red")  
abline(h=0.00,col="blue")  
zz<-z[z>=0.00 & z<=2.00]  
n<-length(zz)  
for(j in 1:n){  
  lines(x=c(zz[j],zz[j]),y=c(0,dnorm(zz[j])))  
}
```



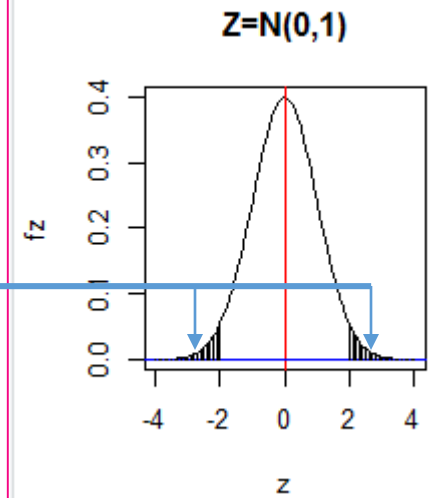
เฉลย

```
#### z02_plot.png ####
z<-seq(-4,4,by=0.1)
fz<-dnorm(z)
plot(z,fz,type="l",main="Z=N(0,1)" )
abline(v=0.00,col="red")
abline(h=0.00,col="blue")
zz<-z[z>=0.00]
n<-length(zz)
for(j in 1:n){
  lines(x=c(zz[j],zz[j]),y=c(0,dnorm(zz[j])))
}
```



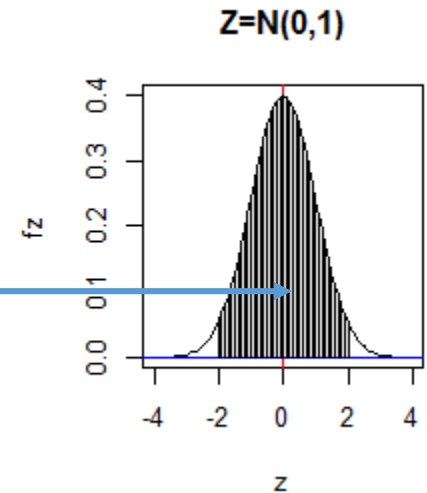
เฉลย

```
#### z03_plot.png ####
z<-seq(-4,4,by=0.1)
fz<-dnorm(z)
plot(z,fz,type="l",main="Z=N(0,1)" )
abline(v=0.00,col="red")
abline(h=0.00,col="blue")
zz<-z[z<=-2.00 | z>=2.00]
n<-length(zz)
for(j in 1:n){
  lines(x=c(zz[j],zz[j]),y=c(0,dnorm(zz[j])))
}
```



เฉลย

```
#### z04_plot.png ####
z<-seq(-4,4,by=0.1)
fz<-dnorm(z)
plot(z,fz,type="l",main="Z=N(0,1)" )
abline(v=0.00,col="red")
abline(h=0.00,col="blue")
zz<-z[z>=-2.00 & z<=2.00]
n<-length(zz)
for(j in 1:n){
  lines(x=c(zz[j],zz[j]),y=c(0,dnorm(zz[j])))
}
```



เฉลย

กรณีแบ่งรูปภาพเป็น 4 รูปเพิ่มคำสั่ง `par(mfrow=c(2,2))`

