

R컴퓨팅

12강

R 그래프 I

정보통계학과 장영재 교수

- 1 막대 그림(Bar Chart)과 barplot 함수
- 2 원그림(Pie Chart)
- 3 상자그림(Box Plot)
- 4 줄기잎 그림(Stem and Leaf Plot)
- 5 점도표-stripchart
- 6 히스토그램
- 7 산점도와 plot 함수
- 8 abline, arrow, segment 함수로 선 그리기

1

막대 그림(Bar Chart)과 barplot 함수

1 막대 그림(Bar Chart)과 barplot 함수

- 막대 그림은 범주형 자료(순서형 또는 명목형 자료)의 빈도수 또는 상대도수를 기둥의 높이로 표현하기 위한 그림으로 아래 함수를 사용

```
barplot(height, names.arg = NULL, legend.text = NULL,  
beside = FALSE,  
horiz = FALSE, density = NULL, angle = 45, col =  
NULL, ...)
```

- height: 각 기둥의 높이에 해당하는 값을 저장한 벡터 또는 행렬

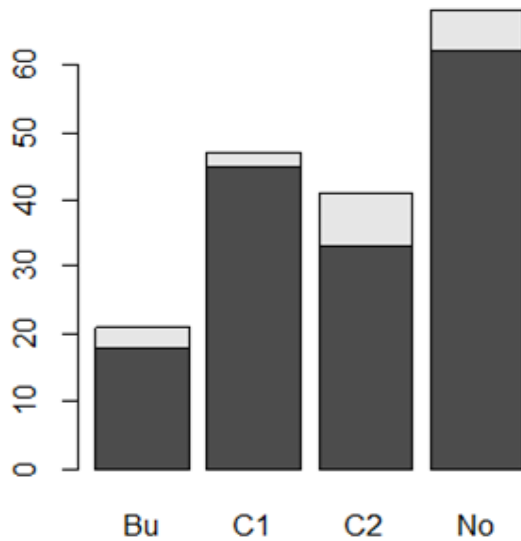
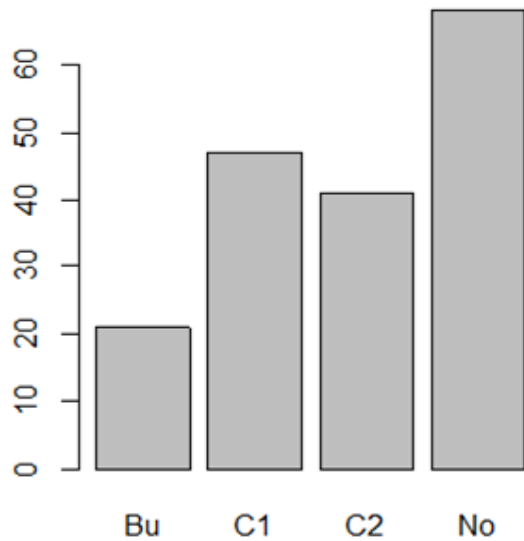
1 막대 그림(Bar Chart)과 barplot 함수

- 보기 12-1 : height에 빈도수 벡터가 설정된 경우와 행렬이 설정되는 경우의 차이

```
> barplot.test1 <- function() {  
+ par(mfrow=c(1,2))      # 한 그래픽 장치에 그릴 그림의 개수를 설정  
+ barplot(table(BMI$religion))  
+ barplot(table(BMI$gender, BMI$religion))  
+ }                      # end function
```

1 막대 그림(Bar Chart)과 barplot 함수

※ height에 빈도수 벡터가 설정되는 경우와
행렬이 설정되는 경우



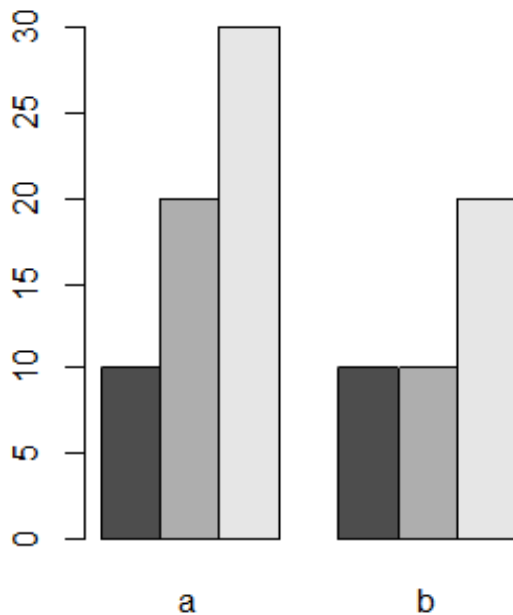
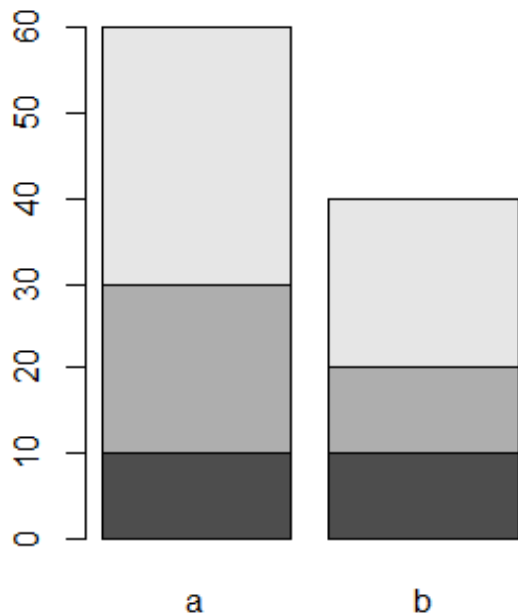
1 막대 그림(Bar Chart)과 barplot 함수

➤ 보기 12-2: beside 속성이 FALSE일 때와 TRUE일 때 비교

```
>barplot.test2 <- function() {  
+ par(mfrow=c(1,2))  
+ a <- c(10,20,30); b <- c(10, 10, 20)  
+ barplot(cbind(a,b))  
+ barplot(cbind(a,b), beside=T)  
+ }      # end function
```

1 막대 그림(Bar Chart)과 barplot 함수

※ 옵션이 없는 경우(좌)와 beside = TRUE인 경우(우)



2 원그림(Pie Chart)

2 원그림(Pie Chart)

- 원그림은 주로 범주형 자료의 상대빈도수를 원의 면적(또는 각도)에 비례하게 그림

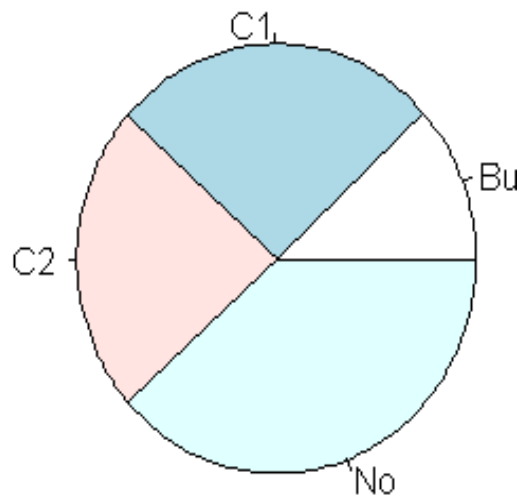
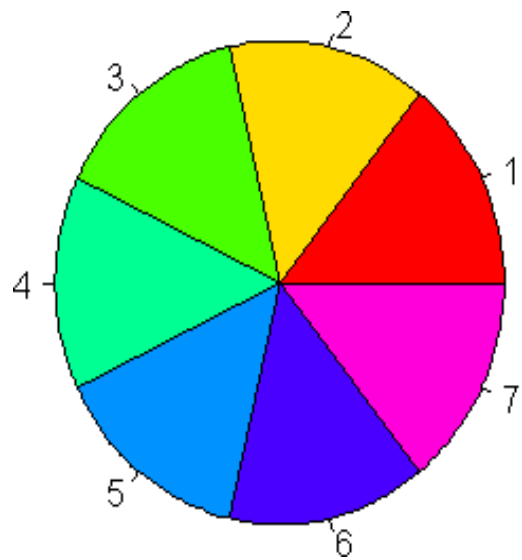
```
pie(x, labels = names(x), edges = 200, radius = 0.8,  
clockwise = FALSE, init.angle = if(clockwise) 90 else 0,  
col = NULL, ...)
```

- 보기 12-3: 두 개의 원그림의 비교

```
> pie.test <- function() {  
+ par(mfrow=c(1,2))  
+ pie(rep(1, 7), col = rainbow(7), radius = 0.9)  
+ pie(table(BMI$religion))  
+}      # end function
```

2 원그림(Pie Chart)

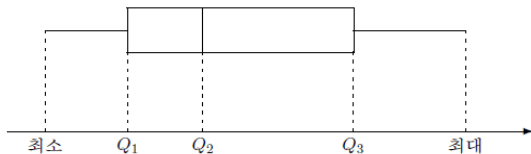
※ radius가 0.9로 크며 rainbow함수로
색을 7개 지정한 원(좌)과 BMI의 종교에 대한 원(우)



3 상자그림(Box Plot)

3 상자그림(Box Plot)

- 상자그림은 사분위에 해당하는 부분은 상자로 표현한 그림



- 상자그림을 그리기 위해 제공되는 R 함수는 `boxplot` 함수이며

```
boxplot(formula, data = NULL, subset,... )
```

또는, 아래와 같이 설정

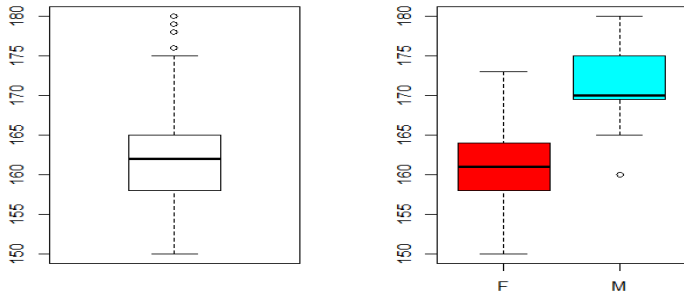
```
boxplot(x, ..., range = 1.5, col = NULL, horizontal =  
FALSE, ...)
```

3 상자그림(Box Plot)

- 보기 12-4: BMI 자료에서 키에 대한 상자그림과, 성별에 따른 키의 상자그림

```
>boxplot.test2 <- function() {  
+   par(mfrow=c(1,2))  
+   boxplot(BMI$height)  
+   boxplot(height ~ gender, data=BMI, col=rainbow(2))  
+ }    # end function
```

※ 상자그림의 자료 입력 방법과 R 모형식(formula) 사용 방법



4 줄기잎 그림(Stem and Leaf Plot)

4 줄기잎 그림(Stem and Leaf Plot)

- ▶ 줄기잎 그림은 자료의 수가 많지 않을 때 전체 자료를 줄기와 잎으로 표현

```
stem(x, scale = 1, width = 80, atom = 1e-08)
```

- ▶ 보기 12-5: 다음은 BMI의 height 변수에 대한 줄기잎 그림을 그린 것

```
> stem(BMI$height)
```

The decimal point is at the |

4

줄기잎 그림(Stem and Leaf Plot)

```
150 | 00
152 | 00000000
154 | 0000000000
156 | 0000000000
158 | 00000000000000000000
160 | 0000000000000000000000000000000000000000
162 | 00000000000000000000000000000000
164 | 00000000000000000000000000000000
166 | 00000000000
168 | 00000000000
170 | 000000
172 | 000
174 | 000
176 | 0
178 | 00
180 | 0
```


4 줄기잎 그림(Stem and Leaf Plot)

- 보기 12-6: scale 과 width의 설정으로 줄기잎 그림의 줄기의 크기/잎의 수를 조절

```
> stem(BMI$height, scale=.4)
```

The decimal point is 1 digit(s) to the right of the |

```
15 | 00222333334
```

```
15 | 555555555666667777788888888888888888899
```

```
16 |
```

```
000000000000000000000000000000000001111112222222222222222233333  
3333333444
```

```
16 | 5555555555555555555666677777788888888899
```

```
17 | 0000002334
```

```
17 | 55689
```

```
18 | 0
```

4 줄기잎 그림(Stem and Leaf Plot)

- 원래의 줄기잎 그림에 16개의 줄기가 있었으나 16의 약 40%인 7개의 줄기만 생성

```
> stem(BMI$height, width=20)
```

The decimal point is at the |

```
150 | 00  
152 | 00000000  
154 | 00000000  
156 | 00000000  
158 | 00000000  
160 | 00000000+16  
162 | 00000000+9  
164 | 00000000+8  
166 | 00000000
```

4 줄기잎 그림(Stem and Leaf Plot)

- 위의 줄기 잎 그림에서 width가 설정되어 160, 162, 164 구간의 자료는 일부만 보여주고 추가로 16, 9 및 8개의 자료가 더 있음을 표시

168 | 00000000

170 | 000000

172 | 000

174 | 000

176 | 0

178 | 00

180 | 0

5 점도표-stripchart

5

점도표-stripchart

- ▶ 자료의 수가 많지 않을 경우 점도표를 사용하여 자료의 분포를 확인

```
stripchart(formula, data= NULL, ...)
```

```
stripchart(x, method = "overplot", pch = 0, col =, par("fg"),  
cex = par("cex"), subset, ...)
```

- ▶ 보기 12-7: BMI 자료에서 성별에 따른 몸무게의 점도표를 서로 다른 방법 적용하여 그리기

5

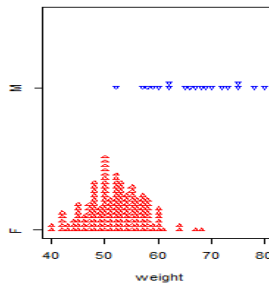
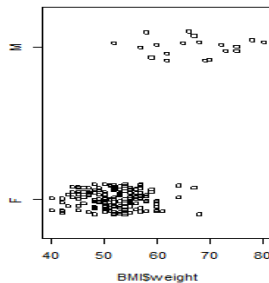
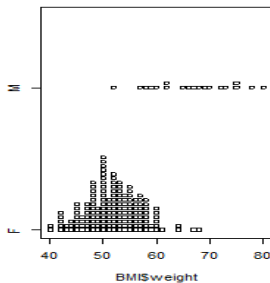
점도표-stripchart

```

> stripchart.test <- function() {
+   par(mfrow=c(1,3))
+   stripchart(BMI$weight~BMI$gender, method="stack")
+   stripchart(BMI$weight~BMI$gender, method="jitter")
+   stripchart(weight~gender, data=BMI, method="stack",
+   pch=24:25, col=c("red", "blue"))
+ }           # end function

```

※ stripchart 함수에서 method 및 col 설정



6 히스토그램

6

히스토그램

- 연속인 변수의 분포를 알기 위해서 변수의 값을 적절한 범위로 구분하여 해당 범위에 속하는 자료의 빈도 또는 상대도수를 기둥의 높이로 하여 생성

```
hist(x, freq = NULL, main = paste("Histogram of" , xname),  
xlim = range(breaks), ylim = NULL, labels = FALSE, ...)
```


6

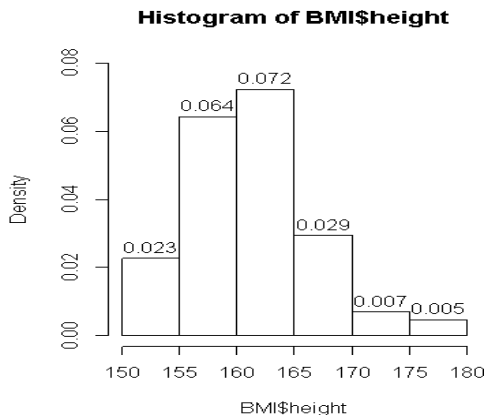
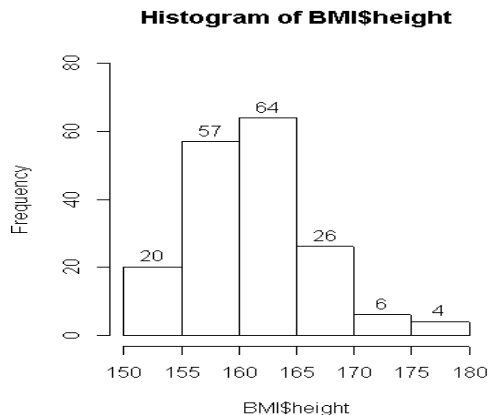
히스토그램

➤ 보기 12-8: 도수히스토그램과 상대도수 히스토그램

```
> hist(BMI$height, labels=T, ylim=c(0, 80) )
```

```
> hist(BMI$height, labels=T, freq=F, ylim=c(0, 0.08))
```

※ plot 함수에서 label 설정에 따른 기둥 설명



7 산점도와 plot 함수

7

산점도와 plot 함수

- n개의 짝으로 얻은 자료 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ 을 x-y 평면에 점으로 나타내는 것

```
plot(x, y, type=, main=, sub=, pch=, lty=, lwd=, xlab=,
     ylab=, ...)
```

- 보기 12-9: 다음 프로그램은 type에 따른 산점도의 모양을 알아보기 위해 $y=\sin(x)$ 함수를 그려본 것이다.

```
> plot.test <- function() {
+   par(mfrow=c(2,2))
+   x<- seq(1, 2*pi, length=10)
+   y <- sin(x)
+   plot(x,y, type="l", main="type=W"lW"")
+   plot(x,y, type="b", main="type=W"bW"")
}
```

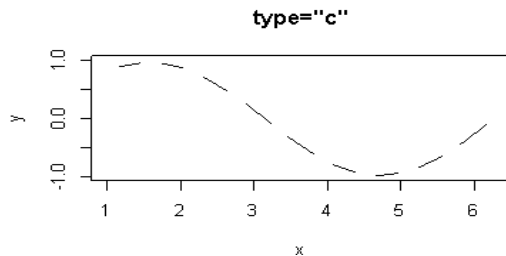
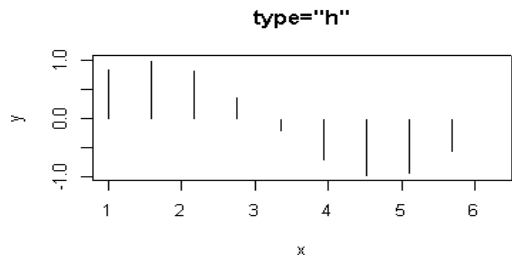
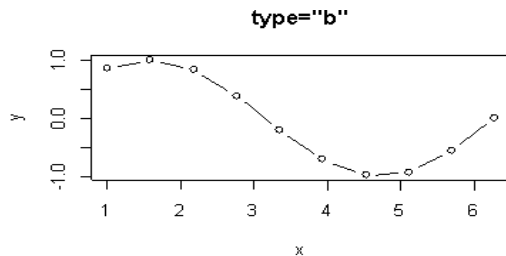
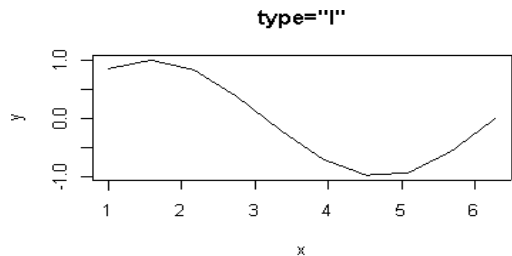
7

산점도와 plot 함수

```
+ plot(x,y, type="h", main="type=\W"h\W")
```

```
+ plot(x,y, type="c", main="type=\W"c\W")
```

```
+ }      # end function
```



7

산점도와 plot 함수

➤ 보기 12-10: plot 함수의 예-pch, sub, main, xlab, ylab 속성의 설정

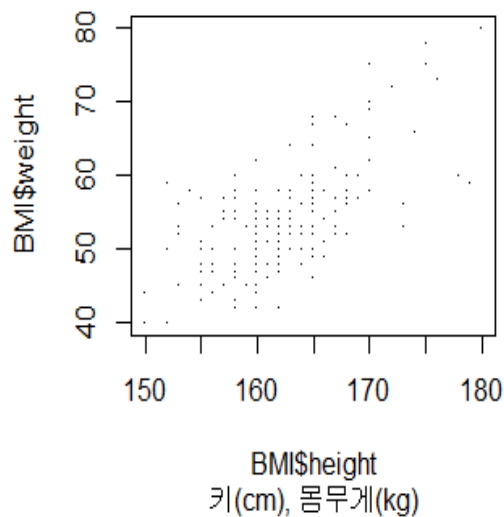
```
> plot.test2 <- function() {  
+   par(mfrow=c(1,2))  
+   plot(BMI$height, BMI$weight, main="키와 몸무게의 산점도", type="p",  
+       pch=".", sub="키(cm), 몸무게(kg)")  
+   plot(BMI$height, BMI$weight, main="키와 몸무게의 산점도", type="p",  
+       pch="o", sub="키(cm), 몸무게(kg)", xlab="몸무게", ylab="키")  
+ }
```

7

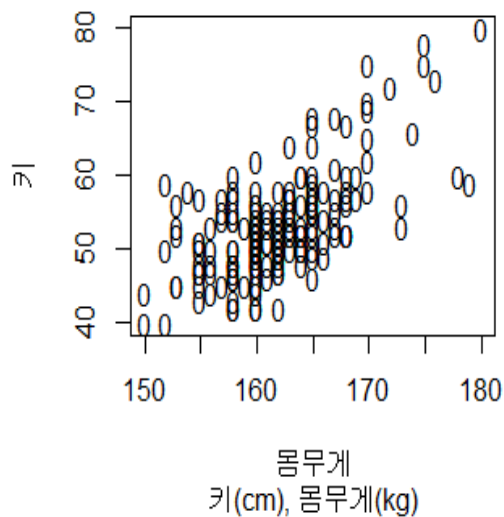
산점도와 plot 함수

※ plot 함수의 예-pch, sub, main, xlab, ylab 속성

키와 몸무게의 산점도



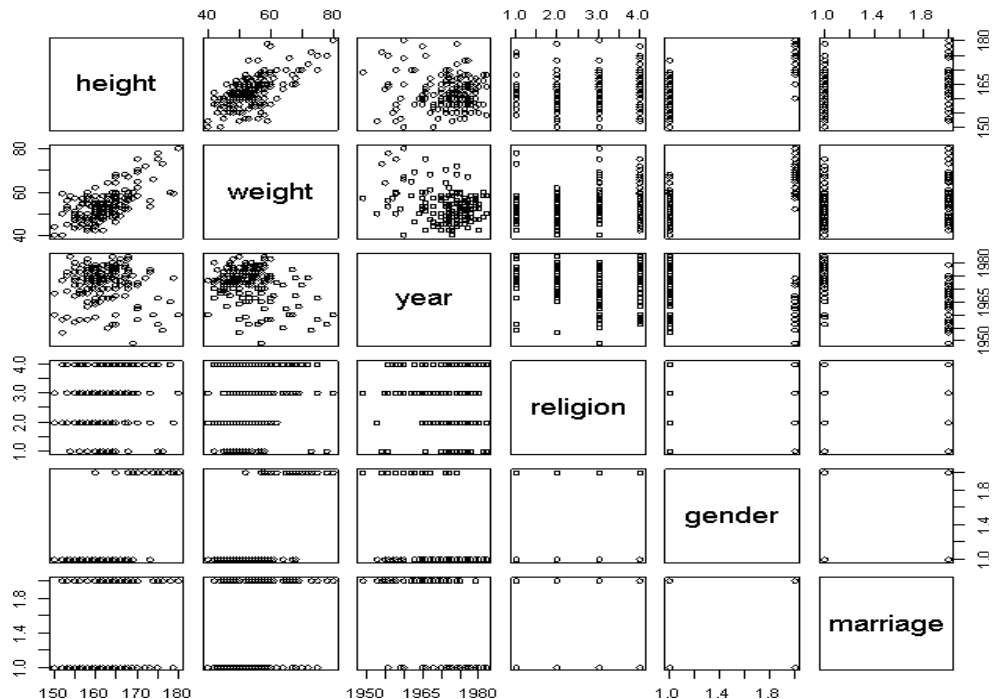
키와 몸무게의 산점도



7

산점도와 plot 함수

※ plot 함수에서 모든 가능한 조합에 대한 산점도



8 abline, arrow, segment 함수로 선 그리기

8 abline, arrow, segment 함수로 선 그리기

1 abline 함수

- 그릴 직선의 기울기와 절편을 설정하여 선을 그릴 수 있는 함수로 plot 등과 같은 함수로 그래픽 장치가 열려 있어야 함

```
abline(a = NULL, b = NULL, h = NULL, v = NULL, coef =  
NULL, ...)
```

- a, b: 그릴 직선의 y 절편 및 기울기
- h: 수평선을 그릴 때 y값만 설정하며 이 때 y값을 h에 설정
- v: 수직선을 그릴 때 x 축의 값
- coef: 위의 a, b값을 벡터로 설정할 때 사용 coef=c(a, b)

8 abline, arrow, segment 함수로 선 그리기

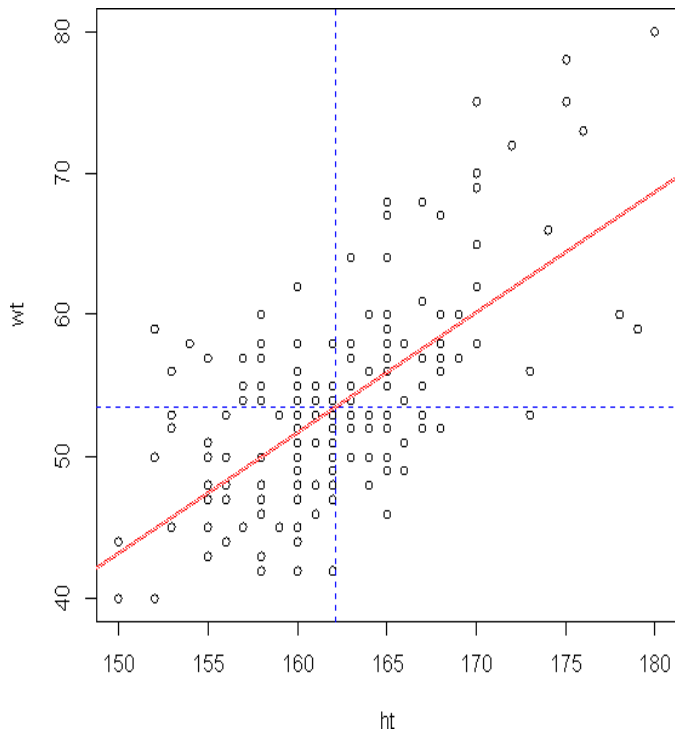
1 abline 함수

➤ 보기 12-11: abline 함수로 직선 추가

```
> abline.test <- function() {  
+   ht <- BMI$height  
+   wt <- BMI$weight  
+   plot(ht, wt)  
+   abline(v=mean(ht), lty=2, col="blue")  
+   abline(h=mean(wt), lty=2, col="blue")  
+   abline(lsfilt(ht, wt), lwd=2, col="red")  
+   abline(lsfilt(ht, wt)$coef, lw=1,col="white")  
+ }           # end function
```

8 abline, arrow, segment 함수로 선 그리기

1 abline 함수



8 abline, arrow, segment 함수로 선 그리기

2 arrows와 segments 함수

➤ 기존 그림에 추가로 선분과 화살표 그리기

```
arrows(x0, y0, x1 = x0, y1 = y0, length = 0.25, angle = 30,  
code = 2, col, lty, lwd. ...)
```

및

```
segments(x0, y0, x1 = x0, y1 = y0, col, lty, lwd. ...)
```

➤ 보기 12-12: 산점도와 회귀직선을 arrow 및 segment 함수를 사용하기

8 abline, arrow, segment 함수로 선 그리기

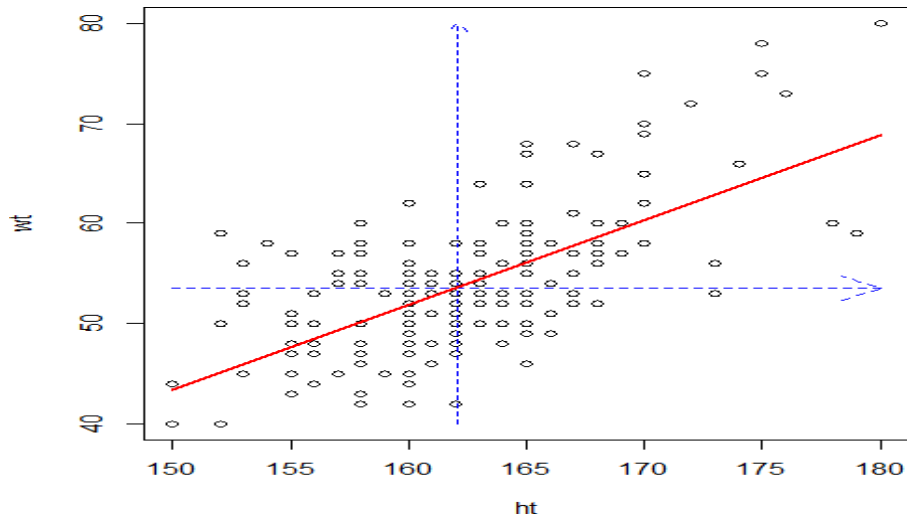
2 arrows와 segments 함수

```
> arrow.segment.test <- function () {  
+ ht <- BMI$height; meanht <- mean(ht); minht <- min(ht);  
+ maxht <- max(ht)  
+ wt <- BMI$weight; meanwt <- mean(wt); minwt <- min(wt);  
+ maxwt <- max(wt)  
+ plot(ht, wt)  
+ arrows(minht, meanwt, maxht, meanwt, lty=2, col="blue")  
+ arrows(meanht, minwt, meanht, maxwt, lty=2, length=0.1,  
+ col="blue")  
+ # 회귀직선이  $y = -84.1 + 0.85x$ 로 추정됨.  
+ segments(minht, -84.1+0.85*minht, maxht, -84.1+0.85*maxht,  
+ lwd=2, col="red")  
+ } # end function
```

8 abline, arrow, segment 함수로 선 그리기

2 arrows와 segments 함수

※ arrow, segment 함수를 사용한 선 그리기



R컴퓨팅

