

Student Crimtab Data Graphic Analysis

coop711

2015년 3월 25일

Data Manipulation

- Data Frame으로 정리하고, 다시 long format 으로 변환. height 를 인치로 변환하면 어떻게 될까?

```
dimnames(crimtab.2)[[2]]<-as.numeric(dimnames(crimtab.2)[[2]])/2.54  
crimtab.2
```

##	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
## 9.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 9.5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 9.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 9.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 9.8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 9.9	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 10	1	0	0	1	2	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
## 10.1	0	0	0	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
## 10.2	0	0	2	2	2	1	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0
## 10.3	0	1	1	3	2	2	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 10.4	0	0	1	1	2	3	3	4	3	3	0	0	0	0	0	0	0	0	0	0	0	0
## 10.5	0	0	0	1	3	7	6	4	3	1	3	1	0	1	0	0	0	0	0	0	0	0
## 10.6	0	0	0	1	4	5	9	14	6	3	1	0	0	1	0	0	0	0	0	0	0	0
## 10.7	0	0	1	2	4	9	14	16	15	7	3	1	2	0	0	0	0	0	0	0	0	0
## 10.8	0	0	0	2	5	6	14	27	10	7	1	2	1	0	0	0	0	0	0	0	0	0
## 10.9	0	0	0	0	2	6	14	24	27	14	10	4	1	0	0	0	0	0	0	0	0	0
## 11	0	0	0	2	6	12	15	31	37	27	17	10	6	0	0	0	0	0	0	0	0	0
## 11.1	0	0	0	3	3	12	22	26	24	26	24	7	4	1	0	0	0	0	0	0	0	0
## 11.2	0	0	0	3	2	7	21	30	38	29	27	20	4	1	0	0	0	0	0	0	0	1
## 11.3	0	0	0	1	0	5	10	24	26	39	26	24	7	2	0	0	0	0	0	0	0	0
## 11.4	0	0	0	0	3	4	9	29	56	58	26	22	10	11	0	0	0	0	0	0	0	0
## 11.5	0	0	0	0	0	5	11	17	33	57	38	34	25	11	2	0	0	0	0	0	0	0
## 11.6	0	0	0	0	2	1	4	13	37	39	48	38	27	12	2	2	0	1	0	0	0	0
## 11.7	0	0	0	0	0	2	9	17	30	37	48	45	24	9	9	2	0	0	0	0	0	0
## 11.8	0	0	0	0	1	0	2	11	15	35	41	34	29	10	5	1	0	0	0	0	0	0
## 11.9	0	0	0	0	1	1	2	12	10	27	32	35	19	10	9	3	1	0	0	0	0	0
## 12	0	0	0	0	0	0	1	4	8	19	42	39	22	16	8	2	2	0	0	0	0	0
## 12.1	0	0	0	0	0	0	0	2	4	13	22	28	15	27	10	4	1	0	0	0	0	0
## 12.2	0	0	0	0	0	0	1	2	5	6	23	17	16	11	8	1	1	0	0	0	0	0
## 12.3	0	0	0	0	0	0	0	0	4	8	10	13	20	23	6	5	0	0	0	0	0	0
## 12.4	0	0	0	0	0	0	1	1	1	2	7	12	4	7	7	1	0	0	1	0	0	0
## 12.5	0	0	0	0	0	0	0	1	0	1	3	12	11	8	6	8	0	2	0	0	0	0
## 12.6	0	0	0	0	0	0	0	0	0	1	0	3	5	7	8	6	3	1	1	0	0	0
## 12.7	0	0	0	0	0	0	0	0	0	1	1	7	5	5	8	2	2	0	0	0	0	0
## 12.8	0	0	0	0	0	0	0	0	0	0	1	2	3	1	8	5	3	1	1	0	0	0
## 12.9	0	0	0	0	0	0	0	0	0	0	0	1	2	2	0	1	1	0	0	0	0	0
## 13	0	0	0	0	0	0	0	0	0	0	3	0	1	0	1	0	2	1	0	0	0	0
## 13.1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
## 13.2	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	3	0	0	0	0	0	0
## 13.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
## 13.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
## 13.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

```

crimtab.3<-crimtab.2
crimtab.3.df<-as.data.frame(crimtab.3, stringsAsFactors = F)
head(crimtab.3.df)

```

```
##   Var1 Var2 Freq
## 1  9.4   56    0
## 2  9.5   56    0
## 3  9.6   56    0
## 4  9.7   56    0
## 5  9.8   56    0
## 6  9.9   56    0
```

```
str(crimtab.3.df)
```

```
## 'data.frame':   924 obs. of  3 variables:
## $ Var1: chr  "9.4" "9.5" "9.6" "9.7" ...
## $ Var2: chr  "56" "56" "56" "56" ...
## $ Freq: int   0 0 0 0 0 0 1 0 0 0 ...
```

```
crimtab.3.df$finger<-as.numeric(crimtab.3.df$Var1)
crimtab.3.df$height<-as.numeric(crimtab.3.df$Var2)
str(crimtab.3.df)
```

```
## 'data.frame':   924 obs. of  5 variables:
## $ Var1 : chr  "9.4" "9.5" "9.6" "9.7" ...
## $ Var2 : chr  "56" "56" "56" "56" ...
## $ Freq : int   0 0 0 0 0 0 1 0 0 0 ...
## $ finger: num  9.4 9.5 9.6 9.7 9.8 9.9 10 10.1 10.2 10.3 ...
## $ height: num  56 56 56 56 56 56 56 56 56 56 ...
```

```
crimtab.3.long<-apply(crimtab.3.df[,4:5], 2, function(x) rep(x, crimtab.3.d
f[,3]))
str(crimtab.3.long)
```

```
## num [1:3000, 1:2] 10 10.3 9.9 10.2 10.2 10.3 10.4 10.7 10 10.1 ...
## - attr(*, "dimnames")=List of 2
## ..$ : NULL
## ..$ : chr [1:2] "finger" "height"
```

- `function(x) rep(x, crimtab.3.df[,3])` 이 어렵게 느껴지면 다음과 같이 세 단계로 나누어 작업할 수도 있음.

```
attach(crimtab.3.df)
finger.long<-rep(finger, Freq)
height.long<-rep(height, Freq)
crimtab.3.long.2<-data.frame(finger=finger.long, height=height.long)
str(crimtab.3.long.2)
```

```
## 'data.frame':   3000 obs. of  2 variables:
## $ finger: num  10 10.3 9.9 10.2 10.2 10.3 10.4 10.7 10 10.1 ...
## $ height: num  56 57 58 58 58 58 58 58 59 59 ...
```

```
detach()
```

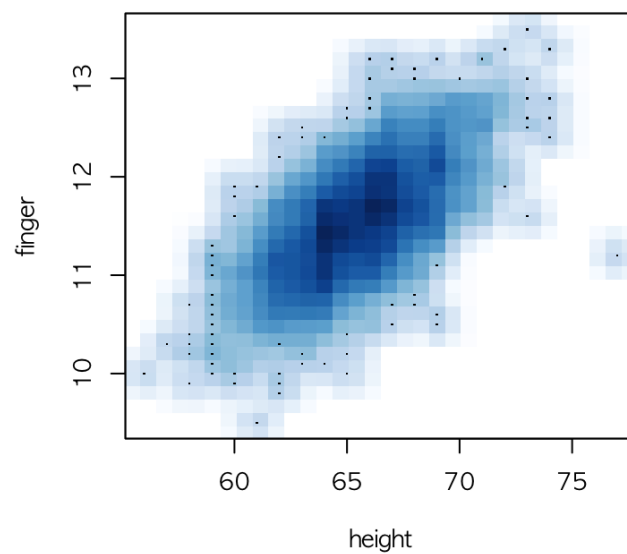
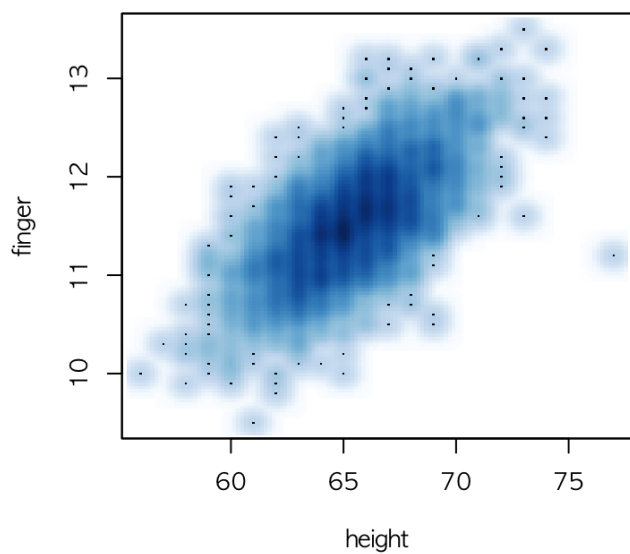
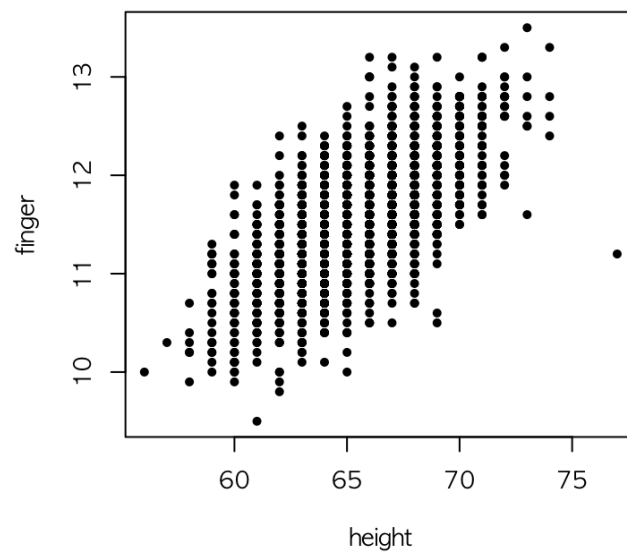
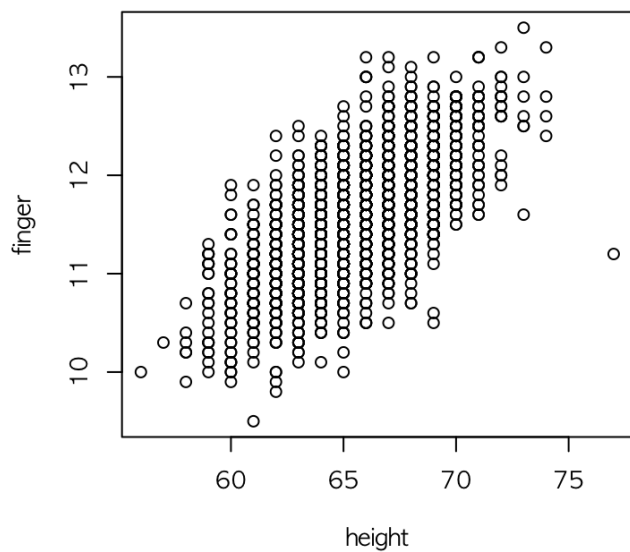
- 산점도를 여러 유형으로 표현하기 위하여 필요한 패키지 설치

```
install.packages("hexbin")
```

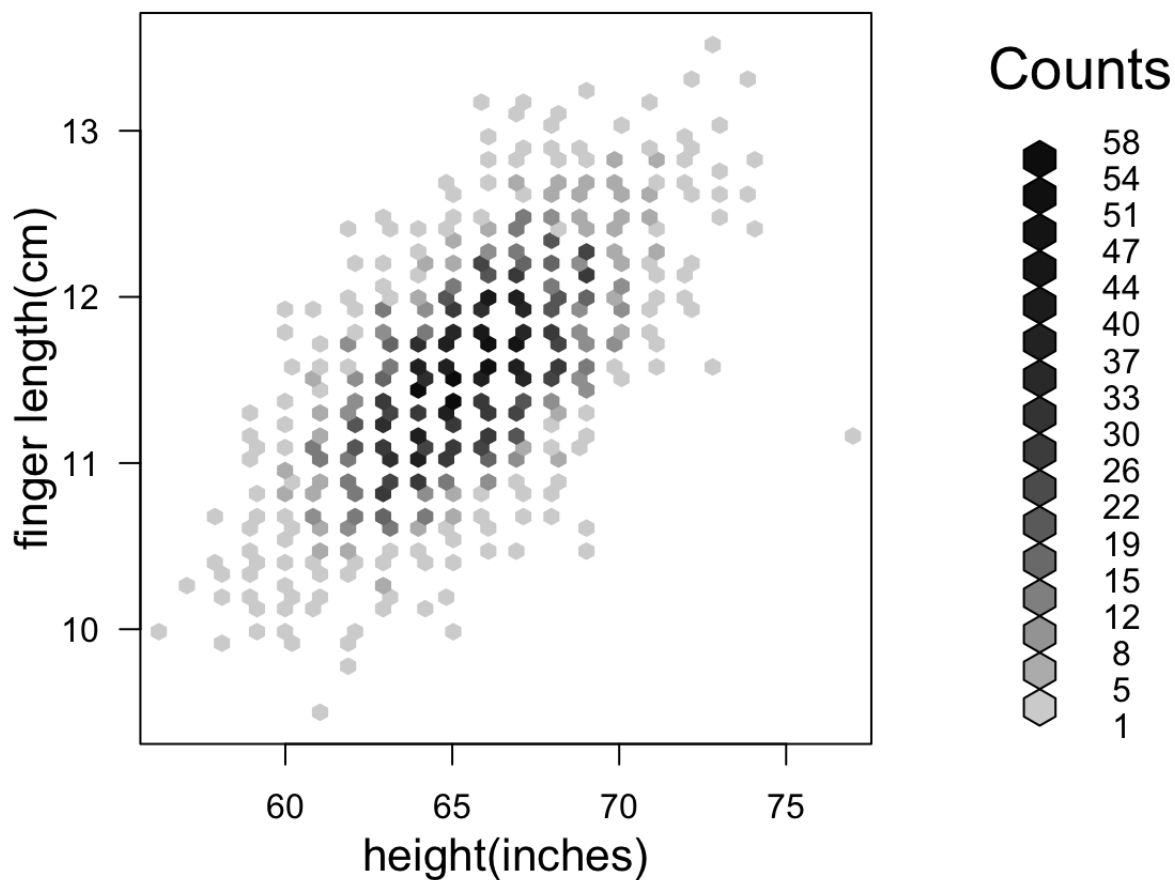
```
library(hexbin)
```

```
crimtab.bin<-hexbin(crimtab.3.long[, "height"], crimtab.3.long[, "finger"], xbins=50)
```

```
par(mfrow=c(2,2))  
plot(finger~height, data=crimtab.3.long)  
plot(finger~height, data=crimtab.3.long, pch=20)  
smoothScatter(crimtab.3.long[, "height"], crimtab.3.long[, "finger"], xlab="height", ylab="finger")  
smoothScatter(crimtab.3.long[, "height"], crimtab.3.long[, "finger"], nbins=32, xlab="height", ylab="finger")
```

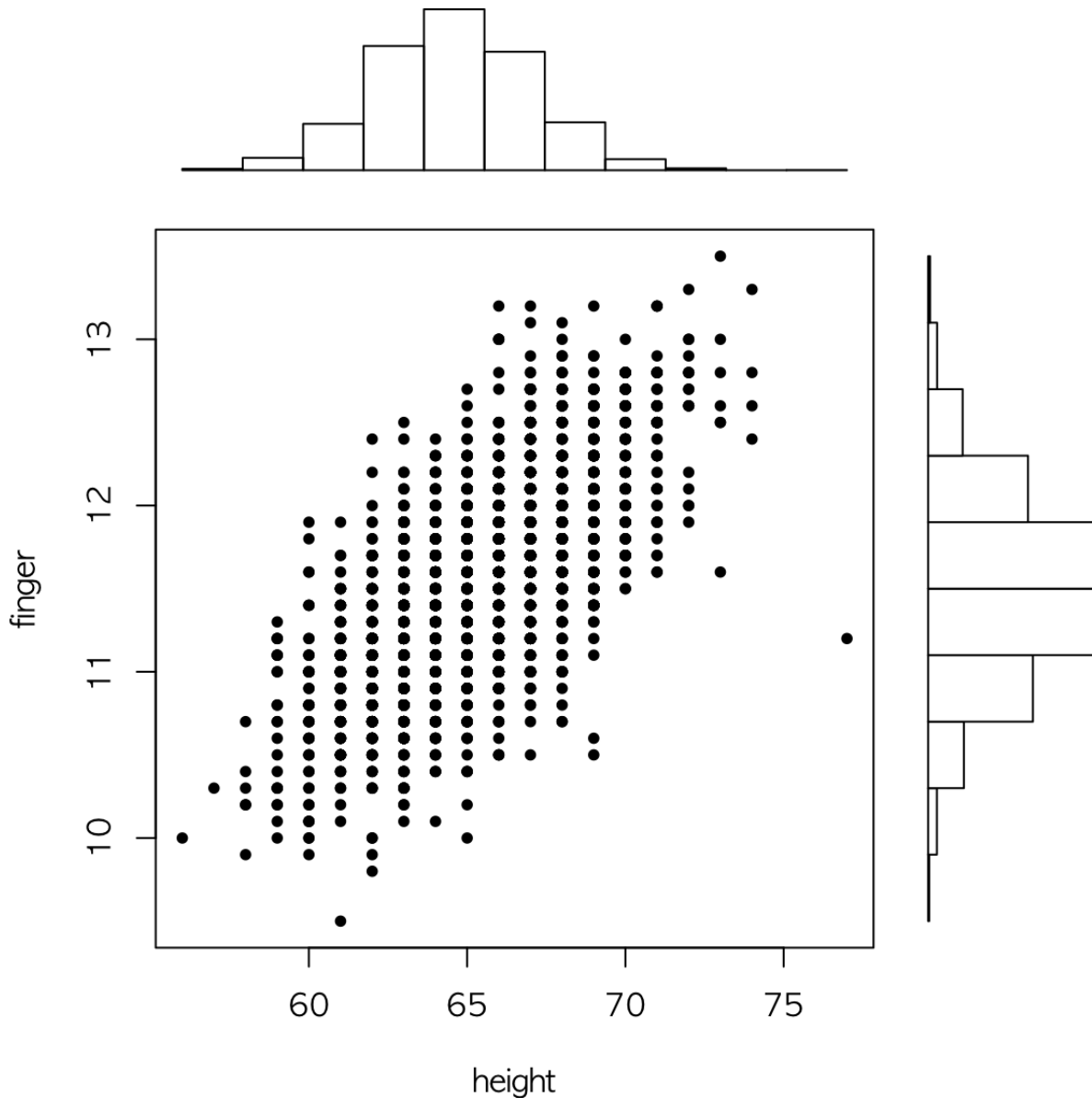


```
par(mfrow=c(1,1))
plot(crimtab.bin, xlab="height(inches)", ylab="finger length(cm)")
```



- 산점도와 함께 주변분포 표시

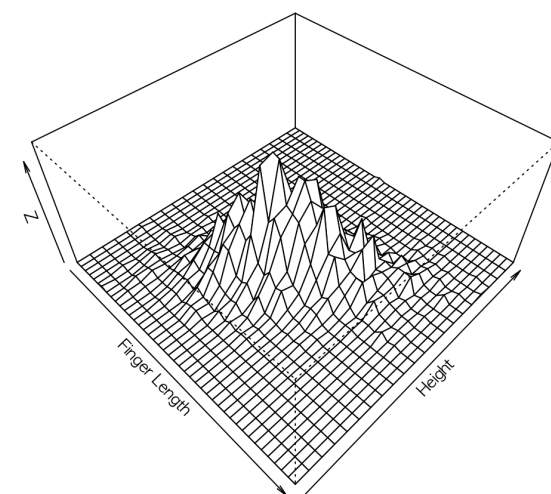
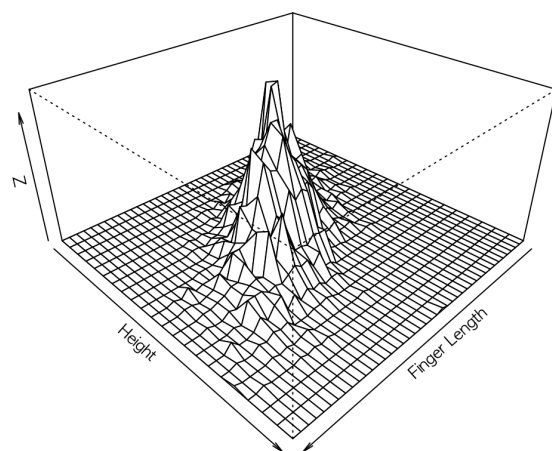
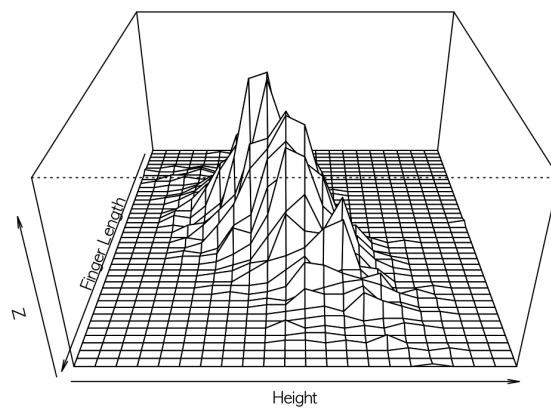
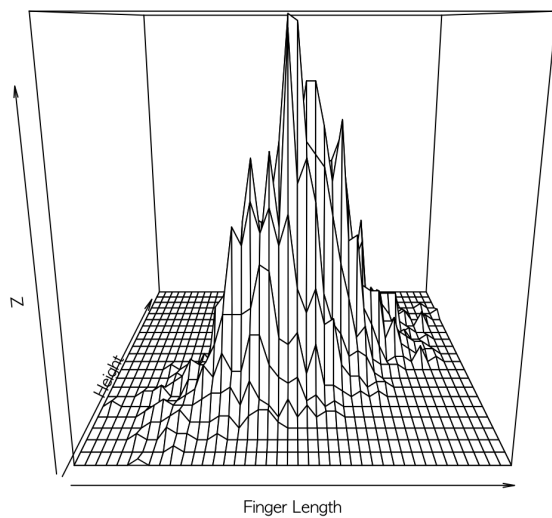
```
par(mar=c(4,4,1,1))
par(fig=c(0,0.8,0,0.8))
plot(finger~height, data=crimtab.3.long, pch=20)
par(fig=c(0,0.8,0.68,1), new=TRUE)
hist(crimtab.3.long[, "height"], axes=F, ann=F)
par(fig=c(0.68,1,0,0.8), new=TRUE)
barplot(table(cut(crimtab.3.long[, "finger"], breaks=10)), space=0, col="white",
horiz=T, axes=F, axisnames=F)
```



```
par(fig=c(0,1,0,1))
par(mar=c(5,4,1,1)+0.1)
```

- `persp()` 를 활용하면 다양한 각도에서 3차원 겨냥도를 그려볼 수 있음. x 축은 행, y 축은 열에 펼쳐진 격자를 0에서 1까지로 조정. θ 와 ϕ 는 박스를 돌려보는 각도이고, `expand`는 박스 높이의 상대적인 비율임. x 축과 y 축의 라벨 이외에는 디폴트값을 적용시킨 겨냥도와 적절히 조정한 겨냥도를 비교해 볼 것,

```
par(mfrow=c(2,2))
persp(crimtab.3, xlab="Finger Length", ylab="Height")
persp(crimtab.3, xlab="Finger Length", ylab="Height", theta=90, phi=30, expand=0.5, scale=TRUE)
persp(crimtab.3, xlab="Finger Length", ylab="Height", theta=135, phi=30, expand=0.5, scale=TRUE)
persp(crimtab.3, xlab="Finger Length", ylab="Height", theta=45, phi=45, expand=0.5, scale=TRUE)
```



```
par(mfrow=c(1,1))
```