Student 3000 Criminal Data

coop711

2015년 10월 7일

Structure of Data

Data Manipulation

- W. S. Gosset 이 t-분포를 유도하느라고 모의실험에 활용한 자료는 다음과 같음.
    - `table` 은 본질적으로 `matrix` 임. 행과 열의 이름에 측정값을 글자로 새겨 넣었기 때문에 long format으로 변환할 때 많은 주의가 필요함.

```
crimtab
```

```
##      142.24 144.78 147.32 149.86 152.4 154.94 157.48 160.02 162.56 165.1
## 9.4       0      0      0      0     0      0      0      0      0     0
## 9.5       0      0      0      0     0      1      0      0      0     0
## 9.6       0      0      0      0     0      0      0      0      0     0
## 9.7       0      0      0      0     0      0      0      0      0     0
## 9.8       0      0      0      0     0      0      1      0      0     0
## 9.9       0      0      1      0     1      0      1      0      0     0
## 10        1      0      0      1     2      0      2      0      0     1
## 10.1      0      0      0      1     3      1      0      1      1     0
## 10.2      0      0      2      2     2      1      0      2      0     1
## 10.3      0      1      1      3     2      2      3      5      0     0
## 10.4      0      0      1      1     2      3      3      4      3     3
## 10.5      0      0      0      1     3      7      6      4      3     1
## 10.6      0      0      0      1     4      5      9     14      6     3
## 10.7      0      0      1      2     4      9     14     16     15     7
## 10.8      0      0      0      2     5      6     14     27     10     7
## 10.9      0      0      0      0     2      6     14     24     27    14
## 11        0      0      0      2     6     12     15     31     37    27
## 11.1      0      0      0      3     3     12     22     26     24    26
## 11.2      0      0      0      3     2      7     21     30     38    29
## 11.3      0      0      0      1     0      5     10     24     26    39
## 11.4      0      0      0      0     3      4      9     29     56    58
## 11.5      0      0      0      0     0      5     11     17     33    57
## 11.6      0      0      0      0     2      1      4     13     37    39
## 11.7      0      0      0      0     0      2      9     17     30    37
## 11.8      0      0      0      0     1      0      2     11     15    35
## 11.9      0      0      0      0     1      1      2     12     10    27
## 12        0      0      0      0     0      0      1      4      8    19
## 12.1      0      0      0      0     0      0      0      2      4    13
## 12.2      0      0      0      0     0      0      1      2      5     6
## 12.3      0      0      0      0     0      0      0      0      4     8
## 12.4      0      0      0      0     0      0      1      1      1     2
## 12.5      0      0      0      0     0      0      0      1      0     1
## 12.6      0      0      0      0     0      0      0      0      0     1
## 12.7      0      0      0      0     0      0      0      0      0     1
## 12.8      0      0      0      0     0      0      0      0      0     0
## 12.9      0      0      0      0     0      0      0      0      0     0
## 13        0      0      0      0     0      0      0      0      0     0
## 13.1      0      0      0      0     0      0      0      0      0     0
## 13.2      0      0      0      0     0      0      0      0      0     0
## 13.3      0      0      0      0     0      0      0      0      0     0
## 13.4      0      0      0      0     0      0      0      0      0     0
## 13.5      0      0      0      0     0      0      0      0      0     0
##      167.64 170.18 172.72 175.26 177.8 180.34 182.88 185.42 187.96 190.5
## 9.4       0      0      0      0     0      0      0      0      0     0
## 9.5       0      0      0      0     0      0      0      0      0     0
## 9.6       0      0      0      0     0      0      0      0      0     0
## 9.7       0      0      0      0     0      0      0      0      0     0
## 9.8       0      0      0      0     0      0      0      0      0     0
## 9.9       0      0      0      0     0      0      0      0      0     0
## 10        0      0      0      0     0      0      0      0      0     0
## 10.1      0      0      0      0     0      0      0      0      0     0
## 10.2      0      0      0      0     0      0      0      0      0     0
```

```
## 10.3       0       0       0       0       0       0       0       0       0       0
## 10.4       0       0       0       0       0       0       0       0       0       0
## 10.5       3       1       0       1       0       0       0       0       0       0
## 10.6       1       0       0       1       0       0       0       0       0       0
## 10.7       3       1       2       0       0       0       0       0       0       0
## 10.8       1       2       1       0       0       0       0       0       0       0
## 10.9      10       4       1       0       0       0       0       0       0       0
## 11        17      10       6       0       0       0       0       0       0       0
## 11.1      24       7       4       1       0       0       0       0       0       0
## 11.2      27      20       4       1       0       0       0       0       0       0
## 11.3      26      24       7       2       0       0       0       0       0       0
## 11.4      26      22      10      11       0       0       0       0       0       0
## 11.5      38      34      25      11       2       0       0       0       0       0
## 11.6      48      38      27      12       2       2       0       1       0       0
## 11.7      48      45      24       9       9       2       0       0       0       0
## 11.8      41      34      29      10       5       1       0       0       0       0
## 11.9      32      35      19      10       9       3       1       0       0       0
## 12        42      39      22      16       8       2       2       0       0       0
## 12.1      22      28      15      27      10       4       1       0       0       0
## 12.2      23      17      16      11       8       1       1       0       0       0
## 12.3      10      13      20      23       6       5       0       0       0       0
## 12.4       7      12       4       7       7       1       0       0       1       0
## 12.5       3      12      11       8       6       8       0       2       0       0
## 12.6       0       3       5       7       8       6       3       1       1       0
## 12.7       1       7       5       5       8       2       2       0       0       0
## 12.8       1       2       3       1       8       5       3       1       1       0
## 12.9       0       1       2       2       0       1       1       0       0       0
## 13         3       0       1       0       1       0       2       1       0       0
## 13.1       0       1       1       0       0       0       0       0       0       0
## 13.2       1       1       0       1       0       3       0       0       0       0
## 13.3       0       0       0       0       0       0       1       0       1       0
## 13.4       0       0       0       0       0       0       0       0       0       0
## 13.5       0       0       0       0       0       0       0       1       0       0
##       193.04  195.58
## 9.4        0       0
## 9.5        0       0
## 9.6        0       0
## 9.7        0       0
## 9.8        0       0
## 9.9        0       0
## 10         0       0
## 10.1       0       0
## 10.2       0       0
## 10.3       0       0
## 10.4       0       0
## 10.5       0       0
## 10.6       0       0
## 10.7       0       0
## 10.8       0       0
## 10.9       0       0
## 11         0       0
## 11.1       0       0
## 11.2       0       1
## 11.3       0       0
```

```
## 11.4      0      0
## 11.5      0      0
## 11.6      0      0
## 11.7      0      0
## 11.8      0      0
## 11.9      0      0
## 12        0      0
## 12.1      0      0
## 12.2      0      0
## 12.3      0      0
## 12.4      0      0
## 12.5      0      0
## 12.6      0      0
## 12.7      0      0
## 12.8      0      0
## 12.9      0      0
## 13        0      0
## 13.1      0      0
## 13.2      0      0
## 13.3      0      0
## 13.4      0      0
## 13.5      0      0
```

```
str(crimtab)
```

```
##  'table' int [1:42, 1:22] 0 0 0 0 0 0 1 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:42] "9.4" "9.5" "9.6" "9.7" ...
##   ..$ : chr [1:22] "142.24" "144.78" "147.32" "149.86" ...
```

- 이러한 table 구조의 행과 열에 이름을 붙이려면, `list` 로 부여
    - cm 단위로 되어 있는 키의 계급을 인치 단위로 변환하면 숨어있던 자료구조다 드러남.

```
dimnames(crimtab)
```

```
## [[1]]
##  [1] "9.4"  "9.5"  "9.6"  "9.7"  "9.8"  "9.9"  "10"   "10.1" "10.2" "10.3"
## [11] "10.4" "10.5" "10.6" "10.7" "10.8" "10.9" "11"   "11.1" "11.2" "11.3"
## [21] "11.4" "11.5" "11.6" "11.7" "11.8" "11.9" "12"   "12.1" "12.2" "12.3"
## [31] "12.4" "12.5" "12.6" "12.7" "12.8" "12.9" "13"   "13.1" "13.2" "13.3"
## [41] "13.4" "13.5"
##
## [[2]]
##  [1] "142.24" "144.78" "147.32" "149.86" "152.4"  "154.94" "157.48"
##  [8] "160.02" "162.56" "165.1"  "167.64" "170.18" "172.72" "175.26"
## [15] "177.8"  "180.34" "182.88" "185.42" "187.96" "190.5"  "193.04"
## [22] "195.58"
```

```
crimtab.2 <- crimtab
colnames(crimtab.2) <- as.numeric(colnames(crimtab.2))/2.54
(dimnames(crimtab.2) <- list(finger = rownames(crimtab.2), height = colnames(cr
imtab.2)))
```

```
## $finger
##  [1] "9.4"  "9.5"  "9.6"  "9.7"  "9.8"  "9.9"  "10"   "10.1" "10.2" "10.3"
## [11] "10.4" "10.5" "10.6" "10.7" "10.8" "10.9" "11"   "11.1" "11.2" "11.3"
## [21] "11.4" "11.5" "11.6" "11.7" "11.8" "11.9" "12"   "12.1" "12.2" "12.3"
## [31] "12.4" "12.5" "12.6" "12.7" "12.8" "12.9" "13"   "13.1" "13.2" "13.3"
## [41] "13.4" "13.5"
##
## $height
##  [1] "56" "57" "58" "59" "60" "61" "62" "63" "64" "65" "66" "67" "68" "69"
## [15] "70" "71" "72" "73" "74" "75" "76" "77"
```

```
crimtab.2
```

```
##       height
## finger 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  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  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  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  0  1  2  2  0  1  1  0  0  0  0  0
##   13    0  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  0  1  1  0  0  0  0  0  0  0  0  0
##   13.2  0  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  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  0
##   13.5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
```

```
str(crimtab.2)
```

```
##  'table' int [1:42, 1:22] 0 0 0 0 0 0 1 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ finger: chr [1:42] "9.4" "9.5" "9.6" "9.7" ...
##   ..$ height: chr [1:22] "56" "57" "58" "59" ...
```

- long format 으로 변환
  - 1차적으로 각 손가락 길이와 키의 조합을 갖는 인원수로 구성된 data frame으로 변환하고, case 단위의 long format data frame으로 2단계 변환
  - `as.data.frame()` 에서 `stringsAsFactors = FALSE` 가 매우 중요한 역할을 하는 것임. 이 옵션을 설정하지 않을 경우 `Factor` 로 잡히면 `numeric` 으로 전환할 수 없게 됨. `Factor` 는 본질적으로 음이 아닌 정수로 취급됨.
  - 보통 이 설정은 `stringsAsFactors = default.stringsAsFactors()` 가 default 설정으로 되어 있고, 따라서 `default.stringsAsFactors() =` {r} default.stringsAsFactors()` 에 따라 설정을 바꿔주면 되어야 하나 써 있는 대로 되지 않는 경우가 있어서 가능한 설정해주는 것이 안전함.
  - 단순히 `data.frame()` 으로 변환할 경우 `Factor` 로 설정되어 `numeric` 으로 변환하더라도 의미없는 숫자를 얻게 됨.
  - `as.data.frame()` 의 결과물로 두 개의 character vector 와 counts를 나타내는 새로운 변수 `Freq` 가 나오게 됨. 손가락 길이와 키를 나타내는 character를 numeric으로 전환하고 다음 작업 진행.

```
crimtab.2.df <- as.data.frame(crimtab.2, stringsAsFactors = FALSE)
str(crimtab.2.df)
```

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

```
crimtab.2.df[1:2] <- sapply(crimtab.2.df[1:2], as.numeric)
str(crimtab.2.df)
```

```
## 'data.frame':    924 obs. of  3 variables:
##  $ 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 ...
##  $ Freq  : int  0 0 0 0 0 0 1 0 0 0 ...
```

```
crimtab.2.long <- mapply(function(x) rep(x, crimtab.2.df$Freq), crimtab.2.d
f[c("finger", "height")])
# crimtab.2.long <- mapply(function(x, y) rep(x, y = crimtab.2.df$Freq), crimta
b.2.df[c("finger", "height")])
# crimtab.2.long <- mapply(function(x, y) rep(x, y), y = crimtab.2.df$Freq, cri
mtab.2.df[c("finger", "height")])
# crimtab.2.long <- mapply(function(x) rep(x, crimtab.2.df$Freq), crimtab.2.d
f[, c("finger", "height")])
# crimtab.2.long <- mapply(function(x) rep(x, crimtab.2.df$Freq), crimtab.2.d
f[1:2])
# crimtab.2.long <- mapply(function(x) rep(x, crimtab.2.df$Freq), crimtab.2.d
f[, 1:2])
# crimtab.2.long <- mapply(function(x) rep(x, crimtab.2.df$Freq), crimtab.2.d
f[, 2])
# crimtab.2.long <- mapply(function(x) rep(x, crimtab.2.df$Freq), crimtab.2.d
f$height)
str(crimtab.2.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"
```

```
crimtab.2.long.df <- data.frame(crimtab.2.long)
str(crimtab.2.long.df)
```

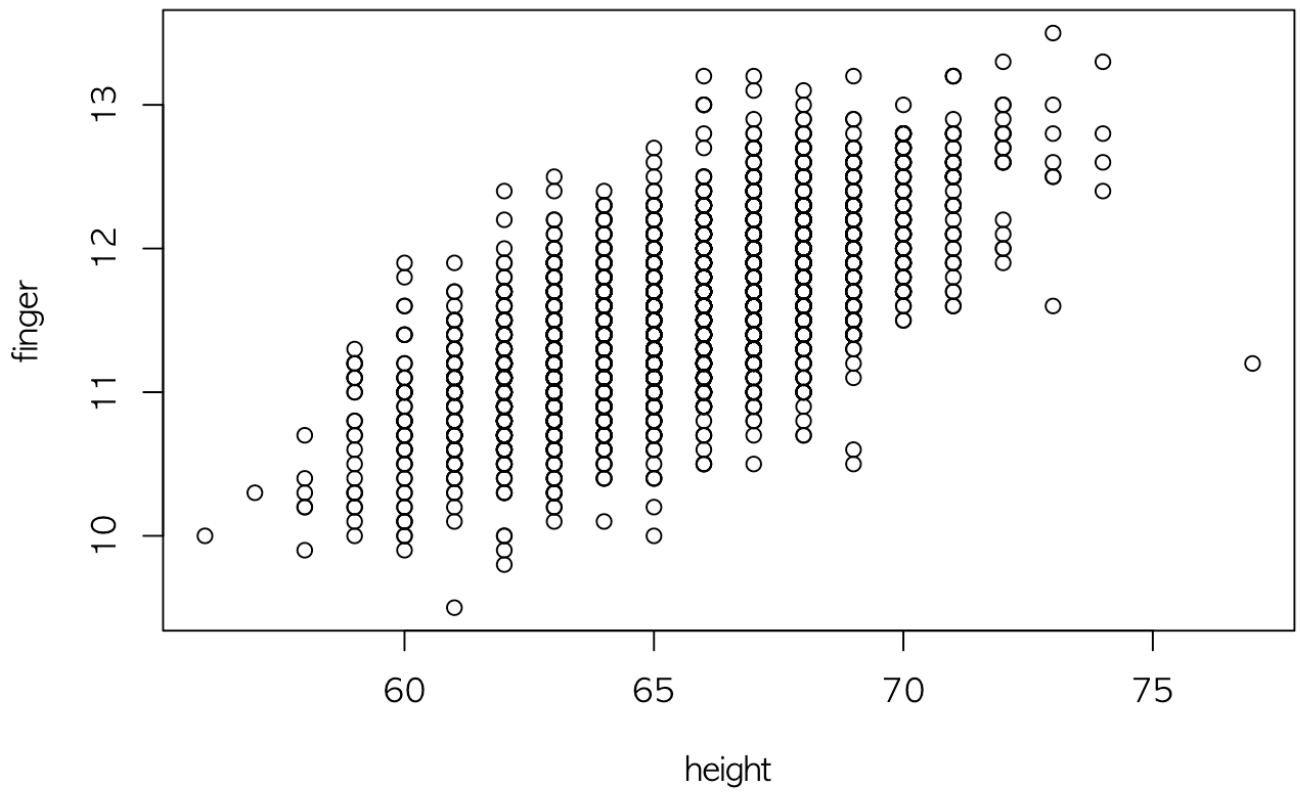
```
## 'data.frame':    3000 obs. of  2 variables:
##  $ finger: num  10 10.3 9.9 10.2 10.2 10.3 10.4 10.7 10 10.1 ...
##  $ height: num  56 57 58 58 58 58 58 58 59 59 ...
```

Graphic Representation

Base Graphics

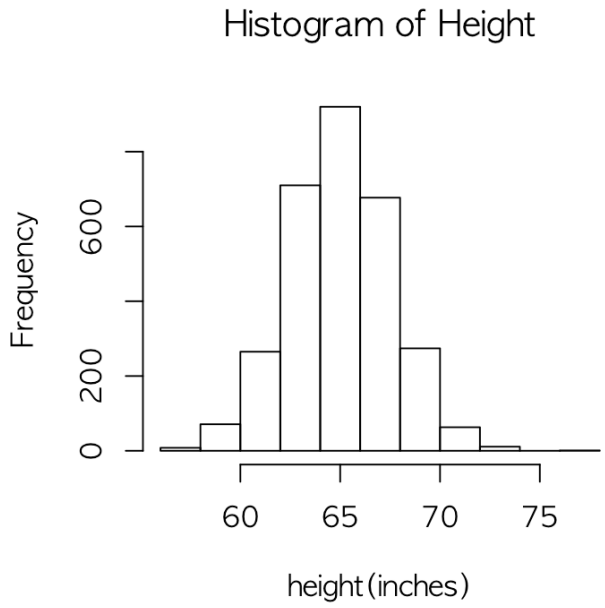
- 키와 손가락길이의 산점도

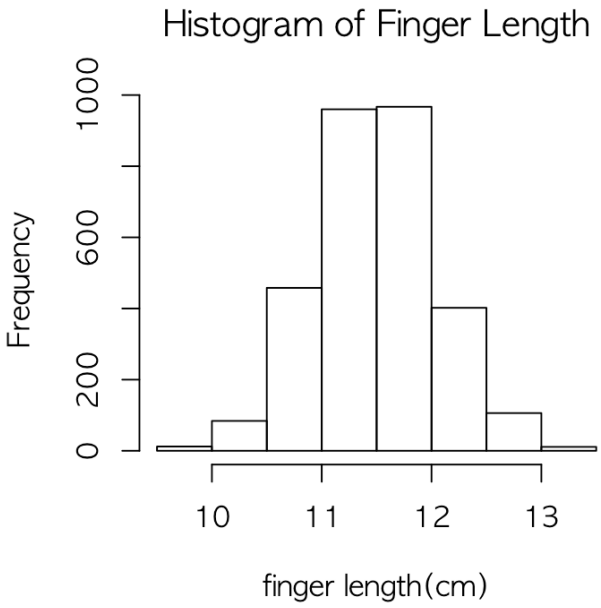
```
plot(finger ~ height, data = crimtab.2.long.df)
```



- 변수 각각의 히스토그램은?

```
par(mfrow=c(1,2))
hist(crimtab.2.long.df$height, main="Histogram of Height", xlab="height(inche
s)")
hist(crimtab.2.long.df$finger, main="Histogram of Finger Length", xlab= "finger
length(cm)")
```






```
# hist(crimtab.2.long.df["height"], main="Histogram of Height", xlab="height(in
ches)")
# hist(crimtab.2.long.df["finger"], main="Histogram of Finger Length", xlab= "f
inger length(cm)")
```

- 평균과 표준편차를 구하면 다음과 같음. 이를 모수로 하는 정규곡선을 덧씌워 볼 것.

```
apply(crimtab.2.long, 2, mean)
```

```
##   finger   height
## 11.54737 65.47300
```

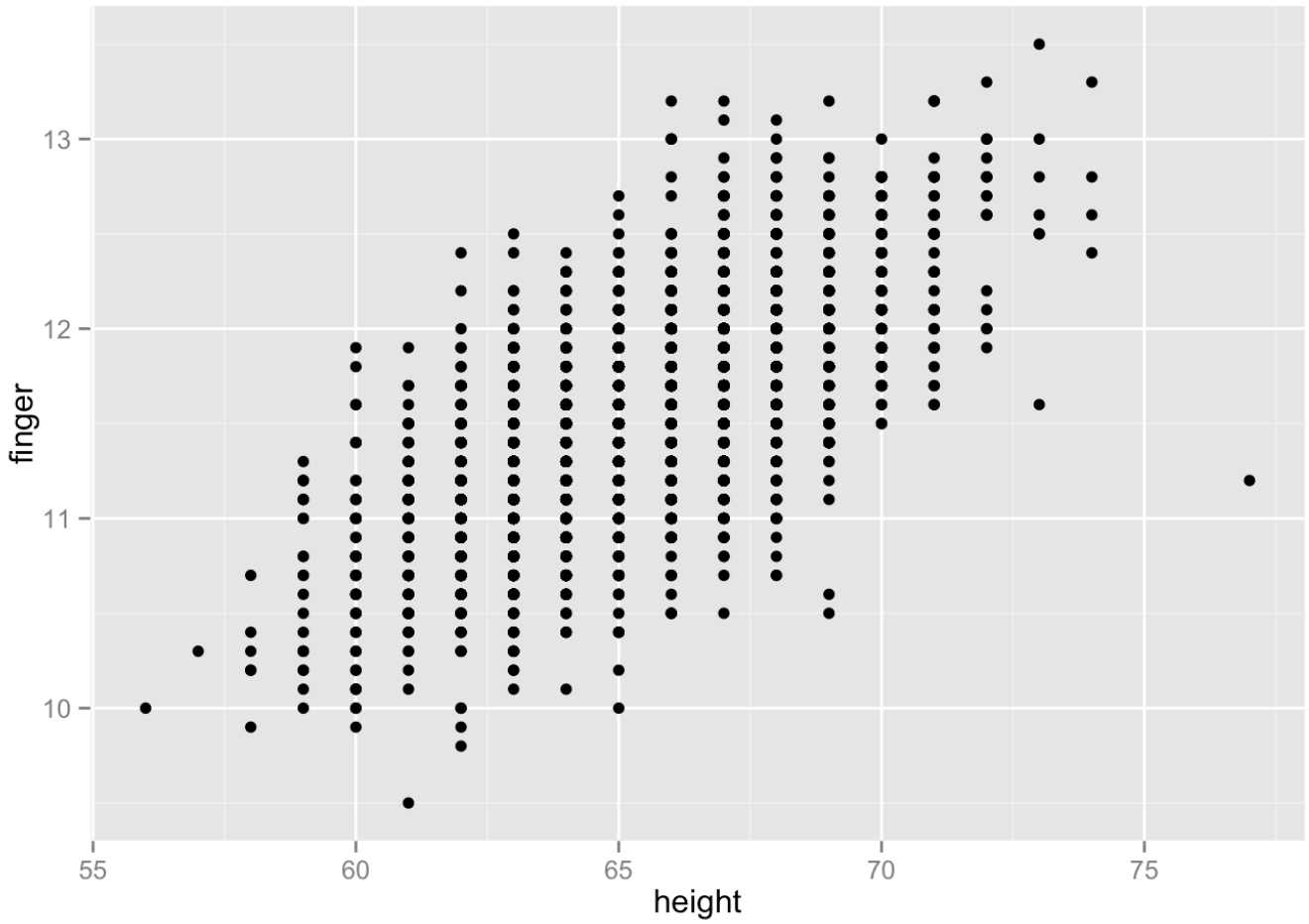
```
apply(crimtab.2.long, 2, sd)
```

```
##    finger    height
## 0.5487137 2.5577565
```

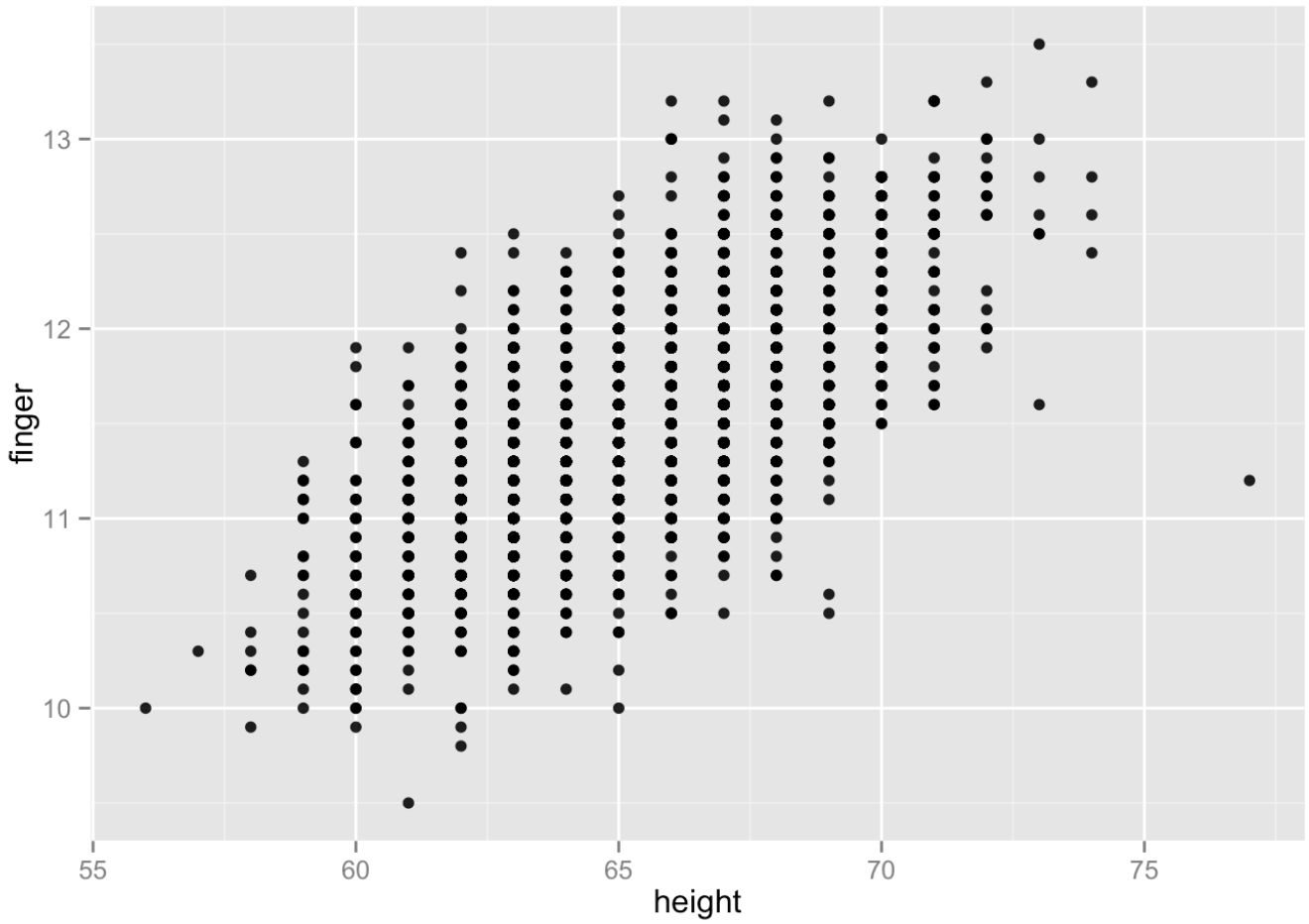
ggplot

- 키와 손가락 길이의 산점도

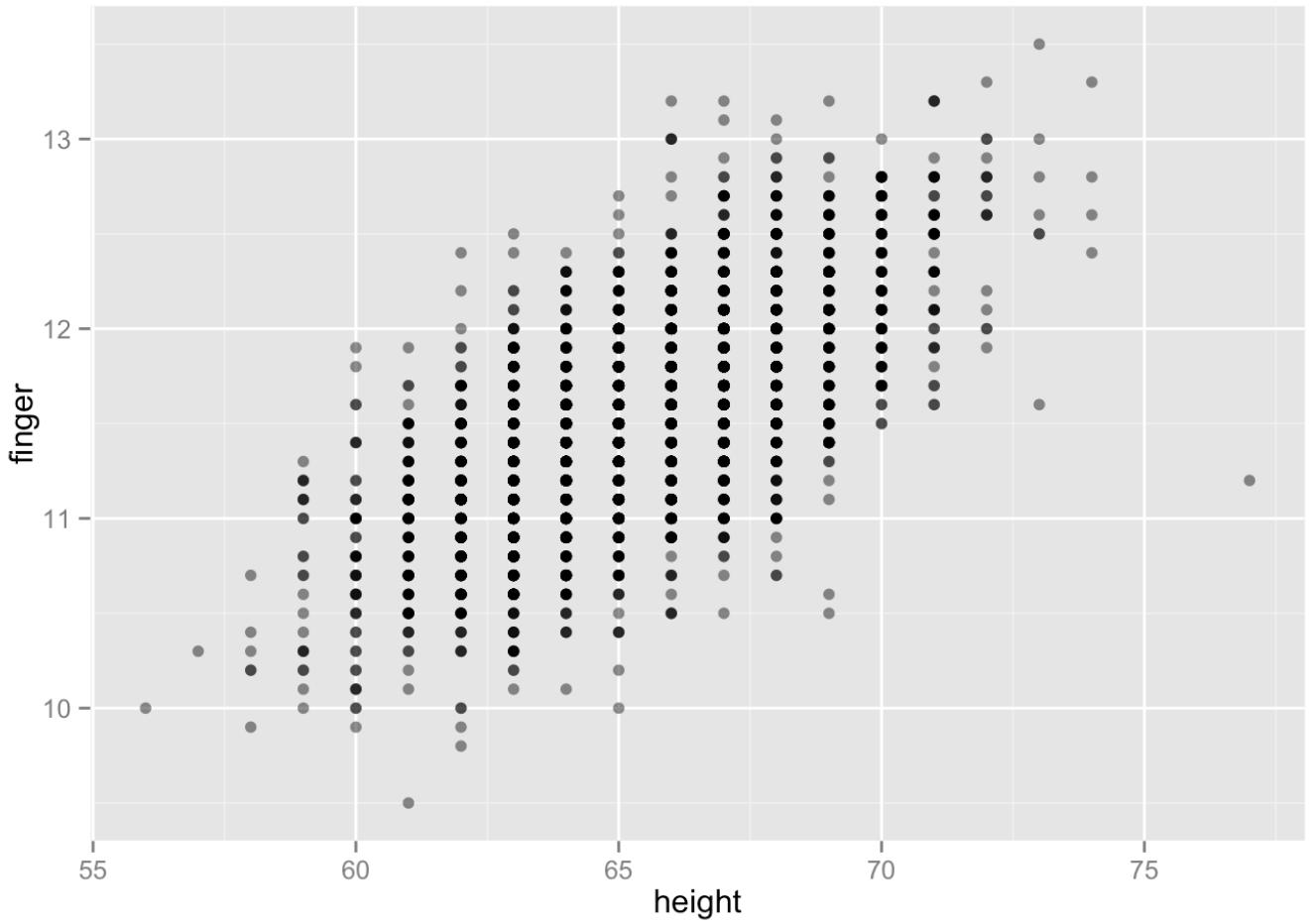
```
library(ggplot2)
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point()
```



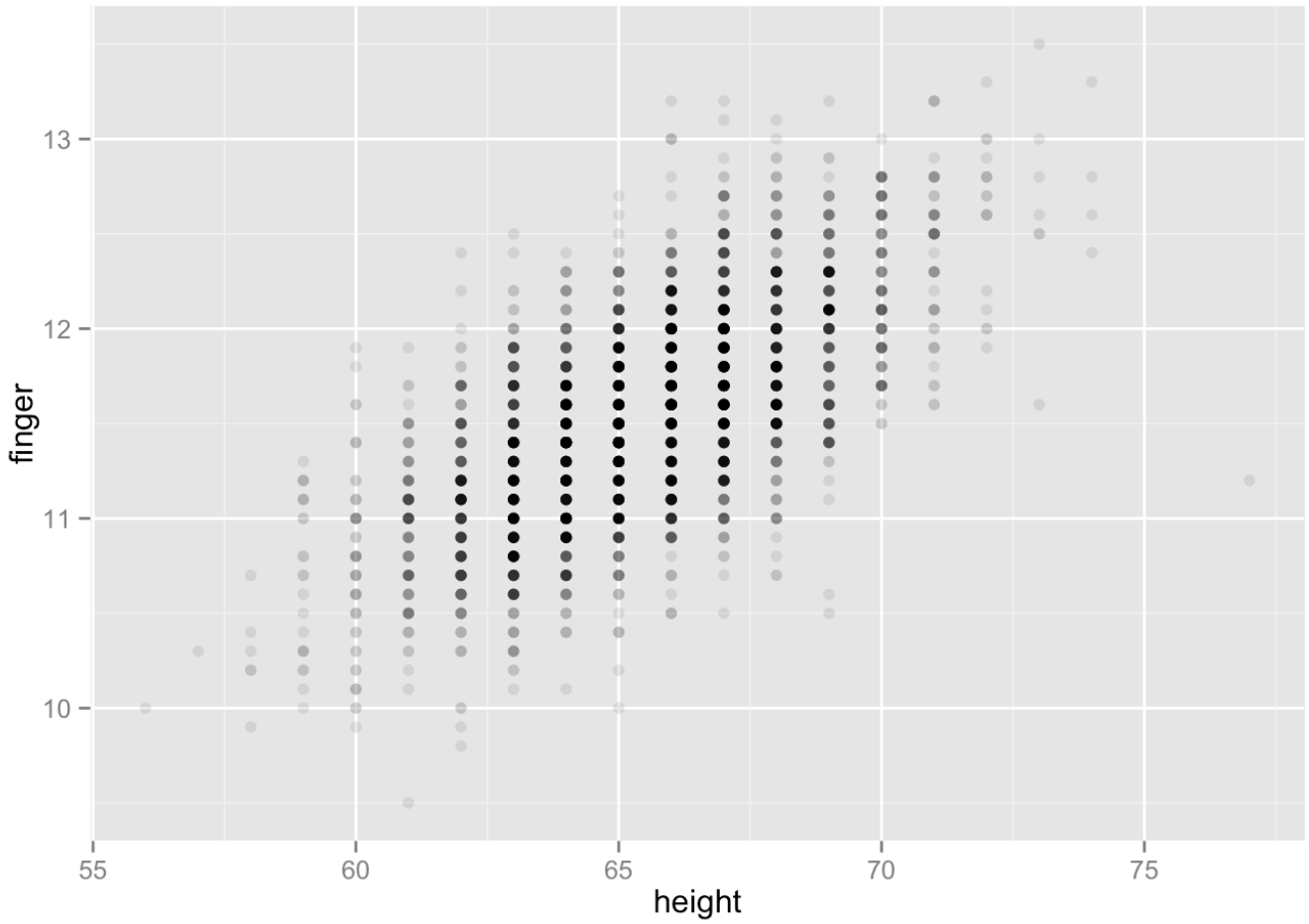

```
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point(alpha =
0.9)
```



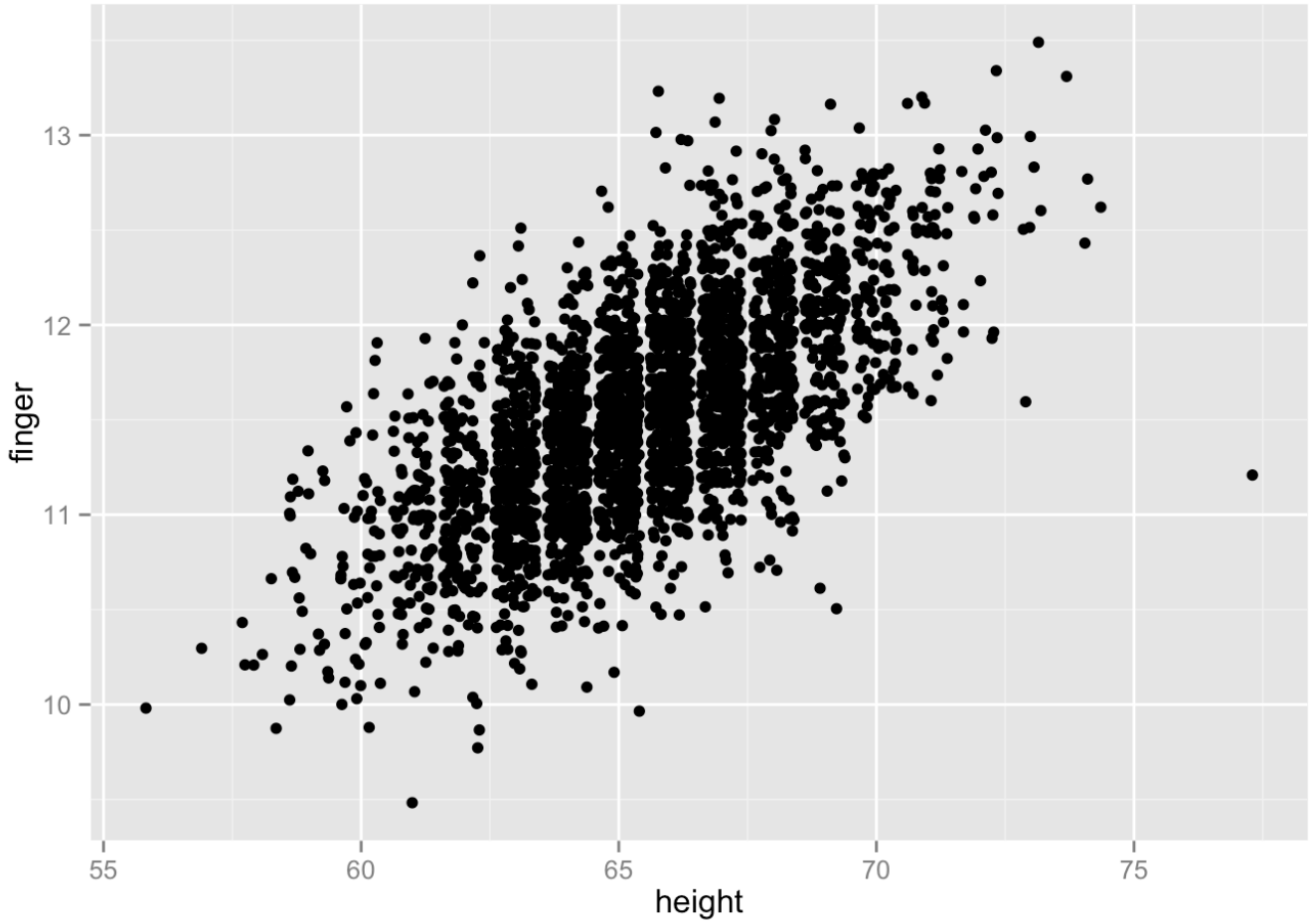

```
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point(alpha =
0.5)
```



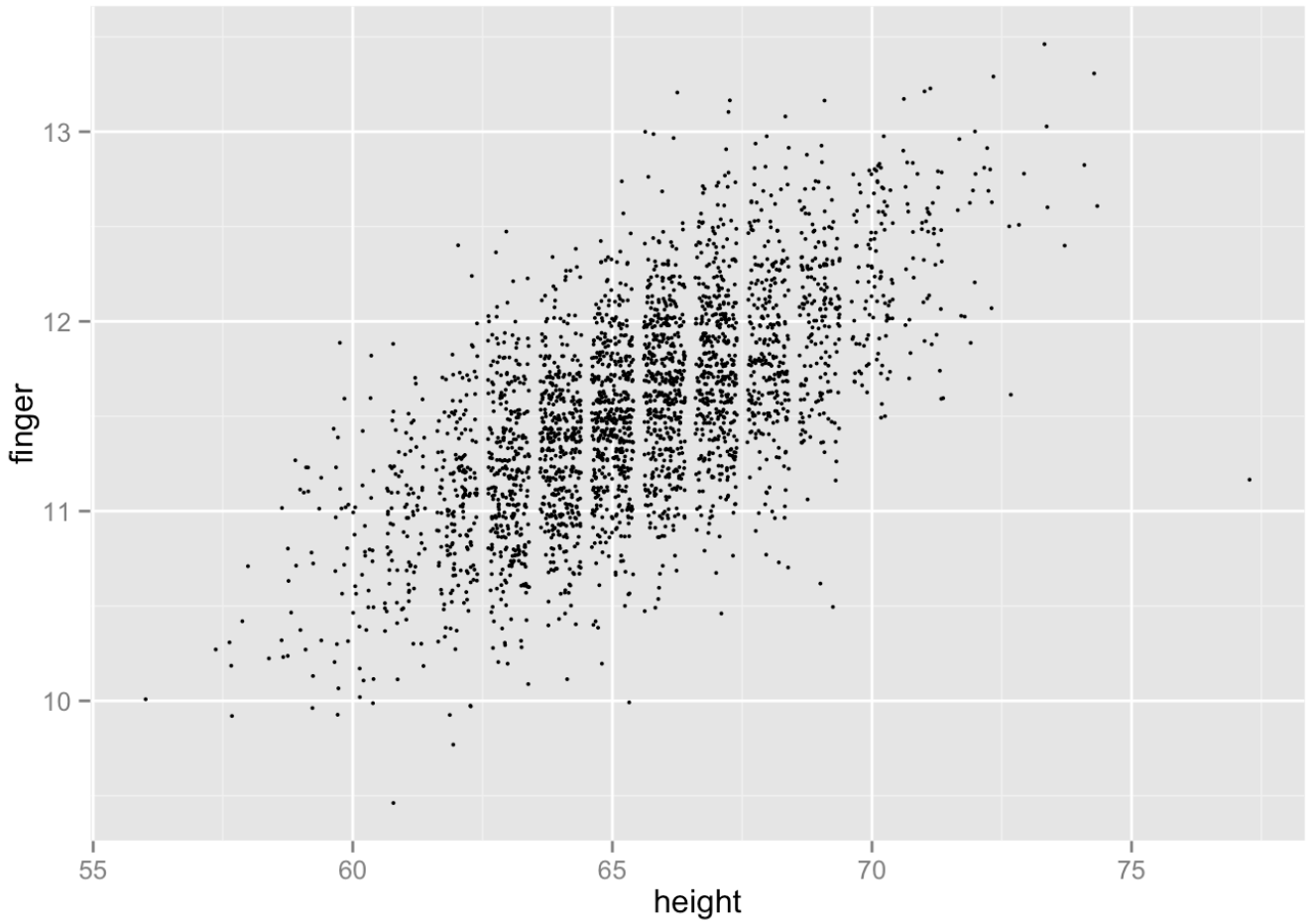

```
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point(alpha =
0.1)
```



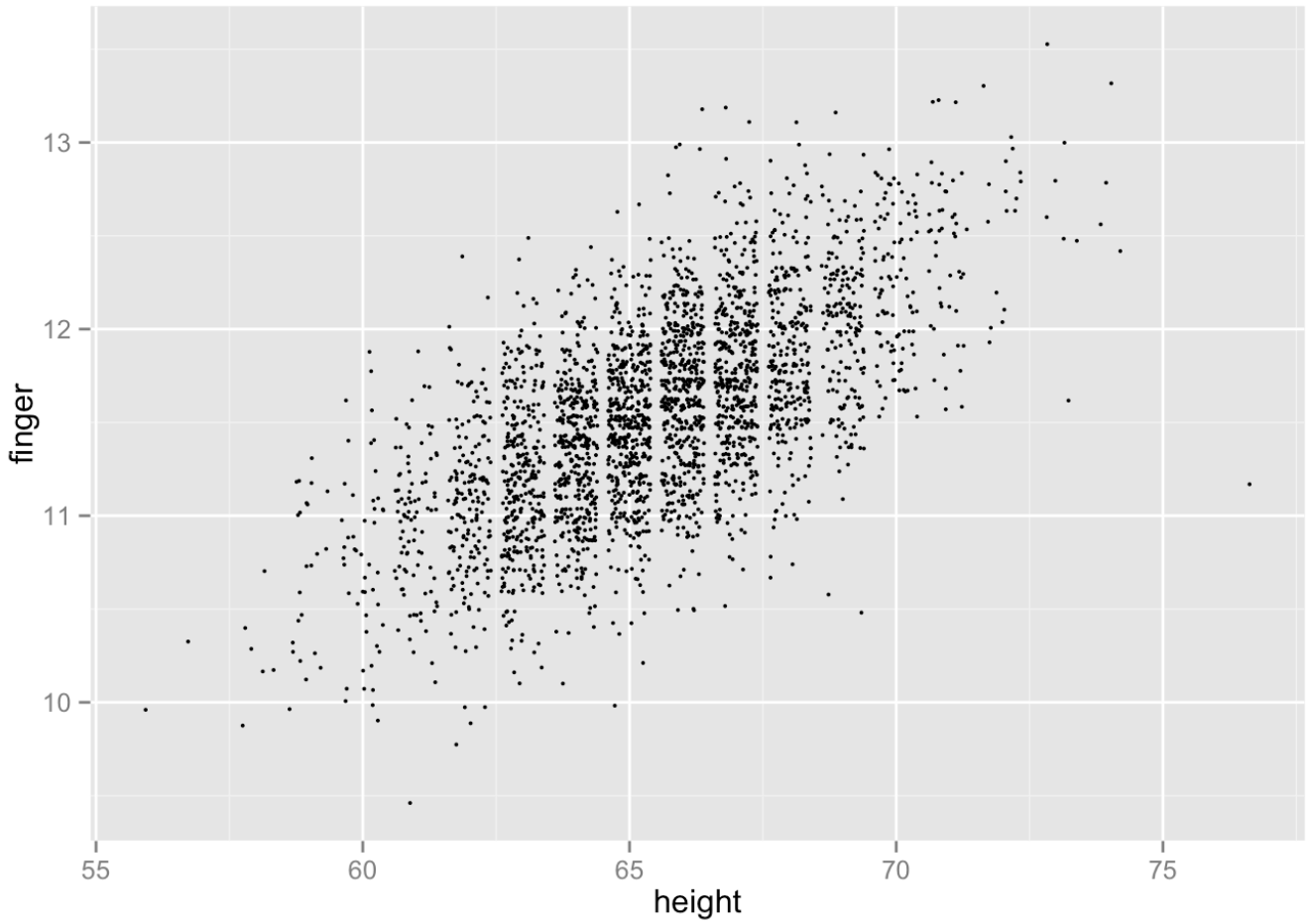

```
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point(position =
"jitter")
```



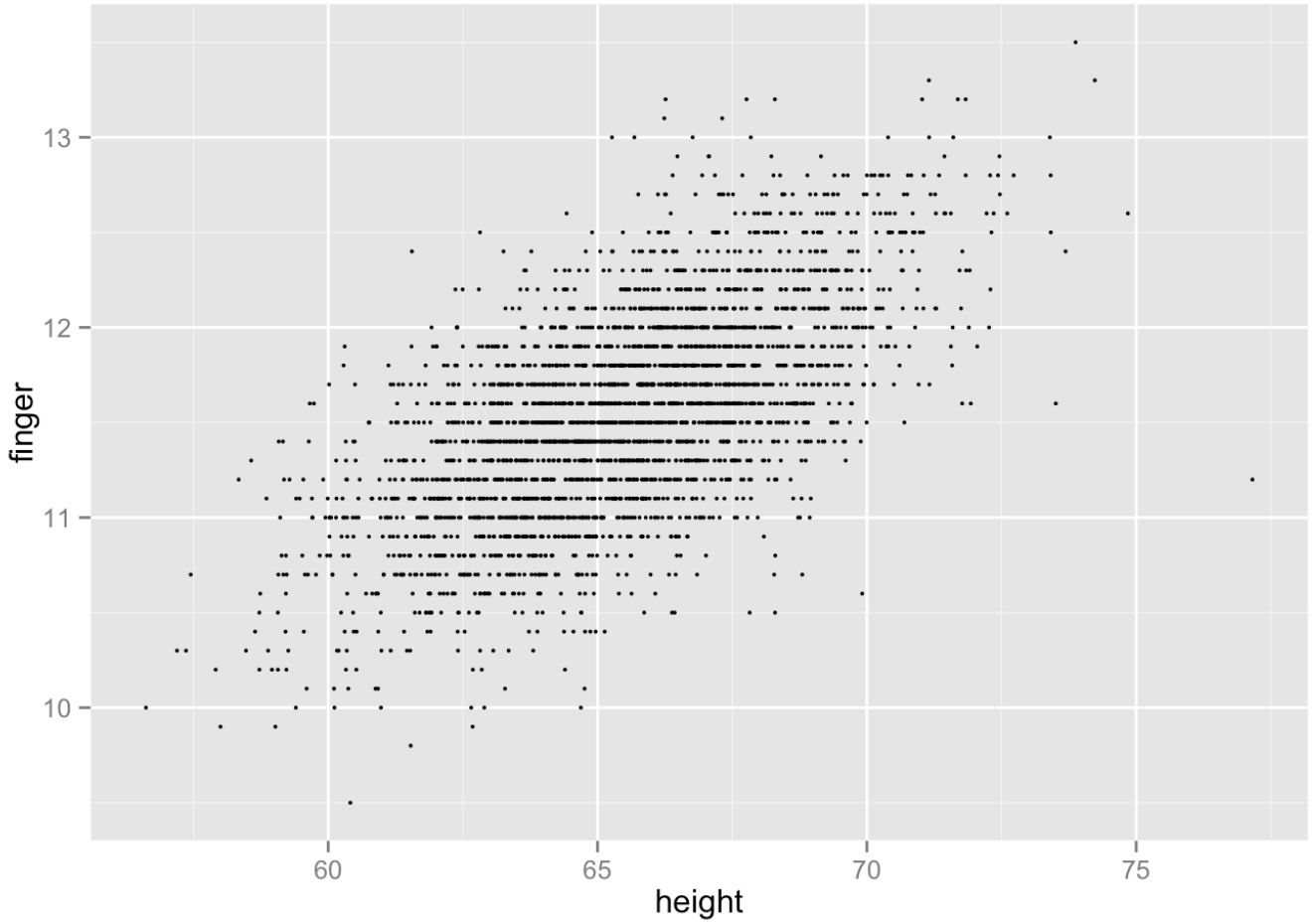

```
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point(position =
"jitter", size = 0.7)
```



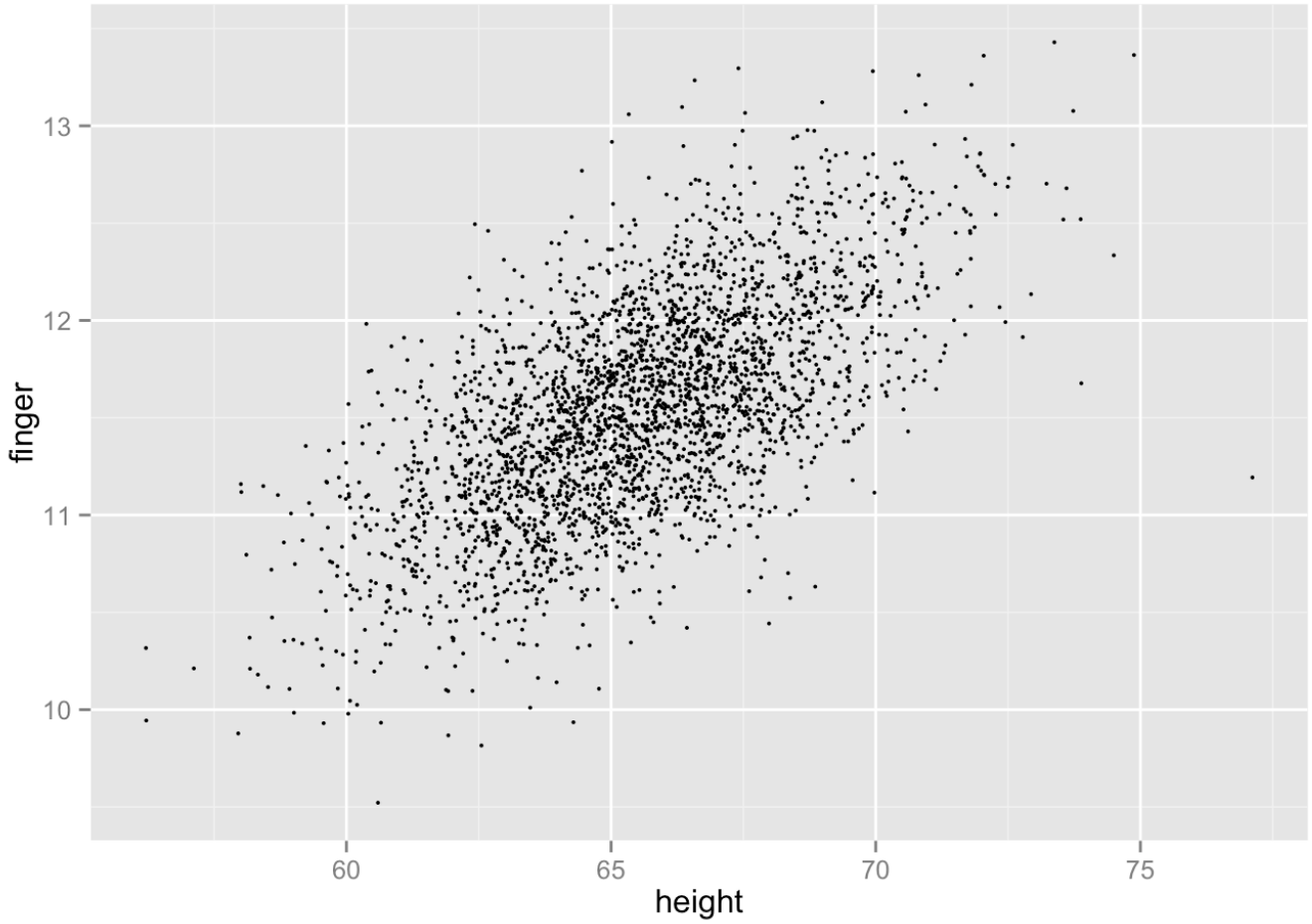

```
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point(position =
position_jitter(), size = 0.7)
```



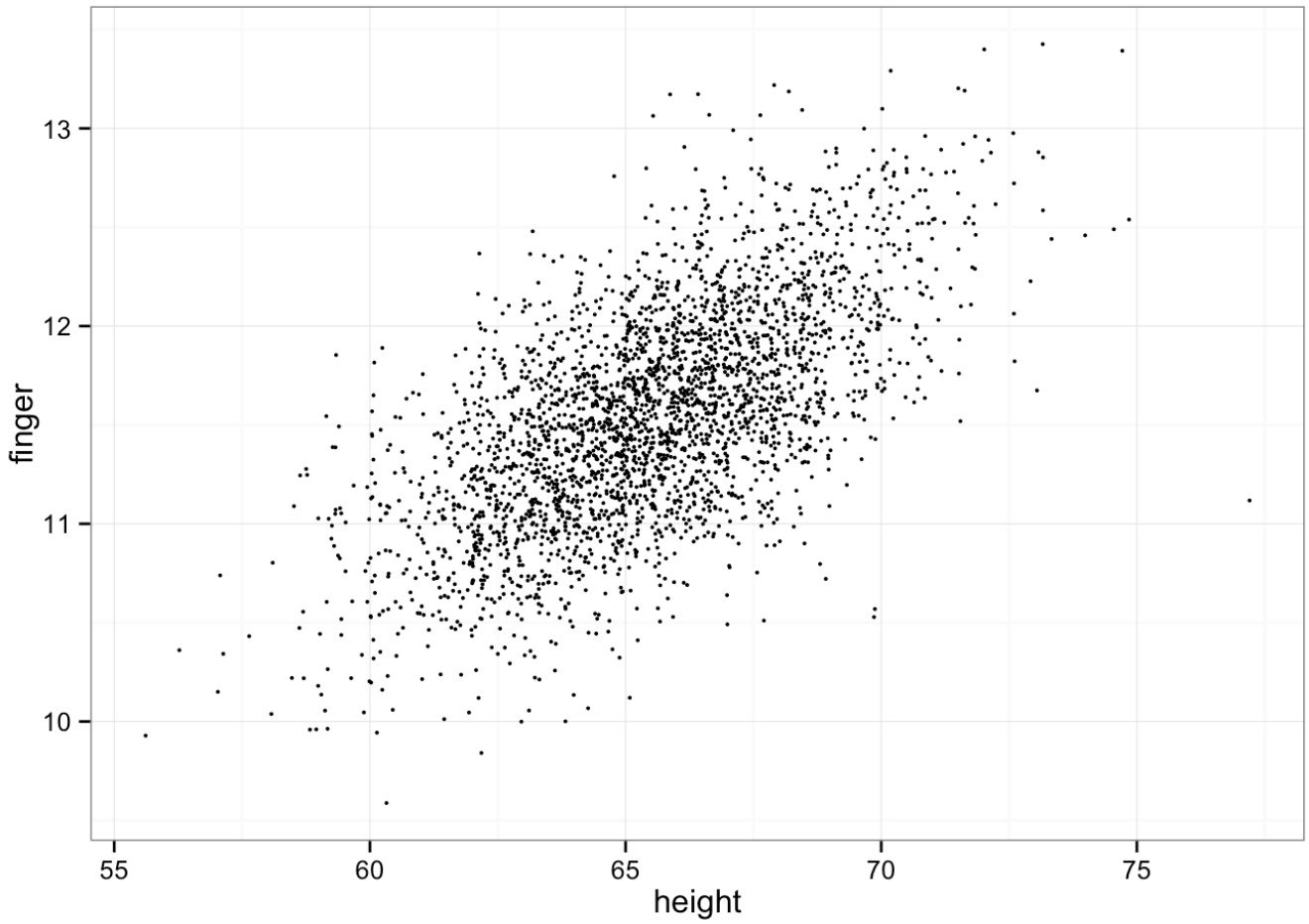

```
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point(position =
position_jitter(width = 1, height = 0), size = 0.7)
```




```
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point(position =
position_jitter(width = 1, height = 0.1), size = 0.7)
```



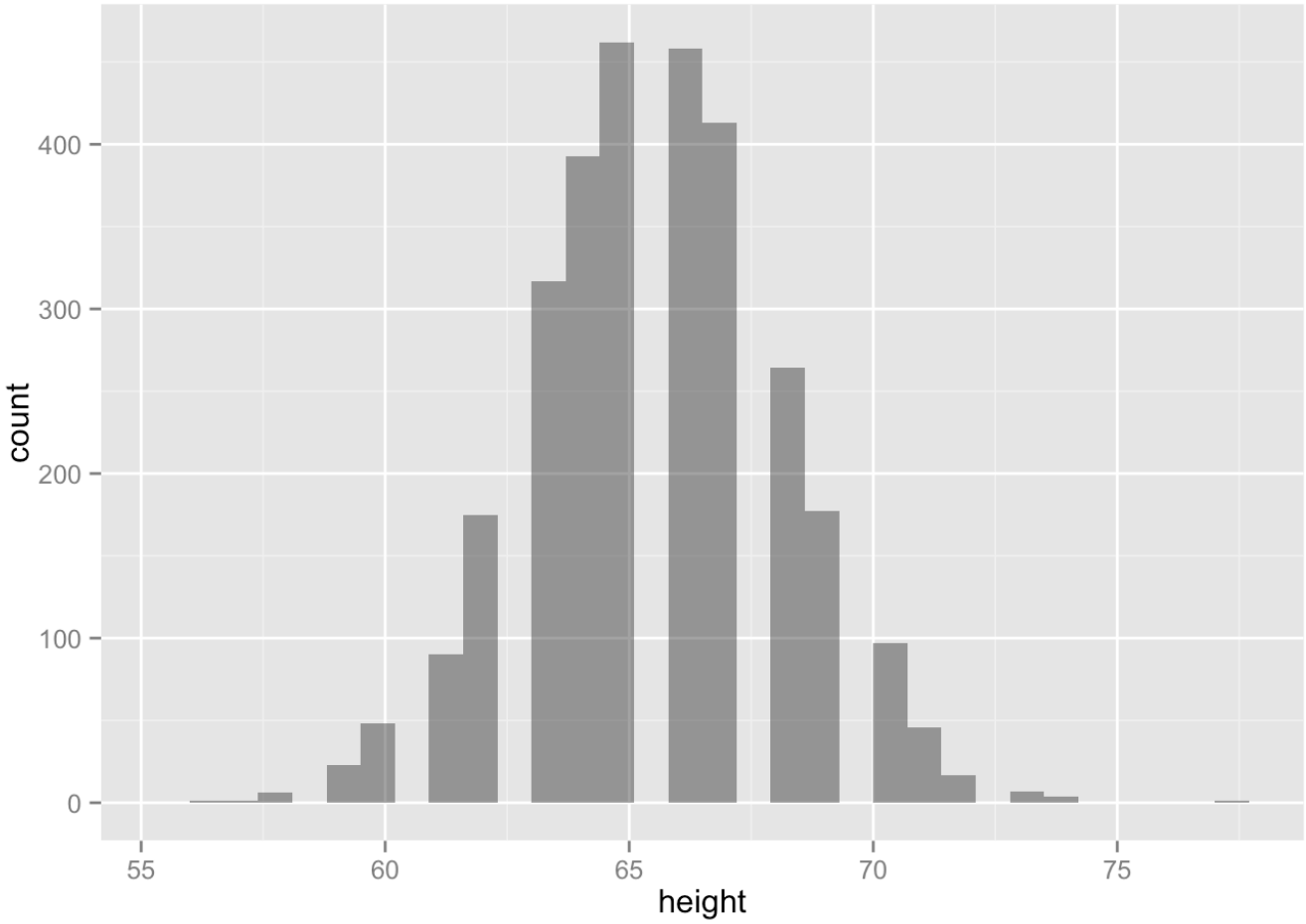

```
ggplot(crimtab.2.long.df, aes(x = height, y = finger)) + geom_point(position =
position_jitter(width = 1, height = 0.1), size = 0.7) +
  theme_bw()
```



- 변수 각각의 히스토그램

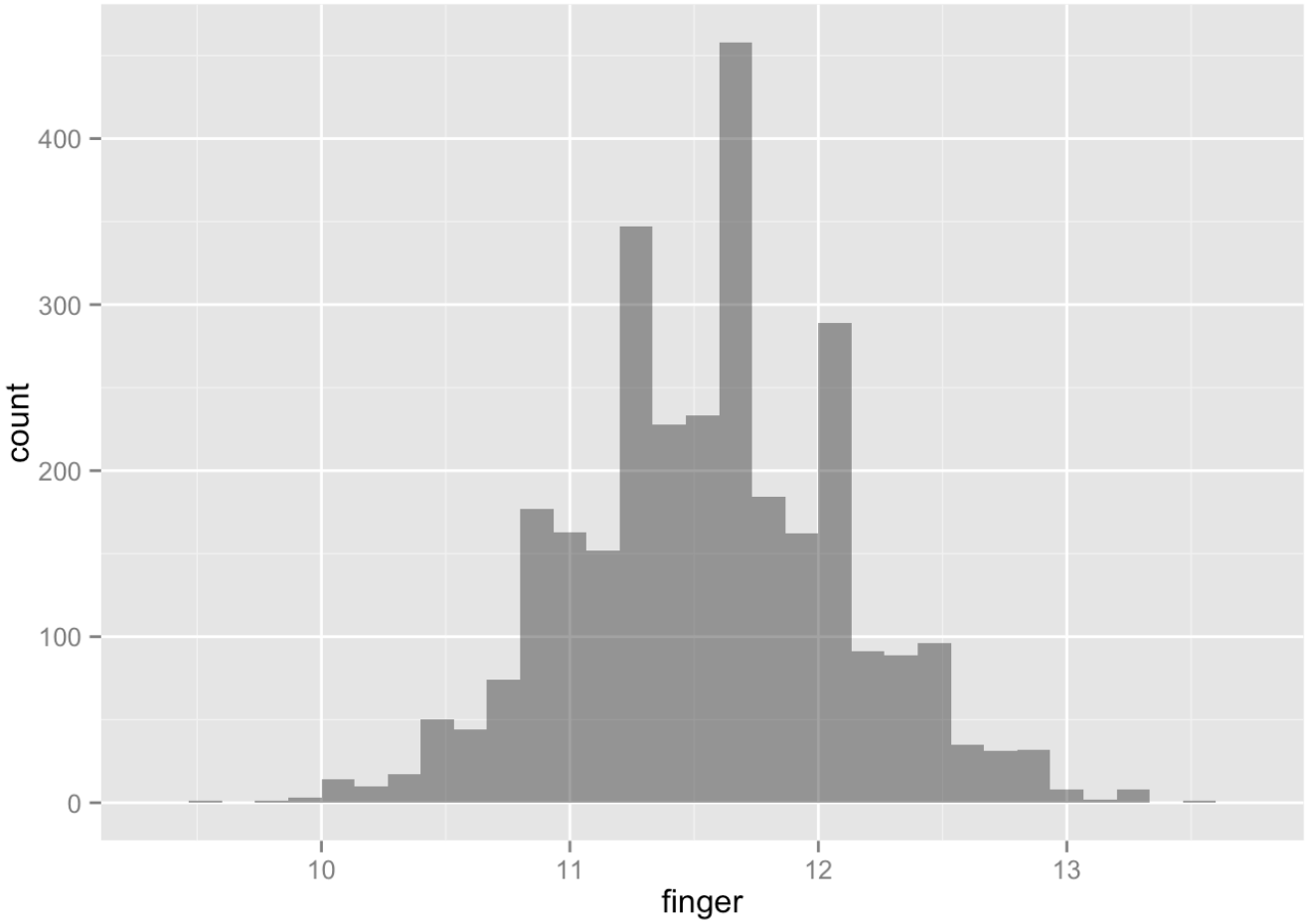
```
ggplot(crimtab.2.long.df, aes(x = height)) + geom_histogram(alpha=0.5)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

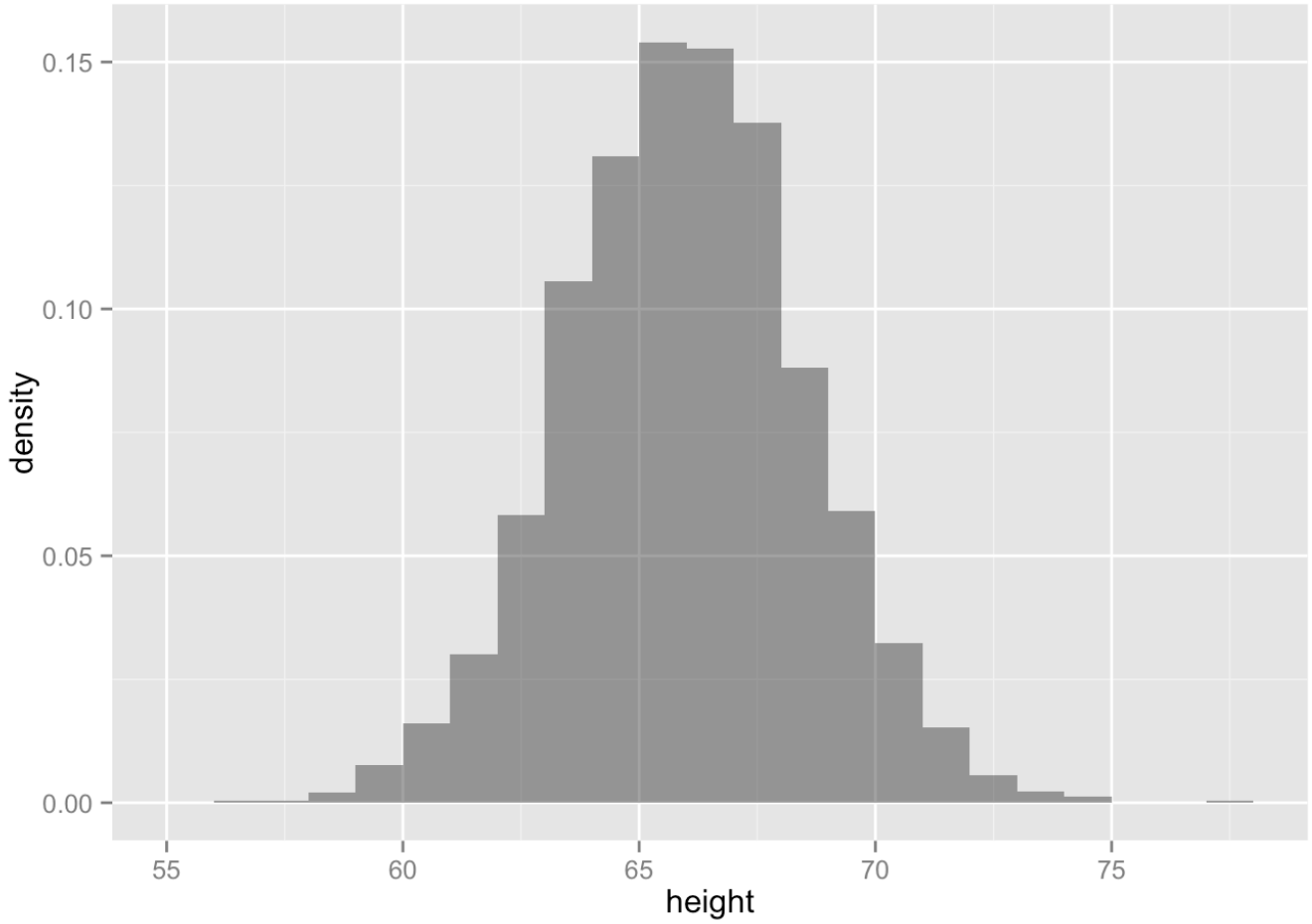



```
ggplot(crimtab.2.long.df, aes(x = finger)) + geom_histogram(alpha=0.5)
```

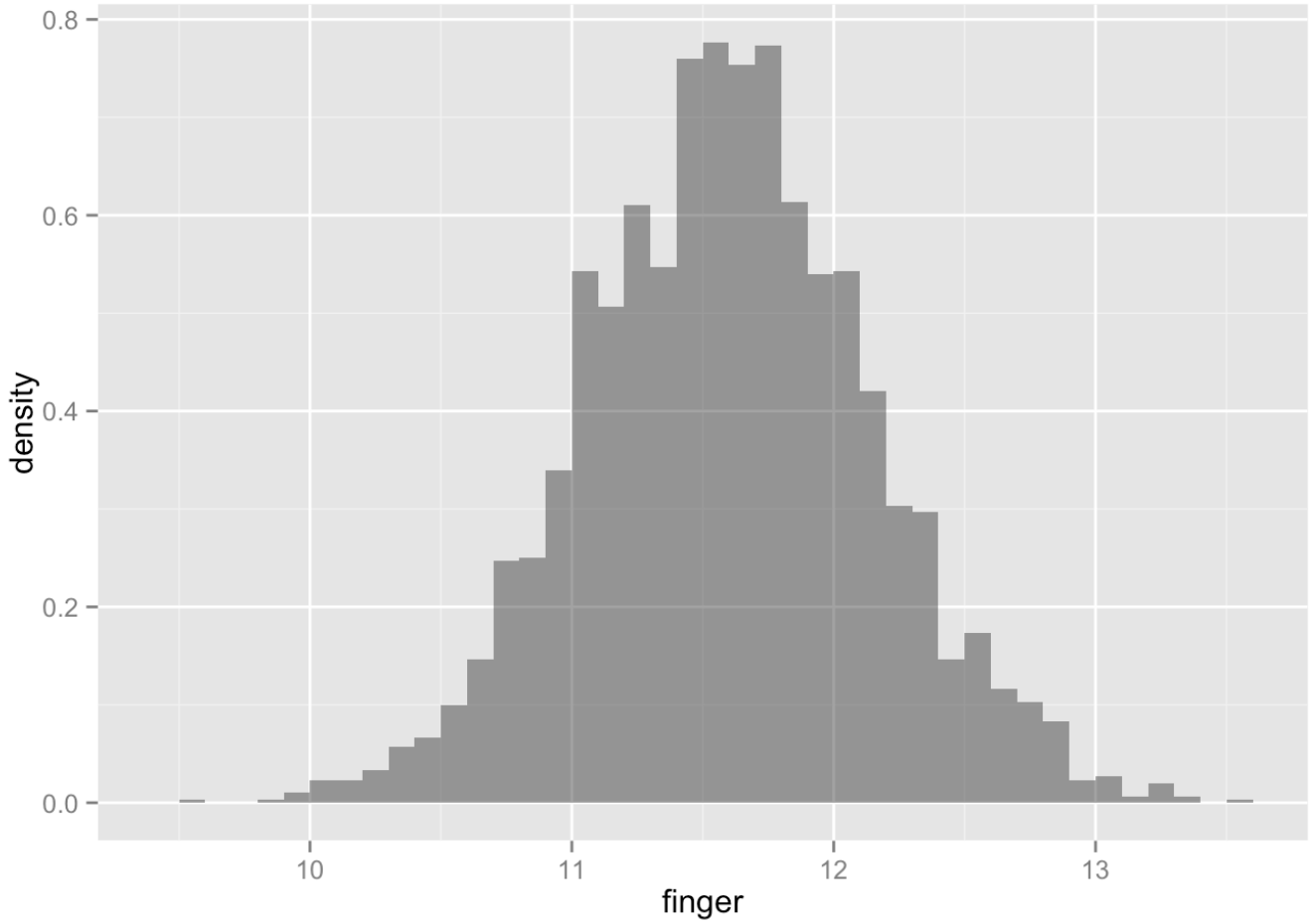
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
ggplot(crimtab.2.long.df, aes(x = height)) + geom_histogram(aes(y = ..densit
y..), binwidth=1, alpha=0.5)
```




```
ggplot(crimtab.2.long.df, aes(x = finger)) + geom_histogram(aes(y = ..densit
y..), binwidth=0.1, alpha=0.5)
```


0.8
0.6
0.4
0.2
0.0
density
10
11
12
13
finger