Student 3000 Criminal Data

coop711

2015년 3월 19일

Structure of Data

- W. S. Gosset 이 t-분포를 유도하느라고 모의실험에 활용한 자료는 다음과 같음.

```
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" ...
```

- 이 자료를 long format 으로 전환하는 과정은 다음과 같음. as.data.frame() 에서 stringsAsFactors=FALSE가 매우 중요한 역할을 하는 것임. 이 옵션을 설정하지 않을 경우 Factor로 잡히면 numeric으로 전환할 수 없게 됨. Factor는 본질적으로 음이 아닌 정수로 취급됨.

```
crimtab.2<-crimtab
crimtab.2.df<-as.data.frame(crimtab.2, stringsAsFactors = F)
crimtab.2.df$finger<-as.numeric(crimtab.2.df$Var1)
crimtab.2.df$height<-as.numeric(crimtab.2.df$Var2)
str(crimtab.2.df)
```

```
## 'data.frame':    924 obs. of  5 variables:
##  $ Var1  : chr  "9.4" "9.5" "9.6" "9.7" ...
##  $ Var2  : chr  "142.24" "142.24" "142.24" "142.24" ...
##  $ 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  142 142 142 142 142 ...
```

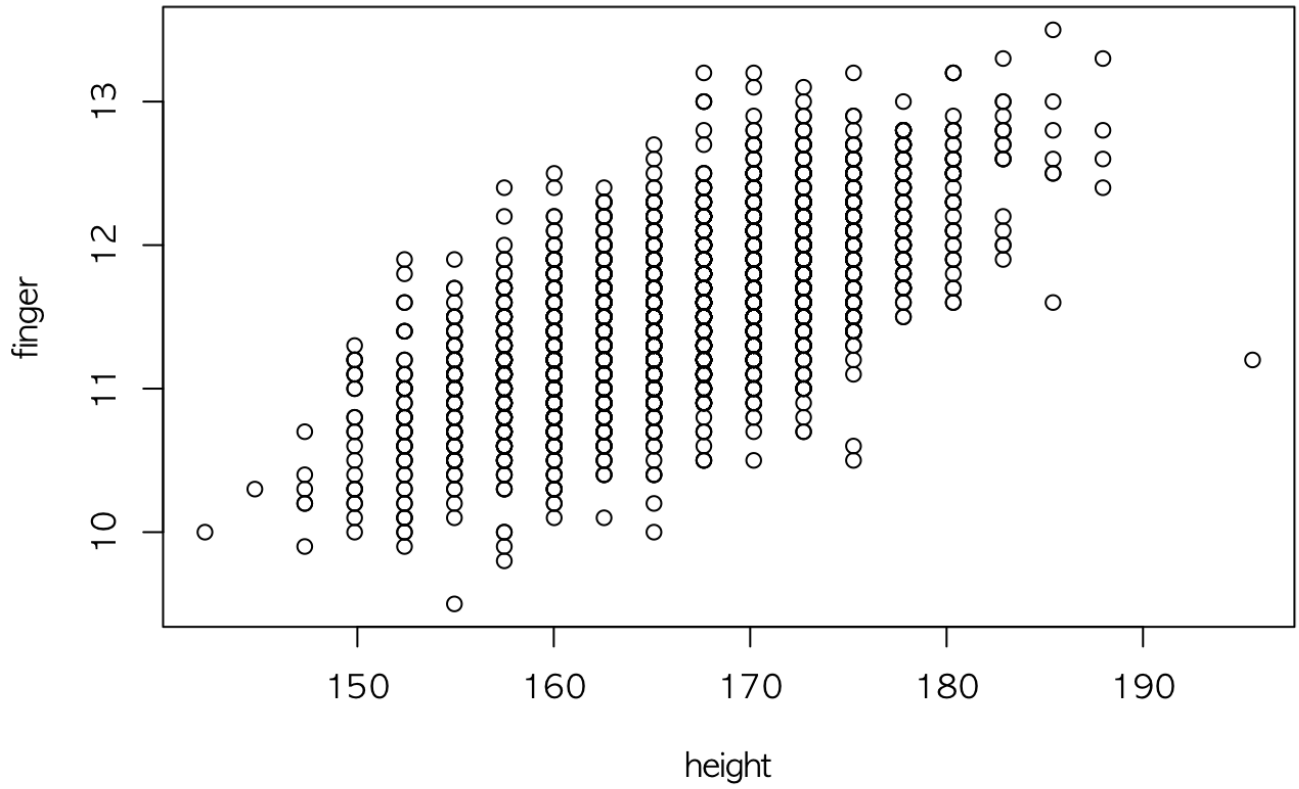
```
crimtab.df<-crimtab.2.df[,3:5]
str(crimtab.df)
```

```
## 'data.frame':    924 obs. of  3 variables:
##  $ 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  142 142 142 142 142 ...
```

```
crimtab.long<-apply(crimtab.df[,2:3],2, function(x)rep(x, crimtab.df[,1]))
str(crimtab.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"
```

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

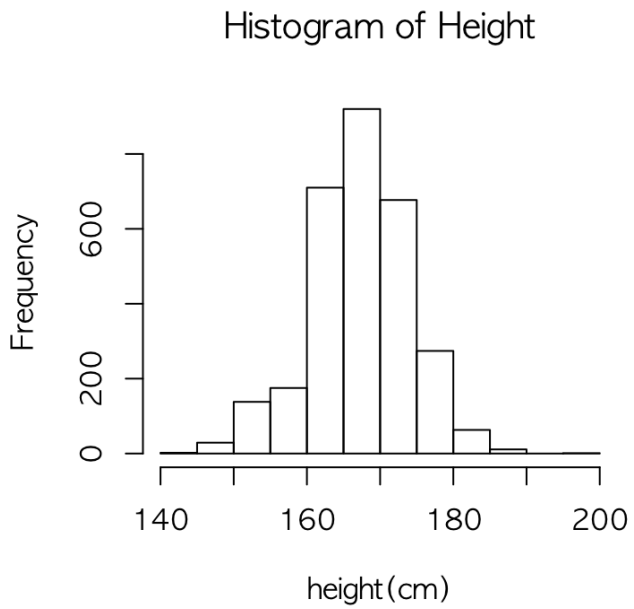


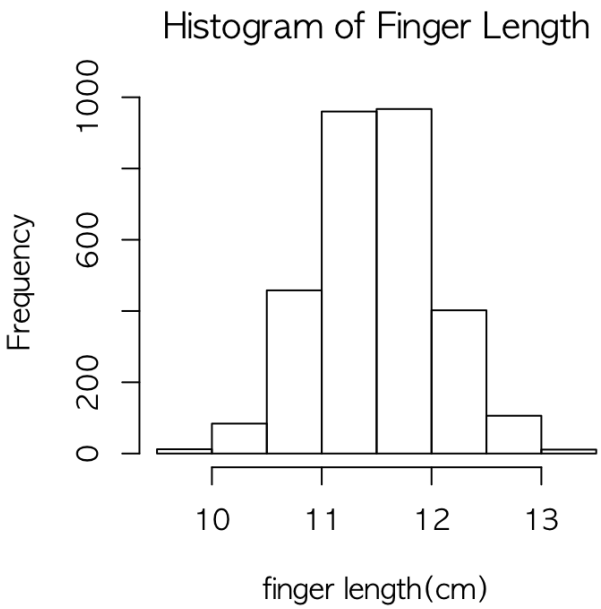

```
str(crimtab.long)
```

```
##  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"
```

- 변수 각각의 히스토그램은?

```
par(mfrow=c(1,2))
hist(crimtab.long[,2], main="Histogram of Height", xlab="height(cm)")
hist(crimtab.long[,1], main="Histogram of Finger Length", xlab= "finger lengt
h(cm)")
```





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

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

```
##    finger    height
##  11.54737 166.30142
```

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

```
##    finger    height
## 0.5487137 6.4967015
```

- Quetelet의 가슴둘레 자료에서 살핀 바와 같이 이 자료를 그대로 ad.test 등에 적용하면 매우 작은 p-value 가 예상됨.

```
library(nortest)
ad.test(crimtab.long[,1])
```

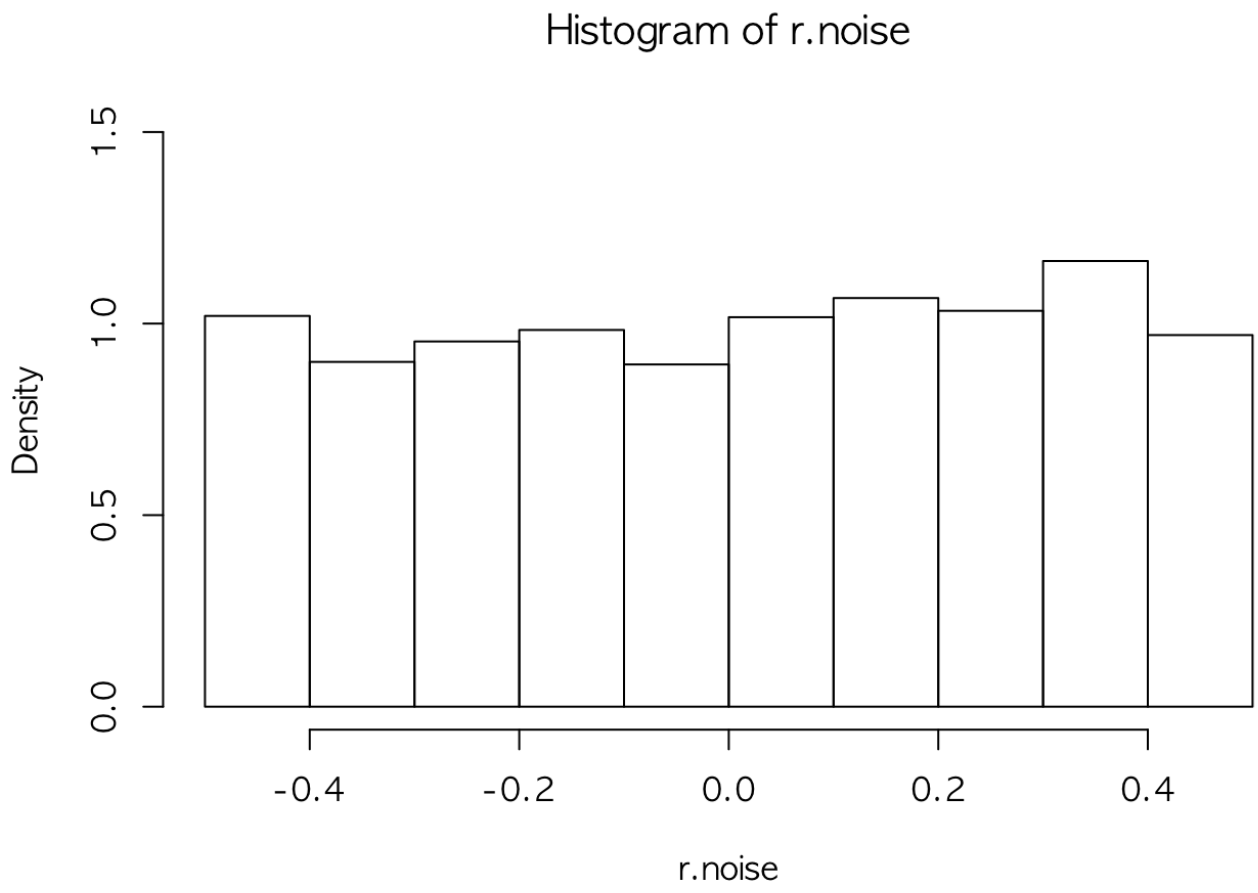
```
##
##  Anderson-Darling normality test
##
## data:  crimtab.long[, 1]
## A = 4.7094, p-value = 1.153e-11
```

```
ad.test(crimtab.long[,2])
```

```
##
##  Anderson-Darling normality test
##
## data:  crimtab.long[, 2]
## A = 18.8368, p-value < 2.2e-16
```

- height의 경우 인치 단위로 측정한 자료를 센티 단위로 변환한 것임. 이 점에 유의하여 원 자료의 모습에 가깝게 noise 를 넣어 적용하면

```
r.noise<-runif(3000)-0.5
hist(r.noise, prob=T, xlim=c(-0.5,0.5), ylim=c(0,1.5))
```




```
ad.test(crimtab.long[,1]+r.noise*2.54)
```

```
##
##      Anderson-Darling normality test
##
## data:  crimtab.long[, 1] + r.noise * 2.54
## A = 0.2784, p-value = 0.6497
```

```
cvm.test(crimtab.long[,1]+r.noise*2.54)
```

```
##
##      Cramer-von Mises normality test
##
## data:  crimtab.long[, 1] + r.noise * 2.54
## W = 0.0368, p-value = 0.7409
```

```
lillie.test(crimtab.long[,1]+r.noise*2.54)
```

```
##
##      Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  crimtab.long[, 1] + r.noise * 2.54
## D = 0.0117, p-value = 0.4046
```

```
ad.test(crimtab.long[,2]+r.noise/10)
```

```
##
##      Anderson-Darling normality test
##
## data:  crimtab.long[, 2] + r.noise/10
## A = 0.6739, p-value = 0.07839
```

```
##
##      Cramer-von Mises normality test
##
## data:  crimtab.long[, 2] + r.noise/10
## W = 0.1029, p-value = 0.1026
```

```
lillie.test(crimtab.long[,2]+r.noise/10)
```

```
##
##      Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  crimtab.long[, 2] + r.noise/10
## D = 0.0143, p-value = 0.1461
```