Chapter 7 Review

2017100057 / 이영노

October 15, 2022

Chapter 7

Chapter7 Intro: how to visualize **DISTRIBUTIIONS**.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.7      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dslabs)
data("heights")
data("murders")
head(heights)
```

```
##      sex height
## 1   Male     75
## 2   Male     70
## 3   Male     68
## 4   Male     74
## 5   Male     61
## 6 Female     65
```

```
str(heights)
```

```
## 'data.frame':    1050 obs. of  2 variables:
##  $ sex   : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 1 1 1 1 2 ...
##  $ height: num  75 70 68 74 61 65 66 62 66 67 ...
```

```
str(murders)
```

```
## 'data.frame':    51 obs. of  5 variables:
##  $ state     : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  $ abb       : chr  "AL" "AK" "AZ" "AR" ...
##  $ region    : Factor w/ 4 levels "Northeast","South",..: 2 4 4 2 4 4 1 2 2 2 ...
```

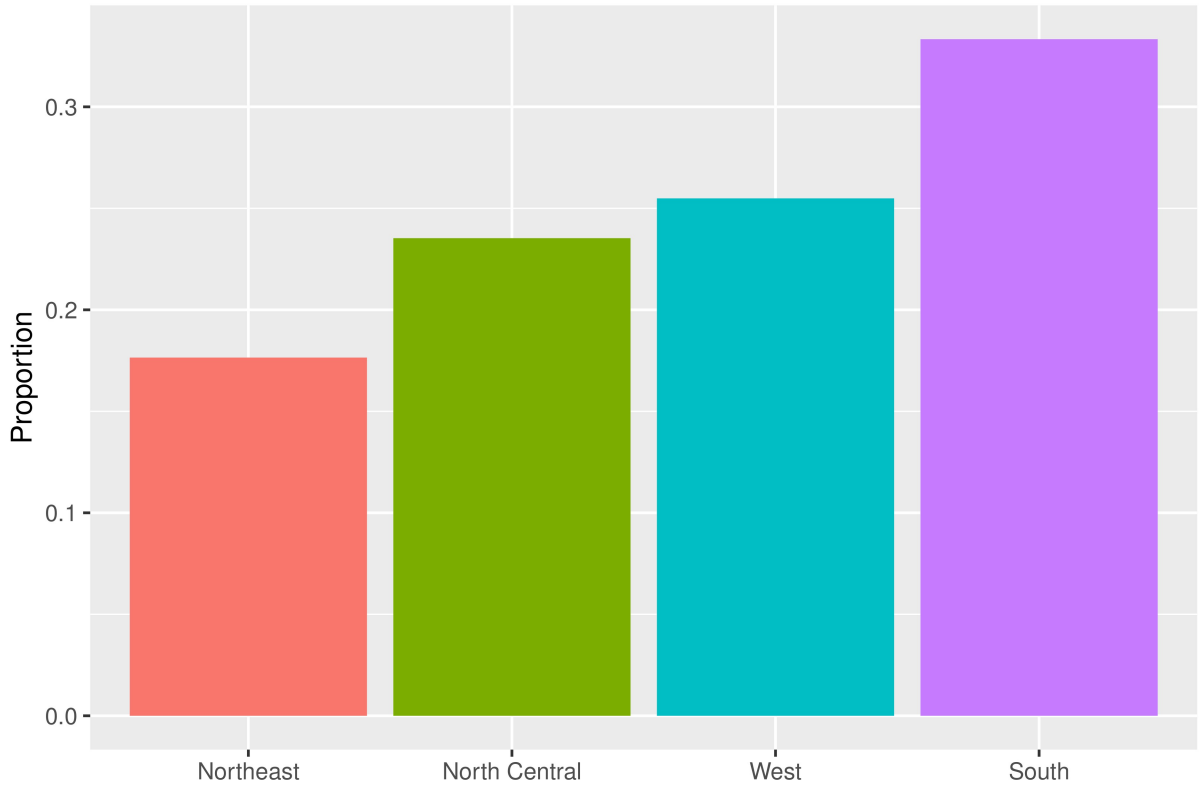
```
##  $ population: num  4779736 710231 6392017 2915918 37253956 ...
##  $ total     : num  135 19 232 93 1257 ...
```

```
library(gridExtra)
```

```
##
## 다음의 패키지를 부착합니다: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

- example 1: Distribution plot using
- `fill()`, `geom_bar` and `group_by()`, `summarize()`

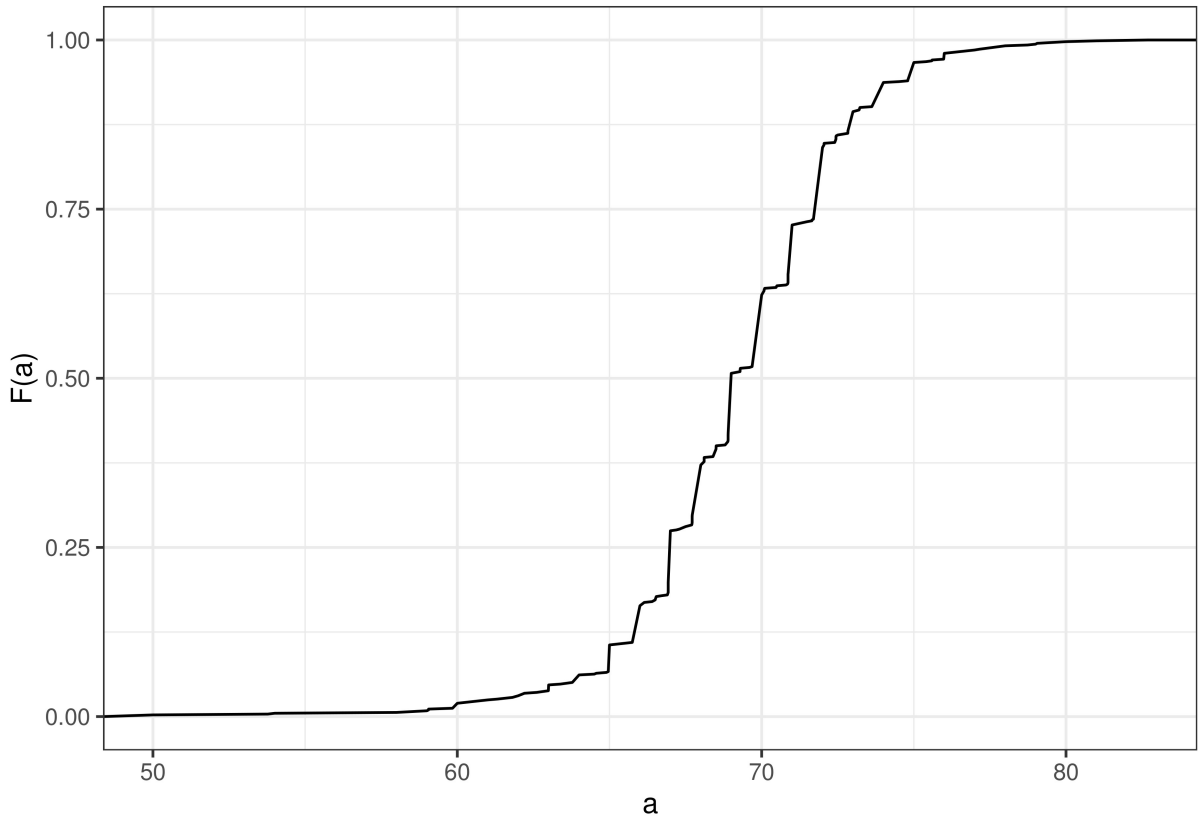
```
murders %>% group_by(region) %>% summarize(n = n()) %>% mutate(Proportion = n/sum(n),
                                                                region = reorder(region,Proporti
  ggplot(aes(x=region, y=Proportion, fill=region)) +
  geom_bar(stat = "identity",show.legend = FALSE) +
  xlab("")
```




```
# `reorder`: region을 proportions의 오름차순(작은게 먼저)로 정렬
```

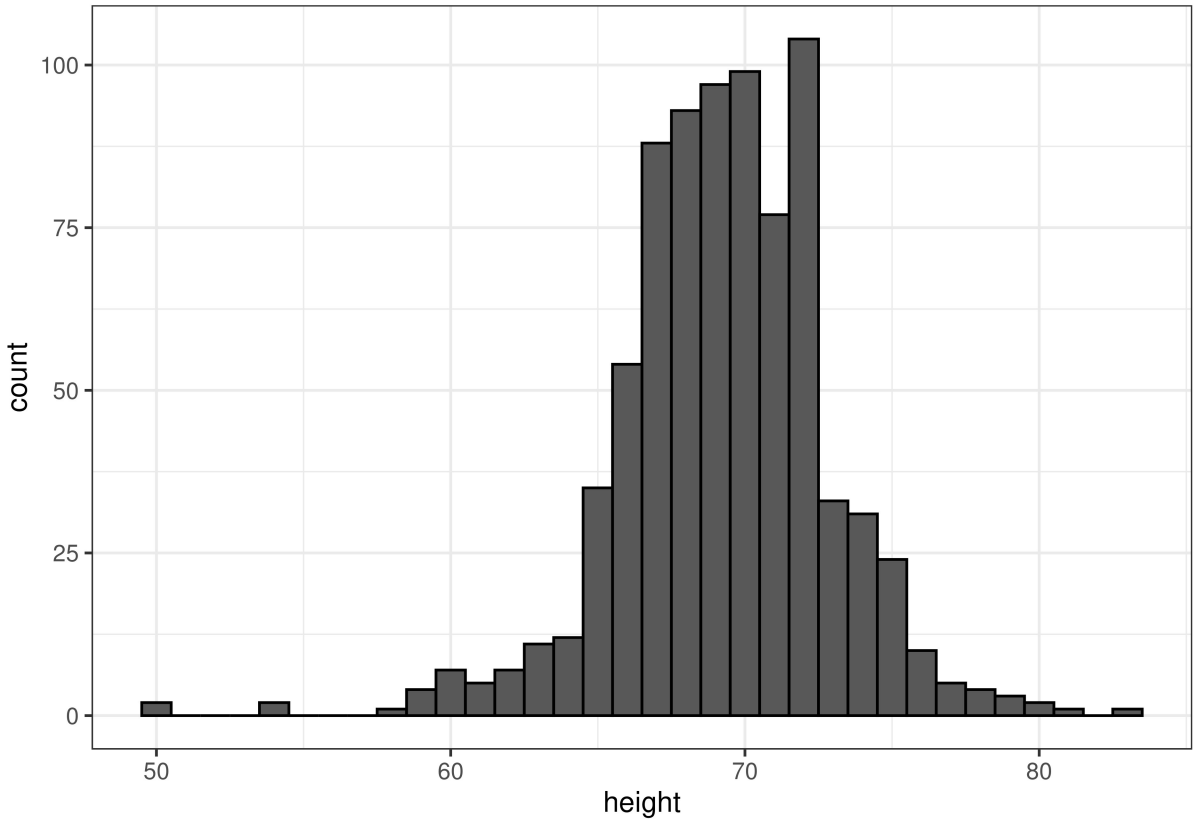
- example 2: CDF

```
ds_theme_set()
heights %>% filter(sex=="Male") %>% ggplot(aes(height)) + stat_ecdf(geom = "line") +
  ylab("F(a)") + xlab("a")
```



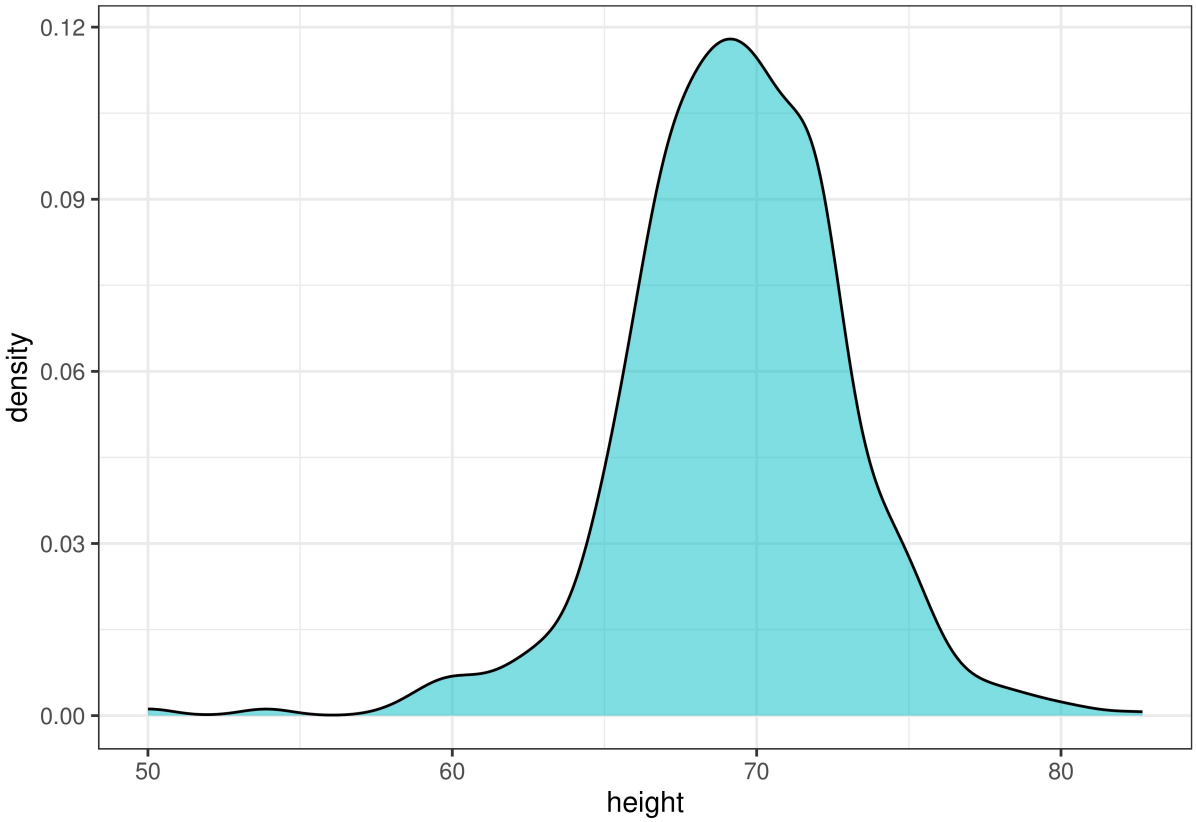
* example 3-1: Histogram (information loss)

```
heights %>% filter(sex=="Male") %>% ggplot(aes(height))+geom_histogram(binwidth=1,color='black
```



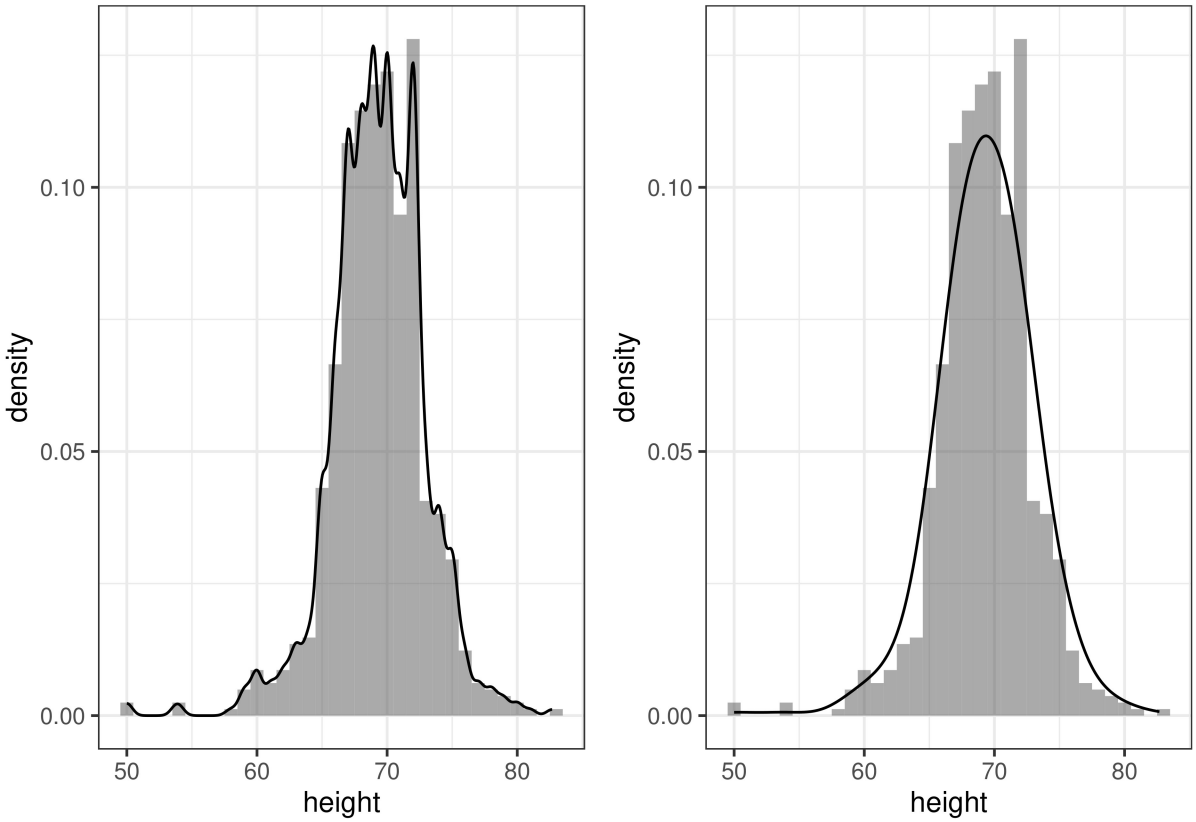
* example 3-2: smooth Histogram

```
heights %>% filter(sex=="Male") %>% ggplot(aes(height))+
  geom_density(alpha=0.5,fill="#00BFC5", color=0) + geom_line(stat='density')
```



* example 3-3: Smoothing controlled

```
p <- heights %>% filter(sex=="Male") %>% ggplot(aes(height))
par(mfrow=c(1,2))
p1=p+geom_histogram(aes(y=..density..),binwidth=1,alpha=0.5)+
  geom_line(stat='density',adjust=0.5)
p2=p+geom_histogram(aes(y=..density..),binwidth=1,alpha=0.5)+
  geom_line(stat='density',adjust=2)
grid.arrange(p1,p2,ncol=2)
```




```
#..density..는 y축을 density로 맞추기 위해 들어감.
#..은 겹쳐서 그릴수 있게 함.
```

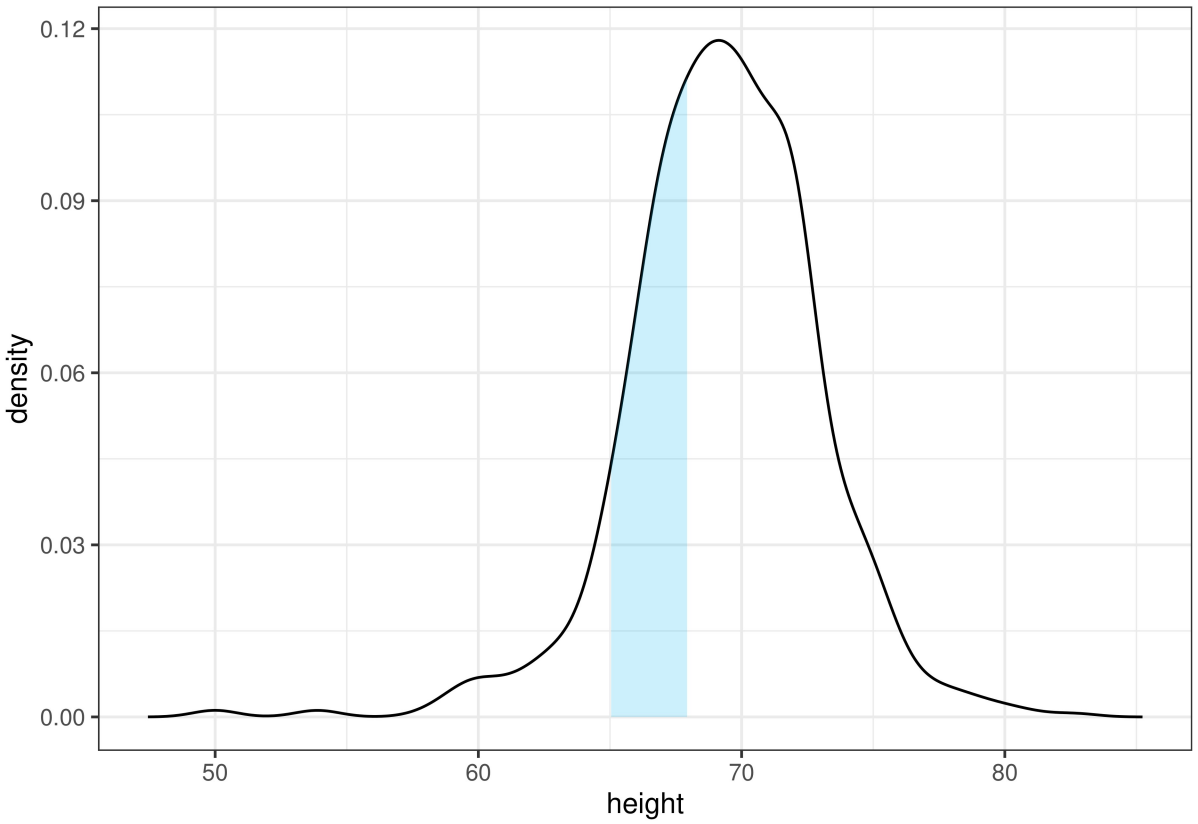
- example 4: interpreting interval area

```
d <- with(heights, density(height[sex=="Male"]))
d
```

```
##
## Call:
##  density.default(x = height[sex == "Male"])
##
## Data: height[sex == "Male"] (812 obs.);  Bandwidth 'bw' = 0.8511
##
##        x                y
##  Min.   :47.45   Min.   :6.530e-06
##  1st Qu.:56.89   1st Qu.:7.249e-04
##  Median :66.34   Median :5.168e-03
##  Mean   :66.34   Mean   :2.644e-02
##  3rd Qu.:75.78   3rd Qu.:3.680e-02
##  Max.   :85.23   Max.   :1.180e-01
```

```
# with 사용하면 매번 코드로 칼럼 적지않고도 각 칼럼에 곧바로 접근할 수 있음
# density: 반환값은 커널밀도에 대한 데이터. 이값을 plot에 주면 밀도그림을 그릴수 있음
```

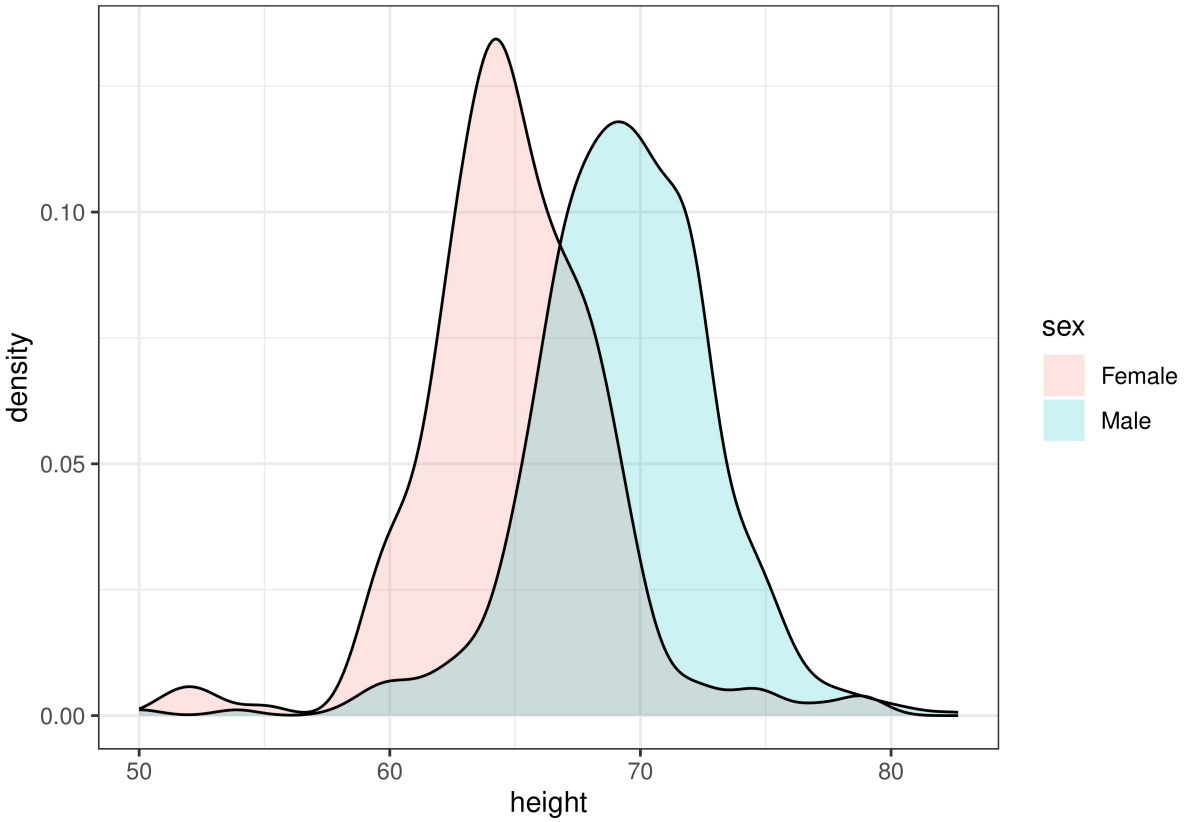
```
tmp <- data.frame(height=d$x, density=d$y)
tmp %>% ggplot(aes(height,density)) + geom_line() +
  geom_area(aes(x=height,y=density),
            data=filter(tmp,between(height,65,68)),
            alpha=.2, fill="#00BCF4")
```




```
# geom_area에 새로운 데이터를 넣었다는 점에 주목
```

density의 장점은 distribution의 comparison이 가능하기 때문이다.

```
heights%>%
  ggplot(aes(height,fill=sex))+
  geom_density(alpha=0.2,color=0)+
  geom_line(stat='density')
```



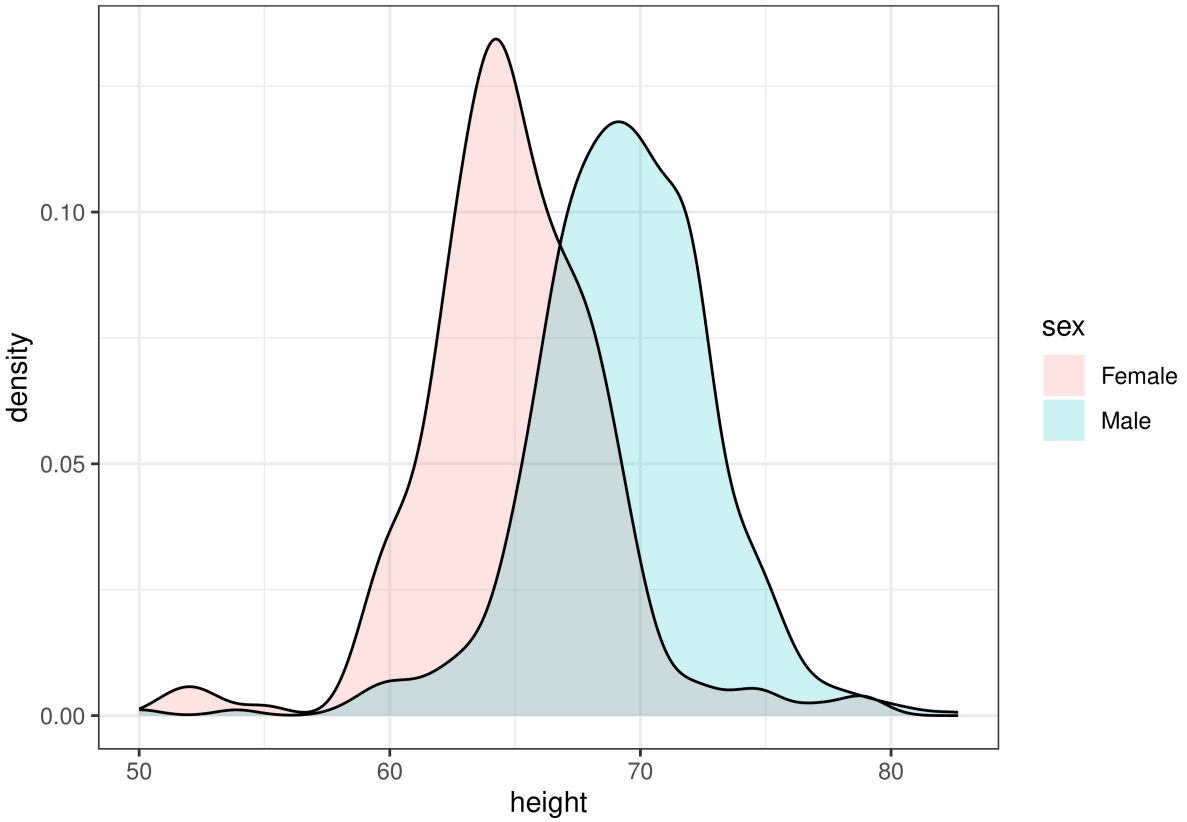

```
heights %>% head()
```

```
##      sex height
## 1   Male     75
## 2   Male     70
## 3   Male     68
## 4   Male     74
## 5   Male     61
## 6 Female     65
```

- exmaple 5: density에서 fill함수로 factor별 density구할때는, fill이 mapping이다!(aes)

차후에 Stratification으로 revisit할 예정

```
heights%>%ggplot(aes(height,fill=sex))+
  geom_density(alpha=.2,color=0)+
  geom_line(stat='density')
```




```
# fill=sex는 sex에 따라 density를 mapping하는 것이므로, aes안에 넣어줘야한다.(중요!)
```

- example 6: Normal Distribution and Quantiles

```
index <- heights$sex =="Male"
x <- heights$height[index] # TRUE/FALSE indexing
m = sum(x)/length(x)
s = sqrt(sum((x-m)^2)/length(x))
c(average=m,sd=s) %>% round(2)
```

```
## average      sd
##   69.31    3.61
```

```
z<-scale(x) # 표준화
mean(abs(z)<2) # Quantile을 비교
```

```
## [1] 0.9495074
```

```
pnorm(-1.96) # -1.96까지의 확률
```

```
## [1] 0.0249979
```

```
qnorm(0.975) # 0.975 확률값을 가지는 z값
```

```
## [1] 1.959964
```

```r
qnorm(0.975, mean = 5, sd = 8)
```

```
## [1] 20.67971
```

```r
mean(x<=69)
```

```
## [1] 0.5073892
```

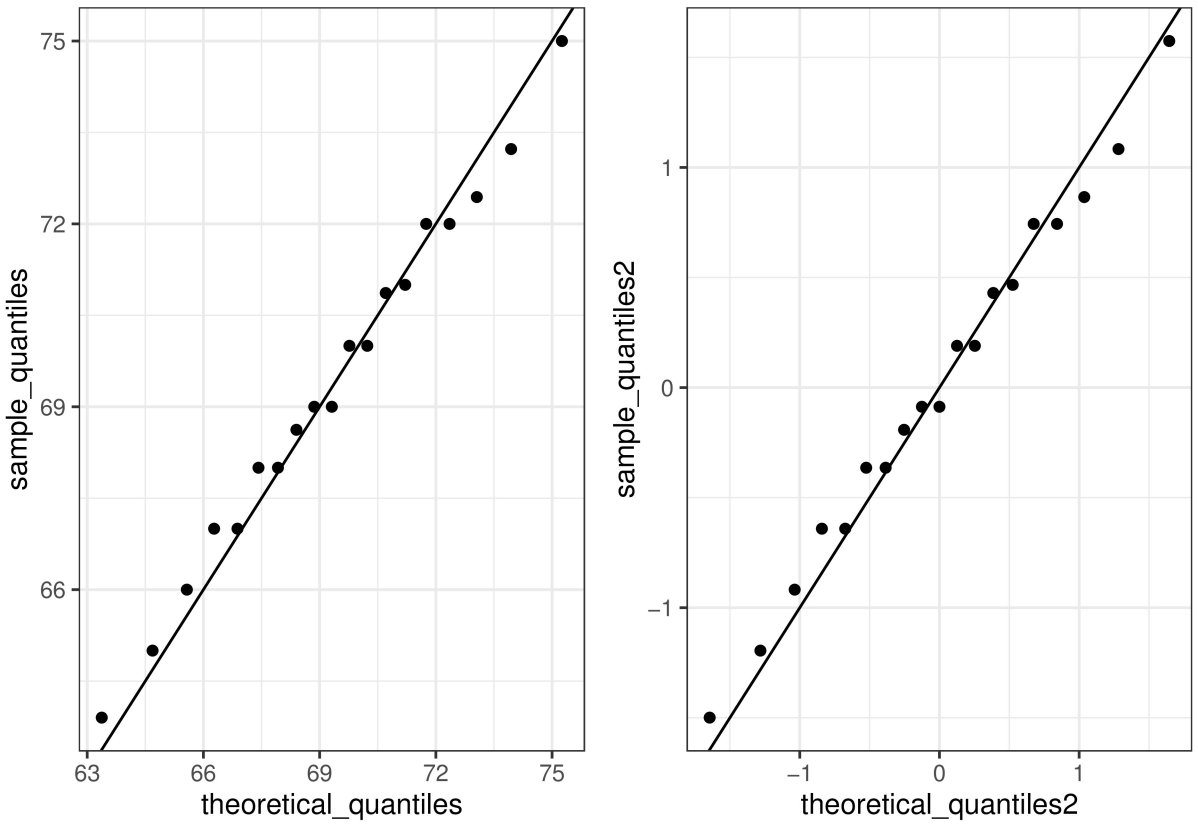
```r
# concept of QQ_plot
p <- seq(0.05,0.95,0.05)
sample_quantiles <- quantile(x,p)
sample_quantiles %>% head()
```

```
##       5%      10%      15%      20%      25%      30%
## 63.90079 65.00000 66.00000 67.00000 67.00000 68.00000
```

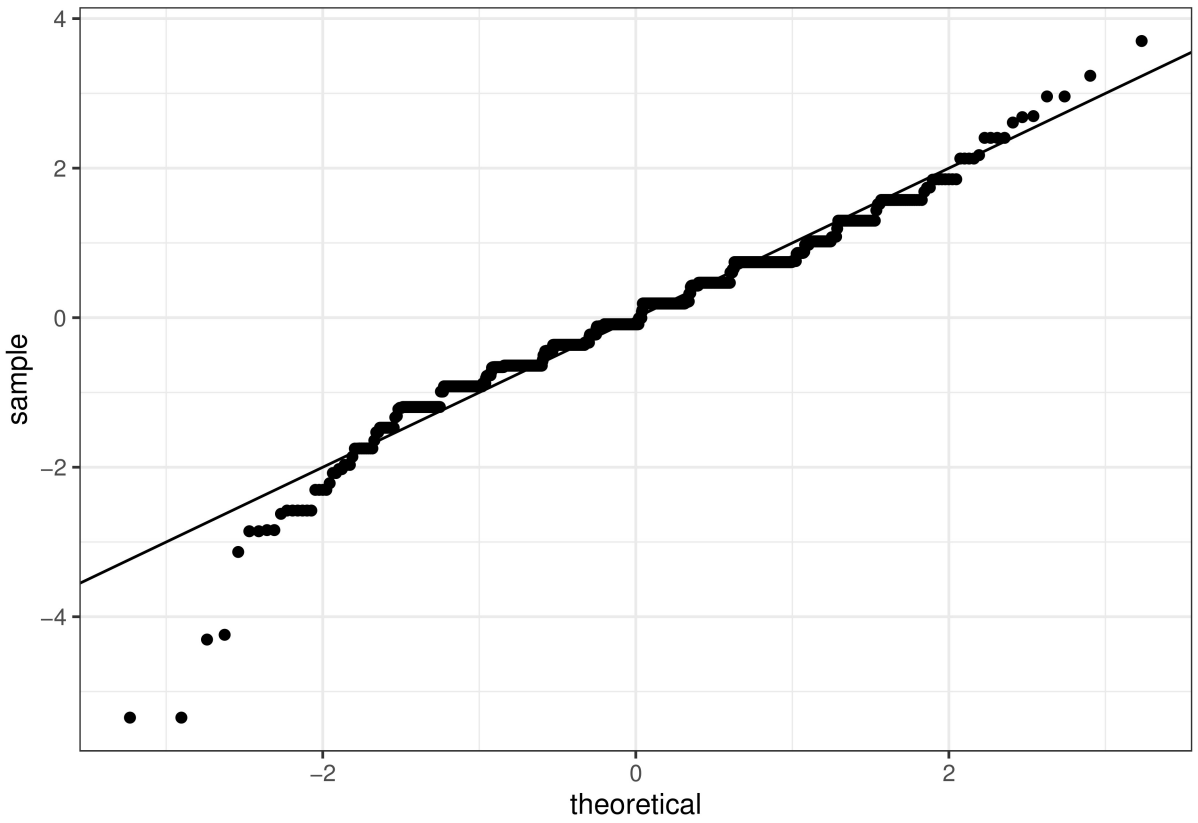
```r
theoretical_quantiles <-qnorm(p,mean=mean(x),sd=sd(x))
theoretical_quantiles %>% head() # if quantile follows normal
```

```
## [1] 63.37515 64.68704 65.57217 66.27564 66.87916 67.42113
```

```r
qq1=qplot(theoretical_quantiles,sample_quantiles)+geom_abline()
### same as above
sample_quantiles2 <- quantile(z,p)
theoretical_quantiles2 <- qnorm(p)
qq2=qplot(theoretical_quantiles2, sample_quantiles2)+geom_abline()
grid.arrange(qq1,qq2,ncol=2)
```




```
### which is same as above
heights %>% filter(sex=="Male") %>%
  ggplot(aes(sample=scale(height)))+
  geom_qq()+
  geom_abline()
```

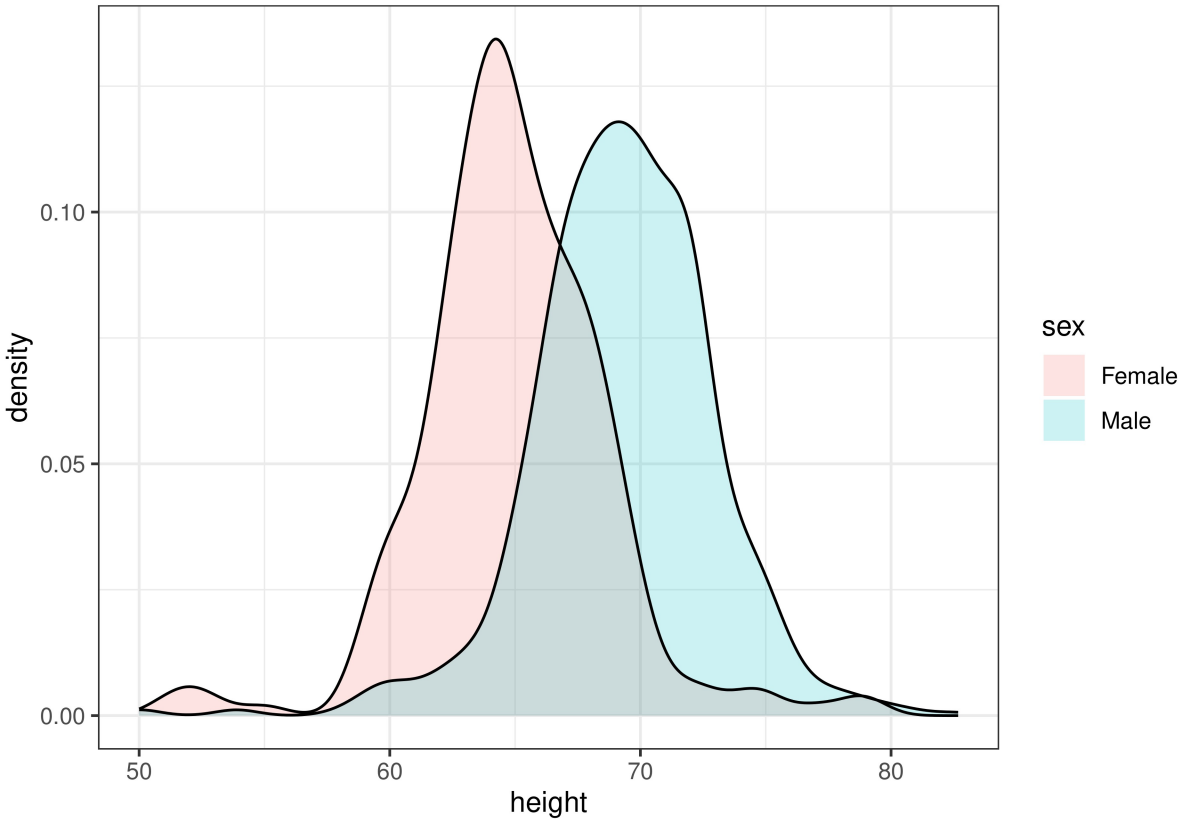



```
# stats qq requires sample as argument name
```

- example 7: Stratification

example 5 REVISETED

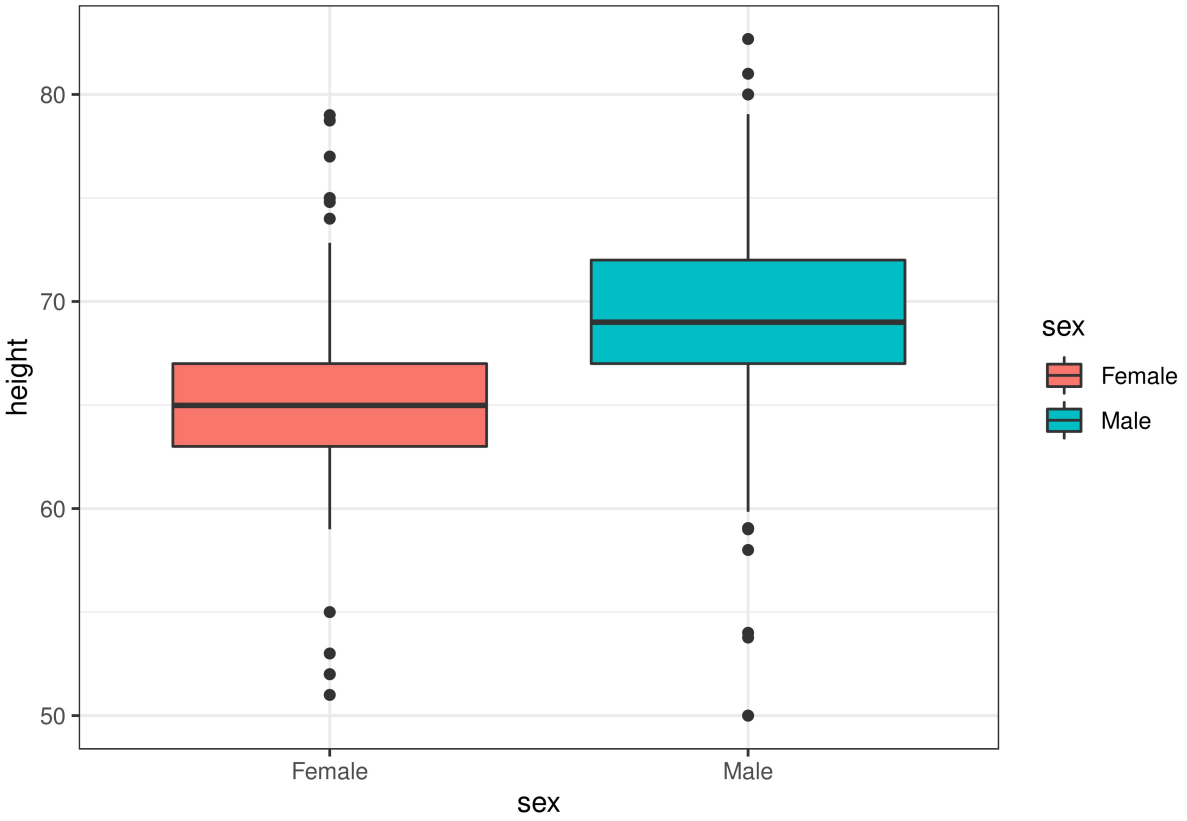
```
heights%>%ggplot(aes(height,fill=sex))+
  geom_density(alpha=.2,color=0)+
  geom_line(stat='density')
```



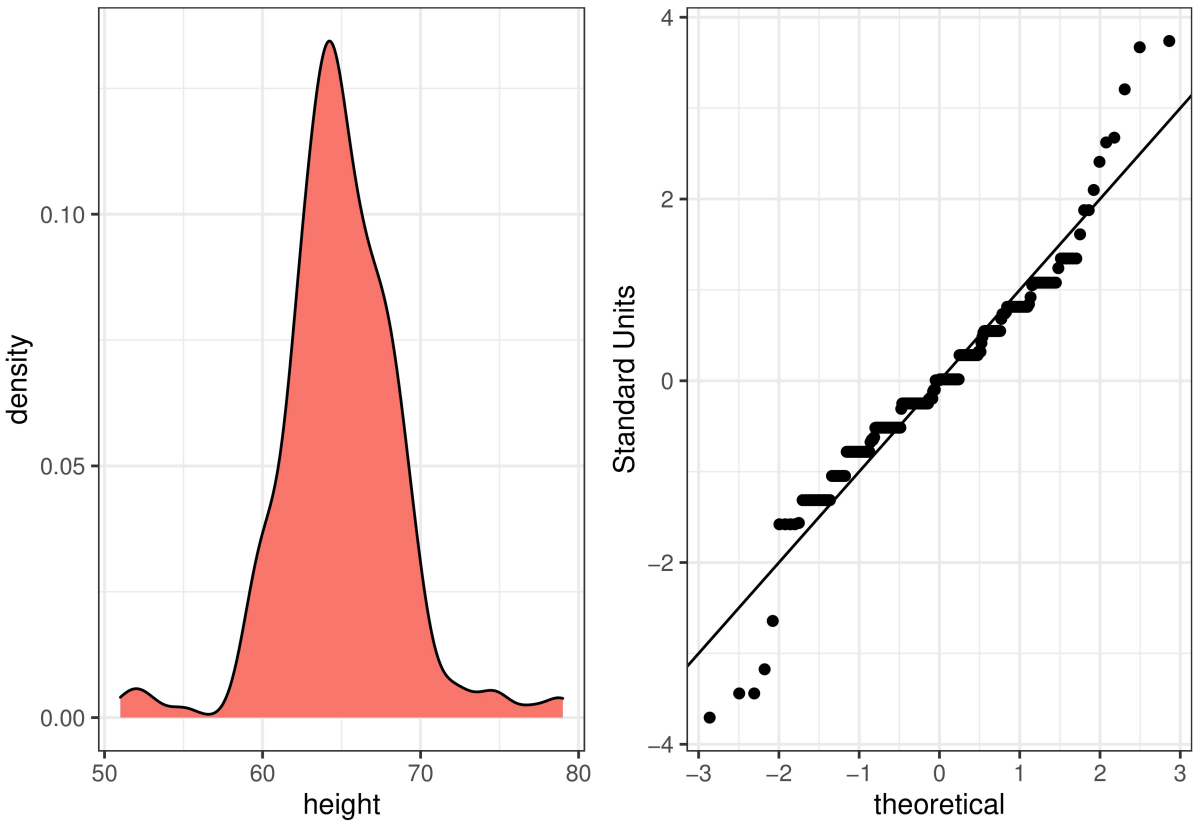

```
# fill=sex는 sex에 따라 density를 mapping하는 것이므로, aes안에 넣어줘야한다.(중요!)
```

factor에 따라 mapping하는 것이므로, mapping(x,y축 설정)함수인 aes안에 fill=sex를 넣어줘야한다

```
heights %>% ggplot(aes(x=sex,y=height,fill=sex))+
    geom_boxplot()
```

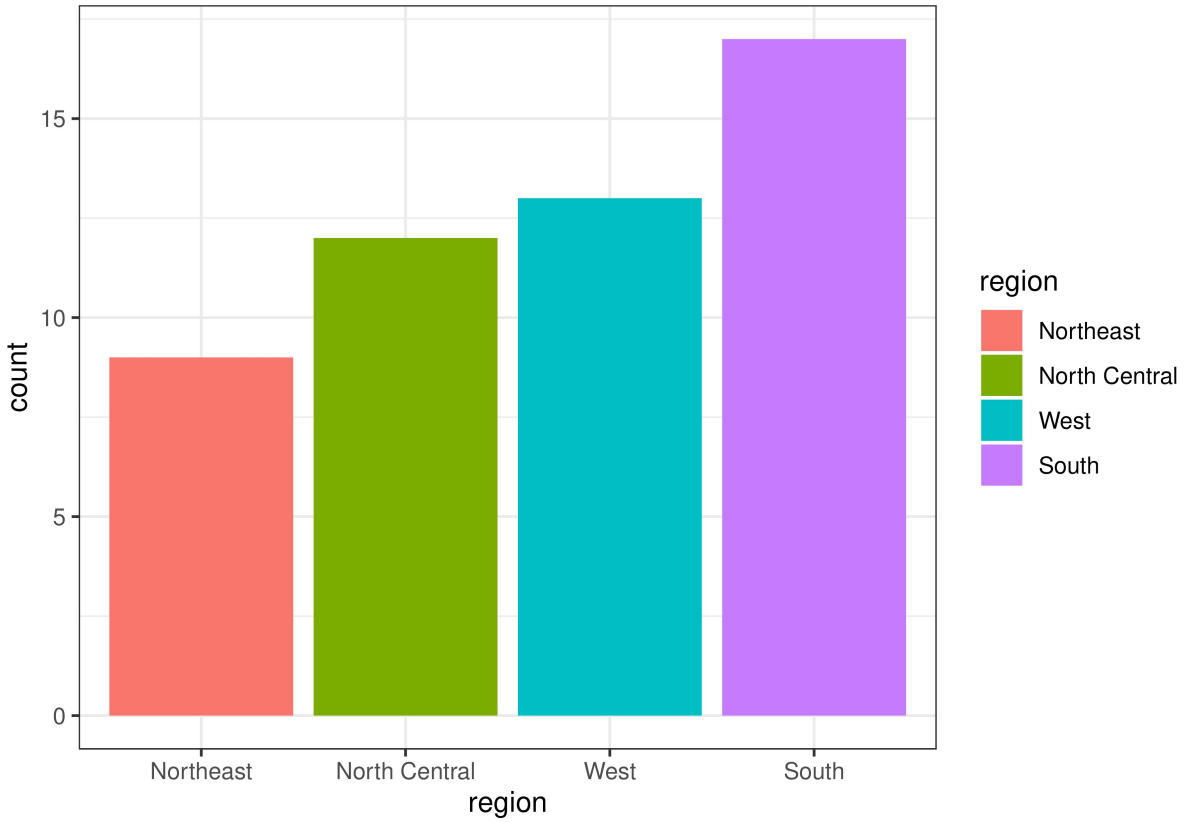



```
#####
p1 <- heights %>% filter(sex=="Female")%>%ggplot(aes(height))+geom_density(fill="#F8766D")
p2 <- heights %>% filter(sex=="Female")%>%ggplot(aes(sample=scale(height)))+
  geom_qq() + geom_abline() + ylab("Standard Units")
grid.arrange(p1,p2,ncol=2)
```



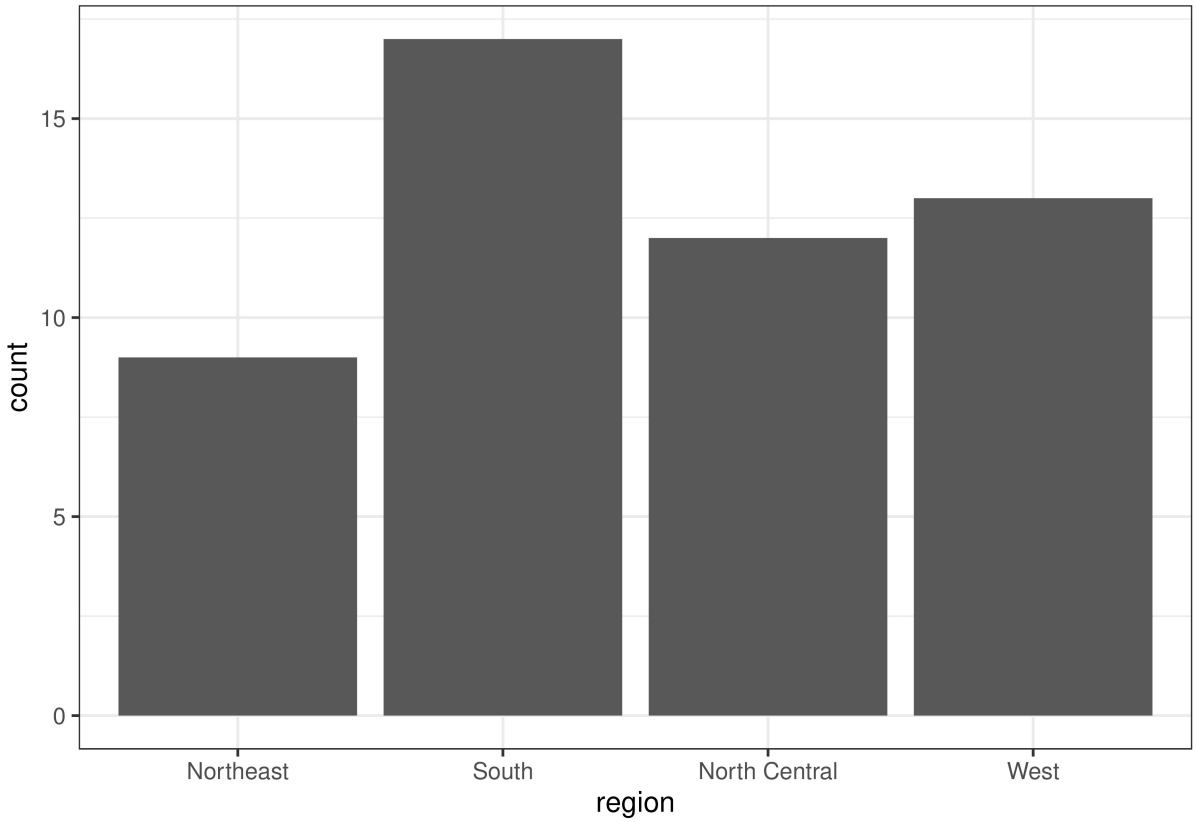
- example 8: barplot

```
# (중요!) region으로 reorder하라!
murders %>% group_by(region) %>% summarize(n=n()) %>%
  mutate(count=n,region=reorder(region,count))%>%
  ggplot(aes(x=region,y=count,fill=region))+geom_bar(stat="identity")
```




```
# stat="identity" : 데이터를 기반으로 barplot그리겠다. 꼭 넣어줘야함
```

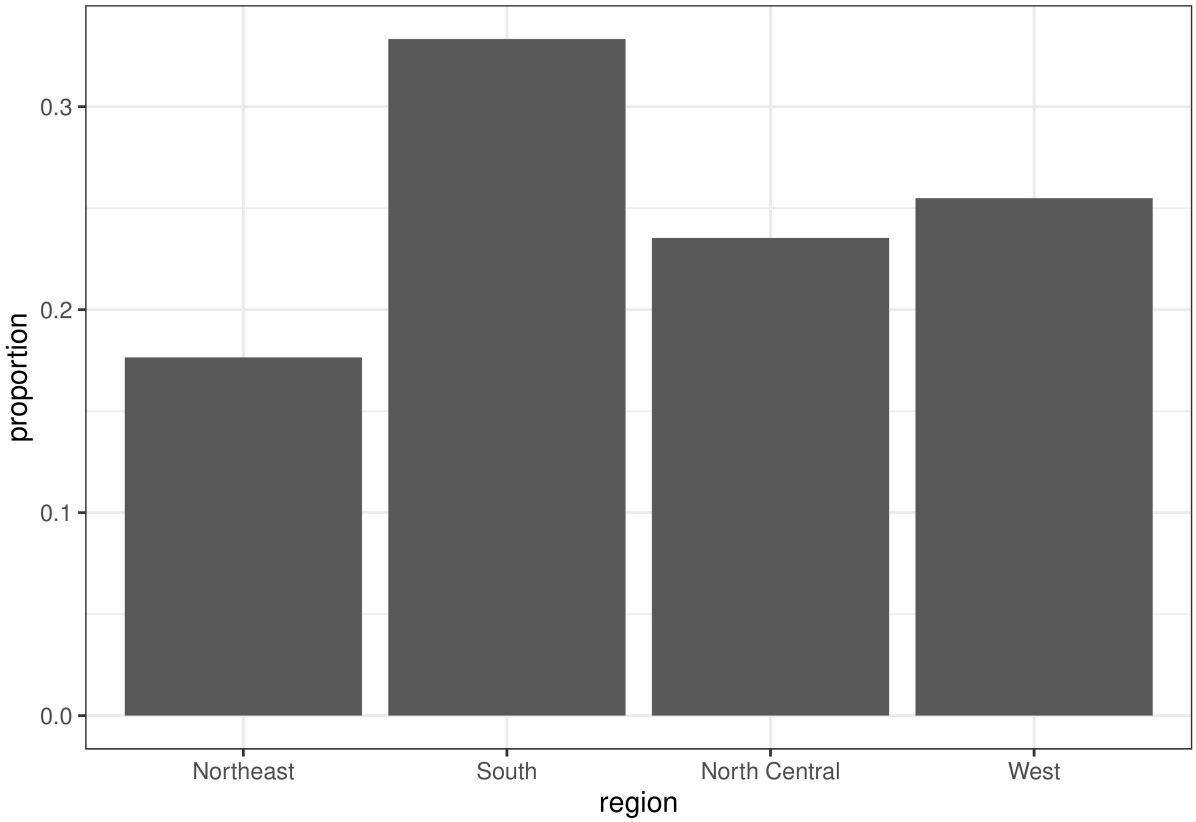
```
murders %>% ggplot(aes(region)) + geom_bar()
```




```
count(murders,region)
```

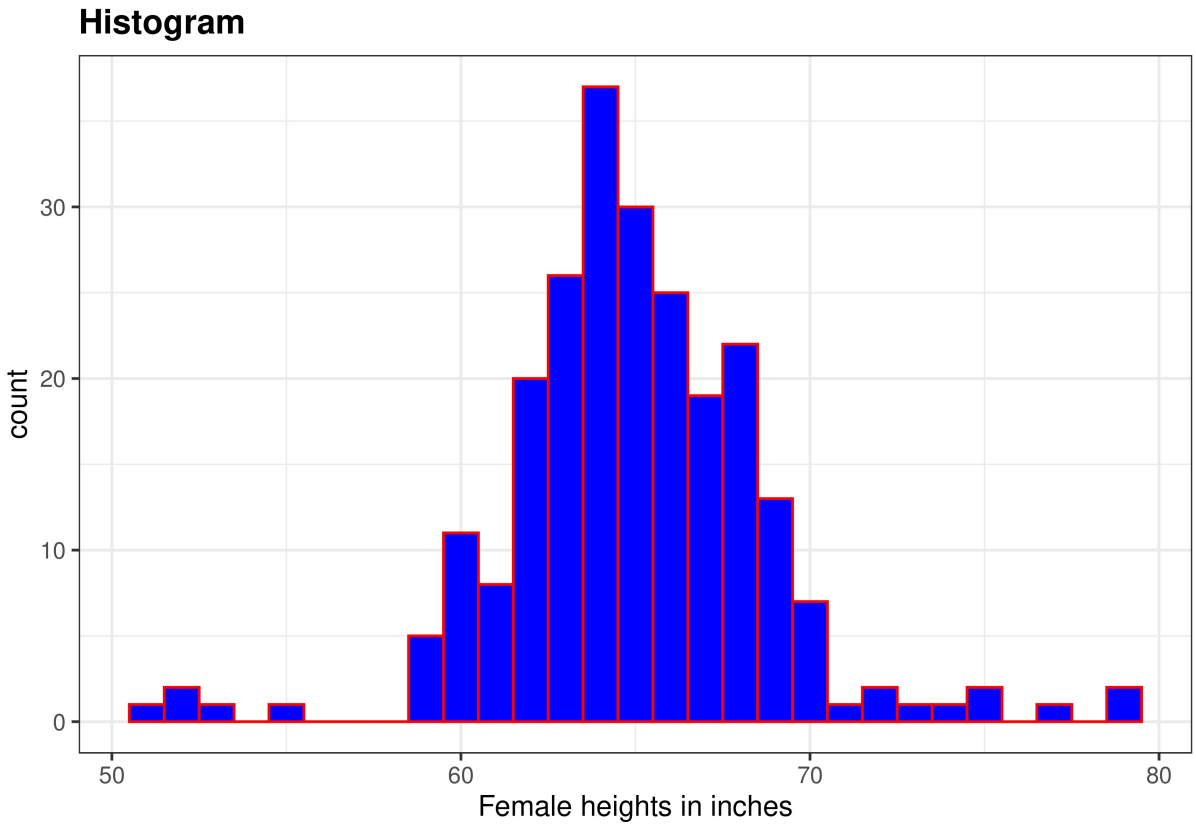
```
##          region  n
## 1     Northeast  9
## 2         South 17
## 3 North Central 12
## 4          West 13
```

```
tab <- murders %>% count(region) %>% mutate(proportion=n/sum(n))
tab %>% ggplot(aes(region,proportion))+geom_bar(stat="identity")
```



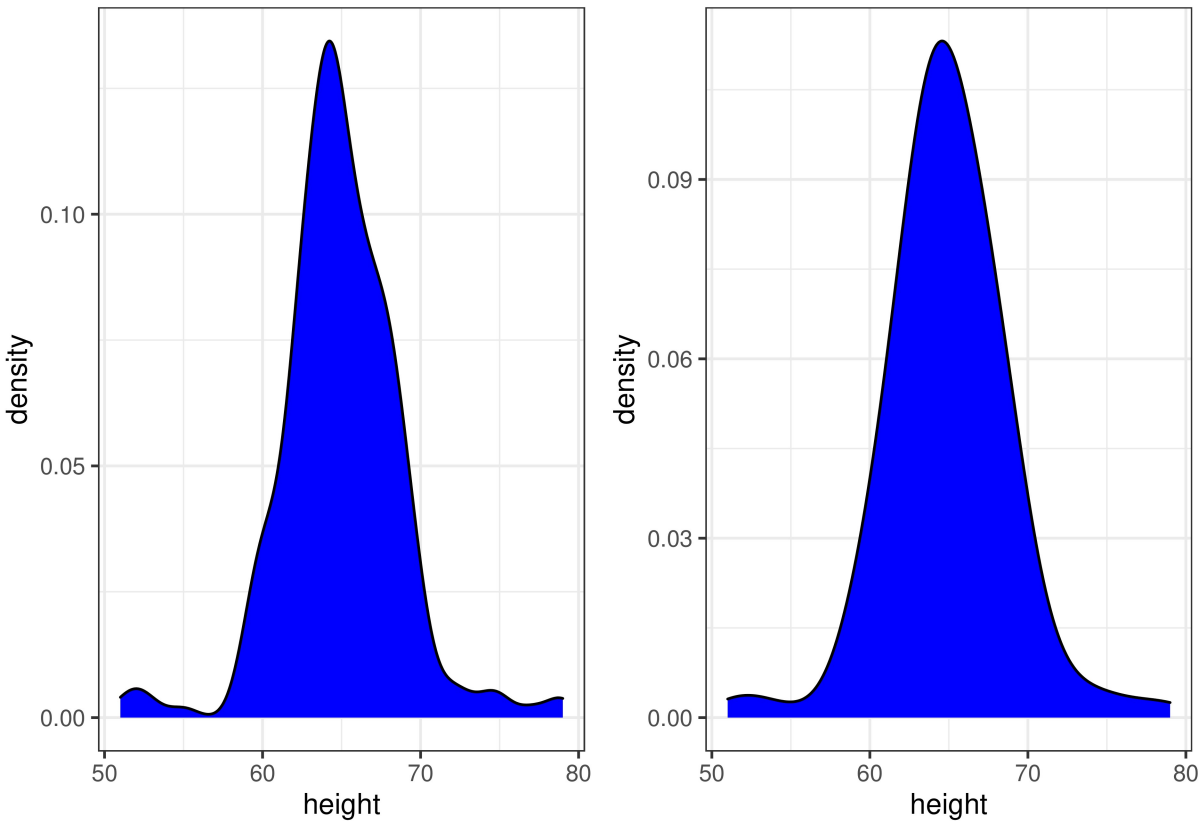
* example 9: Histogram

```
heights %>%
filter(sex == "Female") %>%
ggplot(aes(height)) +
geom_histogram(binwidth=1,fill="blue",col="red")+
  xlab("Female heights in inches")+
  ggtitle("Histogram")
```



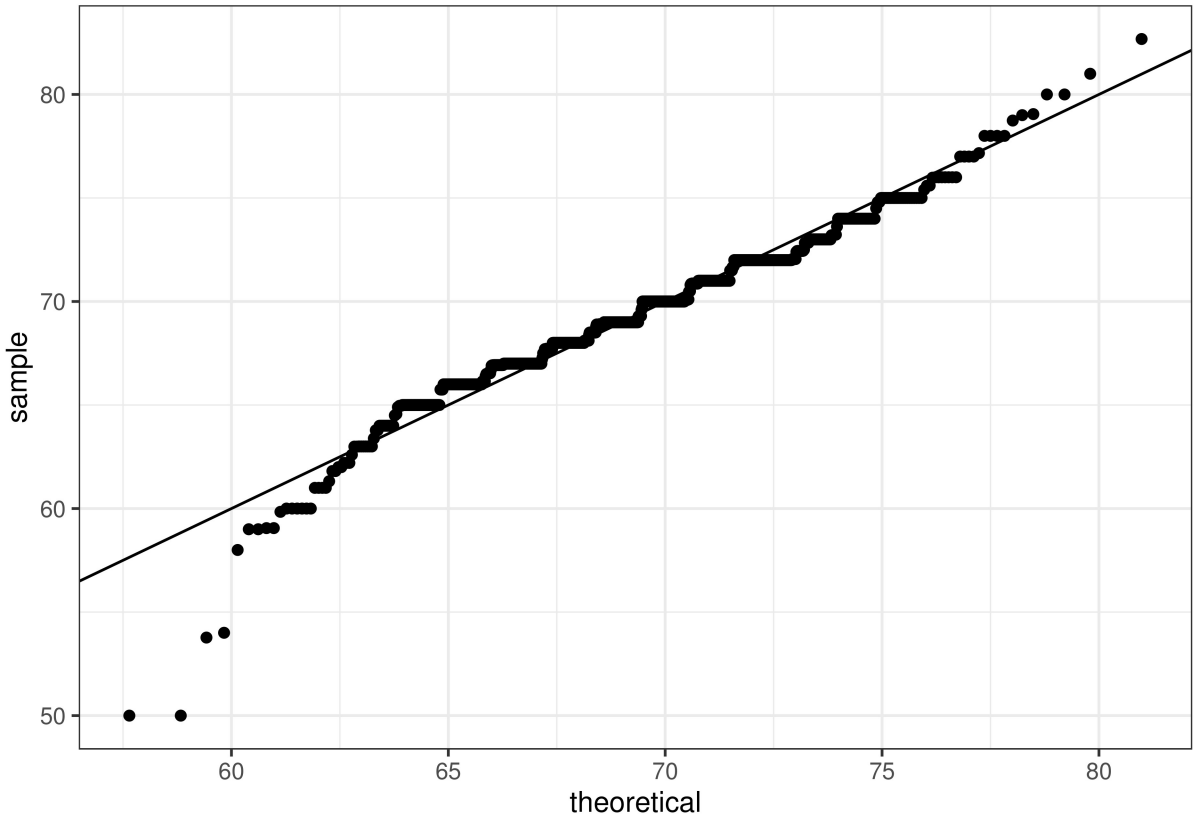
- example 10: Density

```
g1=heights %>%
filter(sex == "Female") %>%
ggplot(aes(height)) +
geom_density(fill="blue")

g2=heights %>%
filter(sex == "Female") %>%
ggplot(aes(height)) +
geom_density(fill="blue", adjust = 2)

grid.arrange(g1,g2,ncol=2)
```



* example 11: QQ plot `dparams`

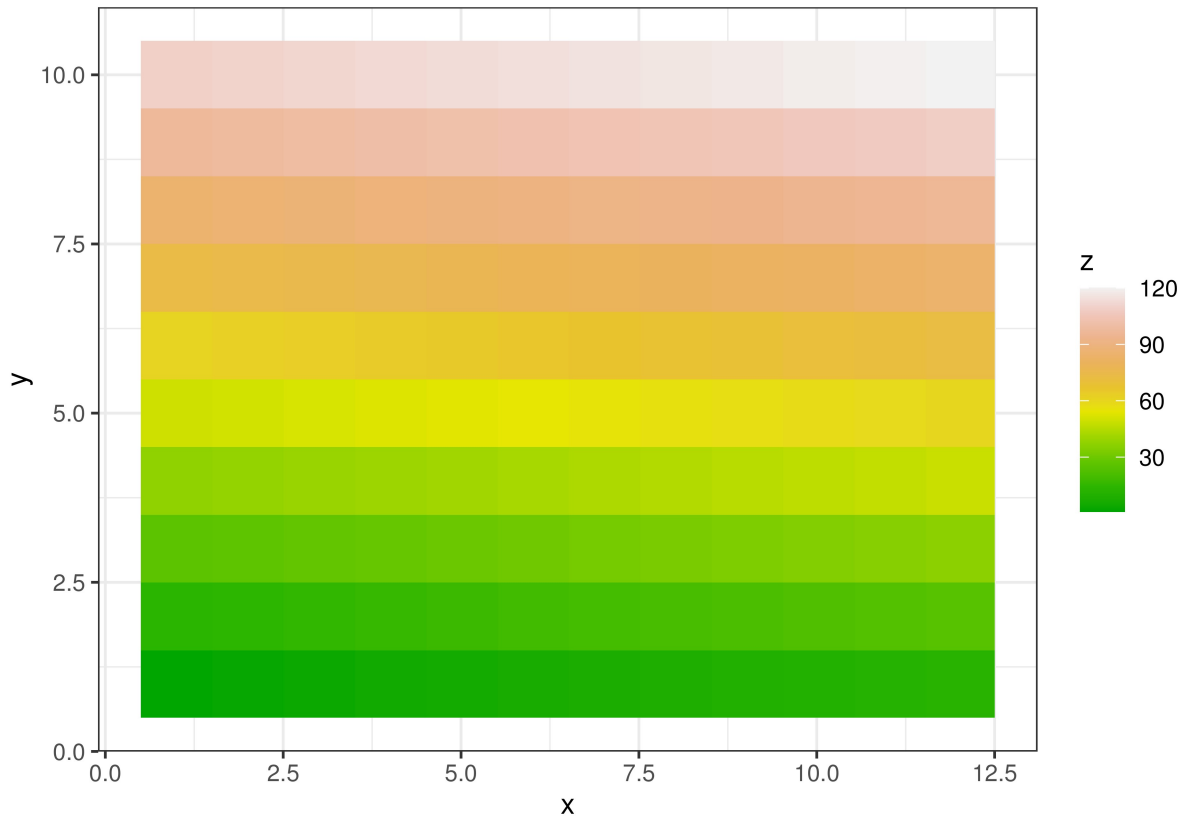
```
params <- heights %>% filter(sex=="Male") %>%
summarize(mean = mean(height), sd = sd(height))
# group_by 대신에 filter로 했음

heights %>% filter(sex=="Male") %>%
ggplot(aes(sample = height)) +
geom_qq(dparams = params) +
geom_abline()
```




```
# 밑에보면 dparams으로 인해서 기준parameter가 바뀜
```

- example 12: Creating IMAGES

```
x <- expand.grid(x=1:12,y=1:10) %>% mutate(z=1:120) # 12x10 data.frame 구조

x %>% ggplot(aes(x,y,fill=z)) + geom_raster() +
  scale_fill_gradientn(colors=terrain.colors(10))
```

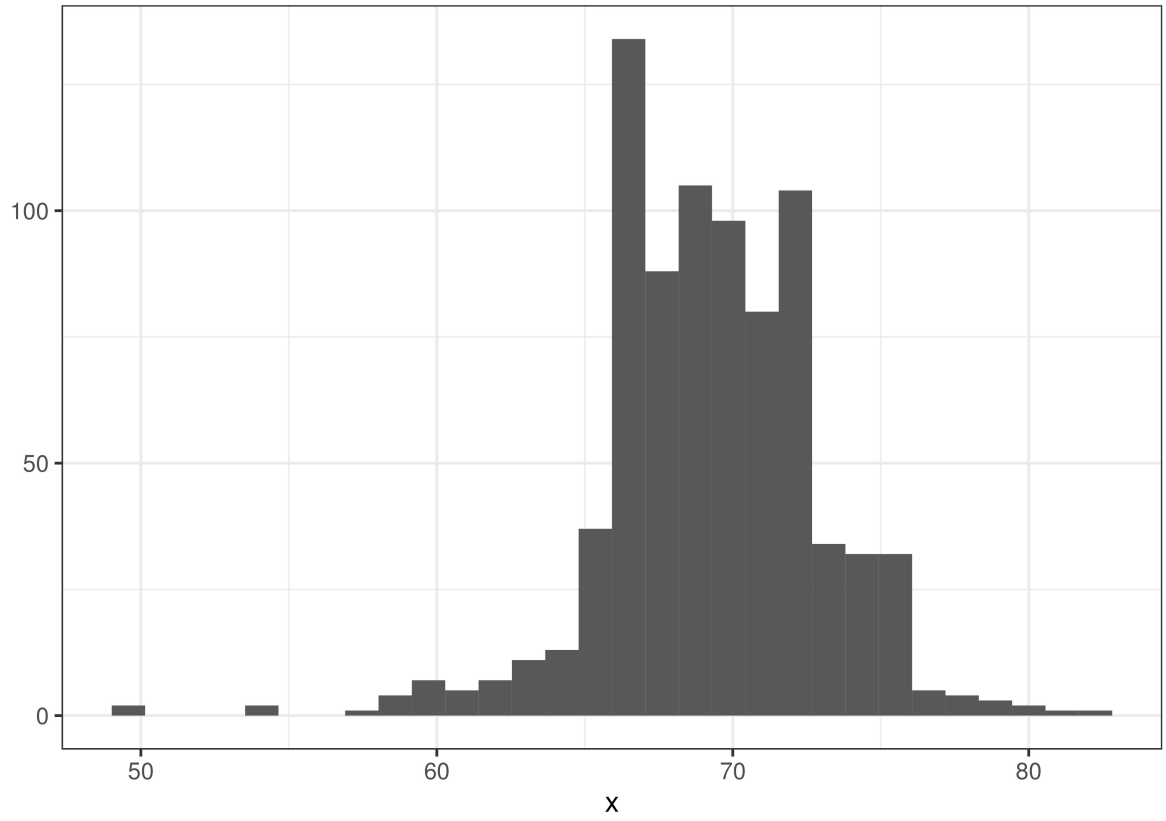



```
# (오류!) %>%는 할당만 하지 자동저장은 안된다는 것을 명심해라.
```

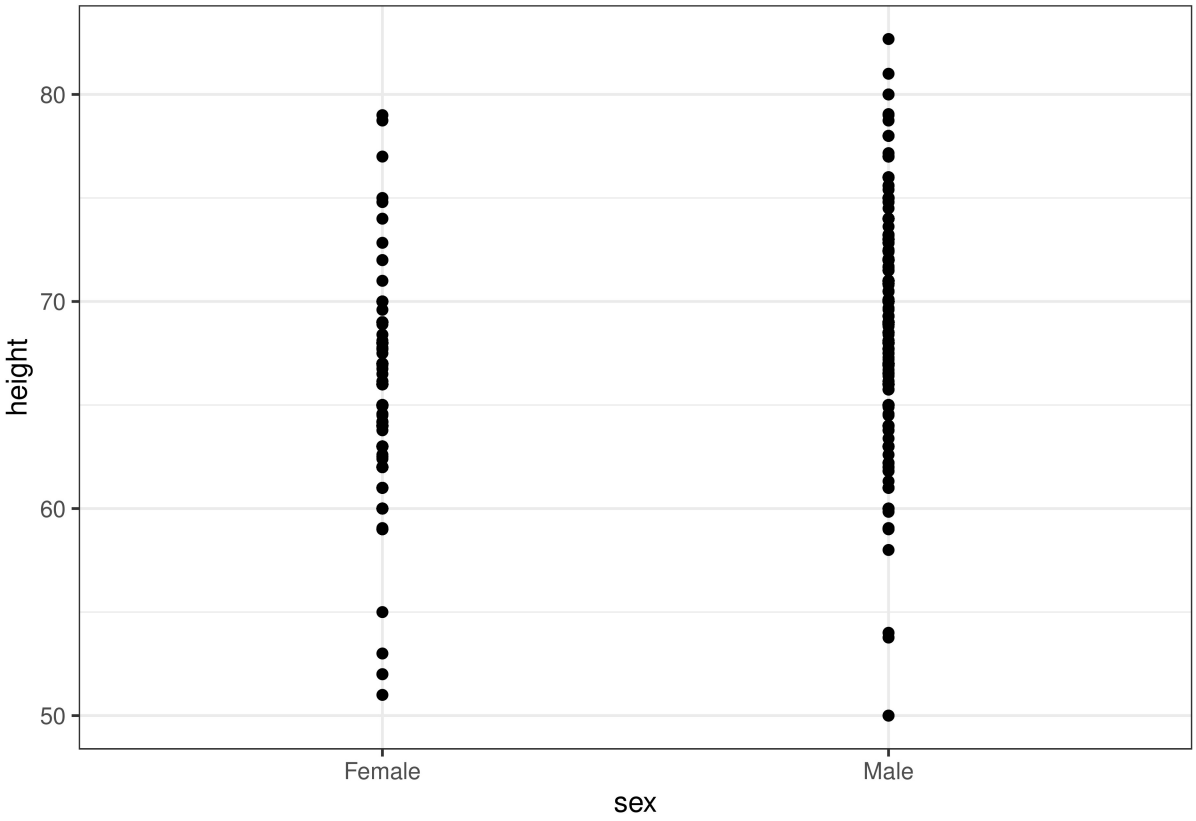
- example 13: Quick plots

```
x <- heights %>% filter(sex=="Male") %>% pull(height)
qplot(x) # factor or numeric vector, we obatin qplot like below.
```

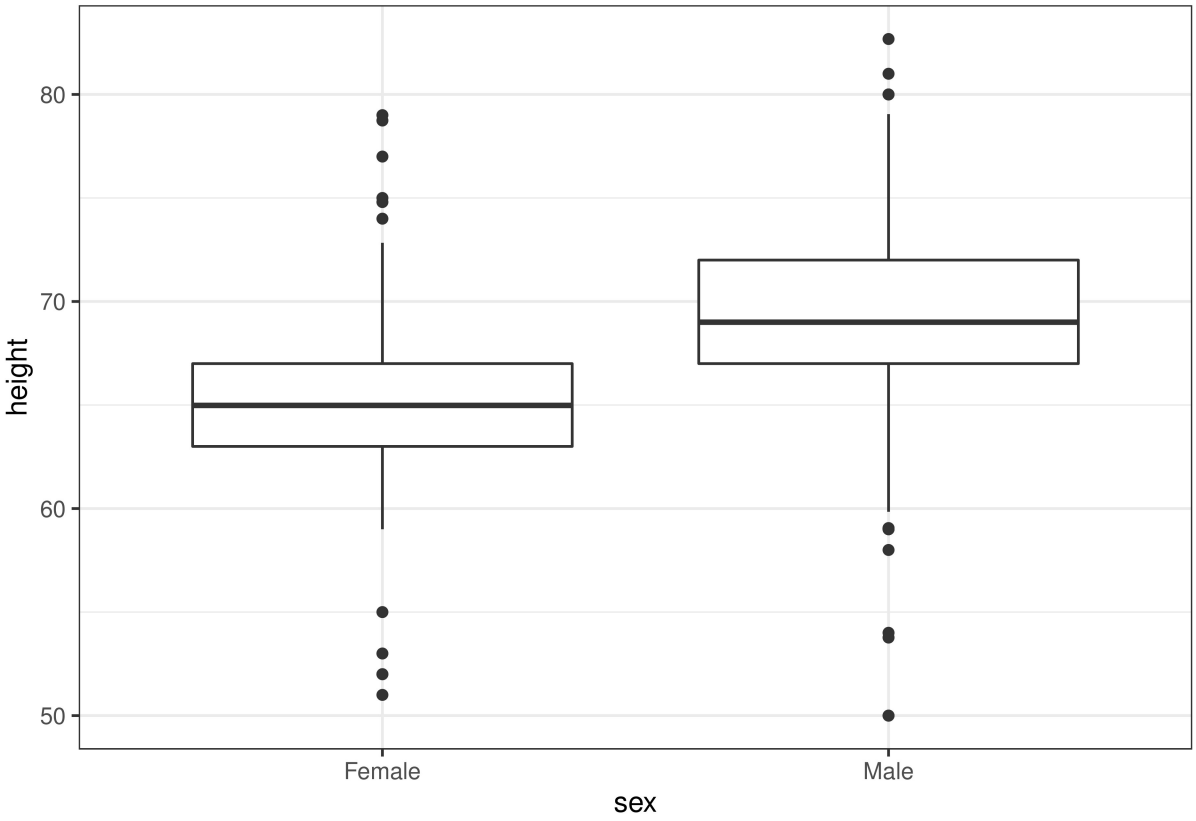
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



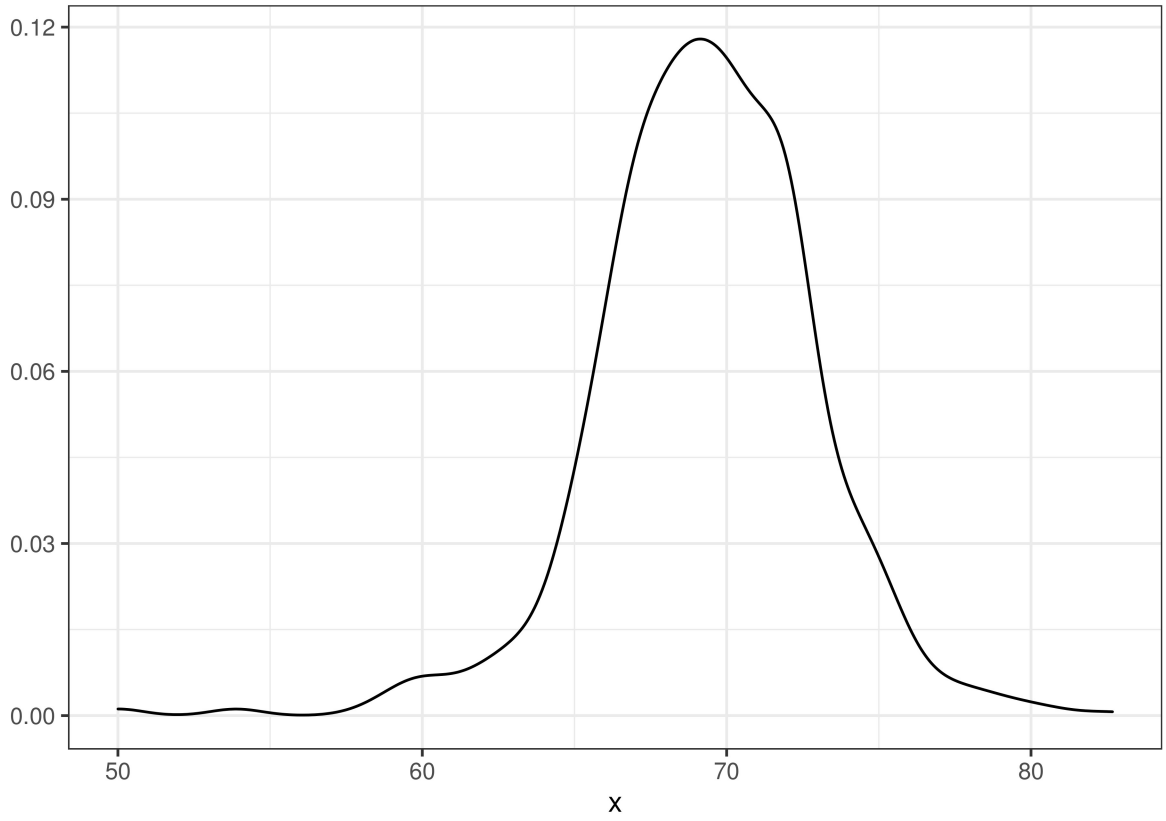

```
heights %>% qplot(sex,height,data=.)
```



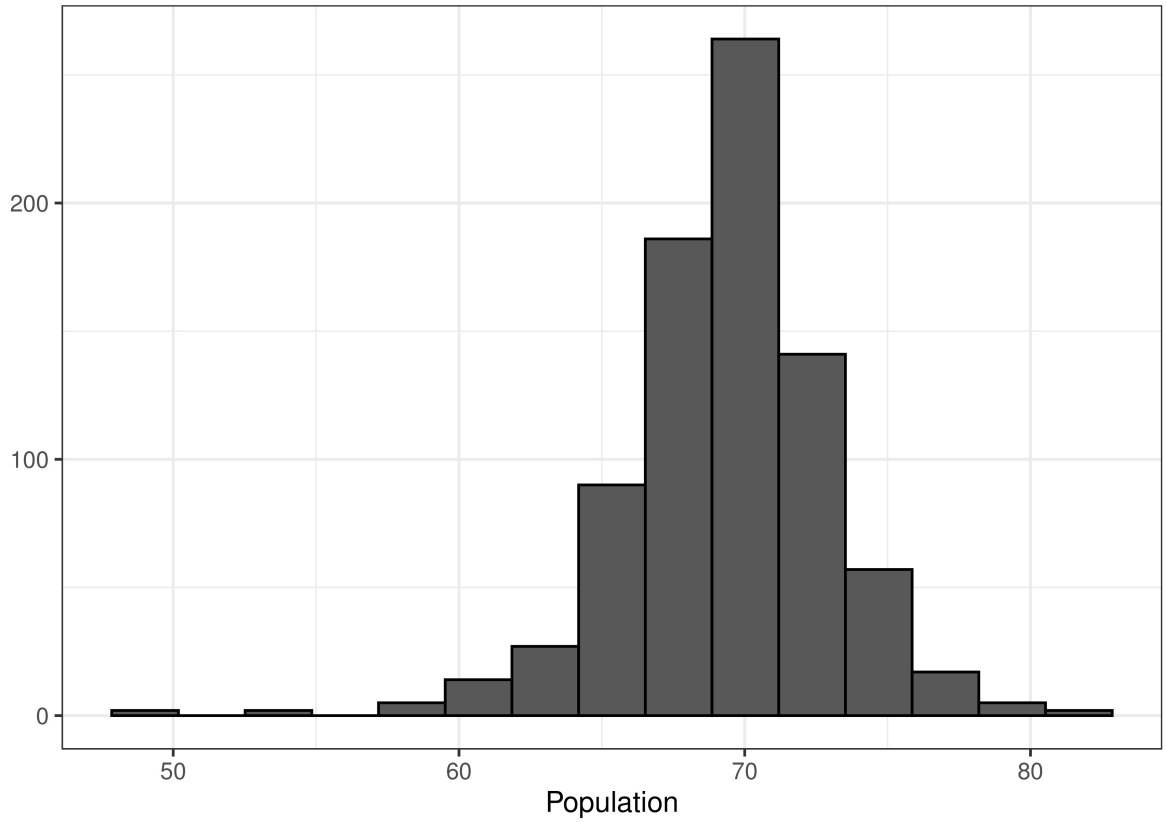

```
# (중요!)dot operator: we do not have to keep naming new objects. ACCESS DATA
# means you get every variable!!

heights %>% qplot(sex,height,data=.,geom='boxplot')
```




```
qplot(x,geom='density')
```




```
qplot(x,bins=15,color=I('black'),xlab='Population')
```




```
# I쓰는 이유: Qplot treat black as "character", rather than converting to "factor".
# which is default within aes function.
# I means "keep it as it is"
```