쌩 초짜를 위한 R

정태훈

2014년 6월 15일

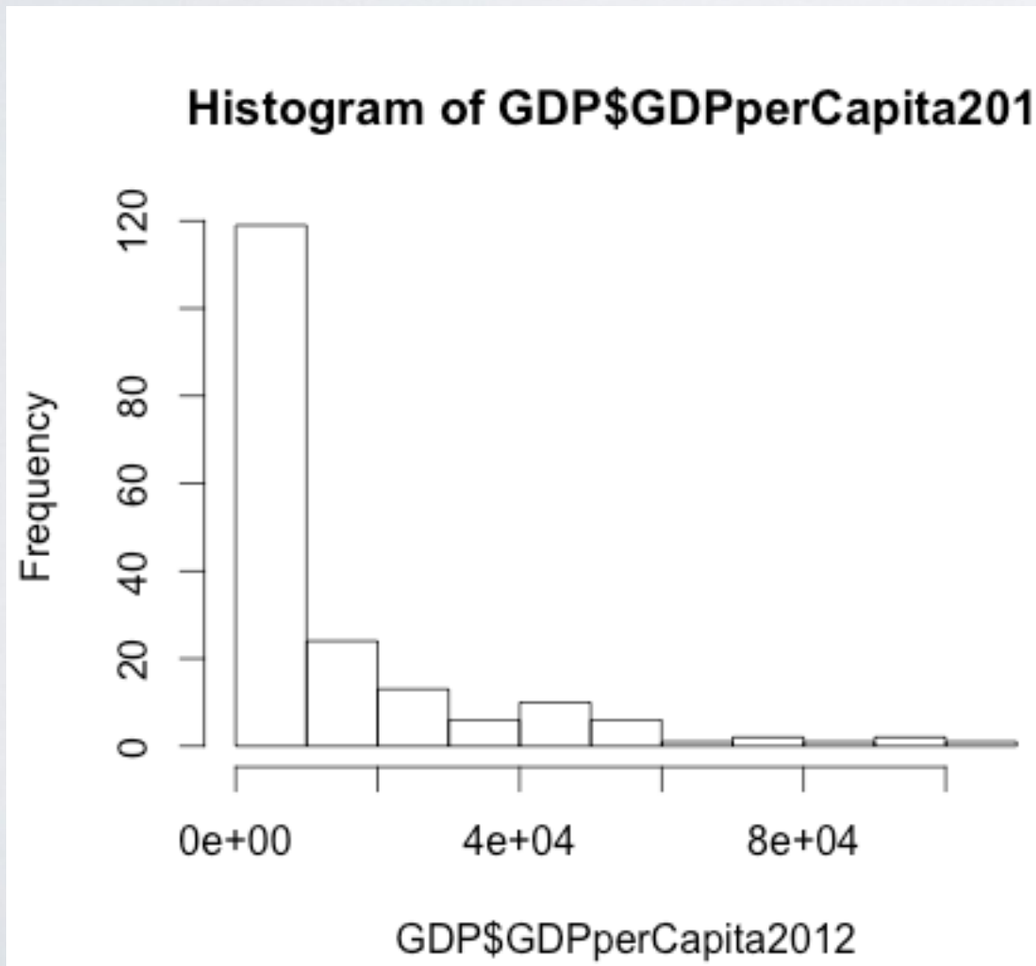
- 앞에선 plot() 함수를 이용해서 데이터를 가지고 어떻게 산포도(scatter plot) 같은 그림을 그릴 수 있는지 살펴 봤습니다. 실제 이 함수는 R의 가장 기본적인 그림 함수로 적절한 매개변수와 조합했을 때 왠만한 도표는 다 그릴 수 있을 만큼 아주 막강한 기능을 가지고 있습니다.
- 하지만 그림 가운데 일부는 편리를 위해 따로 독립된 함수로 제공됩니다. 이번엔 그런 그림을 둘 알아보겠습니다. 바로 히스토그램(histogram)과 상자 그림(boxplot)입니다.
- 히스토그램은 어떤 구간에 얼마나 데이터가 모여 있는지를 파악하는 가장 기본적인 그림이고 상자 그림은 여러 그룹의 데이터가 서로 어떤 분포를 보이는지 파악하는 가장 기본적인 그림입니다. 데이터 이해에 없어서는 안 될 그런 그림들이죠!

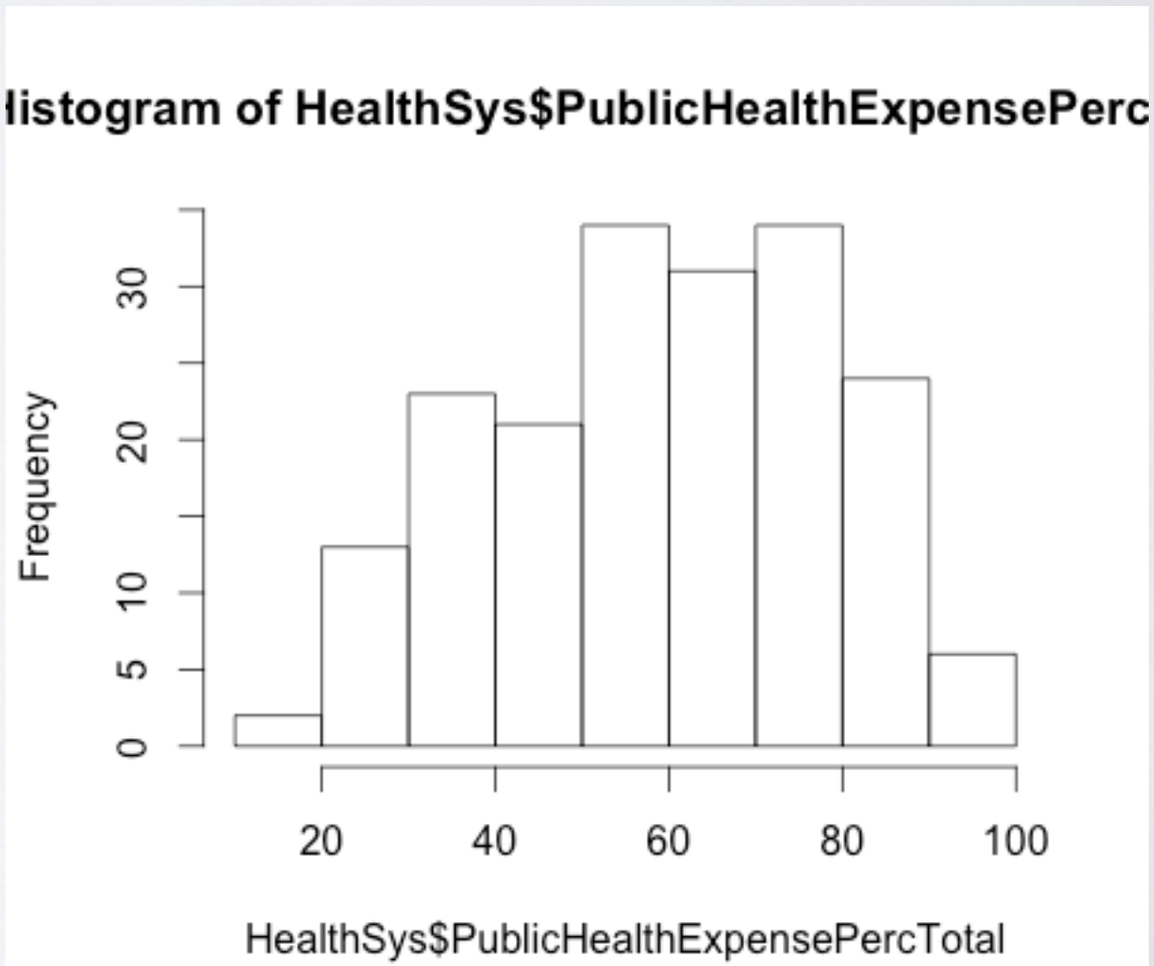
제4과. 그림 그리기

(2) 히스토그램

- 먼저 GDP 데이터의 GDPperCapita2012 변수와 HealthSys 데이터의 PublicHealthExpensePercTotal 변수에 대한 히스토그램을 그려보겠습니다.
- 히스토그램을 그리는 명령어는 간단합니다. hist(..)죠. 그래서 다음과 같이 하면 일단 히스토그램을 그릴 수 있습니다.

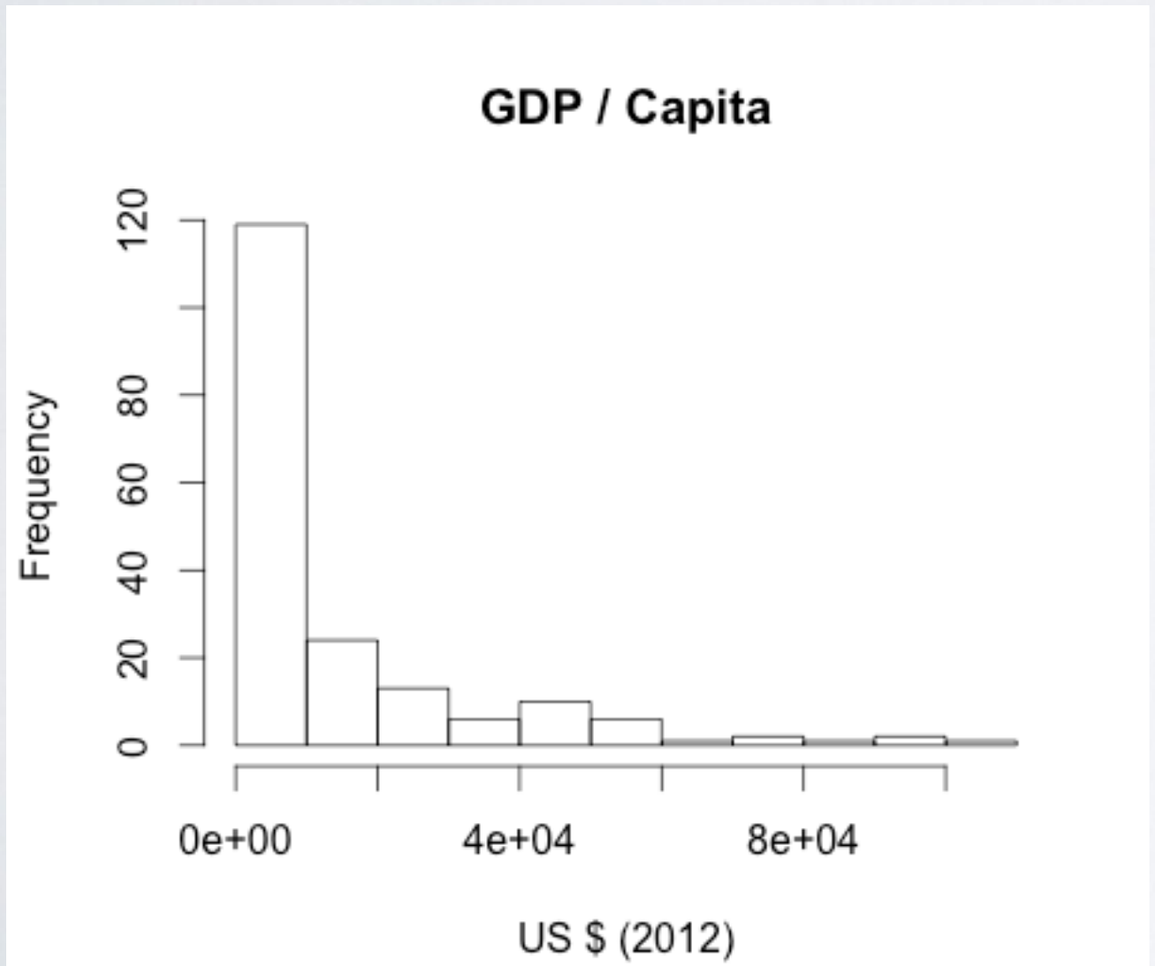
```
> hist(GDP$GDPperCapita2012)
> hist(HealthSys$PublicHealthExpensePercTotal)
```

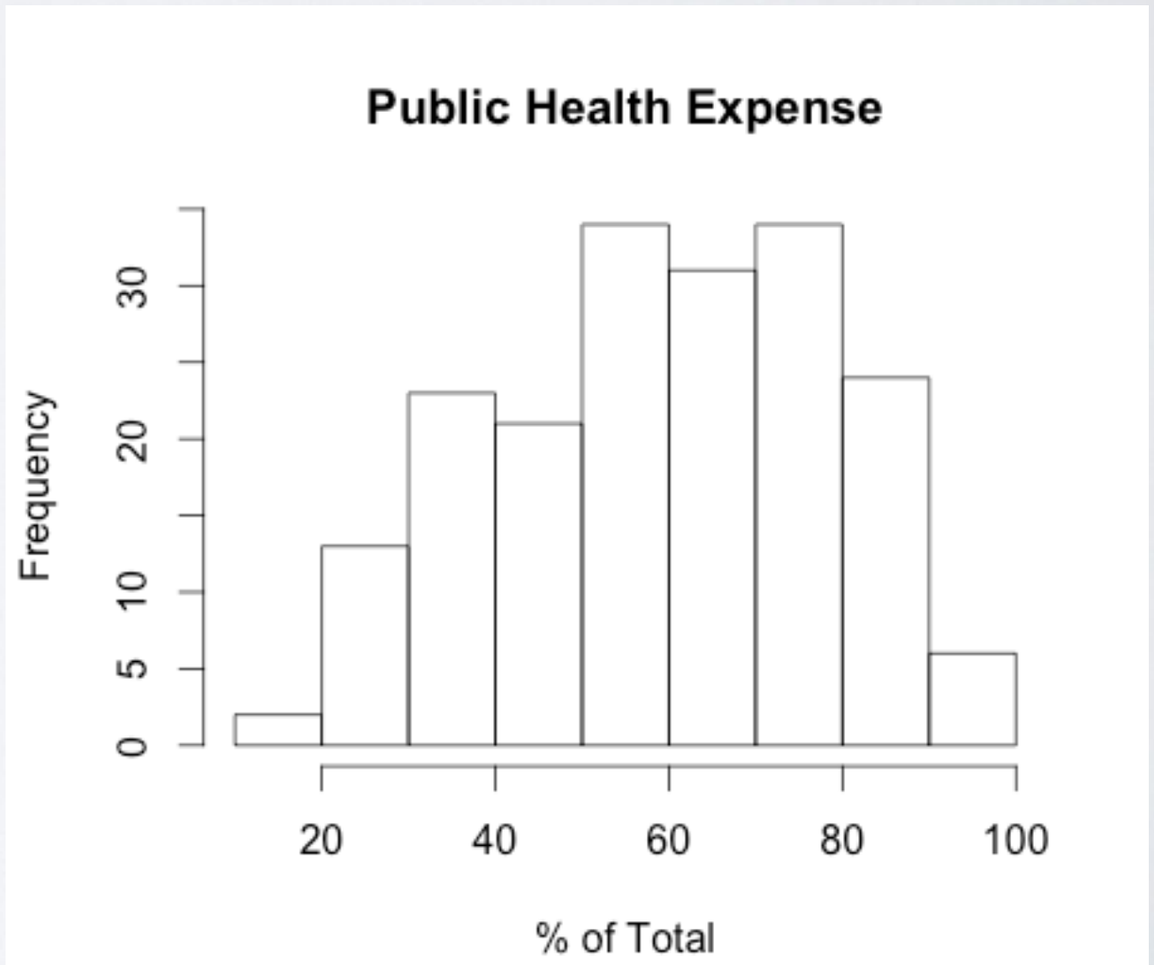




- 문제는 뭐 물론 다른 그림도 마찬가지지만 기본적으로 나오는 히스토그램은 늘 뭔가 5% 부족하다는 겁니다. 우선 그림이나 x축 이름이 별로 마뜩지 않습니다. 그래서 이것들부터 좀 바꿔 보겠습니다. 여기엔 xlab과 main이라는 입력변수가 유용합니다.

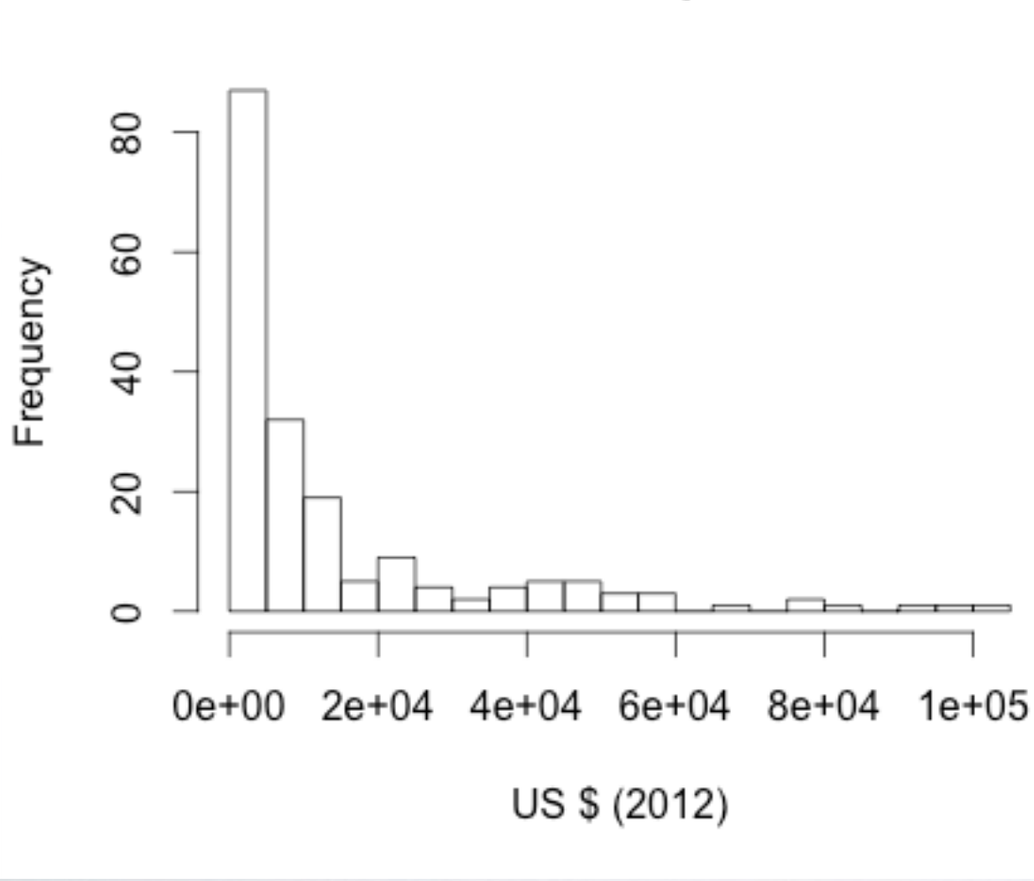
  ```
  > hist(GDP$GDPperCapita2012, xlab="US $ (2012)", main="GDP / Capita")
  > hist(HealthSys$PublicHealthExpensePercTotal, xlab="% of Total", main="Public Health Expense")
  ```

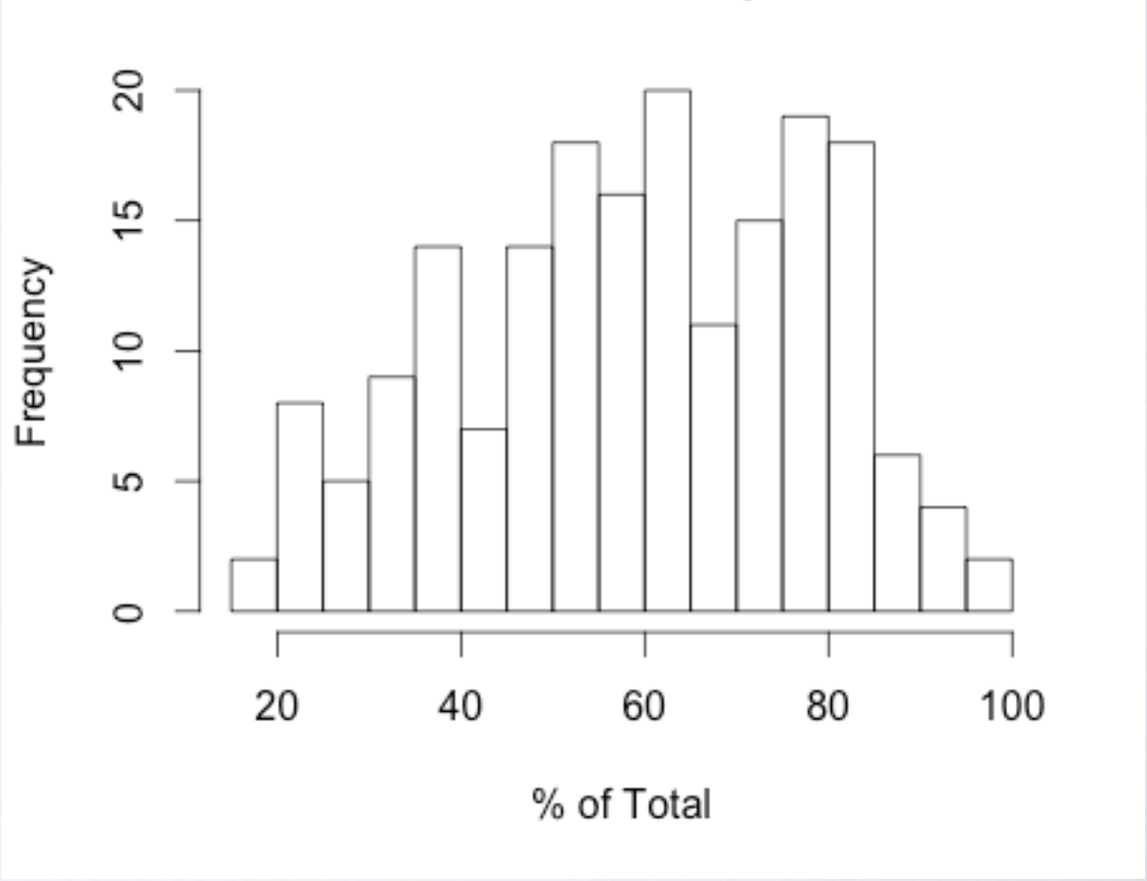




- 그림과 x축의 이름은 이제 좀 나아졌네요. 그런데 히스토그램 구간의 개수가 좀 너무 작은 것 같지 않습니까? 이걸 좀 늘여보겠습니다. 이걸 조절하는 입력변수는 break라는 겁니다! (R은 똑똑해서 입력변수가 구분될 정도만 알려줘도 알아 듣습니다. 그래서 break를 굳이 다 안 쓰고 그냥 br만 써도 됩니다.)

```
> hist(GDP$GDPperCapita2012, xlab="US $ (2012)", main="GDP / Capita", br=25)    ## 구간을 25개로 나눕니다!
> hist(HealthSys$PublicHealthExpensePercTotal, xlab="% of Total", main="Public Health Expense", br=25)    ## 구간을 25개로 나눕니다!
```

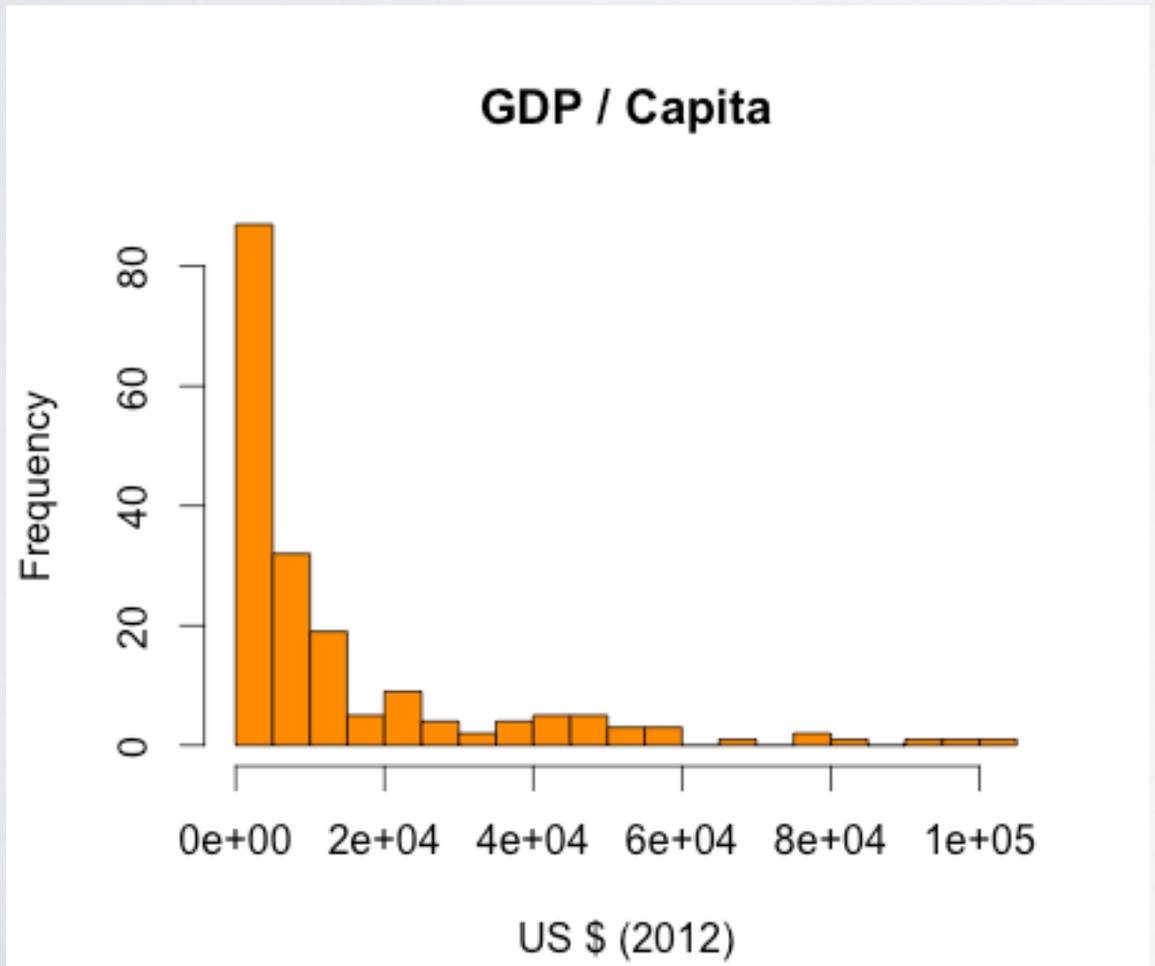


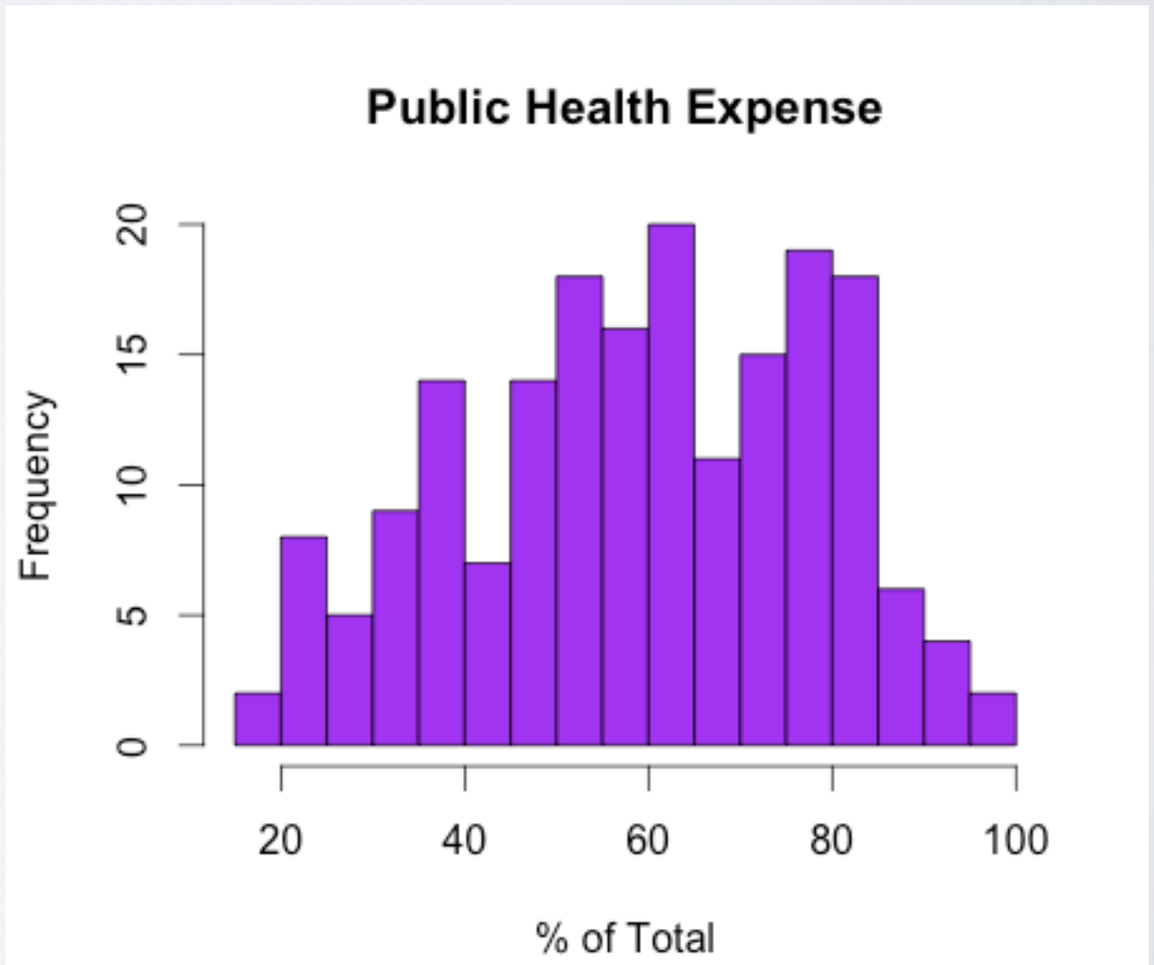


- 그런데 그림에 색깔이 없으니까 너무 밋밋하네요. 그래서 색깔을 한 번 넣어보겠습니다.

```
> hist(GDP$GDPperCapita2012, xlab="US $ (2012)", main="GDP / Capita", br=25,
col="darkorange")  ## 오렌지색!
> hist(HealthSys$PublicHealthExpensePercTotal, xlab="% of Total",
main="Public Health Expense", br=25, col="purple")  ## 보라색!
```

- 그럼 R에서 쓸 수 있는 색깔은 어떻게 알아볼 수 있을까 궁금하시죠? 이 두 사이트를 참고하세요.
  이름으로 지정할 때: http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf
  좀더 일반적인 내용: http://www.stat.tamu.edu/~jkim/Rcolorstyle.pdf

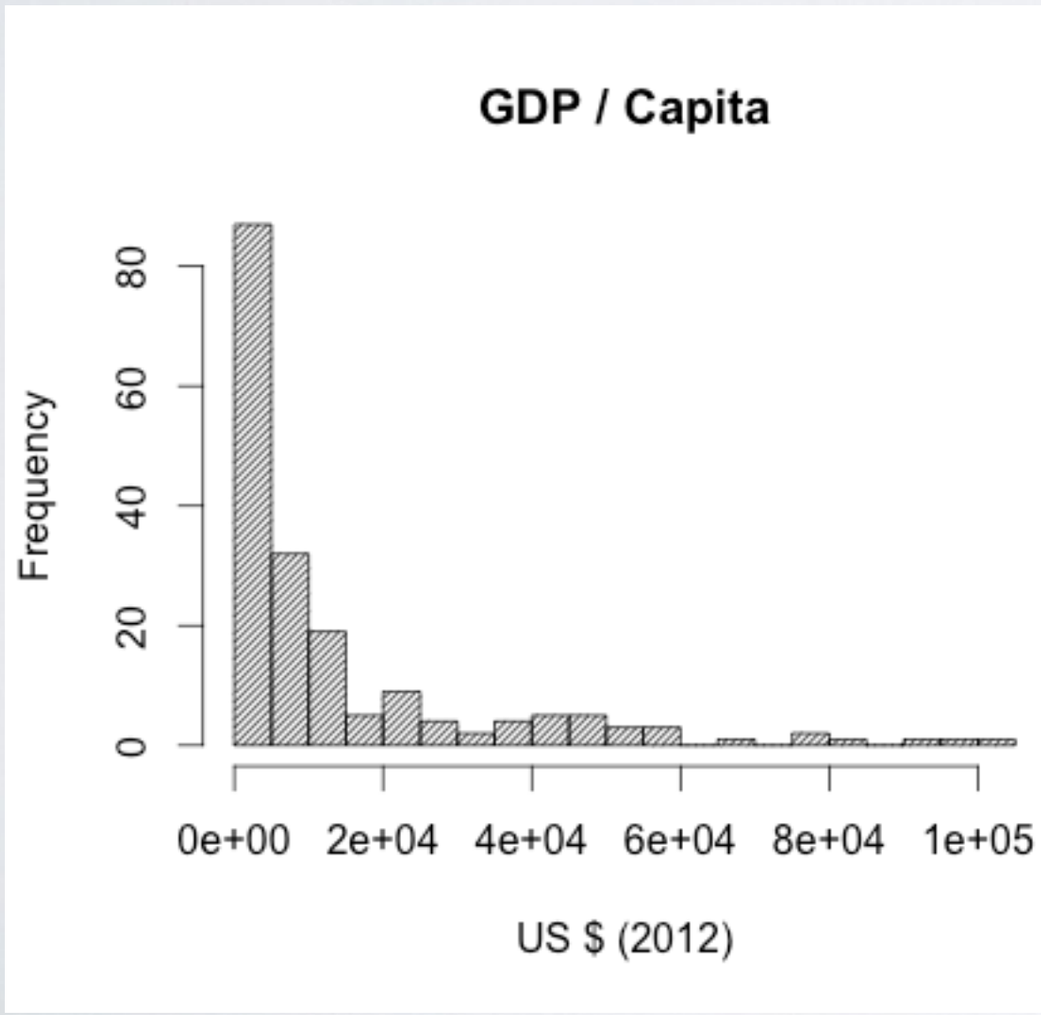


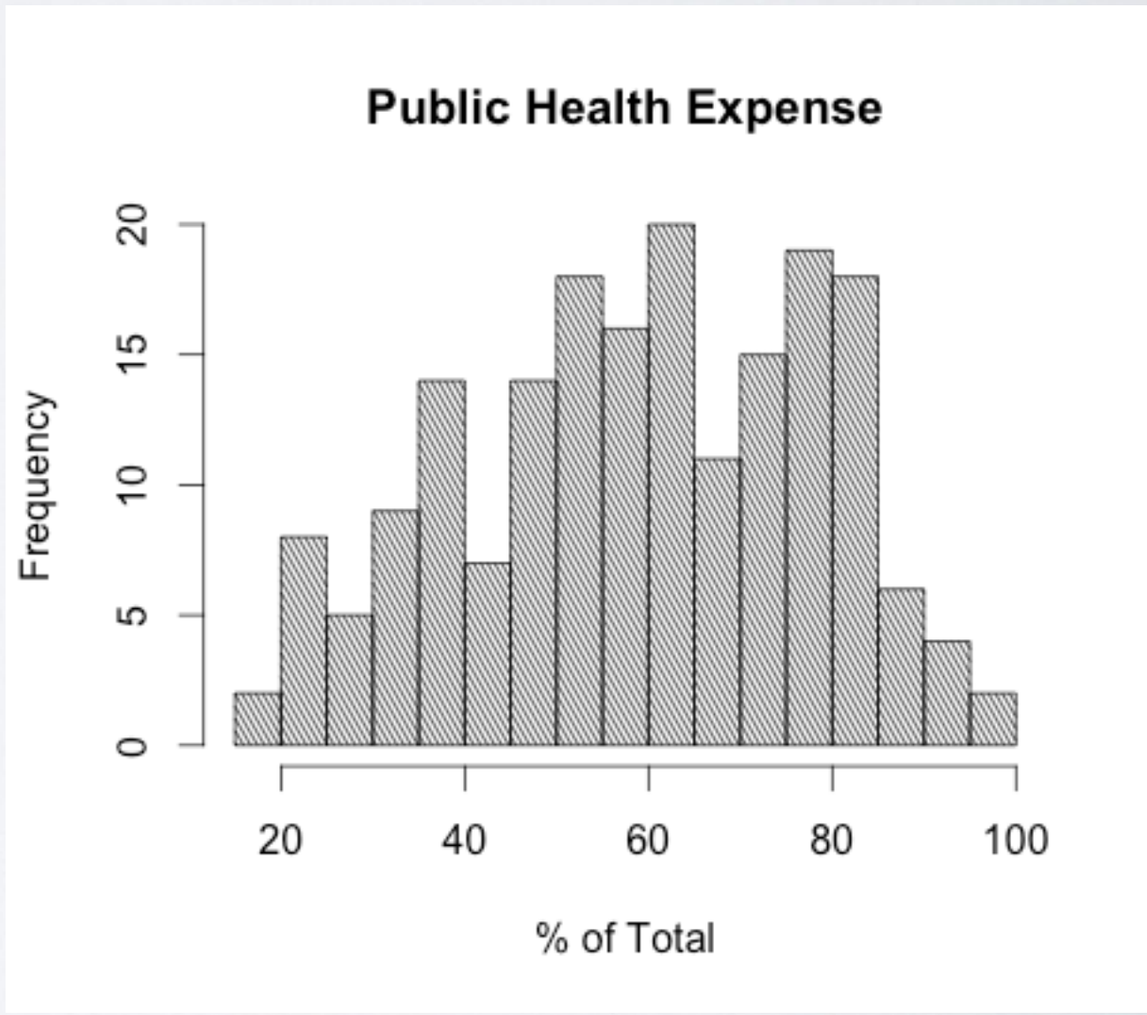


- 참고로 히스토그램에는 색깔만 쓸 수 있는 건 아닙니다. 빗금을 쓸 수도 있습니다!

```
> hist(GDP$GDPperCapita2012, xlab="US $ (2012)", main="GDP / Capita",
br=25, density=30)  ## 1인치에 30개의 45도 빗금!
> hist(HealthSys$PublicHealthExpensePercTotal, xlab="% of Total",
main="Public Health Expense", br=25, density=30, angle=120)  ## 120도 빗금!
```

- "density=30"은 1인치에 30개의 빗금을 넣으라는 소리고 "angle=45"는 빗금의 각도를 45도로 하라는 말입니다. 원래 이 각도가 기본인데 90도면 수직선, 180도면 수평선입니다.

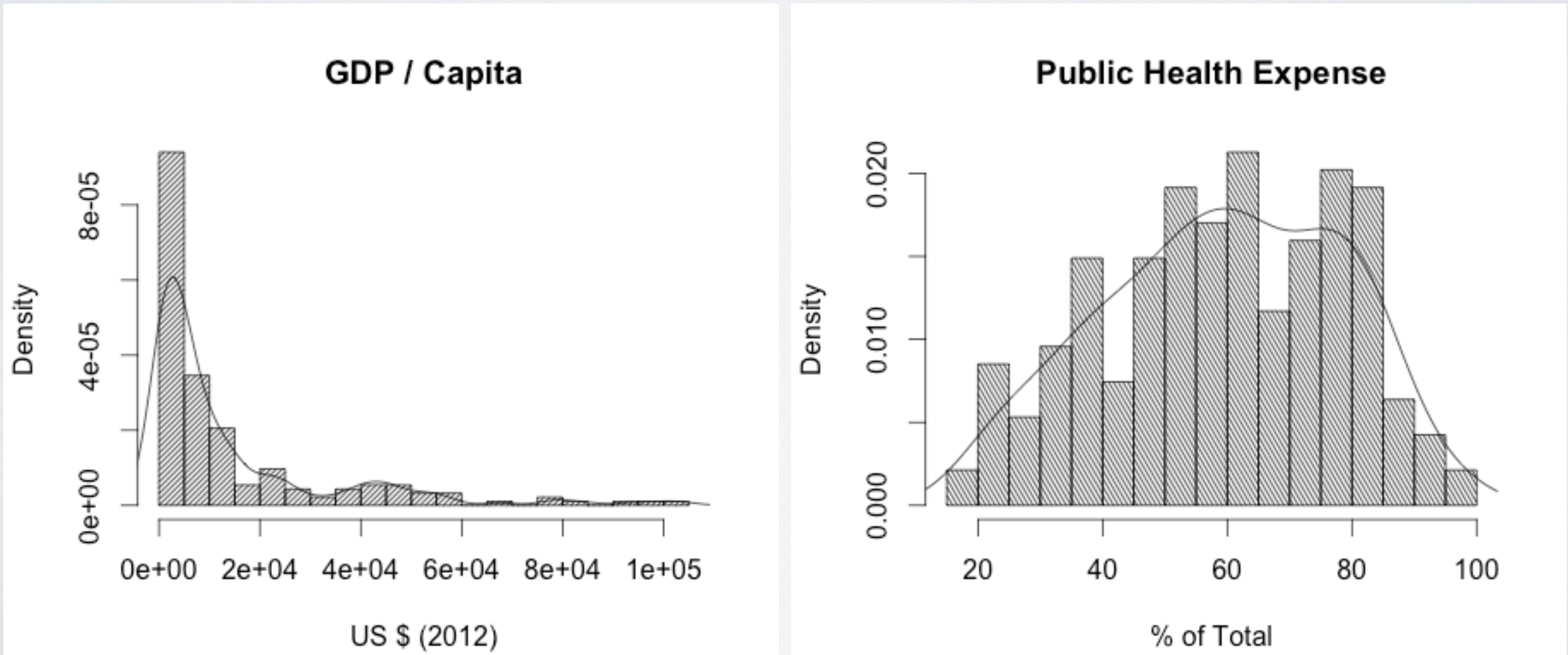




- 히스토그램에 추세선도 넣어 봅시다. 그런데 이건 다음과 같이 두 단계를 거쳐야 합니다.

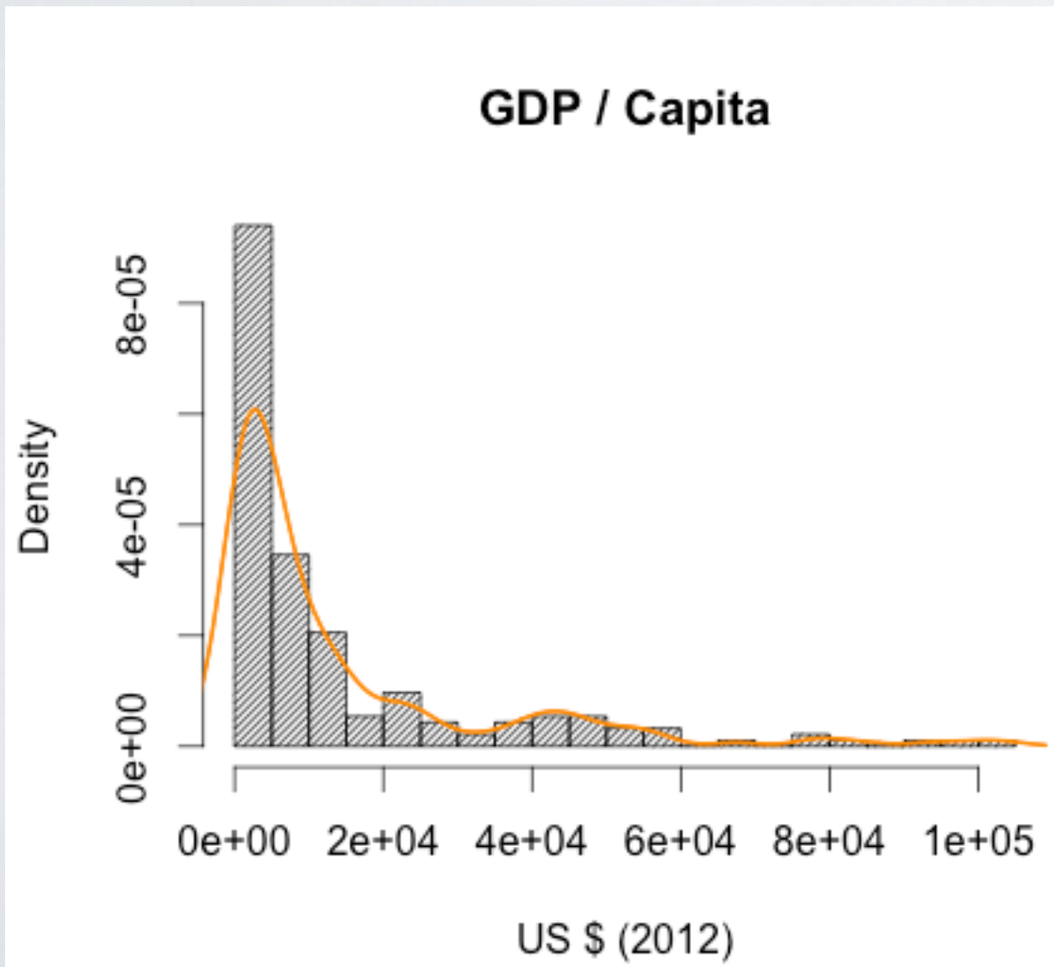
  ```
  > hist(GDP$GDPperCapita2012, xlab="US $ (2012)", main="GDP / Capita", br=25, density=30, freq=F)  ## 각 구간 빈도(frequency) 아니고!
  > lines(density(GDP$GDPperCapita2012, na.rm=T))  ## 추세선을 그려요!
  ```

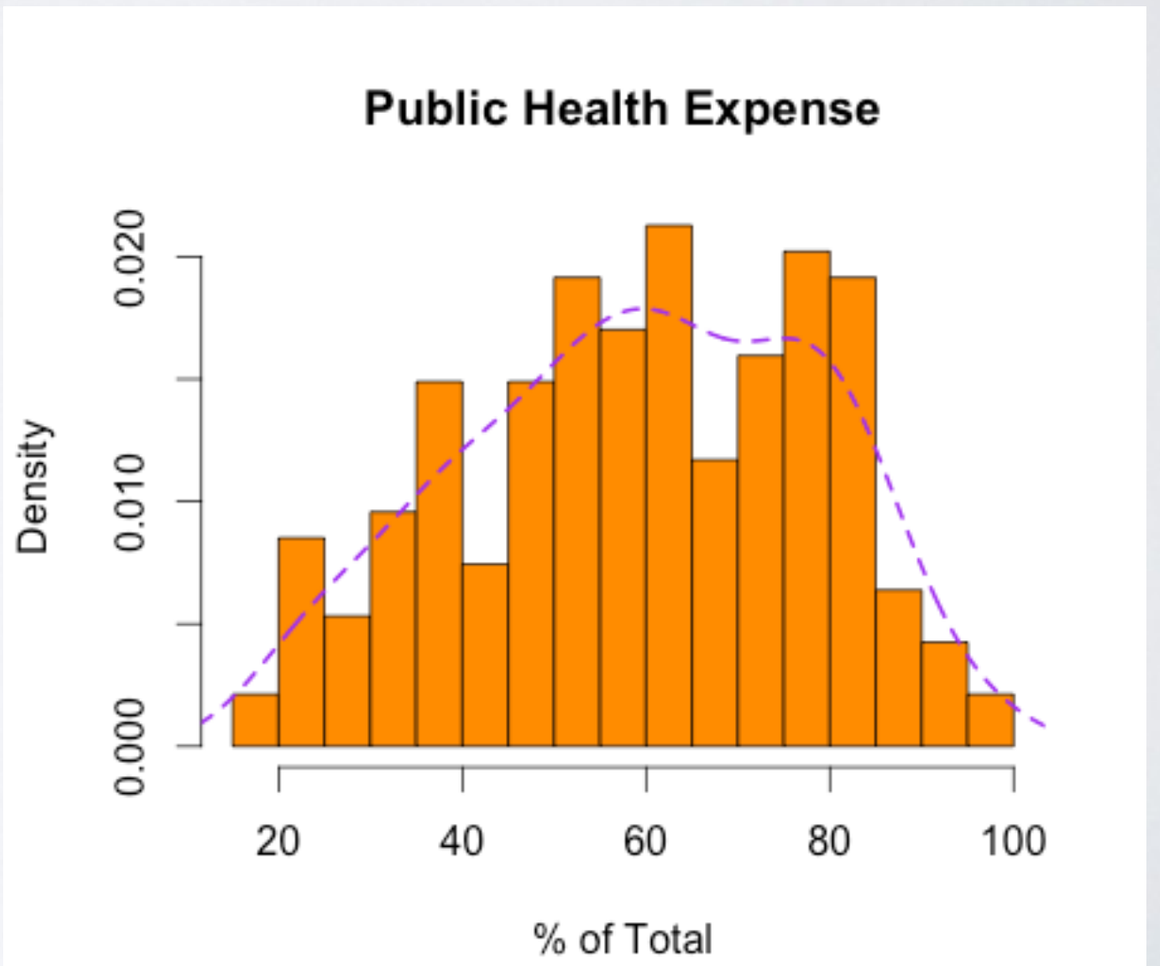
- 추세선을 넣을 땐 각 구간에 든 데이터의 갯수가 아닌 전체 데이터 가운데 몇 개가 들었는지 비율(데이터 밀도 density)을 계산합니다. `hist(…, freq=F)`!!! 그래야 뒤에 `density(..)` 함수를 쓸 수 있습니다. 이게 전체적인 추세선을 계산해 주는데 그 결과를 이용해 `lines(..)`가 선을 그려주는 겁니다. `density(..)` 함수를 그냥 쓰면 없는 값을 불평해서 "`na.rm=T`"를 넣어줬습니다.



- 물론 히스토그램과 추세선에 다양한 변화를 줄 수도 있겠죠!

```
> hist(GDP$GDPperCapita2012, xlab="US $ (2012)", main="GDP / Capita",
br=25, density=30, freq=F)
> lines(density(GDP$GDPperCapita2012, na.rm=T), col="darkorange", lwd=2)
> hist(HealthSys$PublicHealthExpensePercTotal, xlab="% of Total",
main="Public Health Expense", br=25, col="darkorange")
> lines(density(HealthSys$PublicHealthExpensePercTotal, na.rm=T),
col="purple", lwd=2, lty=2)
```

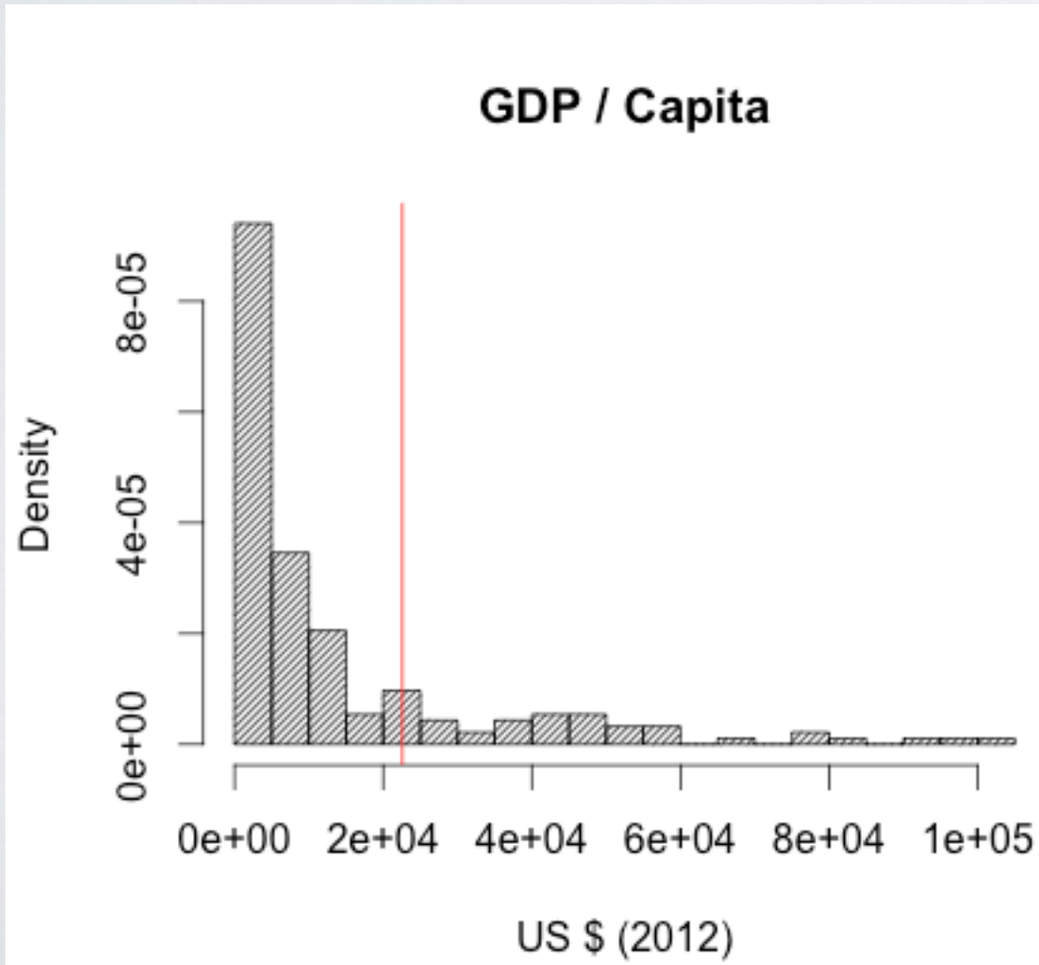


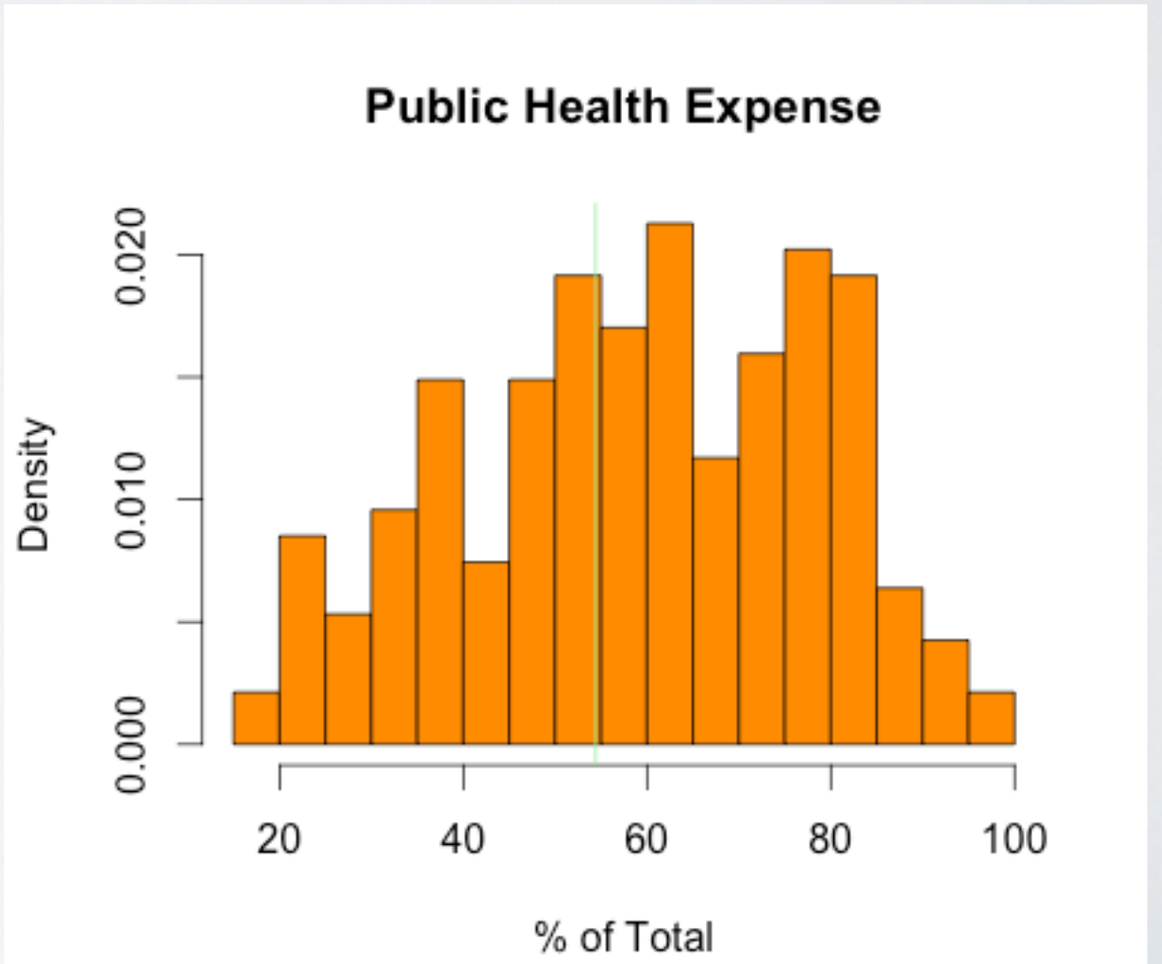


- 그런데 한국의 위치는 어디죠? 이걸 보기 쉽도록 한국 항목의 위치에 수직선을 그어 보겠습니다.

```
> indkor <- which(GDP$CountryCode == "KOR")  ## 한국 항목의 위치
> hist(GDP$GDPperCapita2012, xlab="US $ (2012)", main="GDP / Capita", br=25, density=30, freq=F)  ## 우선 전체 히스토그램을 그리고
> abline(v=GDP$GDPperCapita2012[indkor], col="red")  ## 한국 값에 수직선 긋기
```

- 여기 나온 abline(..) 함수는 원래 y = a + b * x 직선을 그리라는 뜻입니다. 그런데 수직선이나 수평선을 그을 때도 쓸 수 있는데 여기서처럼 abline(v=..)라고 하면 해당하는 값에서 수직선을 긋고 abline(h=..)라고 하면 해당하는 값에서 수평선을 긋는 겁니다.





제4과. 그림 그리기

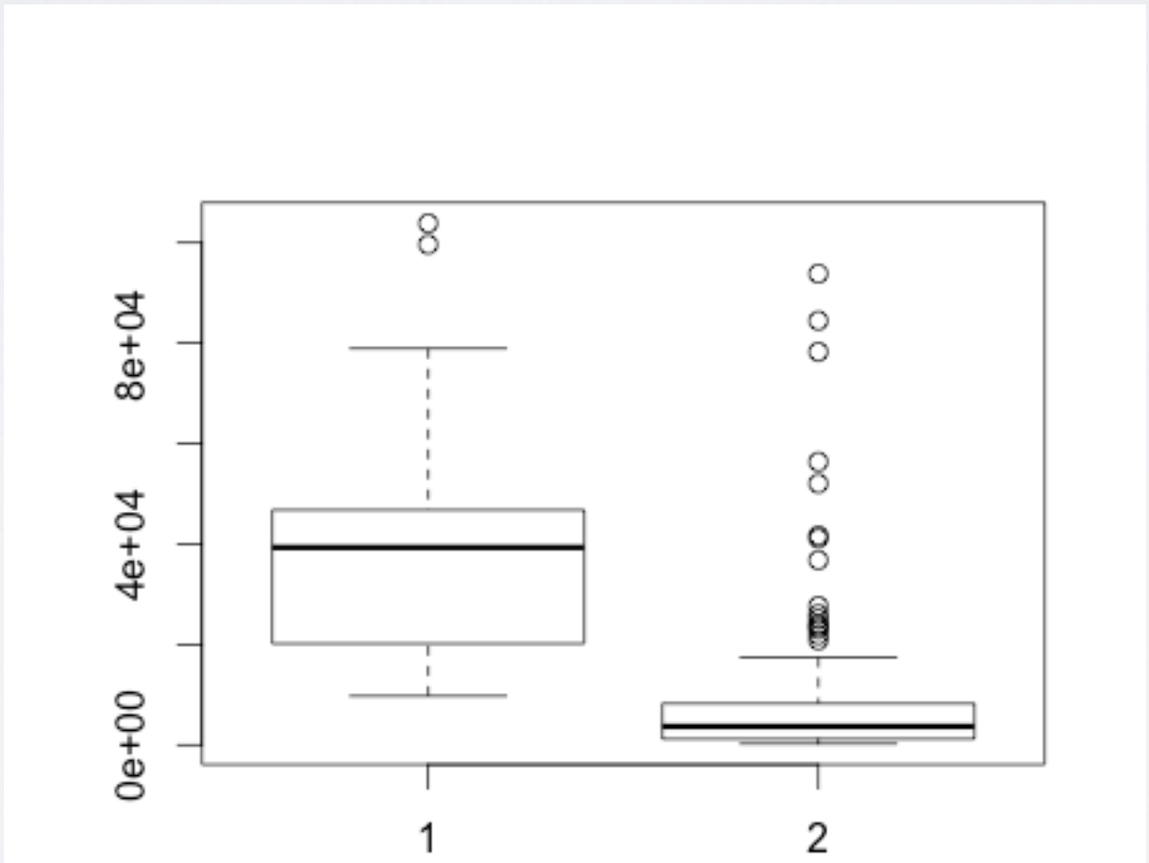
(3) 상자 그림

- 이번엔 상자그림(boxplot)을 그려 보겠습니다. 많이들 보셨을텐데 상자그림이 뭐냐 하면 비교하려는 여러 데이터를 각 데이터별 묶음으로 그려주는 그런 그림입니다. 이렇게 해서 전체적인 분포구간을 데이터별로 비교하기 쉽게 해주는 그런 그림이죠.
- 예를 들어 2012년 일 인 당 GDP를 OECD와 그렇지 않은 나라 이렇게 비교해 보겠습니다. 우선 OECD 국가와 여타 국가의 값은 이렇게 구할 수 있습니다:

```
> gdp.oecd <- GDP$GDPperCapita2012[GDP$OECD == "Y"]  ## OECD인 (== "Y") 국가의 값
> gdp.other <- GDP$GDPperCapita2012[GDP$OECD != "Y"]  ## OECD가 아닌(!= "Y") 국가의 값
```

그리고 상자그림을 그리는 겁니다!

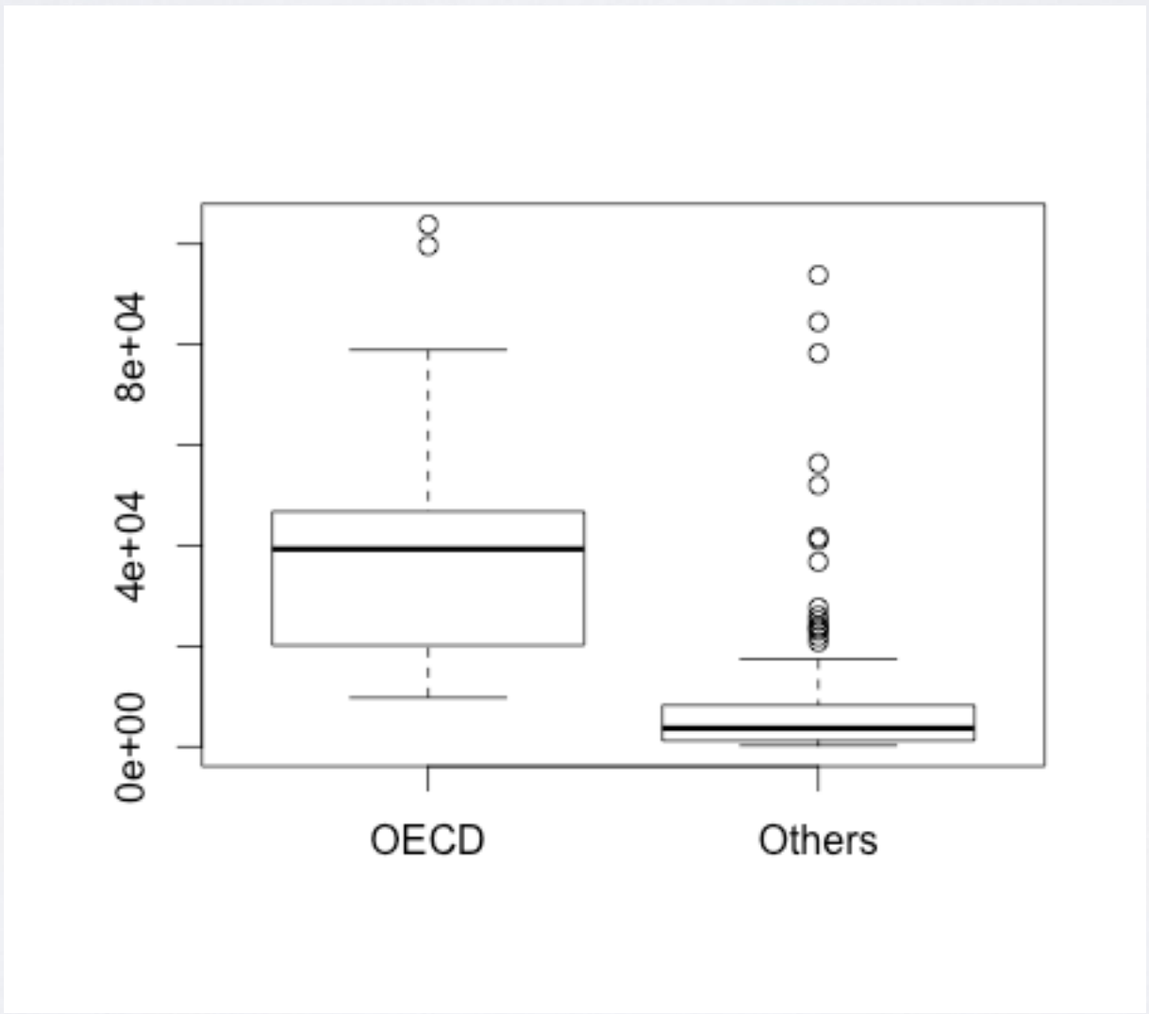
```
> boxplot(list(gdp.oecd, gdp.other))  ## boxplot(..)은 항상 list(..)를 받습니다!
```



- 앞의 상자그림은 너무 허전한데요. 그림에 아무런 설명도 없고.
- 우선 두 상자에 이름을 넣어 보겠습니다. 이건 리스트를 만들 때 이름을 부여하면 됩니다.

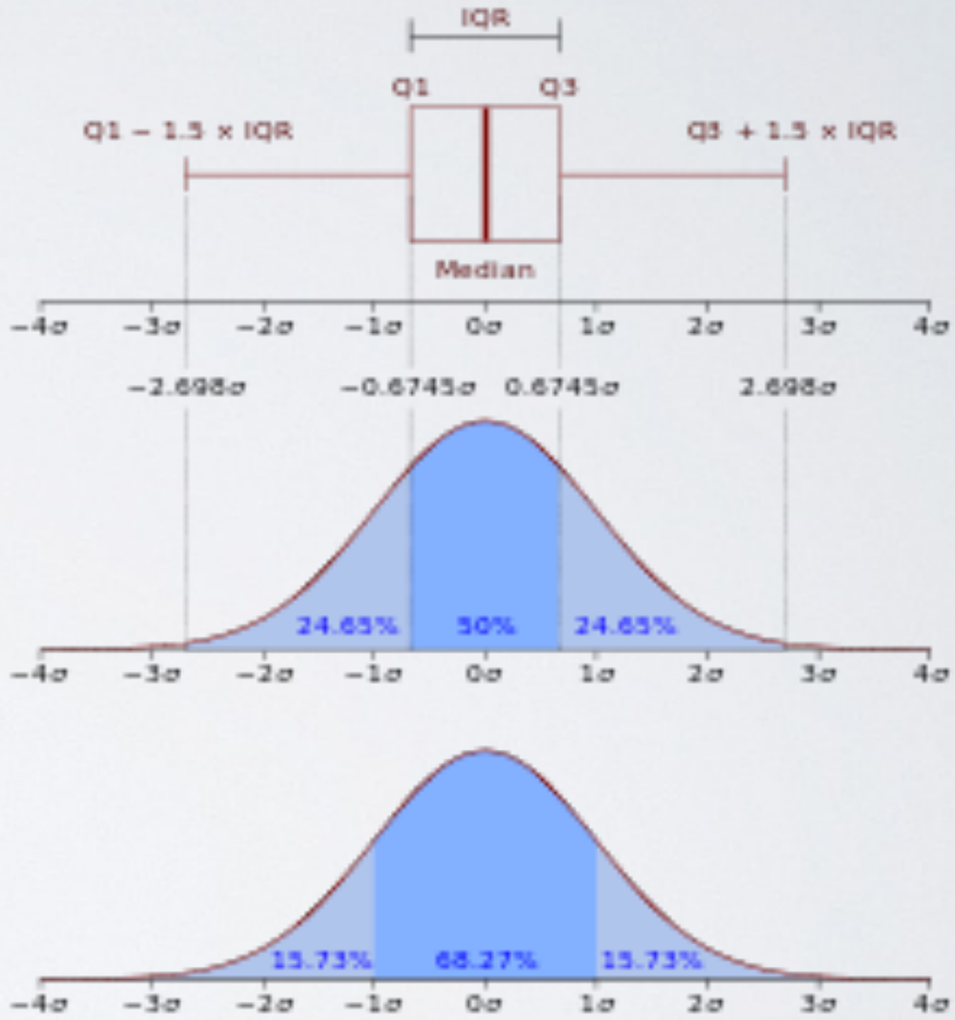
```
> gdplist <- list("OECD"=gdp.oecd, "Others"=gdp.other)
> boxplot(gdplist)
```

- 보시는 것처럼 각 상자의 이름이 들어갔습니다. 이젠 구분이 수월하겠네요. 이렇게 놓고 보니 OECD 그룹이 뭔가 훨씬 큰 값을 가지고 있다는 걸 알 수 있겠습니다.



여기서 잠깐, 상자그림(BOXPLOT)의 이해

- 상자그림을 보실 때는 오른편의 위키피디아 그림 처럼 이해하시면 됩니다.
- 우선 상자 가운데 굵은 선은 중간값(데이터의 한 가운데 값, median)입니다.
- 상자의 맨왼쪽과 맨오른쪽은 (수직으로 그림이 배열된 경우는 맨아래쪽과 맨위쪽) 전체 데이터를 0~100%로 재배열했을 때, 25%와 75%에 해당하는 값입니다. 이 둘 사이의 간격을 IQR이라고 합니다.
- 상자에서 왼쪽, 오른쪽으로 쭉 뻗어나간 선분을 휘스커(whisker)라고 하는데 이건 상자의 양끝에서 각각 1.5 × IQR 만큼씩 뻗어나갑니다. 그리고 이 영역을 벗어난 곳에 있는 값은 소위 극단값(outlier)라고 취급합니다. 상자그림에도 따로 기호로 표시합니다.



위키피디아

여기서 잠깐, 리스트(LIST)란?

- R에서 리스트는 아주 요긴한 녀석인데 왜냐하면 거기에 넣을 녀석들이 굳이 다 같은 모양이 아니어도 상관 없기 때문입니다. 그러니까 첫번째 항목에는 문자, 두번째 항목에는 숫자, 세번째 항목에는 행렬, 네번째는 다른 리스트, … 뭐 이런 식으로 아무렇게나 넣어도 된다는 말이죠.
- 리스트를 만드는 방법은 예를 들어 다음과 같습니다:

  ```
  agent <- list("name"="Jason Bourne", "age"=26, "ops"=list(…))
  ```

  어느 요원의 기록이 이렇게 리스트로 저장되면 이름이 "Jason Bourne"이고 나이는 26, 그간 수행한 작전은 다른 리스트 뭐 이런 식으로 채워지는 겁니다.
- 이럴 때 저 요원의 이름이 궁금하다면 이렇게 하면 됩니다:

  `agent$name` (혹은) `agent[["name"]]` (혹은) `agent[[1]]`

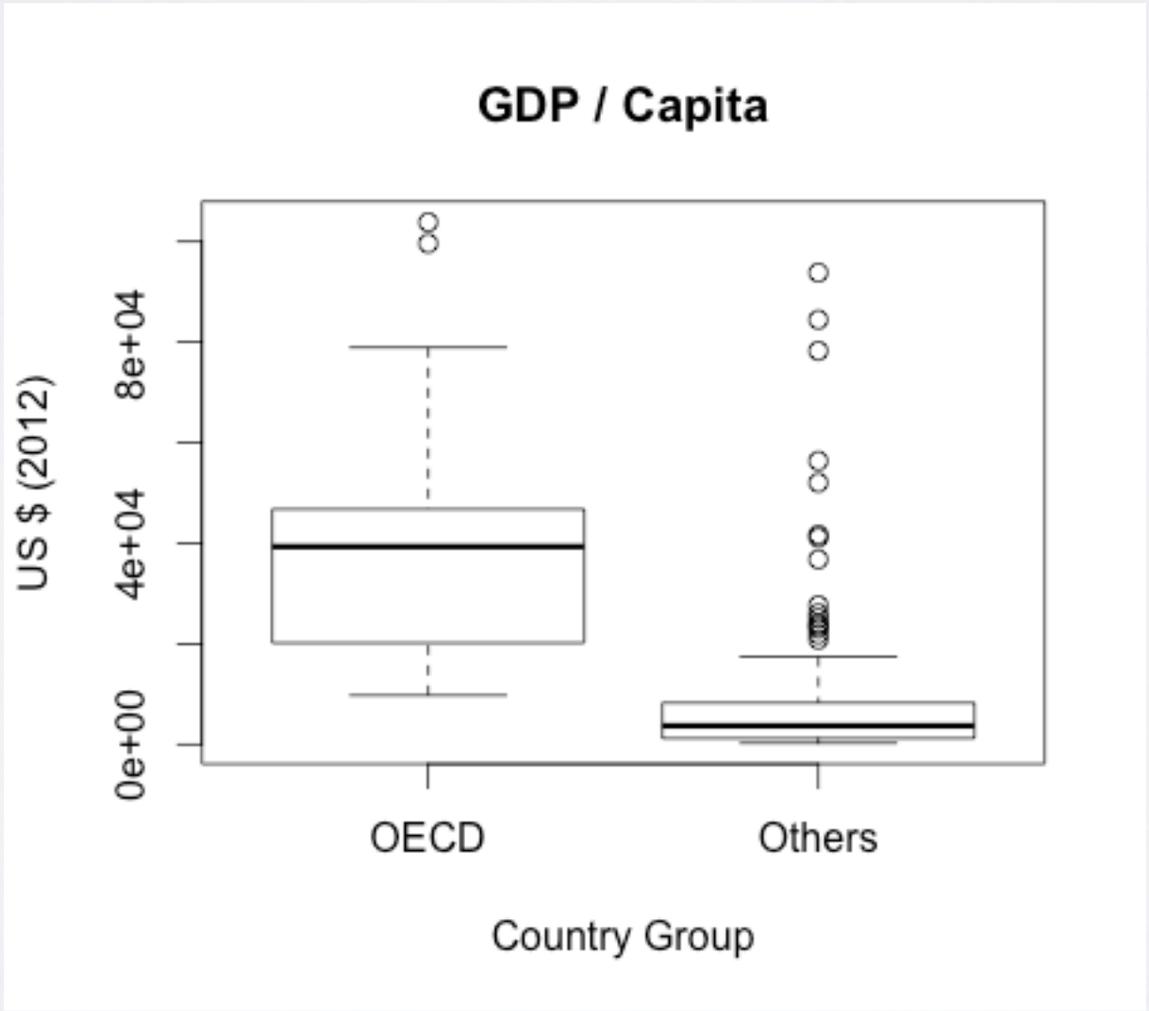
  어디서 본 듯 하지 않습니까? 예, 바로 데이터프레임에서 했던 그 방식입니다. (데이터프레임은 리스트 가운데서도 모든 항목에 들어간 원소의 개수가 동일한 녀석입니다!) 비슷하게 이름은 `agent$age`나 `agent[["age"]]` 이렇게 하시구요.
- 상자그림은 주로 리스트를 받아서 작업을 합니다. 그래서 첫번째 항목에는 OECD 34개국의 일 인 당 GDP를 두번째 항목에는 그외 180개국의 일 인 당 GDP를 넣은 리스트를 직접 만들었습니다. 물론 이렇게도 가능합니다:

  ```
  > gdplist <- list(gdp.oecd, gdp.other)
  > boxplot(gdplist)
  ```

- 이번엔 그림 제목, x-축, y-축 이름 등을 넣어 보겠습니다. 이건 예상하셨겠지만 boxplot(..) 함수 안에서 해결합니다:

  ```
  > boxplot(gdplist, main="GDP / Capita", xlab="Group", ylab="US $ (2012)")
  ```

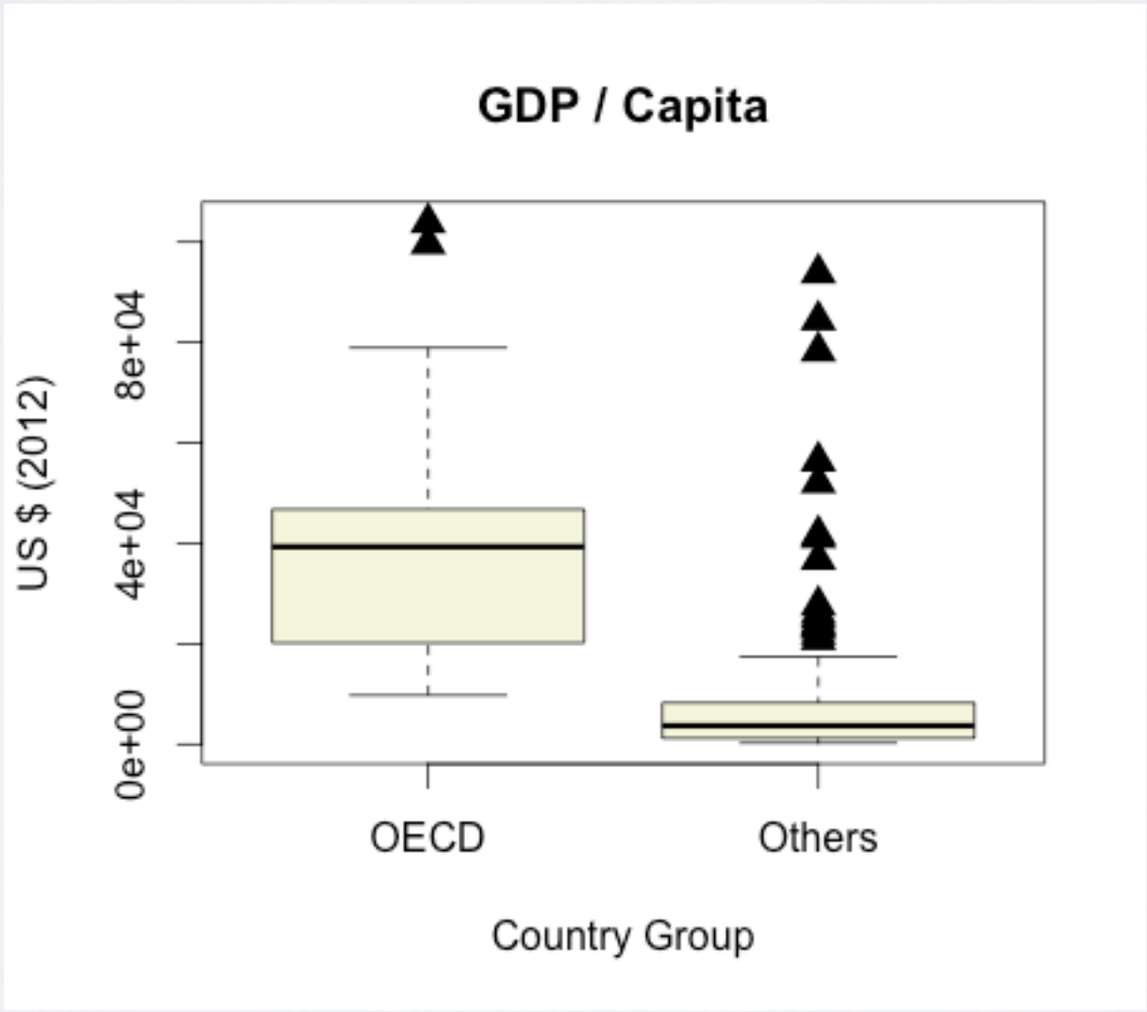
- 보시는 것처럼 각각에 적절한 표시가 들어갔습니다. 그림이 슬슬 모양을 갖춰가고 있습니다.
- 그리고 이렇게 적절히 들어간 표시 덕분에 드디어 OECD 그룹이 일 인 당 GDP가 다른 그룹에 비해 월등히 크다는 게 그림만으로도 명확해 졌습니다!



- 극단값에 해당하는 데이터는 물론 상자도 그리는 방식을 조절할 수 있습니다:

  > boxplot(gdplist, main="GDP / Capita", xlab="Group", ylab="US $ (2012)", pch=17, cex=1.5, bg="darkblue", col="beige")

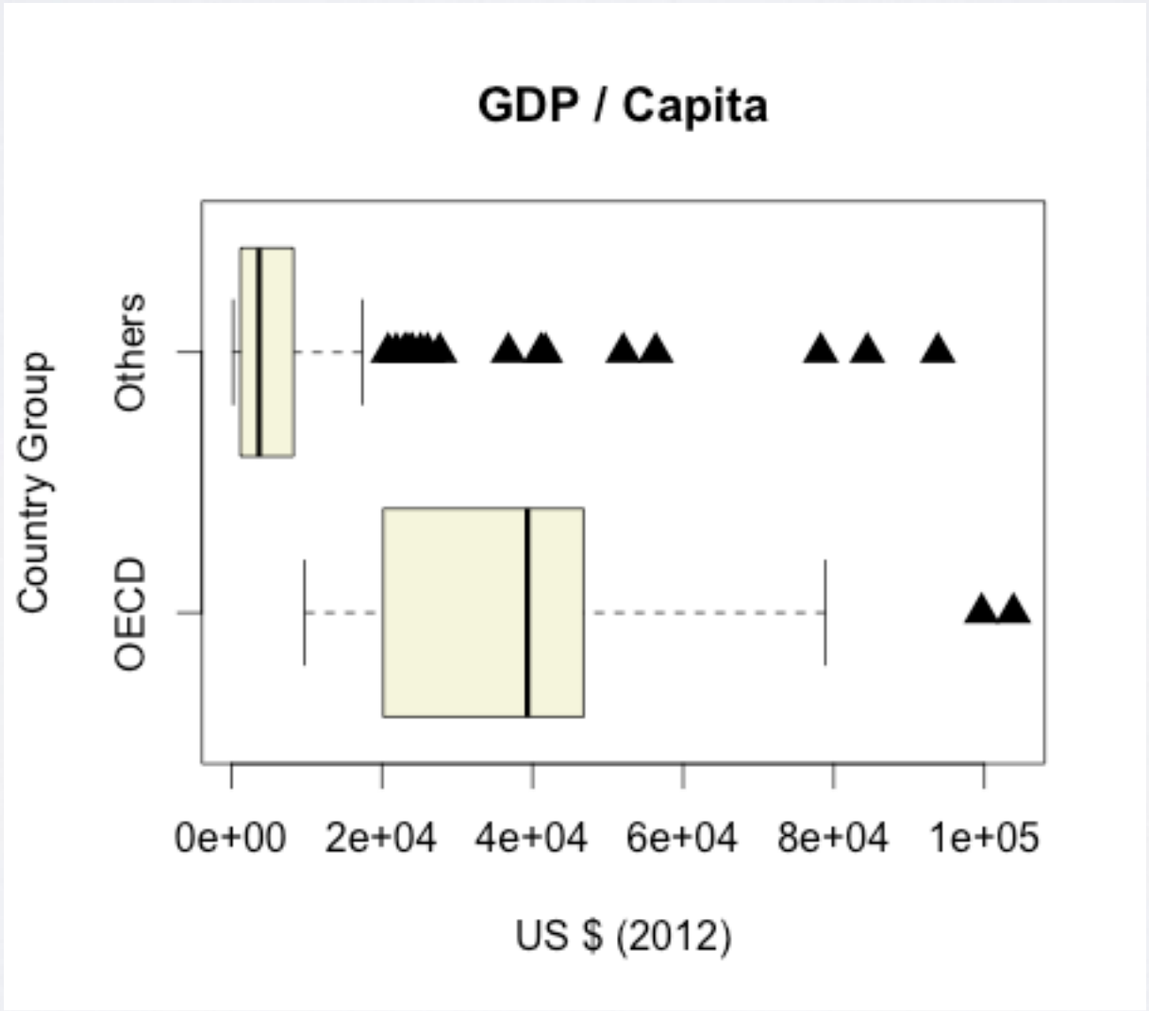
- 극단값은 pch, cex, bg 등으로 각각 심볼 모양, 크기, 색 등을 조절했고 col을 이용해서 상자의 색을 바꿨습니다. (참고로 만약 pch=1 이었다면 bg로는 색을 조절할 수 없고 col로 밖에 조절이 되지 않습니다. 그래서 상자 색이랑 심볼 색이 같아집니다. pch=17이어서 지금 상자 색과 심볼 색이 달라진 겁니다. 이건 심볼의 특성에 따라 달라지는 겁니다.)



- 그리는 방향을 수직 방향에서 수평 방향으로 바꿔 봅니다. 이렇게 하면 x-축과 y-축이 바뀌니까 그 이름도 적절히 바꿔줘야 합니다:

  ```
  > boxplot(gdplist, main="GDP / Capita", ylab="Group",
  xlab="US $ (2012)", pch=17, cex=1.5, bg="darkblue",
  col="beige", horizontal=T)
  ```

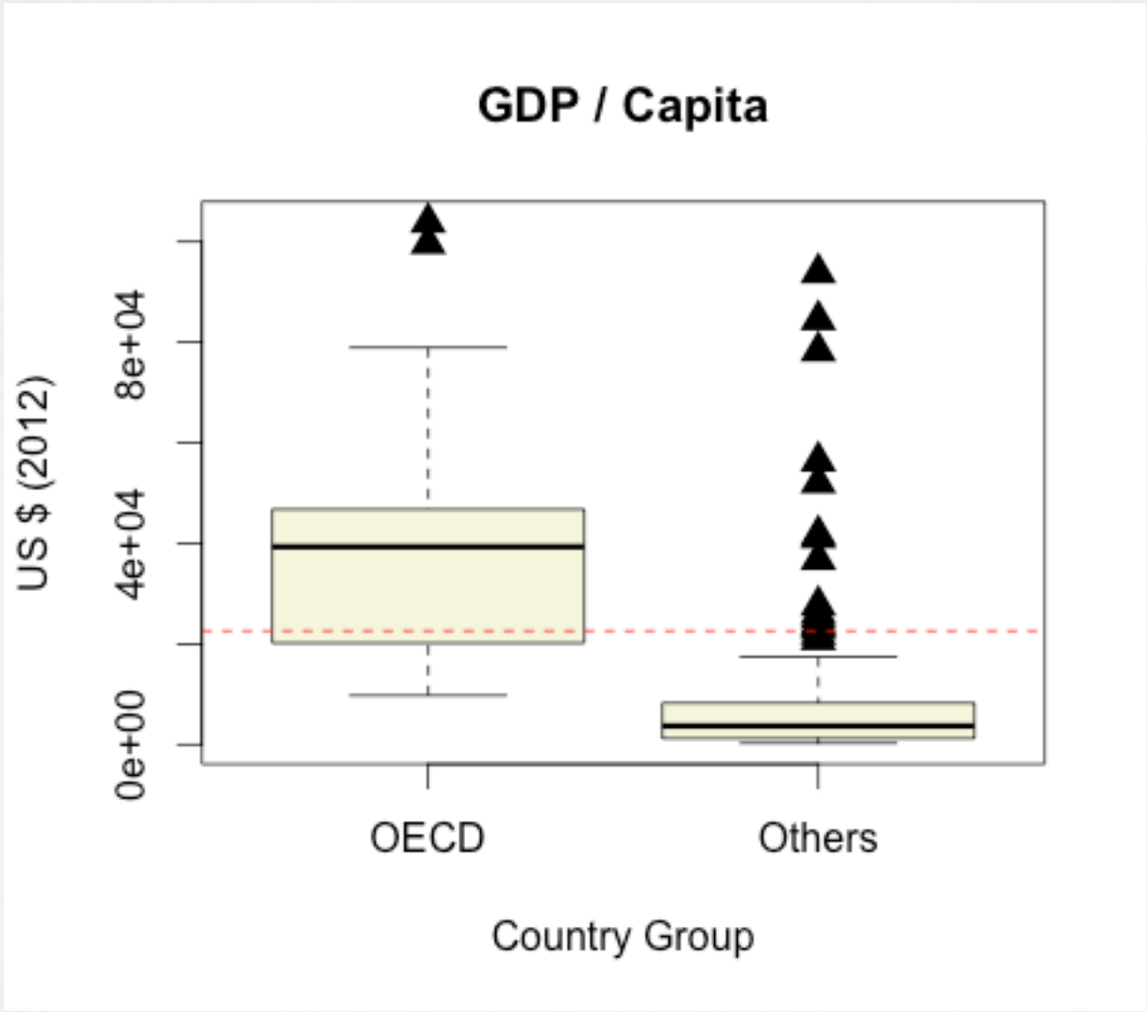
- `horizontal=T` 를 통해 그림의 방향을 수평 방향으로 바꿨습니다!



- 이제 한국의 위치는 어디쯤 되는지 표시해 보겠습니다:

```
> boxplot(gdplist, main="GDP / Capita", xlab="Group",
ylab="US $ (2012)", pch=17, cex=1.5, bg="darkblue",
col="beige")
> abline(h=GDP$GDPperCapita2012[indkor], col="red", lty=2)
```

- 수평 방향으로 선을 그으라고 `abline(h=..)` 를 썼습니다!



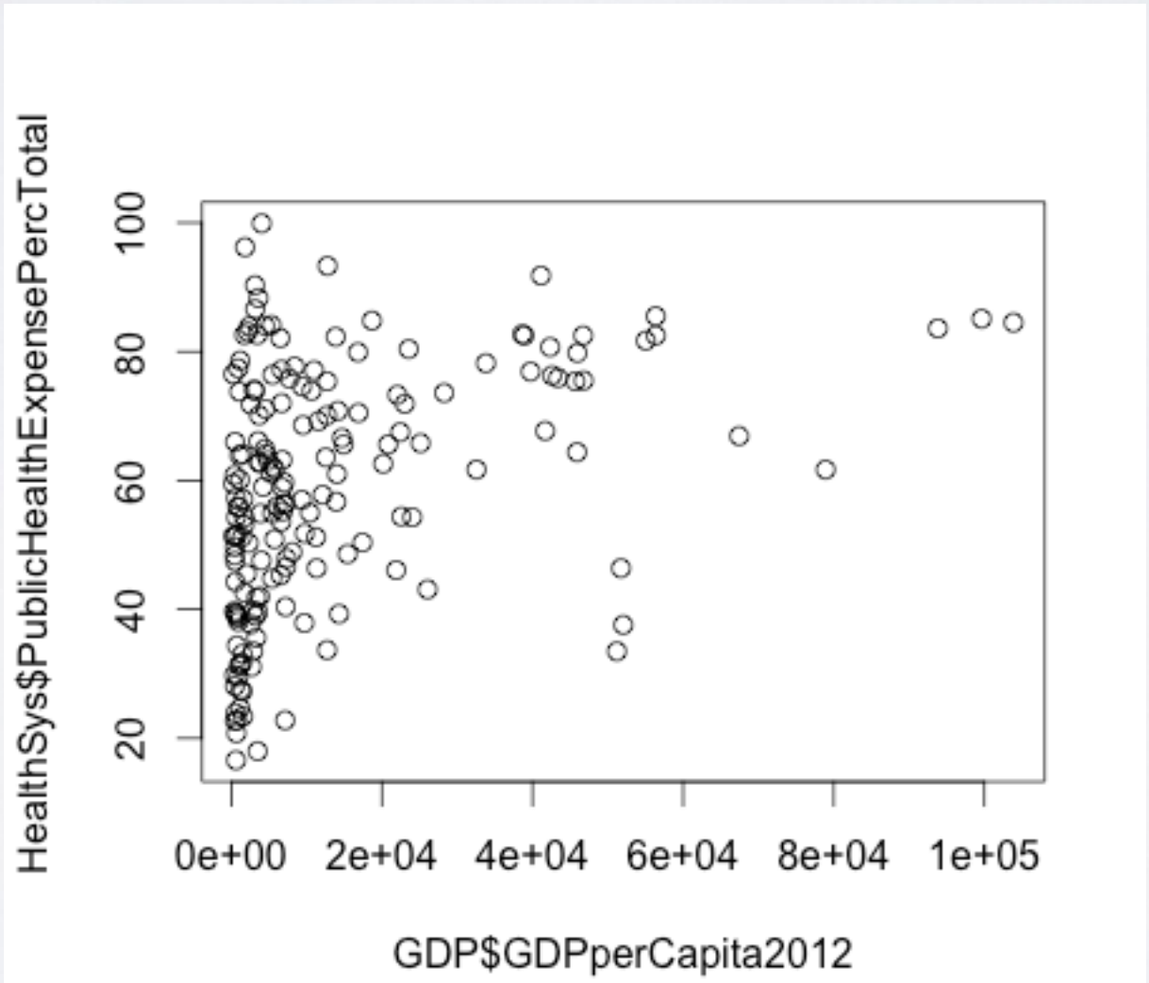
바로 저 붉은색 수평 점선이
한국의 위치입니다.
OECD에서는 낮은 수준이지만
기타 국가에 비해서는
월등히 높은 수준이군요.

제4과. 그림 그리기

(4) 2차원 산포도 한 번 더

- 그럼 마지막으로 GDP와 HealthSys 데이터를 엮어서 그림을 그려보도록 하겠습니다. 그러니까 바로 앞 강좌에서 다뤘던 내용을 그림으로 살펴보는 거죠. 기억을 되살려 보시면 그때 우린 우리나라의 일 인 당 GDP 수준과 공공부문이 담당하는 의료비용 부문을 비교했었는데요 그걸 그림으로 한 번 해보자는 거죠.
- 우선 `GDP$GDPperCapita2012`와 `HealthSys$PublicHealthExpensePercTotal`로 그림을 그려봅시다. 늘 그랬듯이 일단 가장 간단한 그림에서부터 시작합니다:

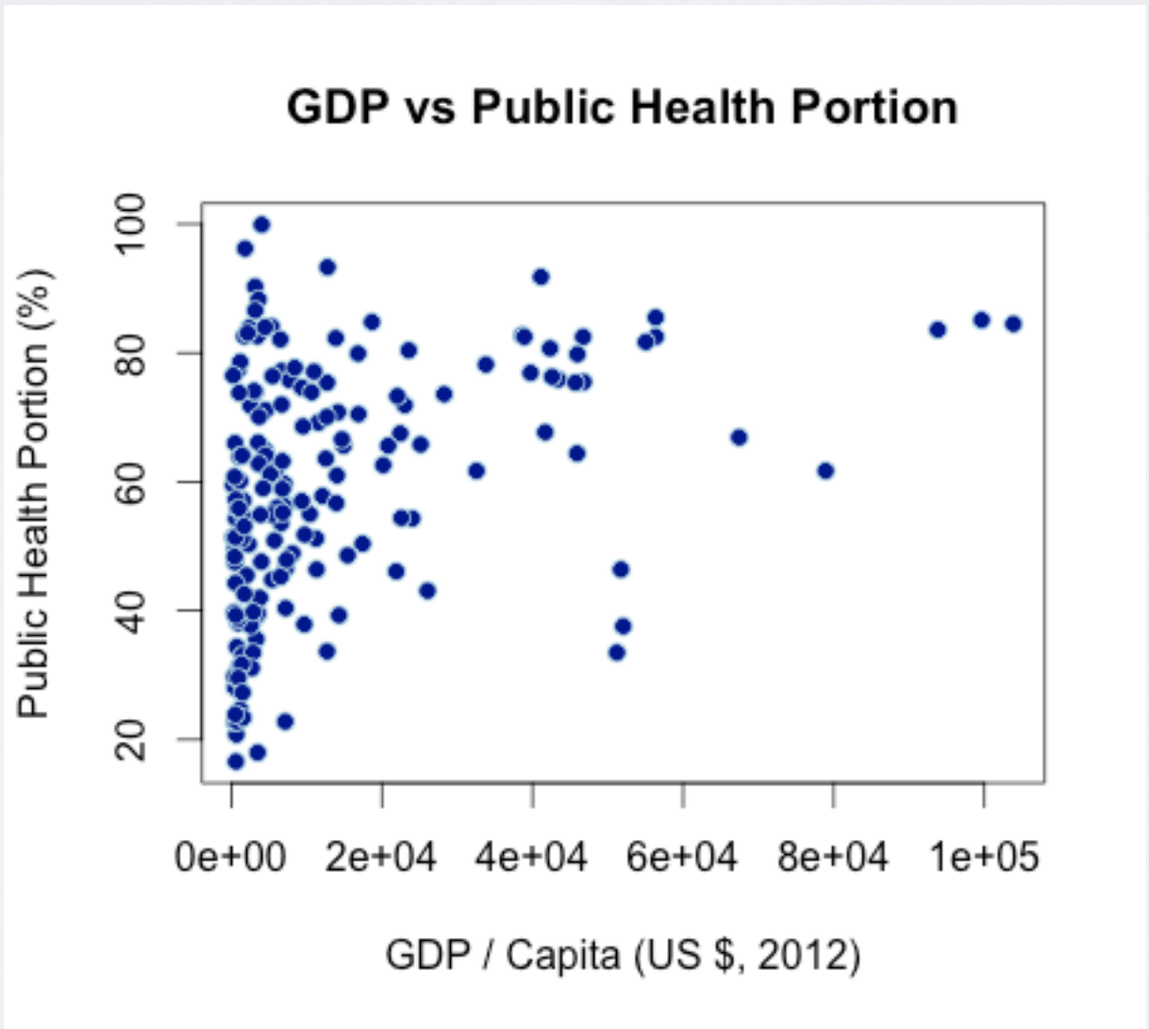
```
> plot(GDP$GDPperCapita2012, HealthSys$PublicHealthExpensePercTotal)
```



히스토그램에서 이미 봤듯이
대부분의 데이터가 GDP가
작은 영역에 집중되어 있습니다!

- 각 축과 그림의 이름을 넣어 좀더 이해가 쉽게 하고 그림의 심볼을 바꾸겠습니다:

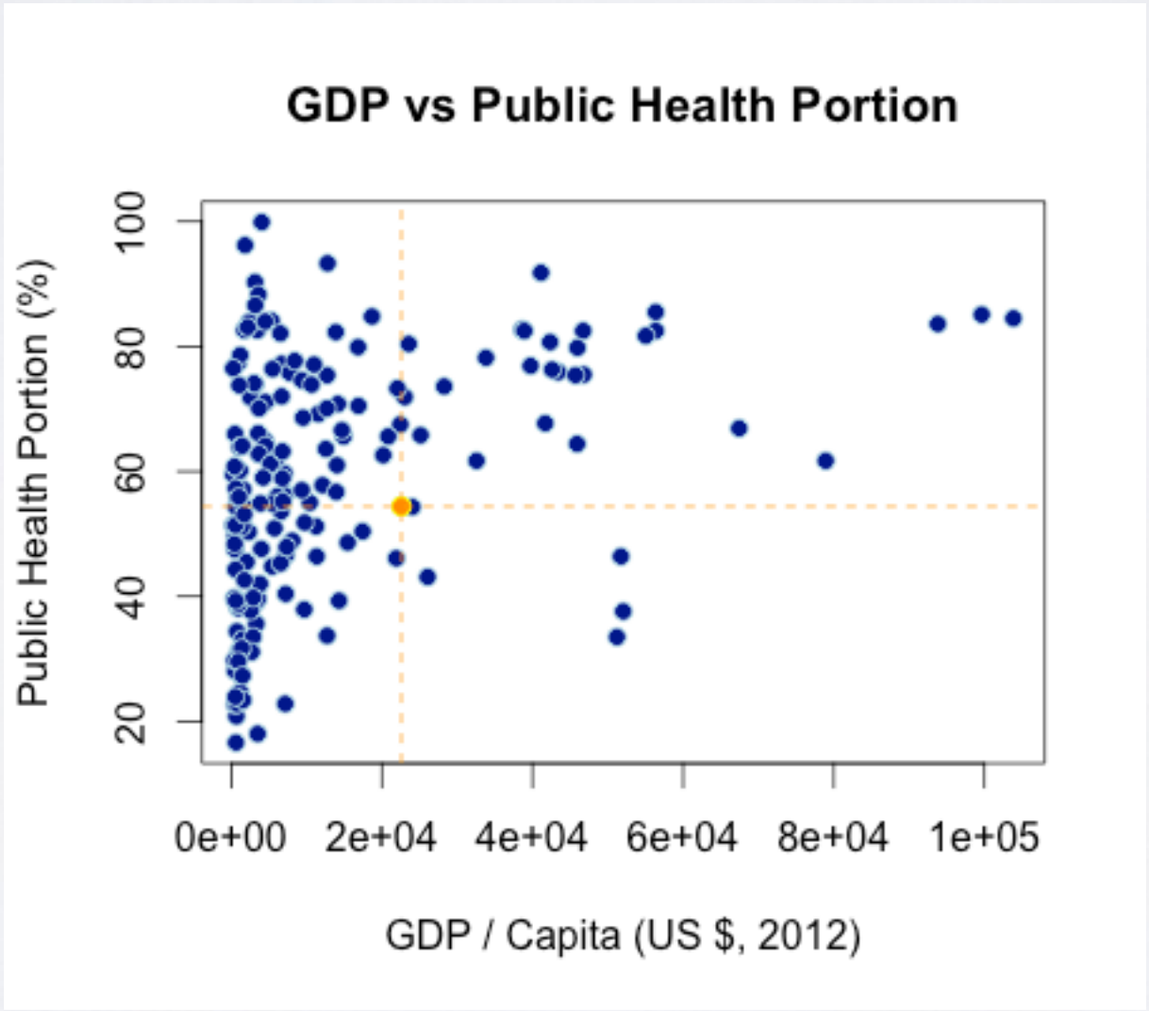
```
> plot(GDP$GDPperCapita2012, HealthSys
$PublicHealthExpensePercTotal, main="GDP vs Public Health
Portion", xlab="GDP / Capita (US $, 2012)", ylab="Public
Health Portion (%)", pch=21, bg="darkblue", col="lightblue")
```



"pch=21"를 했을 때 좋은 점은
bg와 col의 적절한 색깔 조합으로
이렇게 서로 겹쳐 있는 데이터의
앞뒤 관계가 잘 드러난다는 점입니다.

- 이제 한국의 위치를 색다르게 표시해 보겠습니다:

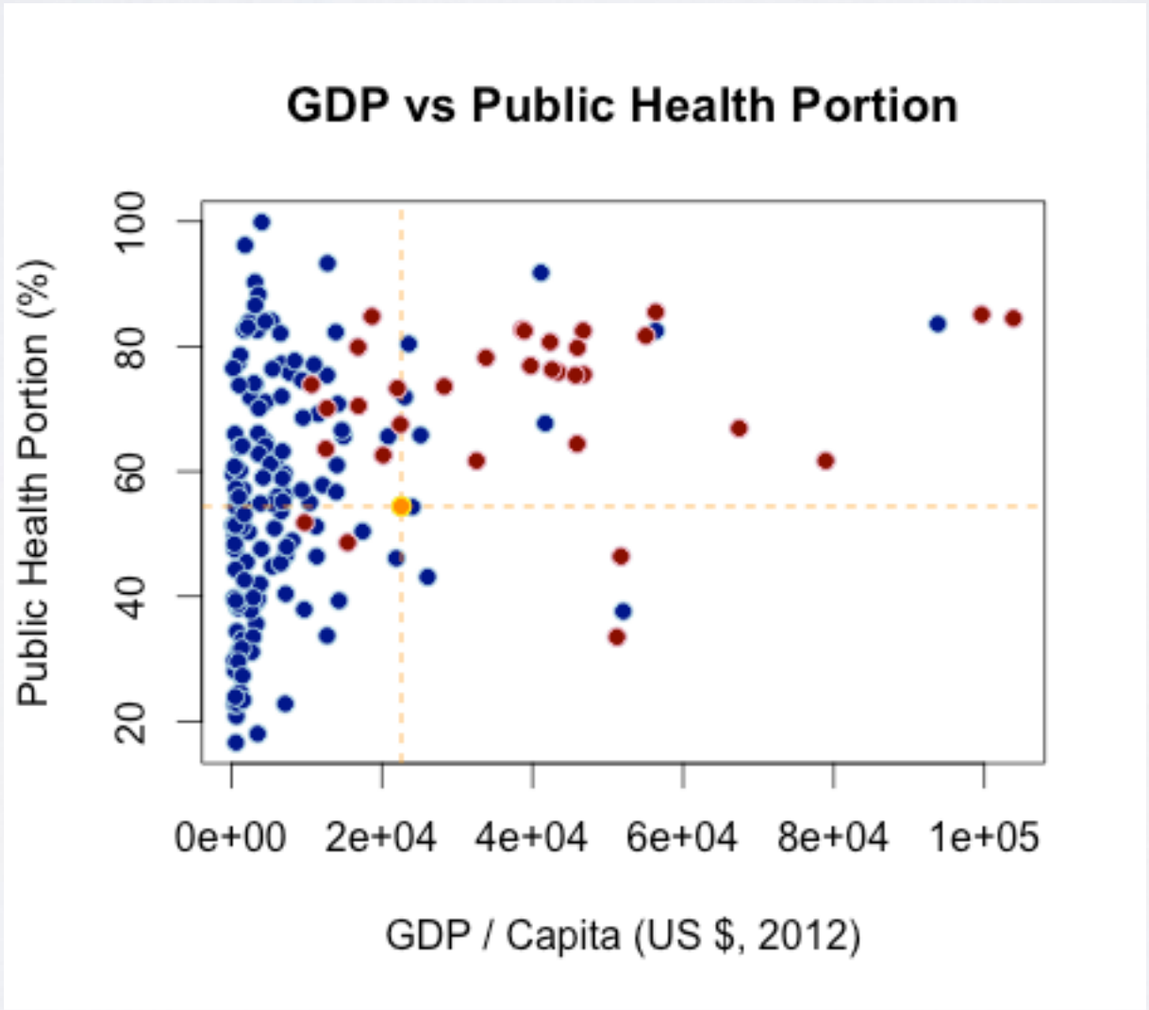
```
> plot(GDP$GDPperCapita2012, …) ## 일단 기본 그림을 그리고
> points(GDP$GDPperCapita2012[indkor],
HealthSys$PublicHealthExpensePercTotal[indkor], pch=21, bg="darkorange",
col="yellow") ## 한국의 값을 따로 표시합니다
> abline(v=GDP$GDPperCapita2012[indkor], lty=2, col="orange")
> abline(h=HealthSys$PublicHealthExpensePercTotal[indkor], lty=2,
col="orange")
```



한국의 위치는 points(…)로 두드러지게 표시하고 그 위치에 맞춰 abline(…)으로 수평, 수직 점선을 그었습니다.

- 이번엔 OECD의 위치를 표시해 보겠습니다:

```
> indoecd <- which(GDP$OECD == “Y”) ## OECD 국가의 위치
> points(GDP$GDPperCapita2012[indoecd], HealthSys
$PublicHealthExpensePercTotal[indoecd], pch=21, bg="darkred", col=“pink”)
> points(GDP$GDPperCapita2012[indkor], HealthSys
$PublicHealthExpensePercTotal[indkor], pch=21, bg="darkorange",
col=“yellow”)
```

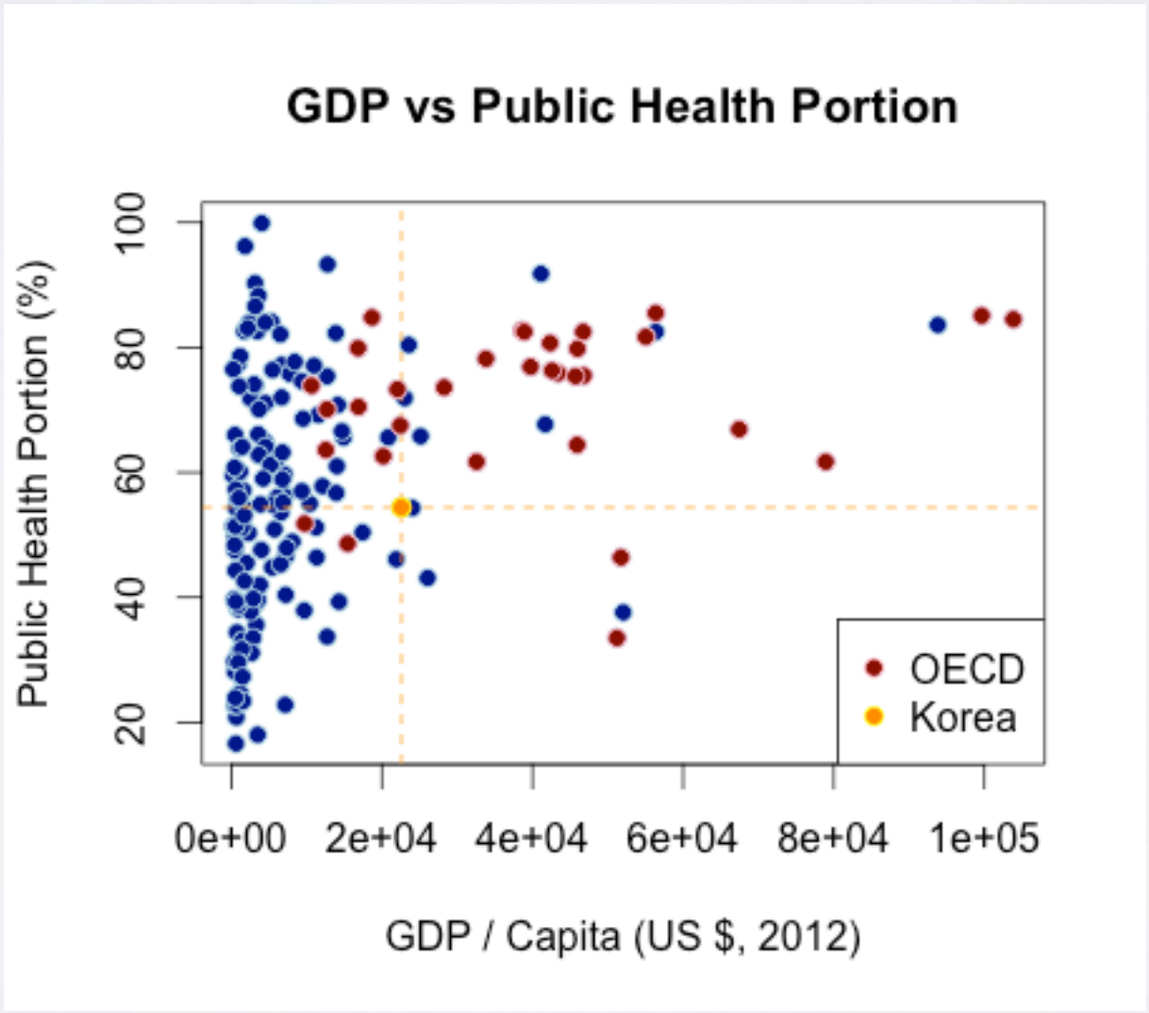


OECD의 점을 표시했더니 한국까지 덮었습니다. 그래서 한국의 점을 다시 그렸습니다! 그림을 보니 OECD 가운데 우리나라보다 공공부문의 건강관련 지출이 적은 나라는 고작 4곳 밖에 없네요!

- 마지막으로 그림에 사용된 심볼을 범례로 표시해 보겠습니다:

  > legend("bottomright", legend=c("OECD", "Korea"), pch=21, col=c("pink", "yellow"), pt.bg=c("darkred", "darkorange"))

  이걸 쓰면 아래의 그림처럼 그림의 바닥 오른쪽에 사용된 심볼을 설명하는 범례가 들어섭니다!



짜잔!!!

여기서 잠깐, 범례(LEGEND)는?

- 범례를 그리는 함수는 `legend(..)`입니다. 이 함수를 쓰는 방법은 다음과 같습니다:
  `legend(<위치>, legend=<범례의 설명문>, <기타 그림 관련 입력 변수>)`

- 위치는 원래 x, y 자표값으로 4, 10 이런 식으로 표현하는데 지금처럼 쓰면 그림의 이쪽 저쪽 구석에 딱 맞춰 범례를 나타냅니다. 이렇게 표현할 수 있는 방식은 왼쪽 위(`"topleft"`), 오른쪽 위(`"topright"`), 왼쪽 아래(`"bottomleft"`), 오른쪽 아래(`"bottomright"`) 등이 있습니다.

- 범례 설명은 설명하는 글자를 죽 묶어서 쓰는데 이때 `c(..)` 함수를 씁니다. 이 함수는 여러 값을 한데 묶어 벡터값을 만들 때 씁니다. `c("a", "b", "c")`는 문자로 된 벡터를 만들고 `c(3, 6, 9, 12, 15)`는 숫자로 된 벡터를 만듭니다.

- 그림의 심벌은 다 같았으니 따로 구분할 필요가 없습니다. (그래서 `pch=21`는 한 값만 있습니다.) 하지만 안쪽의 색과 심벌의 테두리 색은 달랐습니다. 그래서 `col=c("pink", "yellow")`로 심벌의 테두리 선을 표시하고 `pt.bg=c("darkred", "darkorange")`로 안쪽 색을 표시합니다.

- 주의할 점은 `bg="gray"` 이렇게 표현하면 이건 심벌의 안쪽 색을 결정하는 것이 아니라 범례 칸의 배경 색을 결정한다는 겁니다. 그래서 점의 안쪽 색을 결정하도록 `pt.bg`가 따로 있는 거죠.

여기서 잠깐, R에서 그림을 그리는 순서

- 이제 아시겠지만 R에서의 기본적인 그림을 그리는 패턴은 늘 이렇습니다.
  - 기본적인 그림을 `plot(..)`으로 그립니다. (x-, y-축 이름이나 그림 이름 등도 다 여기서 결정됩니다.)
  - 그 위에 선이나 다른 심볼을 덧씌웁니다. 예를 들어 선을 덧씌우려면 `lines(..)`를 심벌을 덧씌우려면 `points(..)`를 그냥 수직, 수평선을 덧씌우려면 `abline(..)`을 씁니다.
  - 마지막에 범례가 필요하다면 `legend(..)`를 써서 마무리합니다.
- 물론 이걸로 R 그림 그리는 끝은 아닙니다만 그래도 이 정도면 어느 정도 원하는 그림을 그릴 기본은 된 것이라고 볼 수 있습니다!