그래프 그리기 (1) - 제목, 축 설정

그래프는 데이터를 보기 쉽게 그림으로 표현한 것으로 데이터의 추세와 경향성이 드러나기 때문에 특징을 쉽게 이해할 수 있으며 그래프를 만드는 과정에서 새로운 패턴이 발견되기도 합니다.

#01.그래프 생성을 위한 준비

> 이 단원에서는 기본 그래프를 기준으로 다양한 옵션의 종류를 확인합니다.

1) 패키지 설치 및 참조

```
# 한국 서버를 통해 라이브러리 로드
REPO_URL <- "https://cran.seoul.go.kr/"

# 그래프 패키지
if (!require(ggplot2)) install.packages("ggplot2", repos=REPO_URL)
library(ggplot2)

# 폰트 설정 패키지
if (!require(extrafont)) install.packages("extrafont", repos=REPO_URL)
library(extrafont)
```

▶ 출력결과

```
Loading required package: ggplot2
Loading required package: extrafont
Registering fonts with R
```

2) 한글 사용을 위한 폰트 로드

아래 구문 실행시 y/n를 묻는 화면이 표시된다. y를 입력해야 실행된다.

시간이 다소 오래 소요되며 장치별로 1회만 수행하면 된다.

나눔고딕 검색하기

```
font_import(pattern = 'NanumGothic')
```

▶ 출력결과

```
Importing fonts may take a few minutes, depending on the number of fonts and the speed of the system.
Continue? [y/n]  y

Scanning ttf files in C:\WINDOWS\Fonts ...
Extracting .afm files from .ttf files...
C:\Windows\Fonts\NanumGothic.ttf => D:/leekh/R-3.6.1/library/extrafontdb/metrics/NanumGothic
C:\Windows\Fonts\NanumGothic_0.ttf => D:/leekh/R-3.6.1/library/extrafontdb/metrics/NanumGothic_0
C:\Windows\Fonts\NanumGothicBold.ttf => D:/leekh/R-3.6.1/library/extrafontdb/metrics/NanumGothicBold
C:\Windows\Fonts\NanumGothicBold_0.ttf => D:/leekh/R-3.6.1/library/extrafontdb/metrics/NanumGothicBold_0
C:\Windows\Fonts\NanumGothicExtraBold.ttf => D:/leekh/R-3.6.1/library/extrafontdb/metrics/NanumGothicExtraB
C:\Windows\Fonts\NanumGothicExtraBold_0.ttf => D:/leekh/R-3.6.1/library/extrafontdb/metrics/NanumGothicExtr
C:\Windows\Fonts\NanumGothicLight.ttf => D:/leekh/R-3.6.1/library/extrafontdb/metrics/NanumGothicLight
C:\Windows\Fonts\NanumGothicLight_0.ttf => D:/leekh/R-3.6.1/library/extrafontdb/metrics/NanumGothicLight_0
Found FontName for 8 fonts.
Scanning afm files in D:/leekh/R-3.6.1/library/extrafontdb/metrics
Writing font table in D:/leekh/R-3.6.1/library/extrafontdb/fontmap/fonttable.csv
Writing Fontmap to D:/leekh/R-3.6.1/library/extrafontdb/fontmap/Fontmap...
```

설치된 폰트 목록 확인

fonttable() 함수를 통해 반환받는 DataFrame에서 FamilyName 컬럼을 확인하면 R 소스코드에 적용해야 할 폰트 이름을 확인할 수 있다.

출력결과는 시스템에 설치되어 있는 글꼴의 상태에 따라 다를 수 있다.

```
ftable <- fonttable()
ftable$FamilyName
```

▶ 출력결과

1. 'NanumGothic'
2. 'NanumGothic'
3. 'NanumGothic'
4. 'NanumGothic'
5. 'NanumGothic'
6. 'NanumGothic'
7. 'NanumGothic'
8. 'NanumGothic'
9. 'NanumGothicExtraBold'
10. 'NanumGothicExtraBold'
11. 'NanumGothicExtraBold'
12. 'NanumGothicExtraBold'
13. 'NanumGothic Light'
14. 'NanumGothic Light'
15. 'NanumGothic Light'
16. 'NanumGothic Light'

설치된 폰트들 로드하기

```
# mac의 경우 `device="win"` 생략
loadfonts(device="win")
#loadfonts()
```

▶ 출력결과

```
Registering font with R using windowsFonts(): NanumGothic
Registering font with R using windowsFonts(): NanumGothicExtraBold
Registering font with R using windowsFonts(): NanumGothic Light
```

3) 샘플 데이터

```
# 그래프를 표현할 데이터 프레임 구성
df <- data.frame(A=c(0, 1, 2, 3, 4), B=c(0, 1, 2, 3, 4), C=c(0, 2, 4, 6, 8))
df
```

▶ 출력결과

A data.frame: 5 × 3

A	B	C
<dbl>	<dbl>	<dbl>
0	0	0
1	1	2
2	2	4

A	B	C
<dbl>	<dbl>	<dbl>
3	3	6
4	4	8

#04. 그래프 생성하기

```
ggplot(data=데이터프레임) + 그래프종류함수(x축,y축) + 옵션함수1() + 옵션함수2() ...
```

> 이 예제에서는 종류는 선그래프로 제한하고 옵션들의 종류를 확인합니다.

1) 그래프 기본 크기 및 불필요한 경고 메시지 끄기

```
options(repr.plot.width=5, repr.plot.height=5, warn=-1)
```

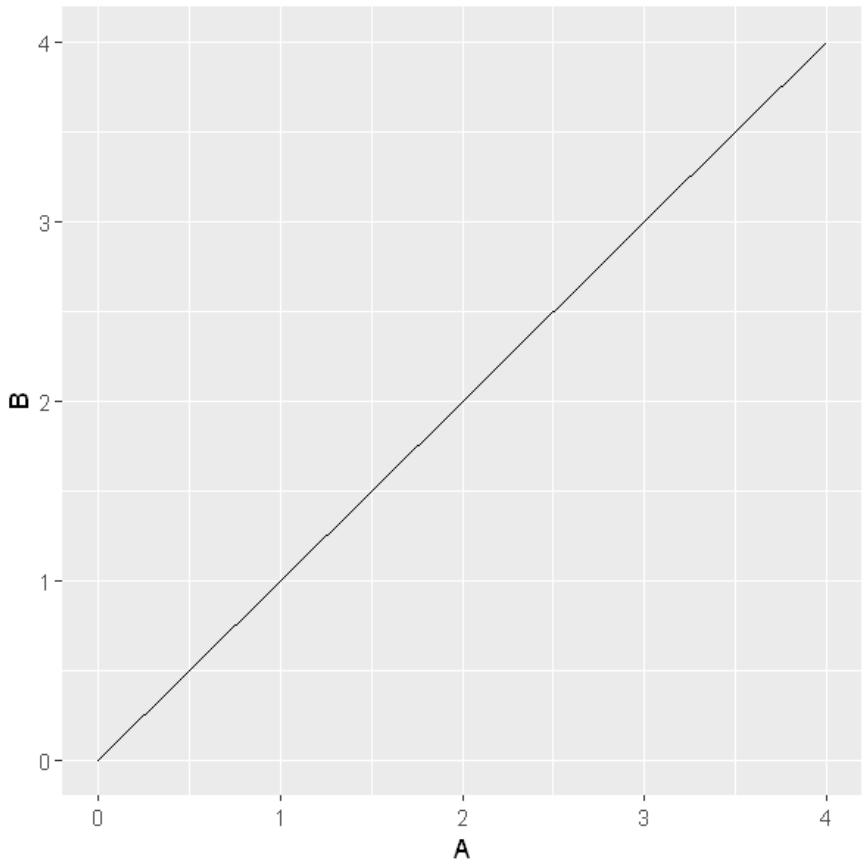
2) 기본 선 그래프

`geom_line(aes(x=x축컬럼, y=축컬럼))` 함수는 선 그래프를 생성한다.

직접 출력하기

```
ggplot(data=df) + geom_line(aes(x=A, y=B))
```

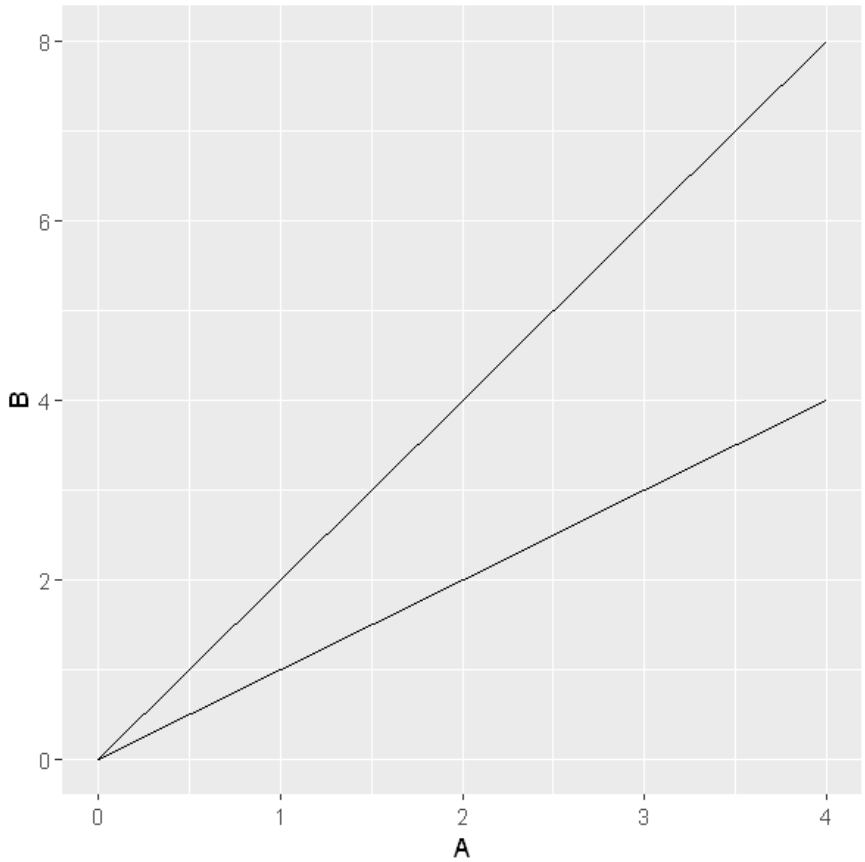
▶ 출력결과



두 개의 선 그래프를 한 번에 표시하기

```
ggplot(data=df) + geom_line(aes(x=A, y=B)) +
                  geom_line(aes(x=A, y=C))
```

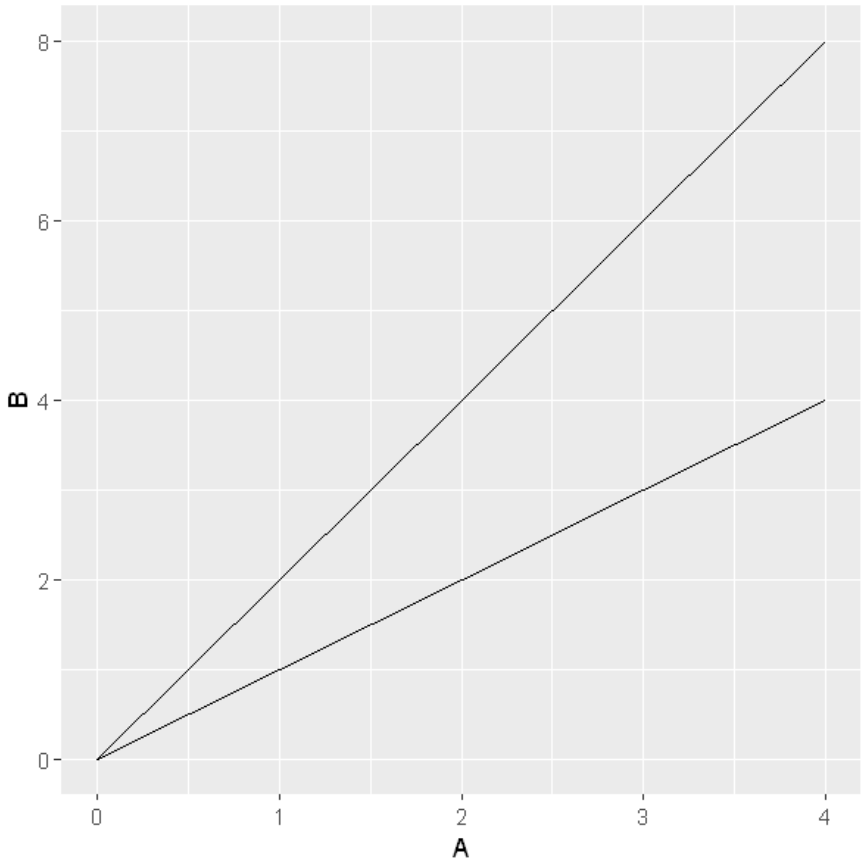
▶ 출력결과



다른 변수에 그래프를 할당 후 화면에 표시하기

```
graph <- ggplot(data=df) + geom_line(aes(x=A, y=B)) +
                           geom_line(aes(x=A, y=C))
graph
```

▶ 출력결과



#05. 그래프 옵션 설정

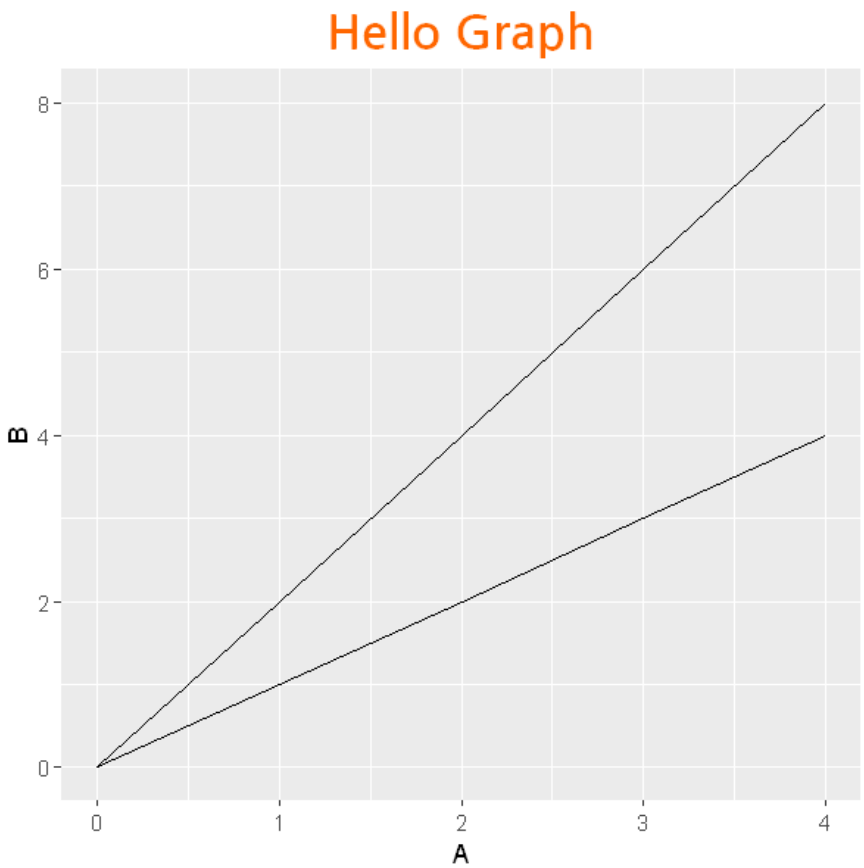
1) 타이틀 입력하고 꾸미기

`ggtitle()` 함수로 그래프의 제목을 설정할 수 있다.

`theme()` 함수를 통해 제목에 대한 글꼴, 모양, 크기, 색상 등을 설정할 수 있다.

```
graph + ggtitle("Hello Graph") +                # title문자열 설정 (줄바꿈은 \n 으로 설정)
  theme(plot.title = element_text(
                           family="NanumGothic", # 글꼴이름
                           face="bold",          # bold=굵게, italic=기울임
                           hjust=0.5,            # 0=왼쪽, 0.5=가운데, 1=오른쪽
                           vjust=1,              # 기본위치 0을 기준으로 0보다 작으면 아래쪽, 0보다 크면 위쪽으로 이동
                           size=20,              # 글자크기
                           color="#ff6600"))     # 글자색상
```

▶ 출력결과



2) 축(Axis) 설정

축 제목, 모양 설정하기

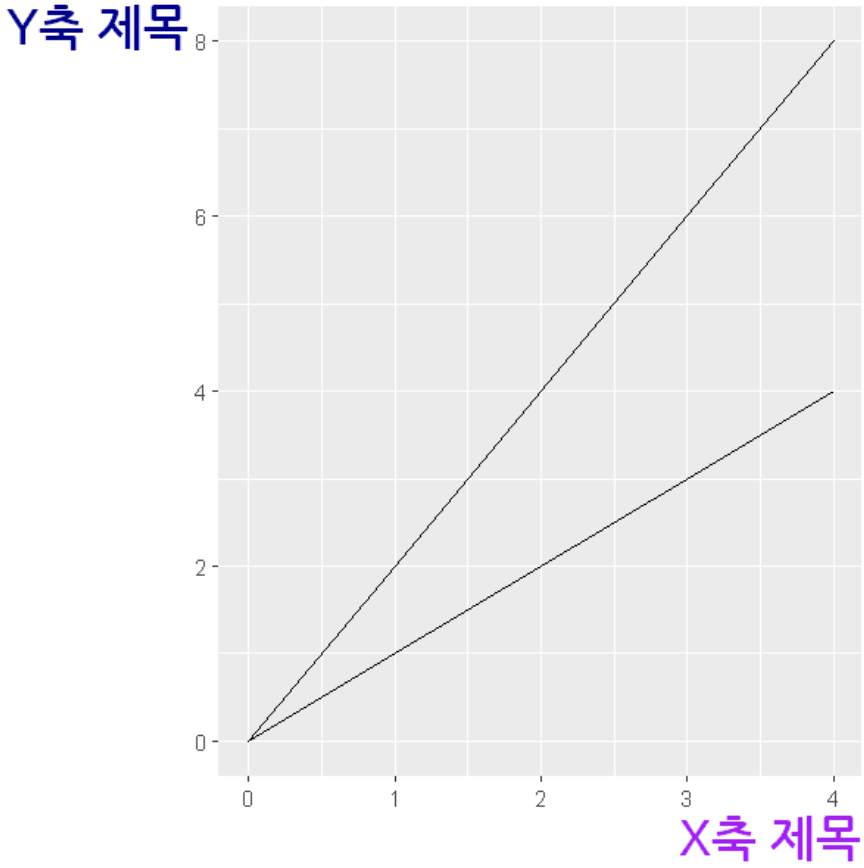
`xlab()`, `ylab()` 함수로 축의 제목을 설정할 수 있다. 설정하지 않을 경우 컬럼의 이름이 기본값으로 사용된다. `theme()` 함수를 통해 x축 제목과 y축 제목에 대한 글꼴, 모양, 크기, 색상 등을 설정할 수 있다.

angle 속성과 hjust, vjust 속성의 관계

축	angle	hjust/vjust
x	0	hjust : 0=왼쪽정렬, 0.5=중앙정렬, 1=우측정렬
x	90	vjust : 1=왼쪽정렬, 0.5=중앙정렬, 0=우측정렬
y	0	vjust : 0=하단정렬, 0.5=중앙정렬, 1=상단정렬
x	90	hjust : 0=하단정렬, 0.5=중앙정렬, 1=상단정렬

```
graph + xlab("X축 제목") +
        ylab("Y축 제목") +
        theme(axis.title.x=element_text(family="NanumGothic",  # 글꼴이름
                                        face="bold",           # "bold" or "italic"
                                        hjust=1,               # 0=왼쪽정렬, 0.5=중앙정렬, 1=우측정렬
                                        size=20,               # 글자 크기
                                        color="purple",        # 글자 색상
                                        angle=0),              # 글자 회전 (0~360)
              axis.title.y=element_text(family="NanumGothic",
                                        face="bold",
                                        vjust=1,               # 0=하단정렬, 0.5=중앙정렬, 1=상단정렬
                                        size=20,
                                        color="darkblue",
                                        angle=0))
```

▶ 출력결과



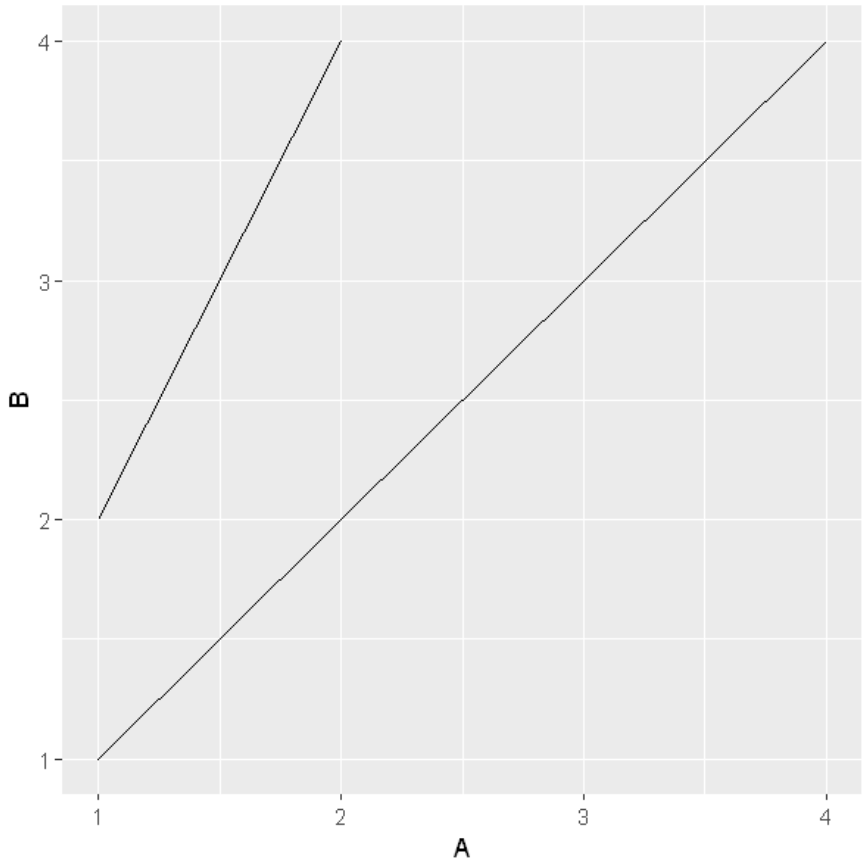
축 표현 범위 설정하기

`lims()` 함수를 사용하여 제한하고자 하는 축의 시작과 끝 범위를 입력한다.

이 함수를 사용할 때에는 데이터를 표현하는 범위를 제한하기 때문에 범위에 속하지 않는 데이터가 제거된다는 경고 메시지가 뜰 수 있다.

```
graph + lims(x=c(1, 4), y=c(1, 4))
```

▶ 출력결과



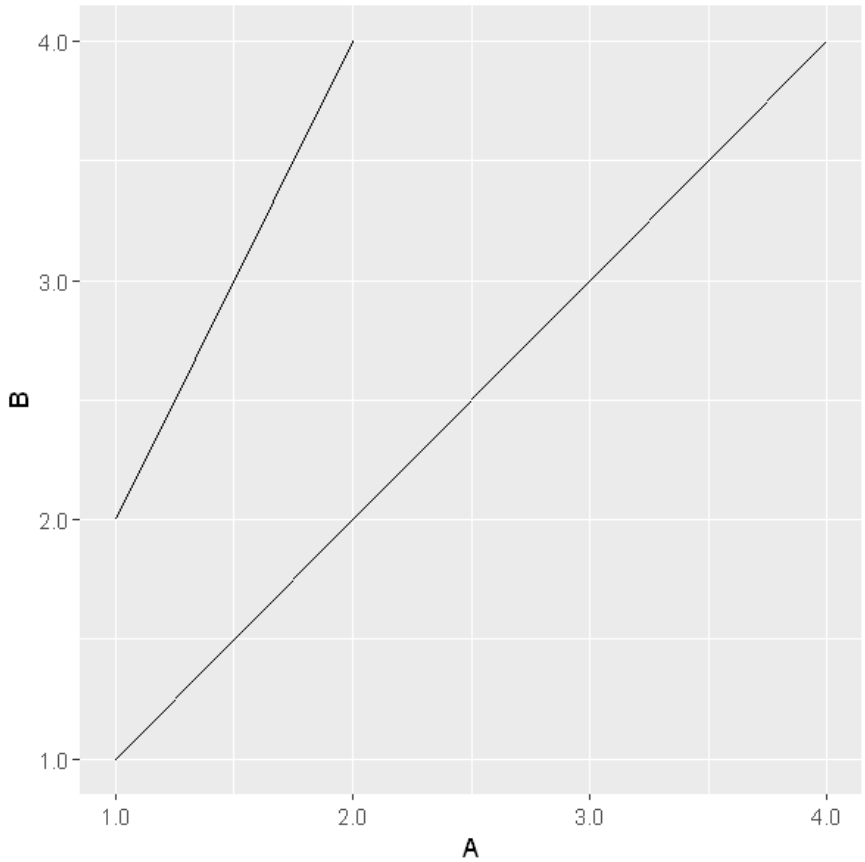
축 범위를 설정하는 또 다른 방법

`scale_x_continuous()` 함수나 `scale_y_continuous()` 함수에 limits 파라미터를 사용해도 축의 범위를 설정할 수 있다.

만약 숫자값이 너무 커서 지수형태로 표시된다면 `labels=scales::comma` 파라미터를 사용하여 정수형태로 변경할 수 있다.

```
graph + scale_x_continuous(limits=c(1, 4), labels=scales::comma) +
        scale_y_continuous(limits=c(1, 4), labels=scales::comma)
```

▶ 출력결과



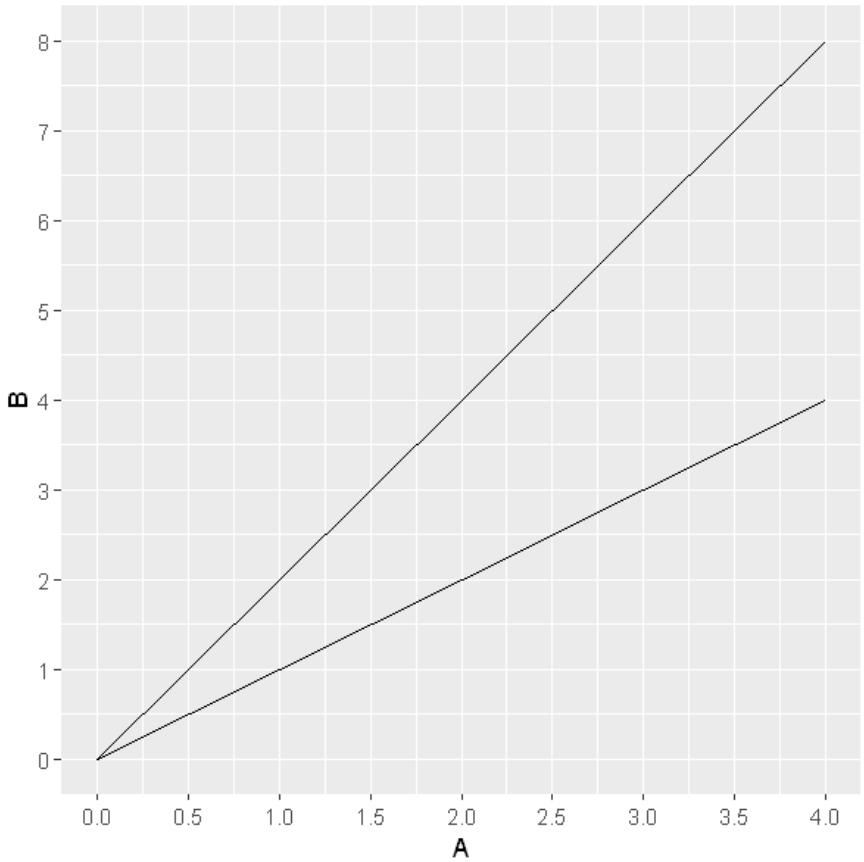
축 간격 조절

그래프의 x,y축에 대한 좌표 간격은 ggplot2 라이브러리가 임의로 설정하게 된다.

만약 축에 대한 간격을 조절해야 할 필요가 있다면 `scale_x_continuous()` 함수나 `scale_y_continuous()` 함수에 breaks 파라미터를 사용한다.

```
graph + scale_x_continuous(breaks=c(0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5)) + # x축 간격 조절
        scale_y_continuous(breaks=c(0, 1, 2, 3, 4, 5, 6, 7, 8))   # y축 간격 조절
```

▶ 출력결과

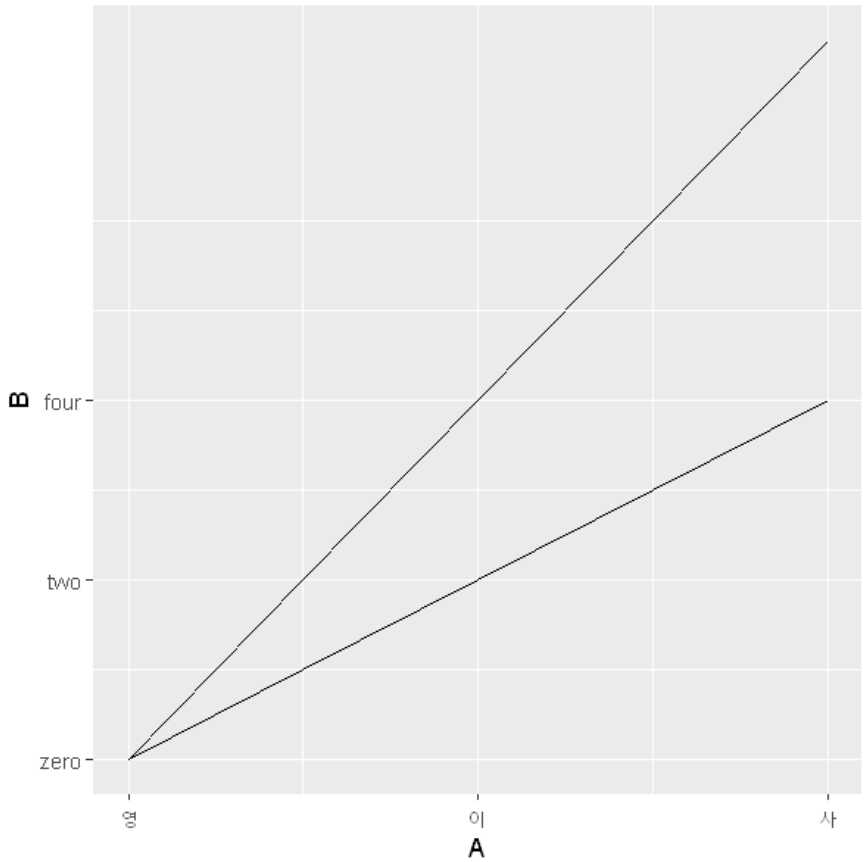


축 라벨 변경하기

`scale_x_continuous()`, `scale_y_continuous()` 함수에 labels 파라미터를 사용하면 라벨의 텍스트를 변경할 수 있다.

```
graph + scale_x_continuous(breaks=c(0, 2, 4), labels=c('영','이','사')) + # x축 간격 조절
        scale_y_continuous(breaks=c(0, 2, 4), labels=c('zero','two','four'))   # y축 간격 조절
```

▶ 출력결과



축 위치 변경하기

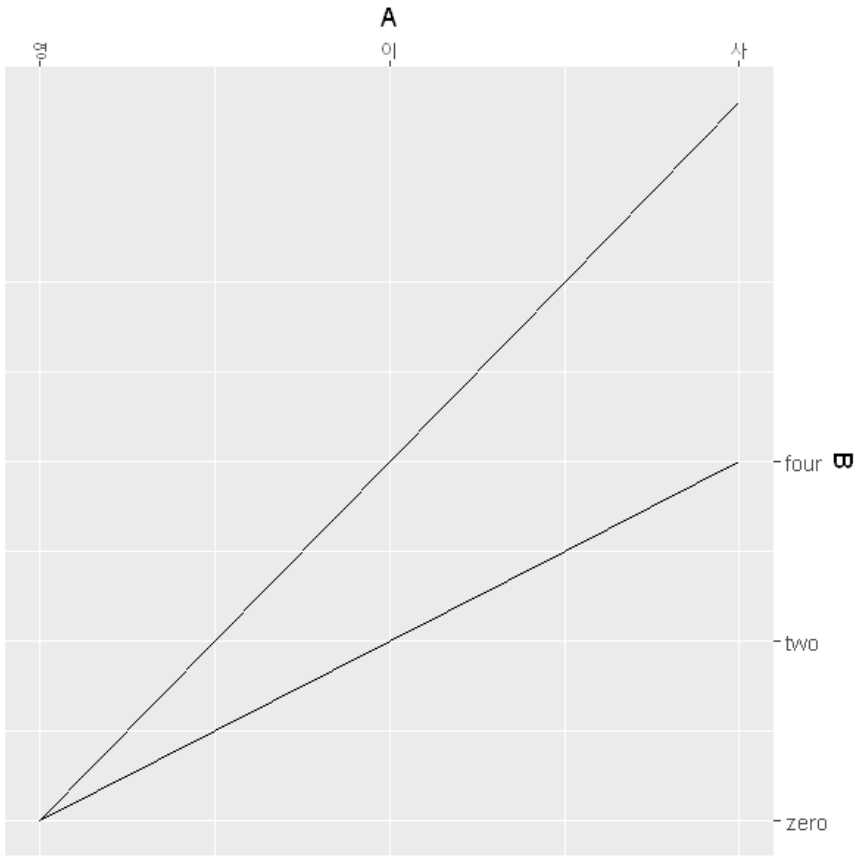
기본 그래프에서 X축은 아래, Y축은 좌측에 위치한다.

이러한 축의 위치는 X축은 위로, Y축은 우측으로 변경이 가능하다.

라벨의 위치를 바꿀 때도 라벨이 속한 축에 따라 `scale_x_continuous()` 함수나 `scale_y_continuous()` 함수에 position 파라미터를 통해 라벨을 원하는 위치로 변경할 수 있다.

```
graph + scale_x_continuous(breaks=c(0, 2, 4), labels=c('영','이','사'), position = "top") +
        scale_y_continuous(breaks=c(0, 2, 4), labels=c('zero','two','four'), position = "right")
```

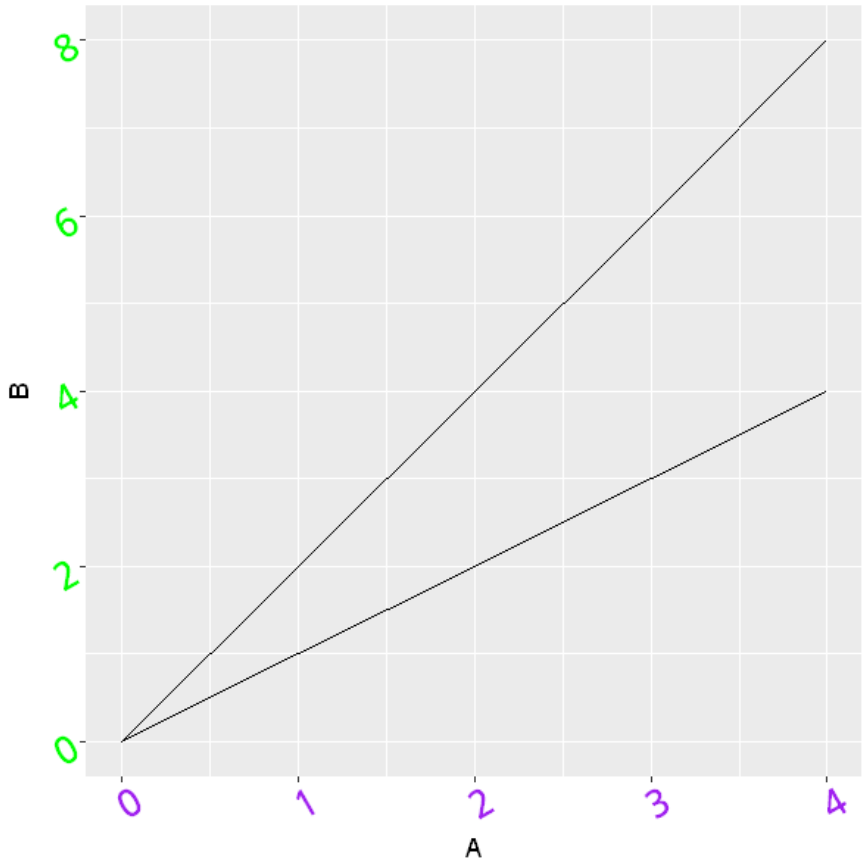
▶ 출력결과



축 텍스트 꾸미기

```
graph + theme(axis.text.x=element_text(family="NanumGothic",  # 글꼴이름
                                       face="bold",           # "bold" or "italic"
                                       size=15,               # 글자 크기
                                       color="purple",        # 글자 색상
                                       angle=30),             # 글자 회전 (0~360)
              axis.text.y=element_text(family="NanumGothic",  # 글꼴이름
                                       face="bold",           # "bold" or "italic"
                                       size=15,               # 글자 크기
                                       color="green",         # 글자 색상
                                       angle=30))             # 글자 회전 (0~360)
```

▶ 출력결과



3) 그래프 안에 글자/선/도형 넣기

`annotate()` 함수를 통해 그래프 안에 글자나 선, 사각형 등의 추가가 가능하다.

한 번에 여러 개의 항목을 추가할 수도 있으며 도형 위에 글자나 선을 넣을 수 도 있다.

텍스트 넣기 구문형식

```
그래프객체 + annotate("text", x=x좌표, y=좌표, label=표시할문자열)
```

사각형 넣기

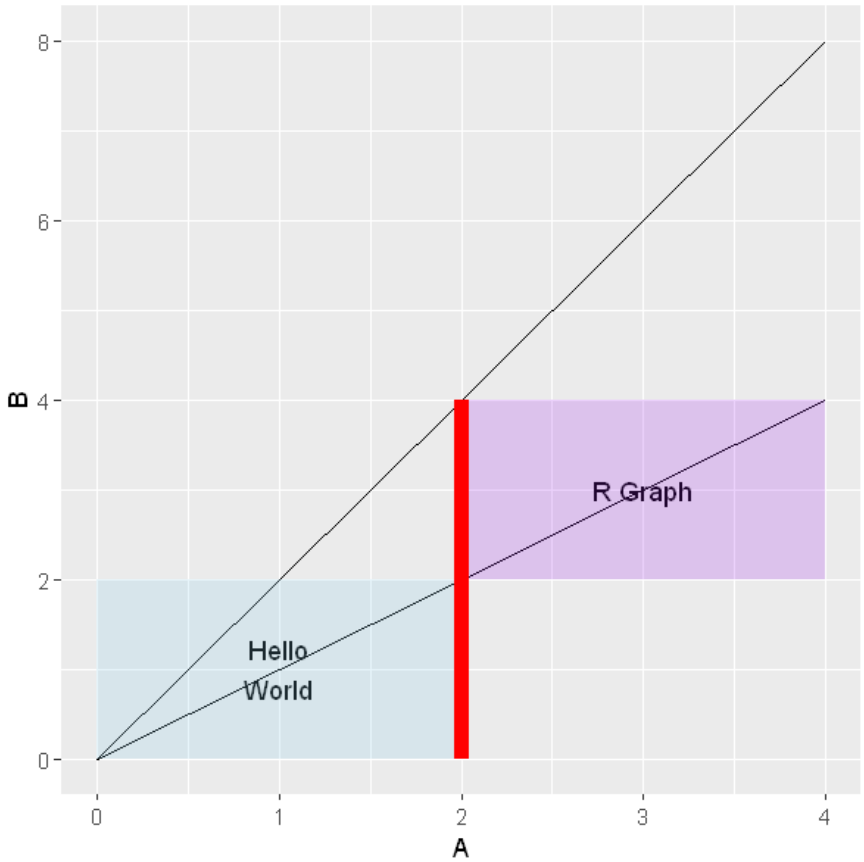
```
그래프객체 + annotate("rect", xmin=x시작좌표, ymin=y시작좌표, xmax=x끝좌표, ymax=y끝좌표, alpha=투명도, fill=색상)
```

선 긋기

```
그래프객체 + annotate("segment", x=x시작좌표, y=y시작좌표, xend=x끝좌표, yend=y끝좌표, colour=색상, size=굵기)
```

```
graph + annotate("text", x=1, y=1, label="Hello\nWorld") +
        annotate("text", x=3, y=3, label="R Graph") +
        annotate("rect", xmin=0, ymin=0, xmax=2, ymax=2, alpha=0.2, fill="skyblue") +
        annotate("rect", xmin=2, ymin=2, xmax=4, ymax=4, alpha=0.2, fill="purple") +
        annotate("segment", x=2, y=0, xend=2, yend=4, colour="red", size=3)
```

▶ 출력결과

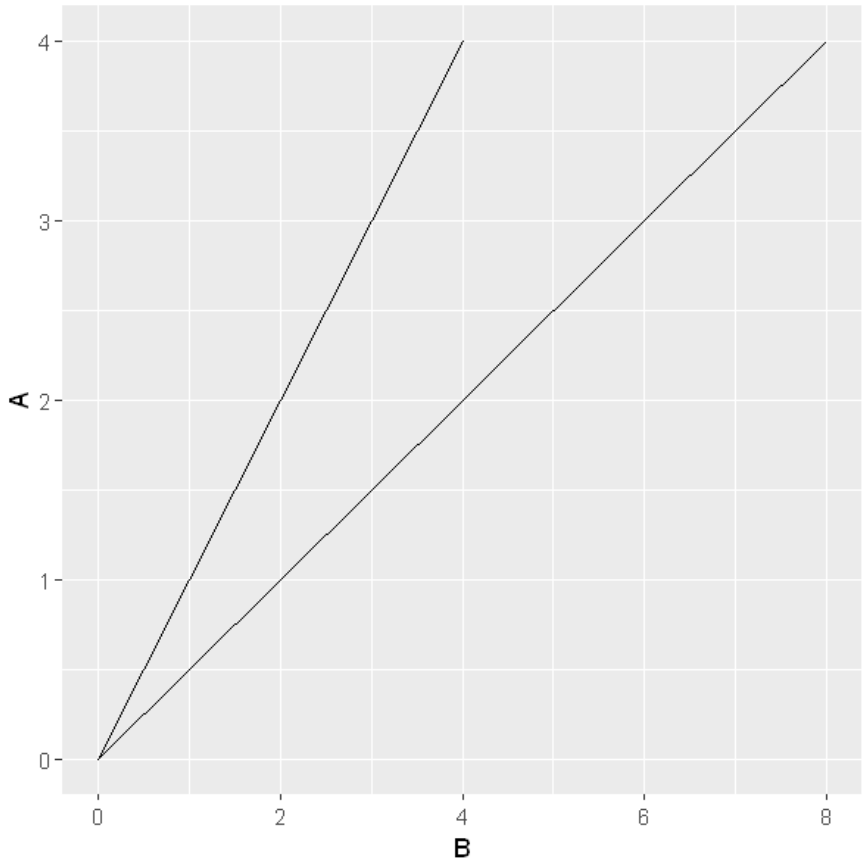


4) 그래프 X축, Y축 바꿔 그리기

coord_flip() 함수를 사용하면 뒤집어 그리기 위해 일부러 다른 형태의 그래프를 그릴 필요 없이 기존의 그래프에서 X, Y축만 뒤집어 그릴 수 있다.

```
graph + coord_flip()
```

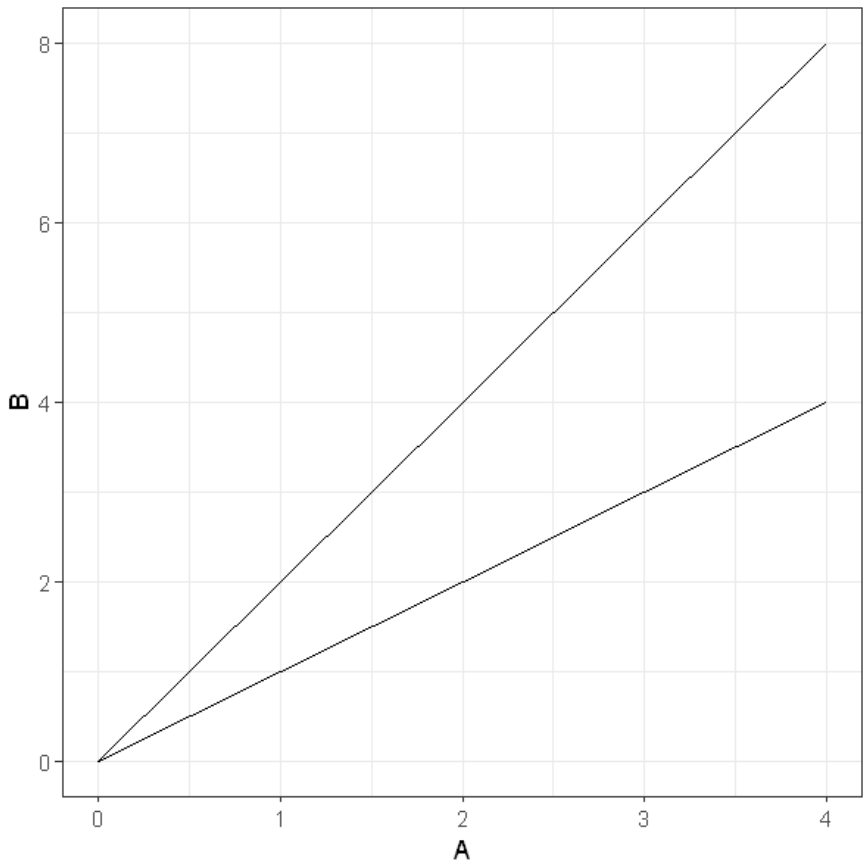
▶ 출력결과



5) 배경을 흰색으로 표시하기

```
graph + theme_bw()
```

▶ 출력결과



6) 그래프를 이미지로 저장하기

```
ggsave(파일이름, 그래프객체 [, scale=비례크기_실수값])
```

```
ggsave("hello.png", graph, scale=1.5)
```

▶ 출력결과

```
Saving 10 x 10 in image
```