

# LaTeX 소개

David Diez, OpenIntro.org  
번역: 이광춘, xwMOOC

## 차례

<b>제 1 절 일반</b>	<b>3</b>
1.1 표제 절(Header section) . . . . .	3
1.2 개요 (Outline) . . . . .	3
1.3 문단 (Paragraphs) . . . . .	3
1.4 공백 (Spacing) . . . . .	4
1.5 텍스트 (Text) . . . . .	4
1.6 매크로 (Macros) . . . . .	4
1.7 글머리표 (Lists) . . . . .	4
1.8 특수 문자 (Special characters) . . . . .	5
1.9 글자 그대로 (verbatim) . . . . .	5
<b>제 2 절 템 (Tabbing)</b>	<b>5</b>
2.1 태깅 기초(Basic tabbing) . . . . .	5
2.2 즉흥 템 (Impromptu tabs) . . . . .	6
2.3 태깅 예제 (Tabbing example) . . . . .	6
<b>제 3 절 표 (Tables)</b>	<b>7</b>
3.1 표 기초(Basic tables) . . . . .	7
3.2 캡션과 참조 (Captions and referencing) . . . . .	7
3.3 array 환경 . . . . .	8
3.4 R 팩키지, xtable . . . . .	8
<b>제 4 절 그림 (Figures)</b>	<b>9</b>
4.1 그림 기초 (Basic figures) . . . . .	9
4.2 캡션과 참조하기 (Captions and referencing) . . . . .	9
4.3 체계적으로 정리하기 (Keeping organized) . . . . .	10

<b>제 5 절 수학 (Math)</b>	<b>11</b>
5.1 텍스트 내부 수학 (Math in text) . . . . .	11
5.2 방정식 환경과 참조 (Equation environment and referencing) . . . . .	12
5.3 줄 맞추기 (Aligning) . . . . .	12
5.4 배열 (Arrays) . . . . .	12
5.5 amsmath 팩키지 사용 시 몇가지 장점 . . . . .	13
<b>제 6 절 실습 (Practice)</b>	<b>13</b>
6.1 시도해 보자 #1 . . . . .	13
6.2 시도해 보자 #2 . . . . .	13
6.3 시도해 보자 #3 . . . . .	14
<b>제 7 절 참고문헌 관련</b>	<b>14</b>

텍스트를 제어하는 방법에 대해 본 `latex-intro-kr.tex` 및 `latex-intro-kr.pdf`를 읽어 친숙해지고, 함께 팔려오는 발표 슬라이드를 미리 읽어보기 강력 추천한다.

## 제 1 절 일반

### 1.1 표제 절(Header section)

`\title`, `\author`, `\date` 을 예외로 하고, LaTeX 문서에 있어 이 부분이 종종 무시된다. 항상, 제목, 저자, 가능하면 날짜를 생신하라. 그렇지 않는 경우 `\maketitle` 명령어를 주석 처리하라. 추가적인 자세한 사항에 대해서는 발표 슬라이드를 참조한다.

### 1.2 개요 (Outline)

명령어 `section` 과 `subsection` 으로 장과 절을 만든다. 예를 들어, “일반”은 장이 되고, “헤더 절(header section)”과 “개요”는 절이 된다.

### 1.3 문단 (Paragraphs)

텍스트 사이에 간단하게 공백 두 줄을 넣으면 새로운 문단이 생성된다. 예를 들어, 이 문단은 “엔터(enter)” 키를 두번 치면 종료된다 (.tex 문서 참조) ... 이것은 새 문단은 아니다 (“엔터(enter)”를 한번만 침).

하지만, 이것은 새 문단이다. 만약 추가 여백이 문단 사이에 필요하면, 역슬래쉬 두개 명령어를 사용하고 나서 “엔터(enter)” 키를 두번 친다...

PDF 가 알아서 마지막 문단과 이 문단 사이에 공백을 삽입한다.

특정 문단에 들여쓰기를 하지 않으려면, 이 문단처럼, `\noindent` 명령어를 사용한다.

일반적으로, 추가로 많은 공백을 넣어도 PDF 출력 텍스트에는 영향이 없다. `\pagebreak` 명령어로를 사용해서 새 페이지를 생성한다.

## 1.4 공백 (Spacing)

수평 텍스트가 `\hspace{0.3cm}` 명령어를 사용해서 텍스트에 삽입될 수 있다. 명령어에 인자는 물론 좀더 크거나 작은 단위로 변경될 수 있다. 즉, 0.1cm, 2.5cm, 1.3in, 등등. 명령어 `\vspace{1.1cm}` 도

비슷한 방식으로 동작한다. LATEX은 다른 선택옵션 몇가지와 더불어, cm, mm, in, 특정한 길이 단위도 받아들인다.

## 1.5 텍스트 (Text)

명령어를 사용해서 텍스트를 조작할 수도 있다. 예를 들어, `\emph` 명령어를 사용해서 텍스트를 강조 (이태릭체) 한다. 텍스트를 이태릭체로 작성하는 방식이 몇가지 있다. { 중괄호 } 를 사용해서 명령어가 먹히는 곳을 잡아낸다. 이태릭체와 유사하게 텍스트를 몇가지 방식으로 굵은 글씨로 표현할 수 있거나 텍스트에 색깔을 입힐 수도 있다. 이 색이 “내가 정의한 색이다”. 또한, 타자수처럼 글자를 타이핑한다.

쉬프트(shift)-아포스트로피 키 조합을 사용해서 ”큰 따옴표” 표시하면 LATEX에서 미려하지 못하다. 대신에 왼쪽 큰 따옴표로 “1 번” 키옆에 아포스트로피를 두번, 오른쪽 큰 따옴표로 따옴표를 조합해서 사용한다. 즉, “왼쪽 큰 따옴표와 오른쪽 큰 따옴표” .

텍스트 크기도 매우 작게(tiny), 스크립트 크기(scriptsize), 주석 크기(footnotesize), 작게(small), 크게(large), 아주 크게(Large), 매우 크게(LARGE) 등등 다양한 크기로 표현 가능하다.

## 1.6 매크로 (Macros)

일반적으로, 이번 절에서 사용된 명령어는 **매크로(Macros)**를 사용하면 찾을 수 있다. 예를 들어, TeXShop 프로그램에서 원하는 폰트 크기를 얻으려면, Macros > Text Styles > size 로 들어간다. 인용(quotation)도 시도해 보라 (Macros > Insertions > quotation ):

보통 문단에 포함하고 싶지 않은 매우 긴 인용될 수 있다. 그래서, 다른 한편으로 별도로 독립시켜서 보통 문단보다 좀더 적을 폭을 갖게 만든다. 만약 원한다면, 인용 시작지점에 `\em` 을 사용해서 이러한 특별한 텍스트 전체를 쉽게 이태릭체로 만들 수 있다.

## 1.7 글머리표 (Lists)

앞선 하위 절은 다음과 같다...

- 공백 (Spacing)
- 텍스트 (Text)
- 매크로 (Macros)

글머리표는 `itemize` 환경을 사용하거나, 만약 숫자형 글머리표를 원한다면, `enumerate` 를 사용한다:

1. 공백 (Spacing)
2. 텍스트 (Text)
3. 매크로 (Macros)

하지만, 숫자는 하위 절과 매칭되어야 되고, 이 작업을 수행하는 선택옵션도 물론 있다....

- 1.4 공백 (Spacing)
- 1.5 텍스트 (Text)
- 1.6 매크로 (Macros)

## 1.8 특수 문자 (Special characters)

`LATEX`코드는 특수 문자를 많이 사용한다. 이것이 의미하는 바는 텍스트에 이러한 특수 문자를 넣으려면, `LATEX`코드의 통상적인 목적에서 문자를 나오게(escape) 만들어야 된다. 예를 들어, 다음 문자를 나타내려면 명령어 각각은 역슬래쉬를 앞에 두는 것이 필요하다: #, \$, {, }, &, %, -. \ 와 ~ 은 약간 더 호들갑을 떨어야 된다. 그리스 문자와 기호는 5 절에서 소개된다.

## 1.9 글자 그대로 (verbatim)

명시적으로 텍스트를 정말 찍고자 한다면, 글자 그대로 (`verbatim`) 기능을 사용한다:

만약 `\LaTeX` 파일로 보고자 한다면, 이 모든 것이 정말 그대로 `\emph{나타날 것이다}`. % 와 주석(`comment`)도 글자 그대로 (`verbatim`)에서는 동작하지 않는다...

# 제 2 절 텹 (Tabbing)

## 2.1 태빙 기초(Basic tabbing)

`indent` 명령어를 사용해서 텍스트를 들여쓰기한다:

이것이 들여쓰기한 텍스트다:

좀더 중요하게, 사용자 정의 태빙도 만들 수 있다. 예를 들어,

첫번째 템 두번째 템 디폴트 기본설정으로 템 사이에 여분 공간은 없다.

1	2	3	4
하나	셋		넷

세번째 줄 두번째 열처럼, 셀은 공백이 될 수도 있다.

`\hspace{0.2cm}` 명령어를 사용해서, 좀더 많은 공백을 넣을 수 있다...

첫번째 템 두번째 템 이제 템 사이에 추가 공백이 생겼다.

1	2	3	4
하나	셋		넷

## 2.2 즉흥 템 (Impromptu tabs)

새로운 템을 중도에 생성할 수도 있다...

첫번째 템 두번째 템 이제 템 사이에 추가 공백이 있다.

1	2	3	4	4.5
하나	셋		넷	이것은 어디서 시작할까요?

그렇게 잘 동작하지 않는다. `\hspace` 을 사용해서 수정한다...

첫번째 템 두번째 템 이제 템 사이에 추가 공백이 있다.

1	2	3	4	4.5
하나	셋	넷		이것은 어디서 시작할까요?

## 2.3 태빙 예제 (Tabbing example)

태빙은 흥미로운 환경이다. 좀더 만만찮은 태빙 생성은 (L<sup>A</sup>T<sub>E</sub>X를 다소 난잡하게 만든다) ... `\hspace` 명령어는 음수 인자를 받아들일 수 있다. 그렇지 않다면, 다음 예제의 구성요소는 허락되지 않는다.

### 테스트 명칭 기술

### 전체 시험 횟수

고정 크기 데이터 수집 시에  $n_f(\alpha, \beta, \delta, \sigma^2)$

만약  $|Z_k| \geq 1.96$ , 정지하고,  $H_0$  를 기각.

그렇지 않으면, 정지하고,  $H_0$  를 기각하지 않는다.

포록 (Pocock) $k = 1, \dots, K - 1$  집단 다음에,  $n_f R_P(K, \alpha, \beta)$

만약  $|Z_k| \geq C_P(K, \alpha)$ , 면추고,  $H_0$  를 기각.

그렇지 않다면, 테스트를 계속한다.

$K$  집단 후에 (마지막 집단)

만약  $|Z_K| \geq C_P(K, \alpha)$  이면  $H_0$  를 기각.

그렇지 않다면,                  면  $H_0$  를 기각하지 않는다.

## 제 3 절 표 (Tables)

### 3.1 표 기초(Basic tables)

기본 표는...

왼쪽 정렬    중간 정렬    오른쪽 정렬

1                      2                      3

표를 가운데 정렬하려면, 표 주위에 가운데 정렬 환경을 생성한다:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.2852	0.8434	-0.34	0.7452
x	0.4192	0.1499	2.80	0.0266

아마도, 수직 분할선을 추가하고 싶을 수도 있다 (원활 경우, 수직 분할선을 더 추가할 수도 있다)...

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.2852	0.8434	-0.34	0.7452
x	0.4192	0.1499	2.80	0.0266

또 다른 표는 다음과 같다...

왼쪽	왼쪽 정렬된다.	오른쪽
1	만약 한 칼럼에 텍스트가 너무 길면, 정렬에 사용되는 1, c, r 대신에 \p{7.5cm} 혹은 이에 상응하는 것을 사용해서 문단이 멋진 형태로 표에 작성되도록 한다. 만약 칼럼 폭을 자유자재로 제어하고자 하면, 매우 편리한 기능이다.	3

### 3.2 캡션과 참조 (Captions and referencing)

표에 캡션을 원합니까? 표 환경을 사용하세요. 이것을 떠다니는 표(floating table)라고 부른다... 만약 주의깊게 제어하지 않는다면, 그리고 때로는 제어할 때조차도 여전히, 페이지 여기저리를 “떠다닌다(float)”.

자동으로 참조를 표(차후에 나올 그림)에 빌드할 수도 있다. 예를 들어, 다음 표는 표 2. 만약 표 번호가 변경되면, .tex 파일을 두번 컴파일한 뒤에 표번호가 자동으로 갱신된다.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.2852	0.8434	-0.34	0.7452
x	0.4192	0.1499	2.80	0.0266

표 1: 이것이 캡션(caption)이다.

왜 두번 컴파일해야 할까요? LATEX이 컴파일할 때 (모두 무시하는 파일들... 컴파일할 때 생성되는 파일 중 하나로부터), 참조(reference)를 불러 읽어온다. 하지만, 불러 읽어오는 파일은 이전 컴파일 결과로부터 만들어졌다. 그래서, 만약 한번만 컴파일하면, 읽어오는 파일이 갱신이 되지 않을 수도 있다. (이제 이해되시죠?)

참조에 대한 추가적인 설명은 `latex-intro-kr.tex` 원본 파일을 참조한다.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.5758	1.4528	-0.40	0.7056
x	0.3775	0.1971	1.92	0.1039
z	1.4042	1.7357	0.81	0.4494

표 2:  $x$  혹은  $z$  어느 변수도 통계적으로 유의하지 않은 것으로 밝혀졌다.

### 3.3 array 환경

5 절에서 보았듯이, 통상 방정식에 사용된다는 것만 제외하면, `array` 환경은 `tabular` 환경과 매우 유사하다.

### 3.4 R 팩키지, xtable

R 산출물을 LATEX에 넣으려는 R 사용자에게 있어 `xtable` 팩키지는 매우 유용하다:

```
> library(xtable) # install.packages('xtable') 명령어를 사용해서 팩키지를 다운로드한다.
> x <- 1:9
> z <- rnorm(9)
> y <- x/7 + z*2 + rnorm(9)
> xtable(summary(lm(y ~ x+z)))
[... LaTeX 으로 복사해서 붙여넣기 할 수 있는 산출물이 많다 ...]
```

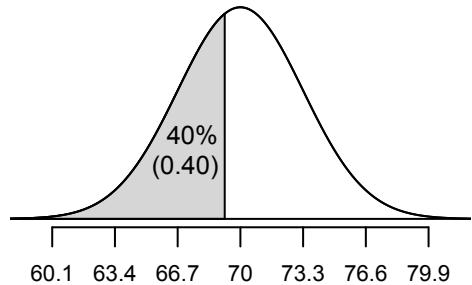
R에서 직접 복사해서/붙여넣은 산출물 표:

이뿐 아니라, 행렬, 데이터프레임, 다른 R 객체에도 사용될 수 있다.

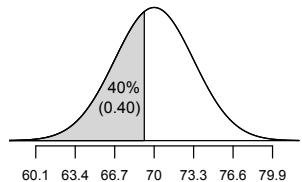
	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.1563	0.6243	-0.25	0.8107
x	0.1094	0.1145	0.96	0.3760
z	2.6170	0.4308	6.08	0.0009

## 제 4 절 그림 (Figures)

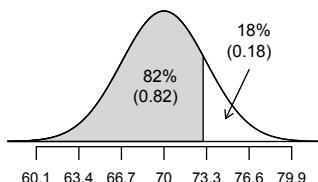
### 4.1 그림 기초 (Basic figures)



\includegraphics 명령어를 사용해서 기본적 그림을 삽입할 수 있음. 선택옵션 space 인자를 사용해서 크기도 제어할 수 있음.



표를 가운데 정렬한 것과 동일한 방식으로 그림도 쉽게 가운데 정렬할 수 있다:



### 4.2 캡션과 참조하기 (Captions and referencing)

표와 마찬가지로, 그림은 “떠다닐(floated)” 수 있고, 캡션/라벨을 갖을 수 있다. Templates 도구로 그래픽 작업을 좀더 멋지게 수행할 수 있다. 그림 1 참조. L<sup>A</sup>T<sub>E</sub>X “Float Figure Template”에는 추가할 필요가 있는 space 선택옵션이 포함되어 있지 않음에 주목한다.

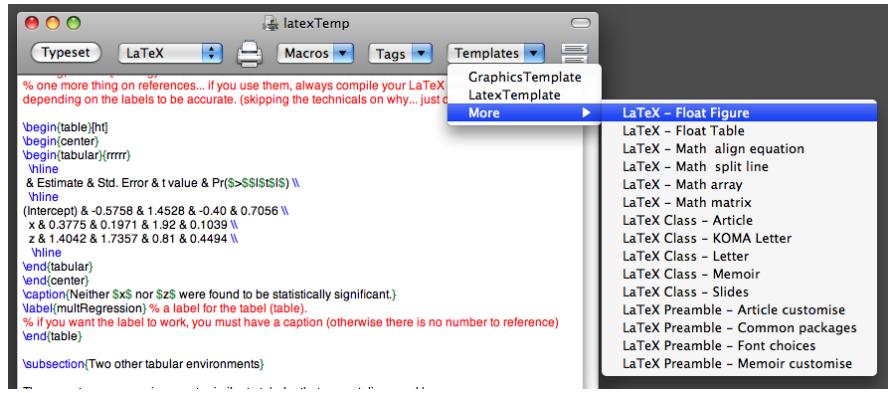


그림 1: 그림 템플릿을 찾는 장소.

### 4.3 체계적으로 정리하기 (Keeping organized)

그림을 별도 디렉토리에 정리할 것을 강력 추천한다. 그림 2 처럼 이미지 파일로 주 작업 디렉토리가 어수선하게 되는 것을 방지한다. 그림 3에 문서 그림에 대한 더 좋게 체계적으로 정리된 구조가 나와 있다.

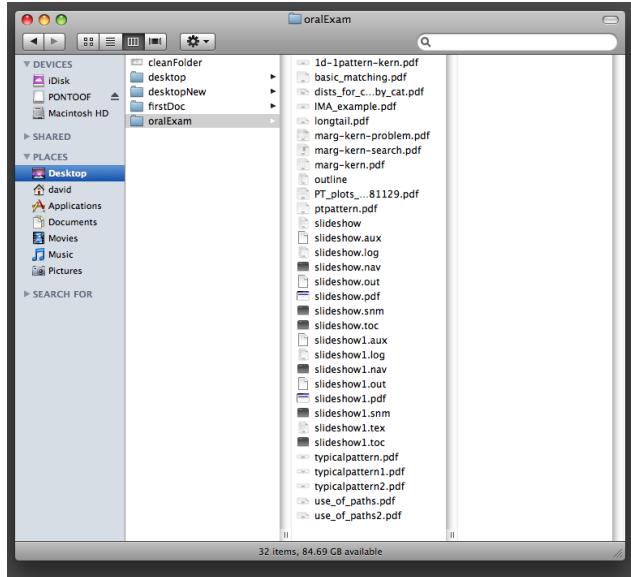


그림 2: 이렇게 하지 말라. “slideshow” 보다는 좀 더 주의깊게 파일명을 짓는다. “slideshow”는 특정되지 않아 향후 혼란의 여지가 많다.

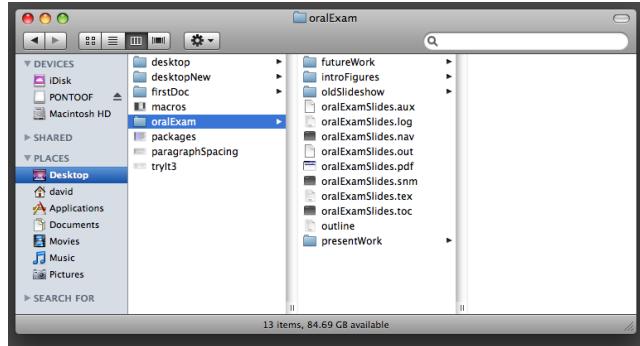


그림 3: 이와 같이 파일을 체계적으로 정리하라.

## 제 5 절 수학 (Math)

### 5.1 텍스트 내부 수학 (Math in text)

LATEX을 사용하면,  $\alpha$ ,  $\zeta$ ,  $\mu$ , 등 그리스 문자를 텍스트 내부에 추가하기 쉽다. 동일한 방식으로, 방정식도 또한 쉽게 추가될 수 있다:  $y = x^3$ ,  $\sum z^j$ ,  $x_1 + \dots + x_n$ .

LaTeX makes it easy to add Greek letters like  $\alpha$ ,  $\zeta$ ,  $\mu$ , etc. into text. In the same way, equations can be added easily as well:  $y=x^3$ ,  $\sum z^j$ ,  $x_1+\dots+x_n$ .

$\alpha, \zeta, y = \sqrt{x} \log(x)$  같은 그리스 문자와 수학 표현식은 달리 기호 두개를 사용해서 쉽게 삽입할 수 있다: 각 수학 표현식과 기호에 대해 어떤 명령어가 대응되는지 단지 기억력과 관련된 문제다. 예를 들어,  $\alpha$  는  $\$\\alpha\$$  명령어로 생성된다.  $\alpha$ 가 생성되는 방식에 기반해서,  $\beta$  는 어떻게 생성할까요?

LaTeX 과 Matrix 패널에는 상당히 많은 수의 기호와 문자 등이 포함되어 있다. 맥 TeXShop, 윈도우 TeXWork에서 alt-command-[dash/underscore key] 혹은 alt-command-[+/= key] 단축키를 통해서, 혹은 메뉴를 통해 해당 패널(TeXShop “Window” 메뉴)로 접근할 수 있다. 생성 할 수 있는 일부 문자와 기호가 다음에 나와 있다...

ჩემი რესულუტაცია არ არის დანართული, მაგრა ვარ და ვარ მას შესაბამისი მოვალეობა.

$\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\pi\varpi\varrho\sigma\varsigma\tau\upsilon\phi\varphi\chi\psi\omega$

이용가능한 기호는 어마어마하다. 만약 특정 기호가 필요하다면, 아마도 LATEX에 존재할 것이다.

수학 표현식을 만들어 내는 방식이 엄청나게 많이 있다...

$$\sqrt{2}, \quad \frac{5}{2+3} = 1, \quad \left( \frac{5}{2+3} \right), \quad 2^1 0 \neq 2^{10} = 1024, \quad x_1 = 3$$

$$\bar{x}, \quad 3 \geq x, \quad \lim_{x \rightarrow 0} \left( \frac{\sin(x)}{x} \right) \rightarrow 1, \quad \frac{\sin(x)}{x} \xrightarrow{x \rightarrow 0} 1$$

## 5.2 방정식 환경과 참조 (Equation environment and referencing)

방정식 환경을 사용해서 방정식을 그 자체 라인으로 넣을 수 있다:

Equations can also be put on their own line using the equation environment:

$$A_{b_{ik}} = \sum_{l=1}^k \sum_{j=1}^i \gamma^{\alpha_{b_{jl}}} \quad (1)$$

그림과 표와 마찬가지로, 방정식도 방정식 1처럼 참조할 수 있다.

만약 방정식에 숫자를 배정하고 싶지 않다면, `eqnarray*` 환경을 사용한다:

$$A_{b_{ik}} = \sum_{l=1}^k \sum_{j=1}^i \gamma^{\alpha_{b_{jl}}}$$

다음에 방정식 2 사례가 하나더 나와 있다...

$$\sum_{k=0}^{\infty} 0.5^k = \frac{1}{1 - 0.5} = 2 \quad (2)$$

## 5.3 줄 맞추기 (Aligning)

방정식이 여러 줄에 걸쳐있고, 줄 맞추기가 필요하다면, 앤퍼샌드(&)를 두개를 사용한다:

$$\begin{aligned} y &= (x - b)^2 + a \\ &= x^2 - 2bx + b^2 + a \end{aligned}$$

만약 앤퍼샌드를 사용하지 않으면, 줄맞추기가 보통 엉망이 된다.

## 5.4 배열 (Arrays)

Matrix 패널을 사용해서 항상 배열을 쉽게 생성한다:

$$\begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \sigma_{1,3} \\ \sigma_{2,1} & \sigma_2^2 & \sigma_{2,3} \\ \sigma_{3,1} & \sigma_{3,2} & \sigma_3^2 \end{pmatrix}$$

수학 기호를 삽입하기 더 쉽다는 점을 제외하면, 배열 생성은 본질적으로 표와 동일하다.

## 5.5 amsmath 팩키지 사용 시 몇가지 장점

amsmath 팩키지는 LATEX 템플릿에 없다. 하지만, 있으면 매우 편리하다. 만약 좀더 긴 방정식이 있고, 한 줄에만 숫자를 넣고 싶다면, \notag 을 사용한다:

$$\begin{aligned} & Cov \left( \left( \bar{X}_A^{(k_1)} - \bar{X}_B^{(k_1)} \right) \sqrt{I_{k_1}}, \left( \bar{X}_A^{(k_2)} - \bar{X}_B^{(k_2)} \right) \sqrt{I_{k_2}} \right) \\ &= Cov \left( \bar{X}_A^{(k_1)} - \bar{X}_B^{(k_1)}, \bar{X}_A^{(k_2)} - \bar{X}_B^{(k_2)} \right) \sqrt{I_{k_1} I_{k_2}} \\ &= Cov \left( \bar{X}_A^{(k_1)} - \bar{X}_B^{(k_1)}, \bar{X}_A^{(k_2)} - \bar{X}_B^{(k_2)} \right) \sqrt{I_{k_1} I_{k_2}} \end{aligned} \quad (3)$$

amsmath 팩키지가 이러한 명령어를 사용하는데 필요하다. \text 를 사용해서 방정식에 텍스트를 추가할 때도 이 팩키지가 필요하다:

$$\bar{x} = \sum_{i=1}^n x_i \quad \text{그리고} \quad \hat{\sigma} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

\text를 갖는 eqnarray\* 의 또다른 사례:

$$\text{추정 시간} = \frac{\text{여행 거리}}{\text{자동차 속도}} + \text{지연 시간}$$

## 제 6 절 실습 (Practice)

새로운 문서를 생성해서 항목 3개를 만드세요. 새로운 문서에 \title 와 \author 정보를 확실히 챙신하세요.

### 6.1 시도해 보자 #1

tabular 환경을 사용해서 그림 4에 나온 출력을 만들어 본다.

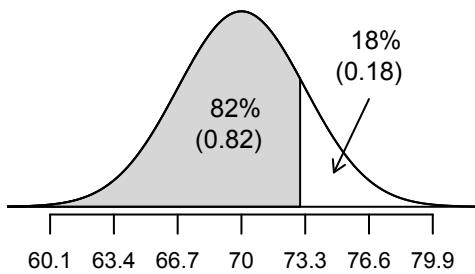
	$\bar{x}$	$\hat{\sigma}$	$n$
$S_1$	6.5	1.3	17
$S_2$	12.2	1.4	25

그림 4: 시도해 보자 #1.

### 6.2 시도해 보자 #2

다음 이미지를 높이 2 센티미터(0.8 인치), 가운데 정렬, 캡션 추가, 참조 추가해서 변형하세요. 그림을 참조하는 문장도 작성하고 LATEX 문서를 두번 컴파일 해서 참조가 정상 동작되도록 한다.

(만약 “Float Figure Template” 을 사용하면, 높이 선택옵션을 추가하는 것을 확실히 하라... 대안으로, 템플릿으로 이전 L<sup>A</sup>T<sub>E</sub>X 예제를 사용할 수도 있다.)



### 6.3 시도해 보자 #3

`eqnarray*` 환경을 사용해서, 그림 5에 나온 방정식을 만들어 보세요.

$$\sum_{i=0}^n p^i = \frac{1 - p^{n-1}}{1 - p}$$

그림 5: 시도해 보자 #3.

## 제 7 절 참고문헌 관련

점 패턴(point pattern)이 점과정(point process) 구현(Daley and Vere-Jones, 2003)으로 기술되고, 점 패턴에 대한 몇가지 일차원 거리함수가 Victor and Purpura (1997)에 기술되어 있다.

## 참고 문헌

Daley DJ and Vere-Jones D (2003). *An Introduction to the Theory of Point Processes, Volume 1: Elementary Theory and Methods*. Springer-Verlag, New York.

Victor J and Purpura K (1997). Metric-space analysis of spike trains: theory, algorithms and application. *Journal of Neuroscience Methods*, 8:127–164.