

bitPublish를 이용하여 한글 책 조판하기

목차

머리말	1
1 예제를 위한 장	3
1.1 데이터	3
1.1.1 측정의 타당도 및 신뢰도	4
1.1.2 척도	4
1.1.3 도수분포표	7
2 bitPublish 소개	15
2.1 Quarto 소개	15
2.1.1 Quarto	15
2.1.2 Quarto extensions	16
2.1.3 bitPublish	17
2.2 bitPublish 소개	17
2.3 조판 설정	17
2.3.1 페이지 레이아웃	17
2.3.2 폰트 설정	18
2.3.3 머리글 바닥글	20
2.3.4 글줄 사이	20
2.4 bitPublish 서식 이용하기	21
2.4.1 멀티 아웃 포맷	21
2.4.2 사용자 정의 블록	21
2.4.3 정보 블록	22
2.4.4 이야기 박스	27
2.4.5 타이틀 박스	28
2.4.6 예제	28

2.4.7	연습문제	30
2.5	bitPublish 활용을 위한 팁	31
2.5.1	이미지 삽입	31
2.5.2	표 삽입	36
2.5.3	수식 입력	39
2.5.4	찾아보기 등록	39
2.5.5	참고문헌 등록	40
2.5.6	자동조사 처리	42
3	Troubleshooting	45
3.1	Quarto 오류	45
3.1.1	TeXlive 버전 이슈	45
3.2	LaTeX 오류	46
3.2.1	찾아보기 오류	46
4	Version	49
4.1	version 0.1	49
4.1.1	version 0.1.2	49
4.2	version 0.2	49
4.2.1	version 0.2.0	49
	참고문헌	51

--- bitPublish를 이용하여

머리말

R은 통계적인 계산과 시각화에 많은 장점을 가지고 있는 도구이다. 개인적으로는 S-PLUS의 경험을 시작으로 해서 R이라는 아주 친한 친구를 알게 된 것을 대단한 행운으로 여기고 있다. R에 대한 깊이가 깊다고 자신할 수는 없지만, 함께해 온 날은 결코 짧지 않다. 1993년 여름, S-PLUS를 통해 S언어를 처음 접한 후 매우 강한 인상을 받았다. 그리고 2000년대 초 어느 서점에서 책을 뒤적이다가 단 몇 줄이지만 S언어의 클론이라는 존재로 R이라는 것이 있음을 알고 전율을 느꼈던 순간을 잊지 못한다. 그리고 아직 까지도 통계적인 계산과 시각화에 R보다 더 좋은 도구가 없다는 생각에는 변함이 없다.

필자는 2005년 “R 그래픽스”를 통해서 R을 이용해서 여러 가지 통계적인 그래프를 그리는 방법과 원리를 제시했고, 2010년 “R 프로그래밍”을 통해서 R로 데이터를 다루고 프로그래밍을 하는 방법을 설명하였다. 그리고 이제 R을 이용해서 통계학의 기본적인 원리를 이해할 수 있도록 예제 중심의 통계학 기본서를 내놓게 되었다. 늘 그렇듯 책을 마무리하는 시점에서는 항상 부족함을 느낀다. 그러나 딱딱한 수리적인 통계 이론이 아닌 R을 이용해서 쉽게 통계학의 원리를 터득할 수 있는 예제를 표현하려고 노력한 것에 대해서는 인정해 주었으면 한다.

기타 등등 블라블라

--- **bitPublish**를 이용하여

2 머리말

제 1 장

예제를 위한 장

이 장에서는 ‘bitPublish’를 이용하여 작성한 한글 책의 형태를 이해할 수 있도록, 여러 조판 방법의 예시를 표현한다. 본문에 표현되는 내용은 저술자인, 가칭 “R로 배우는 기초 통계학”의 일부를 발췌한 것이다.[7] 그러므로 본문의 내용은 완벽한 문맥이 아닌, 중간중간 문단이나 내용이 제거된 불완전한 내용임을 밝혀 둔다.

학습 목표

데이터의 분류 체계로서의 척도를 이해하고, 이를 기반으로 한 R 데이터 객체를 이해한다. CSV 파일을 읽고, 데이터 프레임 객체로 CSV 파일을 생성할 수 있다.

1.1 데이터

데이터(data)를 국어사전에서 찾으면 다음과 같은 세 가지의 뜻을 가지고 있다.

1. 이론을 세우는 데 기초가 되는 사실, 또는 바탕이 되는 자료.
2. 관찰이나 실험, 조사로 얻은 사실이나 정보.

이 세 가지를 정리하면 데이터는 “이론을 세우기 위해 관찰이나 실험, 조사를 통해서 얻은 정보를 컴퓨터가 처리하기 위한 형태로 만든 정보”로 생각해 볼 수 있다. 이 절에서는 데이터의 정의에서 컴퓨터가 처리하기 위한 형태로 된 정보를 R이 처리하기 위한 형태로 된 정보로 간주할 것이다. 또한 위키피디아(Wikipedia)에서는 데이터를 “항목(items)의 집합(set)에 속하는 질적(qualitative) 또는 양적(quantitative) 변수(variables)의 값

(values)”으로 정의하고 있다. 여기서 항목의 집합이란 관심이 있는 객체의 집합을 의미하며, 변수는 항목의 특성을 의미한다. 국어사전에서의 데이터 정의는 데이터의 목적성과 취득 방법 및 형태에 의한 것이라면, 위키피디아에서의 정의는 데이터의 속성에 대한 것이다.

국어사전에서의 정의를 기준으로 R을 이용한 통계분석을 위해서는 몇 단계의 과정이 필요하다.

첫째로, 사건이나 사실 등의 현상을 관찰하여 일정한 규칙에 의한 체계적인 단위의 수치를 부여하는 측정이 필요하다. 측정을 위해서는 자, 저울, 속도계, 온도계, 혈압계와 같은 측정을 위한 도구가 필요할 수도 있으며, 이를 통해서 측정한 현상을 수량화한다. 이처럼 사건이나 사실 등의 현상을 수량화하기 위해서 그 현상에 숫자를 부여한 것을 척도라 한다.

이러한 일련의 과정에 오류가 발생할 수 있다. 측정하는 과정에서 발생할 수 있는 측정의 오류, 측정된 값을 데이터로 코딩하는 과정에서의 오기(타)로 인한 오류도 있을 수 있다. 또한 측정하는 저울, 온도계와 같이 측정하는 도구의 정밀도에 따른 오류도 있을 수 있다. 그러므로 올바른 통계분석을 위해서는 분석 단계 이전에 정밀도가 높은 측정도구를 이용하여 정확하게 측정하고, 측정된 값을 정확하게 옮겨서 데이터화하는 것이 필요하다.

마지막으로 코딩된 데이터를 R의 데이터 객체로 변환하는 방법도 중요하다. 이 작업에서는 적절한 R의 데이터 객체로의 전환이 필요하며, 척도의 적당한 변환도 필요할 수 있다.

1.1.1 측정의 타당도 및 신뢰도

우수한 측정도구는 다음의 두 가지 조건을 만족해야 한다.

양궁으로 과녁을 맞히는 예를 R을 이용해서 그려 보자. 다음의 예제를 실행하면 그림 1.1과 같은 네 가지 종류 플롯을 얻는다. 여기서 가장 이상적인 과녁은 오른쪽 상단의 과녁으로 타당도와 신뢰도가 높음을 알 수 있다.

1.1.2 척도

측정된 데이터는 본질적인 내용이 기호나 수량화되어서 변수로 만들어진다. 이 수치화된 변수값들의 체계를 척도(scale)라 한다. 척도는 다음처럼 명목척도, 서열척도, 구간척도, 비율척도로 나누어진다.

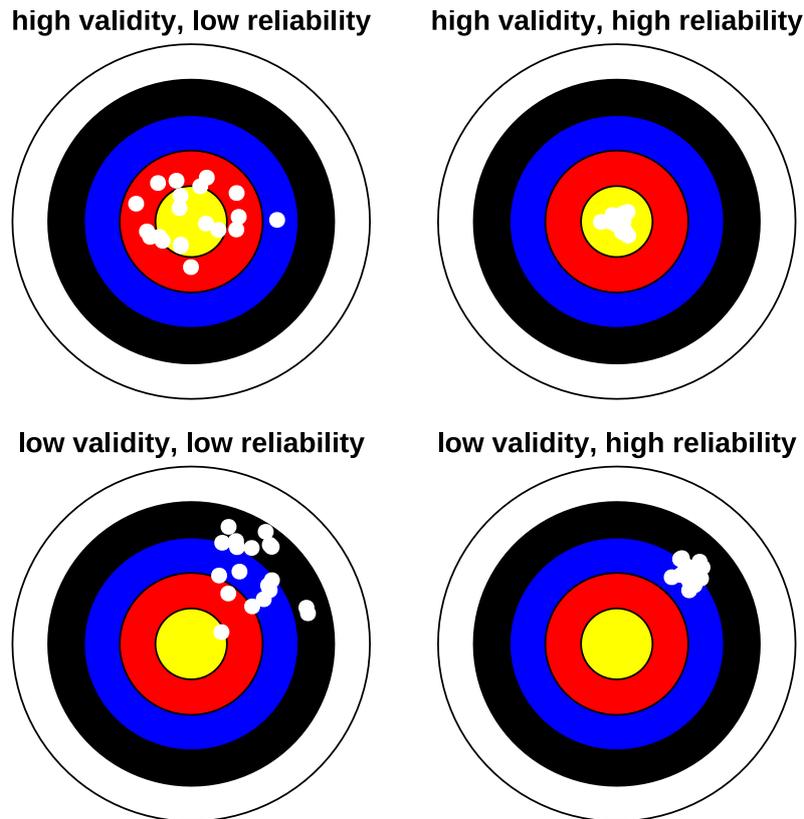


그림 1.1: 타당도와 신뢰도

- 명목척도(nominal scale, 名目尺度)
 - 분류 목적으로 수치화한 것으로 양적인 의미를 전혀 갖지 않고, 명칭을 대신 하는 척도
 - 예) 성별(1-남자, 2-여자)
- 서열척도(ordinal scale, 序列尺度)
 - 명목척도와 같은 개념에 추가적으로 서열(순서) 관계의 속성을 갖는 척도
 - 예) 등급: 1-1등급, 2-2등급, 3-3등급. 크기: 1-대, 2-중, 3-소

빈도수를 이용한 계산만 의미가 있는 명목척도는 일반적으로 범주형 데이터로 만들어진다. 예를 들면 성별이라는 범주형 데이터(categorical data)에서의 남자, 여자와 같은 수준(level)에 대한 도수 및 비율(빈도)을 이용한 분석을 수행할 수 있다. 서열척도 또한 범주형 데이터로 만들어지지만 명목척도와 달리 순서의 의미가 있기 때문에 빈도분석 이외에 중위수 등 순서통계량 기반의 분석을 수행할 수 있다. 구간척도와 비율척도에는 여러 가지의 기술통계 및 추론통계의 방법을 사용할 수 있으며, 우리가 일상생활에서 활

용하는 수치 데이터들의 대부분은 비율척도로 이루어져 있다.

통상적으로 많은 사람들은 '수준을 등급의 높고 낮음을 의미하는 가치의 척도'로 인식하고 있다. 그러나 한국 통계학계에서는 levels을 수준으로 직역하여 사용해 왔다. 혼동스러울 수도 있으니, 수준을 범주형 데이터의 범주로 이해하기 바란다.

예제 1.1.

네 가지의 척도를 R의 데이터 객체로 표현해 보자.

명목척도에 대한 예제다. 예제의 fruits는 세 가지 과일 이름을 갖는 범주형 데이터로 R의 factor() 함수로 만들 수 있다. 이 함수는 factor라는, 명목척도를 표현하는 데이터 객체를 생성한다. 데이터 객체인 factor는 table() 함수를 이용해서 도수와 비율을 구할 수 있다.

명목척도를 표현하는 factor 객체는 apple, banana, pear와 같이 과일 이름으로 표현되지만 R의 내부에서는 1, 2, 3과 같이 인식된다. 그래서 as.numeric() 함수를 이용하면 apple, pear, pear, apple, banana가 1, 3, 3, 1, 2와 같이 인식됨을 알 수 있다. 여기서 각 수준을 인식하는 1, 2, 3은 비록 수치형이지만 각 수준의 크기가 아니라 몇 번째 수준인가를 나타내는 지시자다.

```
fruits <- c("apple", "pear", "pear", "apple", "banana")
fruits <- factor(fruits)
fruits
```

```
[1] apple pear pear apple banana
Levels: apple banana pear
```

```
table(fruits) # 도수분포
```

```
fruits
apple banana pear
 2      1      2
```

```
table(fruits) / sum(table(fruits)) * 100 # 비율 (백분율)
```

```
fruits
apple banana pear
    40    20    40
```

```
as.numeric(fruits)           # 수준 인덱스
```

```
[1] 1 3 3 1 2
```

구간척도와 비율척도를 표현해 보자. 두 척도에서의 차이점은 절대 0의 여부에 따르기 때문에 표현하는 방법은 동일할 뿐, 개념적으로 절대 0을 갖느냐 갖지 못하느냐의 차이로 구분된다. 다음은 섭씨온도 값을 표현한 것과 절대온도 값을 표현한 예제다. 그런데 사실 두 사례는 동일한 온도값이다. 섭씨온도에 273.15를 더해서 절대온도로 변환하였기 때문이다.

1.1.3 도수분포표

도수분포표(frequency table)는 데이터의 분포 형태를 알아보기 위해서, 데이터를 구간별로 나누어 각 구간에 속하는 관찰치의 개수를 표로 나타낸 것이다. 각 구간을 계급(class)이라 하고 계급에 속한 관찰치의 개수를 도수(frequency)라 한다.

데이터가 $x_1, x_2, x_3, \dots, x_n$ 일 때, 도수분포표는 다음의 단계적 방법으로 만든다.

1. 데이터의 개수 n 을 구한다.
2. 데이터의 최댓값 Max와 최솟값 Min을 구한다. 데이터를 오름차순으로 정렬하여 순서통계량 $x_{(1)}, x_{(2)}, x_{(3)}, \dots, x_{(n)}$ 을 만들면 최댓값 Max와 최솟값 Min은 각각 $x_{(n)}$ 과 $x_{(1)}$ 이다.
3. 계급의 개수 계산 (Sturges 공식을 이용한 계급의 개수 계산)

$$n_{\text{class}} = 1 + \frac{\log n}{\log 2} \text{ 근처의 정수}$$

데이터의 개수가 너무 적으면 데이터의 분포에 대한 정보를 많이 잃게 되며, 너무 많으면 데이터의 전반적인 형태를 파악하기 어렵다.

4. 데이터의 최소단위(Unit) u 를 구한다. 예를 들어 168.2, 175.3, 171.2 등의 최소단위는 0.1이다.
5. 누적도수, 상대도수, 누적상대도수 등을 구한다.
 - 누적도수: 각 계급별로 누적된 도수로 마지막 계급의 누적도수는 n 이 된다. k 번째

계급의 누적도수:

$$F_k = \sum_{i=1}^k f_i$$

- 상대도수: 계급도수를 데이터 개수로 나눈 비율로 전체 데이터 중에서 해당계급이 차지하는 비율을 의미한다. 상대도수의 합은 1이 된다.

$$f_i^R = \frac{f_i}{n}, i = 1, 2, \dots, n_{\text{class}}$$

$$\sum_{i=1}^{n_{\text{class}}} f_i^R = \frac{f_1}{n} + \frac{f_2}{n} + \dots + \frac{f_{n_{\text{class}}}}{n} = \sum_{i=1}^{n_{\text{class}}} f_i \times \frac{1}{n} = n \times \frac{1}{n} = 1$$

- 누적상대도수: 각 계급에서의 누적된 상대도수의 값으로 누적 비율을 의미한다. 마지막 계급에서는 그 값이 1이 된다. k번째 계급의 누적상대도수:

$$CF_k = \sum_{i=1}^k f_i^R = \frac{F_k}{n}$$

클래스 상속과 ordered factor

상속(inheritance)은 부모 클래스의 내용(속성과 메소드)을 자식 클래스가 물려받는 것으로 객체지향형 언어의 특징이다. 예를 들면, 자동차라는 부모 클래스를 상속받아 세단과 SUV라는 자식 클래스를 생성하면, 자식 클래스인 세단과 SUV는 부모 클래스인 자동차가 갖고 있는 속성인 ‘핸들’, ‘브레이크’, ‘바퀴’와 같은 속성과 ‘액셀을 밟다’, ‘핸들을 돌리다’, ‘브레이크를 밟다’와 같은 메소드를 사용할 수 있다. 세단은 ‘가벼운 중량’, ‘낮은 차체’ 속성을, SUV는 ‘무거운 중량’, ‘높은 차체’ 속성을 추가로 만들어서 부모 클래스로부터 자식 클래스를 정의한다.



R에서도 클래스 상속의 개념이 있다. ordered() 함수는 ‘factor’라는 부모 클래스를 상속하여 ‘ordered’라는 자식 클래스를 만든다. 그러므로 ‘ordered’는 factor의 내용에, 수준의 순서가 서열로 처리되는 특성이 추가된 클래스다. 속성 중 class의 값이 “ordered” “factor”인 것은, 부모 클래스인 factor로부터 상속받아 만들어진 ordered 클래스라는 의미다. 오른쪽이 부모 클래스고, 왼쪽이 자식 클래스다.

CP949 인코딩과 fileEncoding 인수



MS Windows에서는 별도의 설정이 없으면, 텍스트 파일의 한글을 **CP949**로 인코딩하여 저장한다. 그러므로 MS Windows에서 만들어진 한글이 포함된 텍스트 파일을 읽을 때에는, 'fileEncoding' 인수에 "cp949"를 지정해야 한다. 그런데 그림처럼 Excel에서 CSV 파일을 저장할 때, 인코딩을 **UTF-8**로 설정하면 read.table() 함수에서 fileEncoding 인수로 인코딩을 "cp949"로 지정하지 않아도 된다.

외산 소프트웨어에서 한글을 문제없이 사용하는 것은 매우 번거롭고 까다롭다. 최선의 방법 중 하나는 한글이 포함된 데이터 파일을 **UTF-8** 인코딩으로 저장하는 것임을 잊지말자.

나는 통계계산이 수리영역인 줄만 알았다. 그런데, 이제는 논리적인 사고도 필요한 논리영역임을 느낀다. 그래서 논리적 사고로 통계적 데이터 분석을 위한 성능 좋은 연장이 필요하기 시작했다.

tidyverse 패키지는 R의 대표적인 패키지로 데이터 분석의 일련과정을 지원하는 여러 유용한 패키지의 모음이다. 그래서 패키지군이라 부르는 게 맞을듯 하다. tidyverse 패키지군을 구성하는 대표적인 패키지에는 dplyr 패키지와 ggplot2 패키지가 있다. dplyr 패키지는 논리적이고 순차적인 문법으로 데이터를 조작하거나 집계하는 패키지며 ggplot2 패키지는 데이터 시각화 패키지다. 우리는 dplyr 패키지를 이용해서 끊임없는 논리적 흐름의 전개로 도수분포표를 계산할 것이다.

개별 파이프의 목적은 파이프의 직경과 길이에 해당하는 수돗물을 특정 위치까지 이동시키는 것이다. 그리고 순차적인 데이터 처리의 흐름은 그림 1.2처럼 해당 처리를 담당하는 파이프들을 순차적으로 연결한 구조의 파이프라인으로 정의된다.

input -> process -> output 구조의 단일 파이프를 연결하기 위해서 tidyverse 패키지군의 magrittr 패키지가 파이프 연산자인 %>%를 제공한다. 이 연산자의 역할은 연산자 이전 함수(process)에서의 결과(output)를 다음 함수(process)의 입력(input)으로 전달하는 역할을 수행한다. 결국 이 연산자를 통해서 파이프라인이 만들어지며 **input -> process -> output -> input -> process -> output**과 같은 구조를 만들 수 있게 된다. 즉, 파이프 연산자 %>%는 파이프라인을 만드는 배관공이라 할 수 있다.

이러한 이유들로 인해서 파이프 연산자는 특히 dplyr 패키지의 데이터 처리에서 두각을 보였으며, 많은 R 사용자들의 R 스크립트에서 사랑받게되었다. 그래서 이제는 외부

파이프



파이프라인

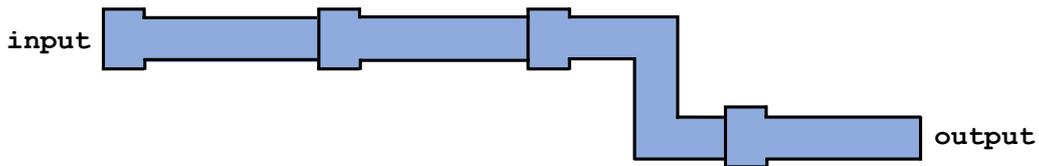


그림 1.2: 파이프와 파이프라인

(3rd-party) 패키지가 아니라 R 엔진 자체에 파이프 연산자를 포함시켰고, `|>`로 표현한다.

// 이 책에서는 수리적인 알고리즘의 이해를 위한 작업이 아니라면, 가급적으로 R의 내장 파이프 연산자를 `|>`를 사용하여 파이프라인 처리를 하겠다. 유용한 기능이기에 적극적으로 습득하는 것을 추천한다. **//**

`state.region`은 내장 데이터셋으로 미국 50개주를 'Northeast', 'South', 'North Central', 'West'라는 4개 권역으로 분류한 범주형 데이터다. 이 데이터를 `dplyr` 패키지의 함수와 파이프 연산자를 이용해서 도수분포표를 만들어보겠다.

먼저 `head()` 함수로 데이터의 앞부분 몇 건을 조회해 본다.

```
head(state.region)
```

```
[1] South West West South West West
Levels: Northeast South North Central West
```

대략 데이터의 구조가 이해되었으니, `tidyverse` 패키지를 불러들여서 작업을 위한 환경을 구성한다. `library()` 함수로 `tidyverse` 패키지를 불러오면 `dplyr`, `ggplot2`를 비롯한 몇 개의 패키지가 자동으로 로드된다.

```
library(tidyverse)
```

다음은 도수분포표를 생성하기 위한 `dplyr` 패키지 함수와 파이프 연산자를 이용한 스크

립트다.

```
state.region |>
  data.frame() |>
  count(지역 = state.region, name = "도수") |>
  mutate(누적도수 = cumsum(도수)) |>
  mutate(상대도수 = 도수 / sum(도수)) |>
  mutate(누적상대도수 = cumsum(상대도수))
```

	지역	도수	누적도수	상대도수	누적상대도수
1	Northeast	9	9	0.18	0.18
2	South	16	25	0.32	0.50
3	North Central	12	37	0.24	0.74
4	West	13	50	0.26	1.00

열거형 도수분포표는 복잡한 수리적인 계산 알고리즘이 필요없다. 논리적인 사고로 데이터 처리 프로세스를 다음처럼 순차적으로 풀어나가면 된다.

1. factor를 데이터 프레임 객체로 변환한다.
 - data.frame()
2. 집계 대상 변수의 수준(levels)별로 도수를 계산한다.
 - count()

이 사례에서 중요한 것은 파이프 연산자로 심리스(seamless)¹하게 데이터를 처리하는 순차 연산을 구현한다는 것이다.

¹뭔가를 연결할 때 마치 원래 하나였던 것처럼 끊김없이 매끄럽게 연결하는 것을 의미한다.

X \ Y	1	2	...	J	Total
1	n_{11}	n_{12}	...	n_{1J}	$n_{1.} = \sum_{j=1}^J n_{1j}$
2	n_{21}	n_{22}	...	n_{2J}	$n_{2.} = \sum_{j=1}^J n_{2j}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
I	n_{I1}	n_{I2}	...	n_{IJ}	$n_{I.} = \sum_{j=1}^J n_{Ij}$
Total	$n_{.1} = \sum_{i=1}^I n_{i1}$	$n_{.2} = \sum_{i=1}^I n_{i2}$...	$n_{.J} = \sum_{i=1}^I n_{iJ}$	$n = \sum_{i=1}^I \sum_{j=1}^J n_{ij}$

표 1.1: 2차원 $I \times J$ 분할표

1.1 연습문제

1. 다음의 데이터를 R의 객체로 만들어 보아라.

- (1) 합격, 불합격, 불합격, 합격, 불합격
- (2) 1등급, 2등급, 3등급, 2등급, 5등급, 3등급, 4등급, 4등급, 3등급, 3등급
- (3) 12cm, 23cm, 32cm, 22cm, 19cm, 21cm, 20cm

2. 다음의 데이터 객체를 외부 파일로 출력해 보아라.

- (1) R의 `datas` 패키지에 포함되어 있는 `women` 데이터 프레임을 `women.csv` 파일로 출력해 보아라.
- (2) (1)의 파일을 다시 읽어 들여서 `women2`라는 이름의 데이터 프레임을 만들어라.

--- bitPublish를 이용하여

제 2 장

bitPublish 소개

이 장에서는 'bitPublish'의 활용을 위한 특징점과 기능을 소개합니다.

학습 목표

bitPublish가 제공하는 기능을 이해하고, bitPublish를 이용해서 원하는 책을 저작할 수 있다.

2.1 Quarto 소개

2.1.1 Quarto

Quarto^[3]는 오픈 소스의 과학 및 기술 출판 시스템으로 다음과 같은 기능을 제공합니다.¹

- Jupyter 노트북, RStudio를 사용하거나 선호하는 편집기로 마크다운 문서를 작성합니다.
- Python, R, Julia 및 Observable을 사용하여 동적 콘텐츠를 만듭니다.
- HTML, PDF, MS Word, ePub 등으로 재현 가능한 프로덕션 품질의 기사, 프리젠테이션, 웹사이트, 블로그 및 책을 게시합니다.
- Posit Connect, Confluence 또는 기타 게시 시스템에 게시하여 조직 전체에서 지식과 통찰력을 공유합니다.

¹<https://quarto.org/>

--- bitPublish를 이용하여

- 방정식, 인용, 상호 참조, 그림 패널, 콜아웃, 고급 레이아웃 등을 포함한 팬독 마크다운을 사용하여 작성합니다.

Quarto[1]를 요약하자면 그림 2.1처럼, Quarto 문서를 knitr 또는 jupyter로, 마크다운(Markdown) 문서로 렌더링한 다음 팬독(Pandoc)에 의해 마크다운 문서를 HTML, PDF 또는 Word 문서를 비롯한 여러 출력 포맷으로 변환해주는 시스템입니다.

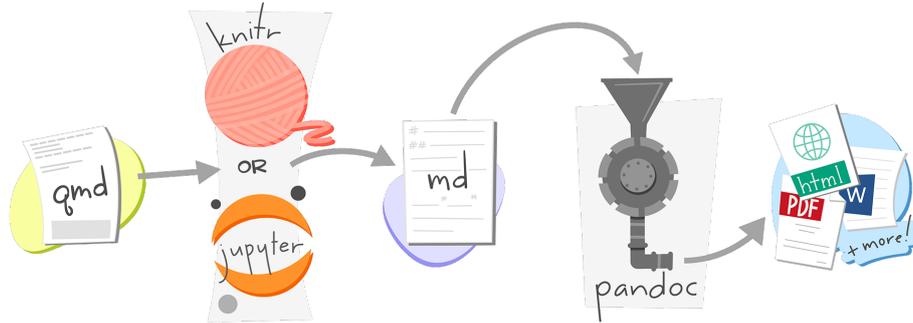


그림 2.1: Quarto 매커니즘, Allison Horst.

2.1.2 Quarto extensions

extensions은 Quarto의 동작을 수정하고 확장할 수 있는 방법을 제공하며, 다음과 같은 유형으로 구분합니다.[2]

영문 유형	국문 유형	설명
Shortcodes	단축 코드	다양한 유형의 콘텐츠를 생성하는 특수 마크다운 지시문. 예를 들어 짧은 코드를 만들어 문서에 트윗이나 동영상을 삽입
Filters	필터	새로운 전역 동작 또는 새로운 마크다운 렌더링 동작을 도입하기 위한 도구. 예를 들어 출력 접기, 이미지 슬라이더 등의 필터 제작
Journal Articles	저널 기사	마크다운으로 전문적인 저널 기사를 작성. LaTeX(PDF) 및 HTML 버전을 모두 생성할 수 있음

영문 유형	국문 유형	설명
Custom Formats	맞춤 형식	문서 옵션, 템플릿, 스타일시트 및 기타 콘텐츠를 함께 묶어 새로운 출력 형식을 생성
Revealjs Plugins	Revealjs 플러그인	Revealjs로 만든 HTML 프레젠테이션의 기능을 확장
Project Types	프로젝트 유형	표준 콘텐츠와 옵션을 함께 묶는 새로운 프로젝트 유형을 만들거나, 사용자 지정 HTML 형식으로 웹 사이트 제작
Starter Templates	시작 템플릿	템플릿과 예제 콘텐츠를 제공하여 사용자가 새 프로젝트를 시작할 수 있도록 지원

2.1.3 bitPublish

bitPublish는 Quarto extensions로 R 기반으로 한글 책을 저작해 줍니다. 또한 한글 책 저작을 위한 예제 콘텐츠를 제공합니다. 그러므로 extensions 분류 상으로는 맞춤 형식, 시작 템플릿으로 분류할 수 있습니다.

2.2 bitPublish 소개

Quarto를 이용해서 책으로 만들 수 있는 포맷에는 HTML, PDF, MS Word, EPUB, AsciiDoc이 있습니다. bitPublish는 이 중에서 PDF 포맷의 책을 생성합니다.

현재 bitPublish의 기본 설정에서는 4x6 배판 판형을, 갖는 영어를 포함한 라틴계열 언어와 한국어, 중국어를 혼용한 책을 저작할 수 있습니다.

2.3 조판 설정

2.3.1 페이지 레이아웃

현재 bitPublish의 기본 페이지 레이아웃 설정은 4x6 배판으로 설계되었습니다. 4x6 배판 판형의 레이아웃 설정을 변경하려면, `_extensions/bit2r/bitPublish/_extension.yml` 파일의 다음 행들을 수정합니다. LaTeX의 geometry 패키지를 사용합니다.

--- bitPublish를 이용하여

```
geometry:                                # 4x6 배판 도서를 위한 설정
- paper=a4paper
- layoutwidth=190mm
- layoutheight=260mm
- layoutoffset=10mm
- layoutvoffset=18.5mm
- showcrop
- top=20mm
- headsep=10mm
- bottom=30mm
- footskip=15mm
- left=25mm
- right=25mm
- centering
```

2.3.2 폰트 설정

bitPublish는 전자책이 아닌 종이에 출력되는 책의 저작을 목적으로 만들어졌기 때문에, 폰트의 선택에서 TTF(True Type Font)가 아닌 OTF(Open Type Font)를 선택하였습니다.

2.3.2.1 폰트 선정의 기준

- 출판 시 발생할 라이선스 이슈 제거
 - 모든 폰트를 상업적으로 이용이 가능한 폰트로 선정
 - serif² 영어 폰트인 Times New Roman 저작권의 이슈
 - * 가독성이 뛰어난 범용적인 세리프체로 영문 에세이와 학위 논문의 표준 폰트로 자리잡았으나
 - * 상업용으로 쓸 땐 저작권 면책 조건을 확인해야 함
 - sans-serif³ 영어 폰트인 Helvetica 저작권의 이슈
 - * 대표적인 산세리프 글꼴로 20세기에 널리 쓰였음

²serif 폰트는 바탕체 폰트를 의미합니다. 글자의 획에서 부리처럼 날카롭게 튀어나온 부분을 세리프(serif)라 합니다. 영어는 로만체, 한국어의 경우에는 과거에 명조체라는 이름으로 통용되던 바탕체 서체를 의미합니다.

³serif 폰트는 돌움체 폰트를 의미합니다. 프랑스어로 sans는 “~가 없이”라는 뜻으로, sans-serif는 세리프가 없는 서체입니다. 한국어의 경우에는 과거에 고딕체라는 이름으로 통용되던 돌움체 서체를 의미합니다.

- 출판 인쇄물의 미적 완성도
 - 가독성, 심미성 등

2.3.2.2 한국어 폰트

- serif 폰트:
 - **KoPubWorld바탕체_Pro**
- sans-serif 폰트:
 - **KoPubWorld돋움체_Pro**

2.3.2.3 영어 폰트

- serif 폰트:
 - Times New Roman 대체 폰트
 - **STIX Two Text**
- sans-serif 폰트:
 - **Nimbus Sans L**

2.3.2.4 한자 폰트

- **KoPubWorld돋움체_Pro**

2.3.2.5 Mono space 폰트

- mono space 폰트
 - **D2Coding**
 - * 영어/한국어 모두 동일한 폰트 사용
 - * 영어 mono space와 한국어 mono space를 혼용하면 자간격이 틀어짐

2.3.2.6 Math 폰트

- STIX Two Math

2.3.2.7 R 그래픽스 이미지 폰트

- 한국어 폰트
 - 나눔스퀘어
- 영어 폰트

--- bitPublish를 이용하여

- Nimbus Sans L

2.3.2.8 주의사항

모든 폰트는 `_extensions/bit2r/bitPublish/fonts` 디렉토리에 위치합니다. 그래서 따로 설치할 필요도 없고, 이미 설치된 폰트도 중복으로 위치하게 됩니다. 그리고 KoPubWorld 바탕체_Pro와 KoPubWorld돋움체_Pro 폰트를 사용하기 위해서는 한국출판인회의 홈페이지인 <https://forms.gle/aQU7b3EoaF53zMKaA>에 사용자 정보를 등록 후 무료로 이용하실 수 있습니다. 이 경우에는 https://www.kopus.org/wp-content/uploads/2021/04/서체_라이선스.pdf 페이지의 라이선스를 일독하시기 바랍니다.

2.3.3 머리글 바닥글

머리글(header)과 바닥글/footer)는 fancyhdr 패키지를 사용해서 정의하였습니다.

`_extensions/bit2r/bitPublish/_extension.yml` 파일에 다음과 같이 설정되어 있습니다.

```
\usepackage{fancyhdr}
\pagestyle{fancy}

중간 생략

\fancyhf{}
\fancyhead[EL]{\changesize \numberfont --- bitPublish를 이용하여}
\fancyhead[OR]{\changesize \numberfont 한글 책 조판하기 ---}
\fancyfoot[EL]{\pagefont\thepage}{\hskip4mm}{\changesize \leftmark}}
\fancyfoot[OR]{\changesize \rightmark}{\hskip4mm}{\pagefont\thepage}}
```

만약에 머리글과 바닥글의 모양을 바꾸려면 직접 LaTeX 스크립트를 수정하면 됩니다. 책의 제목을 바꾸기 위해서는 fancyhead의 EL과 OR을 변경하면 됩니다.

2.3.4 글줄 사이

줄간격, 행간 등으로 부르는 글줄 사이(line spacing)는 1.5로 설정되었습니다. 수정이 필요하다면, `_extensions/bit2r/bitPublish/bitPublish.tex` 파일의 다음 행을 수정합니다.

```
%% 줄간격 정의
\linespread{1.5}
```

2.4 bitPublish 서식 이용하기

bitPublish는 본문에서 예제, 연습문제, 주의, 정보, 인용 등을 표현할 수 있는 여러 LaTeX 서식을 지원합니다. 이 서식은 다음과 같은 구조의 LaTeX 환경(enviroment)으로 제공됩니다.

```
\begin{enviroment name}[optional argument]{main argument}
  text of enviroment
\end{enviroment name}
```

또한 다음과 같은 LaTeX 명령(command)으로도 제공합니다.

```
\command name[optional argument]{main argument}
```

2.4.1 멀티 아웃 포맷

bitPublish 서식은 전자출판이 아닌, 종이로 된 서책의 출판을 염두에 둔 것이어서 LaTeX 을 기반으로 서식을 정의합니다. 그러나 이 경우에는 종이 서책 출판을 위한 PDF 포맷은 지원하지만 전자책 위한 HTML 포맷을 지원하지 못하는 단점이 있습니다.

bitPublish 초기 버전에서 대부분의 서식은 LaTeX 출력을 지원하지만, 전자책을 위한 HTML 출력도 동시에 지원하는 멀티 아웃 포맷(multiple output formats)으로 업그레이드하려 합니다.

2.4.2 사용자 정의 블록

bitPublish 서식의 대표적인 것에 커스텀 블록(custom blocks)^[4]이 있습니다. 사용자 정의 블록은 보고서나 책의 일부 콘텐츠를 블록으로 정의하여 본문보다 돋보이게 하여 독자가 핵심 포인트를 쉽게 이해할 수 있도록 주목도를 높이는 서식입니다.

사용자 정의 블록은 팬독의 fenced Div blocks 기능을 사용합니다. 이 기능은 Div 블록을 HTML과 LaTeX 모두로 변환해줍니다.

--- bitPublish를 이용하여

다음의 팬독 사용자 정의 블록은,

```
::: {.verbatim data-latex=""}  
We show some _verbatim_ text here.  
:::
```

LaTeX 출력일 경우에는 다음과 같이 변환하고,

```
\begin{verbatim}  
We show some \emph{verbatim} text here.  
\end{verbatim}
```

HTML 출력일 경우에는 다음과 같이 변환해 줍니다.

```
<div class="verbatim">  
We show some <em>verbatim</em> text here.  
</div>
```

2.4.3 정보 블록

정보 블록은 멀티 아웃 포맷을 지원하는데, 다음과 같은 네 개의 유형을 지원합니다. 네 개의 유형은 HTML과 LaTeX에서 우주 똑같은 모양은 아니지만, 거의 유사한 모습으로 출력됩니다.

- 정보
 - information
- 주의
 - caution
- 경고
 - warning
- 팁
 - tip

멀티 아웃 포맷을 지원하기 위한 팬독 사용자 정의 블록의 구문은 다음과 같습니다. 그리고 정보 블록에 들어갈 내용은 마크다운 문법으로 기술해야 합니다.

```

::: {.infobox .<type> data-latex="{<type>}{<title>}"}
text of infobox
:::

```

2.4.3.1 정보 블록

본문에서 서술하는 것보다 **유용한 정보**를 기술하는 데 사용하는 정보(information) 블록 블록은 다음처럼 사용합니다.

블록 청크의 옵션 중에 data-latex는 LaTeX에 전달하는 인수입니다. 정보(information)를 의미하는 .information와 {information}가 중복으로 기술된 것은, 멀티 아웃 포맷을 위한 것이므로 반드시 이 형식을 따라야 합니다.

```

::: {.infobox .information data-latex="{information}{information의 용도}"}
information은 독자에게 전하고 싶은 정보나 아이디어를 기술하는 데 사용합니다.
짙은 회색 테두리에 정보를 나타내는 아이콘을 배치하여 주위를 환기시킵니다.
환경의 내부에는 Markdown으로 문장을 기술합니다.
:::

```

information의 용도



information은 독자에게 전하고 싶은 정보나 아이디어를 기술하는 데 사용합니다. 짙은 회색 테두리에 정보를 나타내는 아이콘을 배치하여 주위를 환기시킵니다. 환경의 내부에는 **Markdown**으로 문장을 기술합니다.

다음은 정보 블록에서 제목을 생략한 예제입니다.

```

::: {.infobox .information data-latex="{information}{}" }
information은 독자에게 전하고 싶은 정보나 아이디어를 기술하는 데 사용합니다.
짙은 회색 테두리에 정보를 나타내는 아이콘을 배치하여 주위를 환기시킵니다.
환경의 내부에는 Markdown으로 문장을 기술합니다.
:::

```

i information은 독자에게 전하고 싶은 정보나 아이디어를 기술하는 데 사용됩니다. 짙은 회색 테두리에 정보를 나타내는 아이콘을 배치하여 주위를 환기시킵니다. 환경의 내부에는 **Markdown**으로 문장을 기술합니다..

infobox를 팬독 블록으로 정의할 경우 HTML에서는 제목이 출력되지 않는 버그⁴가 있습니다. 이 경우에는 다음처럼 기술하면 제목이 출력되는 멀티 아웃 포맷이 가능합니다. 즉, 본문에서 제목을 표현하는 방법입니다.

```

::: {.infobox .information data-latex="{information}{}" }
**information의 용도**

information은 독자에게 전하고 싶은 정보나 아이디어를 기술하는 데 사용됩니다.
짙은 회색 테두리에 정보를 나타내는 아이콘을 배치하여 주위를 환기시킵니다.
환경의 내부에는 Markdown으로 문장을 기술합니다.
:::

```

information의 용도

i information은 독자에게 전하고 싶은 정보나 아이디어를 기술하는 데 사용됩니다. 짙은 회색 테두리에 정보를 나타내는 아이콘을 배치하여 주위를 환기시킵니다. 환경의 내부에는 **Markdown**으로 문장을 기술합니다.

또 하나의 방법은 bitPublish가 제공하는 사용자 정의 R 함수 custom_block()를 사용하는 방법이 있습니다.

```

custom_block(msg, type, title)

```

이 방법도 멀티 아웃 포맷을 지원합니다. 다음과 같은 knitr 청크를 이용합니다.

```

```{r}
#| echo: false
#| results: asis
custom_block("
information은 본문에서 서술하는 것보다 유용한 정보를 기술하는 데 사용됩니다.

```

<sup>4</sup>원인을 알고 있으나 저자의 웹 프로그래밍 기술적인 한계로 해결하지 못했음

파란색 테두리와 정보를 나타내는 아이콘을 배치하여 주위를 환기시킵니다.  
 블록 내부를 채울 텍스트는 **Markdown 문법**으로 기술합니다.  
`”, title = "information의 용도”, type = "information")`  
`...`

### information의 용도



**information**은 본문에서 서술하는 것보다 **유용한 정보**를 기술하는 데 사용됩니다. 파란색 테두리와 정보를 나타내는 아이콘을 배치하여 주위를 환기시킵니다. 블록 내부를 채울 텍스트는 **Markdown 문법**으로 기술합니다.

사용자 정의 함수인 `custom_block()`를 사용하기 위해서는 사전에 다음처럼 `init_environments.R` 파일을 로드해야 합니다.

```
```{r}
#| label: init
#| include: false
source(here::here("_extensions", "bit2r", "bitPublish", "init_environments.R"))
```
```

#### 2.4.3.2 주의 블록

당부하고 싶은 **주의사항**을 기술하는 데 사용하는 주의(caution) 블록은 다음처럼 사용됩니다.

```
::: {.infobox .caution data-latex="{caution}{}}
caution의 용도
```

**\*\*caution\*\***은 당부하고 싶은 **\*\*주의사항\*\***을 기술하는 데 사용됩니다.  
 주황색 테두리에 주의를 나타내는 아이콘을 배치하여 주위를 환기시킵니다.  
 블록 내부를 채울 텍스트는 **Markdown 문법**으로 기술합니다.  
`:::`

#### caution의 용도



**caution**은 당부하고 싶은 **주의사항**을 기술하는 데 사용합니다. 주황색 테두리에 주의를 나타내는 아이콘을 배치하여 주위를 환기시킵니다. 블록 내부를 채울 텍스트는 **Markdown 문법**으로 기술합니다.

#### 2.4.3.3 경고 블록

주의사항보다 더 치명적인 문제를 야기할 **경고**를 기술하는 데 사용하는 경고(warning) 블록은 다음처럼 사용합니다.

```
::: {.infobox .warning data-latex="{warning}{}"}
```

**\*\*warning의 용도\*\***

**\*\*warning\*\***은 주의사항보다 더 치명적인 문제를 야기할 **\*\*경고\*\***를 기술하는 데 사용합니다. 붉은색 테두리에 폭탄을 표현한 아이콘을 배치하여 주위를 환기시킵니다.

블록 내부를 채울 텍스트는 **\*\*Markdown 문법\*\***으로 기술합니다.

```
:::
```

#### warning의 용도



**warning**은 주의사항보다 더 치명적인 문제를 야기할 **경고**를 기술하는 데 사용합니다. 붉은색 테두리에 폭탄을 표현한 아이콘을 배치하여 주위를 환기시킵니다. 블록 내부를 채울 텍스트는 **Markdown 문법**으로 기술합니다.

#### 2.4.3.4 팁 블록

유용하게 사용할 수 있는 **아이디어나 비법**을 설명하는 데 사용하는 팁(tip) 블록은 다음처럼 사용합니다.

```
::: {.infobox .tip data-latex="{tip}{}"}
```

**\*\*tip의 용도\*\***

**\*\*tip\*\***은 유용하게 사용할 수 있는 **\*\*아이디어나 비법\*\***을 기술하는 데 사용합니다.

초록색 테두리에 아이디어를 나타내는 전구 아이콘을 배치하여 주위를 환기시킵니다.  
블록 내부를 채울 텍스트는 **\*\*Markdown 문법\*\***으로 기술합니다.

:::

#### tip의 용도



**tip**은 유용하게 사용할 수 있는 **아이디어나 비법**을 기술하는 데 사용합니다. 초록색 테두리에 아이디어를 나타내는 전구 아이콘을 배치하여 주위를 환기시킵니다. 블록 내부를 채울 텍스트는 **Markdown 문법**으로 기술합니다.

#### 2.4.4 이야기 박스

shadequote 환경을 이용합니다. 이야기 박스 구문은 다음과 같습니다.

```
\begin{shadequote}[<alignment>]{<author>}
 text of quote
\end{shadequote}
```

다음처럼 괄호를 비우면, 저자의 이야기를 표시합니다.

```
\begin{shadequote}{}
나는 통계계산이 수리영역인 줄만 알았다. 그런데, 이제는 논리적인 사고도 필요한
논리영역임을 느낀다. 그래서 논리적 사고로 통계적 데이터 분석을 위한 성능 좋은
연장이 필요하기 시작했다.
\end{shadequote}
```

\\ 통상적으로 많은 사람들은 ‘수준을 등급의 높고 낮음을 의미하는 가치의 척도’로 인식하고 있다. 그러나 통계학계에서는 levels을 수준으로 직역하여 사용해 왔다. 혼동스럽다면, 수준을 **범주형 데이터의 범주로 이해**하기 바란다. ||

명사의 명언을 인용하여 이야기할 경우에는 명사의 이름을 첫째 괄호에 기입합니다. 이 예제는 명사의 이름을 오른쪽 정렬하였습니다.

--- bitPublish를 이용하여

```
\begin{shadequote}[r]{스피노자}
자신은 할 수 없다고 생각하는 동안 사실은 그것을 하기 싫다고 다짐하고 있는 것이다.
\end{shadequote}
```

\\ 자신은 할 수 없다고 생각하는 동안 사실은 그것을 하기 싫다고 다짐하고 있는 것이다. ||

스피노자

### 2.4.5 타이틀 박스

앞의 예제에서 ‘학습목표’를 정의한 박스 서식을 타이틀 박스라 부르겠습니다. 여러 용도로 사용될 수 있어 목적으로 이름을 특정할 수 없기 때문입니다. 학습 시나리오, 학습의 목표든 상관없습니다.

다음과 같은 LaTeX 명령으로 제공합니다.

```
\snbox{text of title}{title fill color}{box fill color}{text of box}
```

다음 명령은 학습 목표라는 타이틀을 갖는 박스를 생성합니다.

```
\snbox{학습 목표}{blue!30}{blue!10}{데이터의 분류 체계로서의 척도를 이해하고, 이를 기반으로 한 R 데이터 객체를 이해한다. CSV 파일을 읽고, 데이터 프레임 객체로 CSV 파일을 생성할 수 있다.}
```

#### 학습 목표

데이터의 분류 체계로서의 척도를 이해하고, 이를 기반으로 한 R 데이터 객체를 이해한다. CSV 파일을 읽고, 데이터 프레임 객체로 CSV 파일을 생성할 수 있다.

### 2.4.6 예제

example 환경을 이용합니다. 인용의 구문은 다음과 같습니다.

```
\begin{example}{number of example}
text of example
```

```
\end{example}
```

그러나 예제는 다음처럼 생각보다 복잡합니다.

- 장(chapter)에서 여러 예제가 있어서, 번호를 매겨야 한다.
  - 번호는 자동으로 채번되어야 추가 및 제거 시 번호의 오류를 줄인다.
- 예제를 본문이나 다른 예제에서 참조하는 경우가 있다.
  - 크로스-레퍼런스(cross-reference)를 지원해야 한다.

예제를 위해서 다음과 같이 장(chapter)의 시작부분에 환경변수를 정의해야 합니다. 장의 라벨과 예제의 순번을 위한 카운터를 설정했습니다.

```
\label{chap:bitpublish} % 장 라벨 정의
\newcounter{exam_num_bitpub} % 새로운 카운터 생성
\setcounter{exam_num_bitpub}{0} % 카운터 값 0으로 초기화
```

다음은 예제를 위한 환경 설정과 예제를 구현하는 예제입니다. 크로스-레퍼런스를 위해서 예제에 대해서 ex1이라는 라벨을 example 환경 안에서 정의한 것을 주의깊게 보십시오.

```
\addtocounter{exam_num_bitpub}{1} % 예제를 위한 카운터 1 증가
\begin{example}{\ref{chap:bitpublish}.\arabic{exam_num_bitpub}}
\examplelabel{ex1}{\ref{chap:bitpublish}.\arabic{exam_num_bitpub}}
bitPublish의 서식 중에서 예제, 연습문제, 주의, 정보, 인용을 만들어 보아라.
\end{example}
```

### 예제 2.1.

bitPublish의 서식 중에서 예제, 연습문제, 주의, 정보, 인용을 만들어 보아라.

ex1이라는 라벨을 레퍼런스하는 방법은 다음과 같습니다.

예제 `\ref{ex1}`를 레퍼런스하기 위해서는 레퍼런스 명령 `\ref{}`를 사용합니다.

예제 2.1를 레퍼런스하기 위해서는 레퍼런스 명령 `\ref{}`를 사용합니다.

--- bitPublish를 이용하여

## 2.4.7 연습문제

기술서나 학습서의 경우에는 연습문제를 제공하는 경우가 많습니다. bitPublish에서 연습문제 서식은 Exercise 환경을 이용합니다.

```
\begin{Exercise}
 text of exercise
\end{Exercise}
```

또한 연습문제의 내용을 위해서 tasks 환경과 task 명령을 사용할 수 있습니다.

```
\begin{tasks}[label](1)
 \task text of task
 \task text of task
\end{tasks}
```

다음 명령은 예제장에서 사용한 연습문제를 기술한 LaTeX 스크립트입니다.

```
\begin{Exercise}\label{EX2}
\noindent 1. 다음의 데이터를 R의 객체로 만들어 보아라.
\begin{tasks}[label=(\arabic*)](1)
 \task 합격, 불합격, 불합격, 합격, 불합격
 \task 1등급, 2등급, 3등급, 2등급, 5등급, 3등급, 4등급, 4등급, 3등급, 3등급
 \task 12cm, 23cm, 32cm, 22cm, 19cm, 21cm, 20cm
\end{tasks}

\noindent 2. 다음의 데이터 객체를 외부 파일로 출력해 보아라.
\begin{tasks}[label=(\arabic*)](1)
 \task R의 datas 패키지에 포함되어 있는 women 데이터 프레임을 women.csv 파일로 출력해 보아라.
 \task (1)의 파일을 다시 읽어 들어서 women2라는 이름의 데이터 프레임을 만들어라.
\end{tasks}
\end{Exercise}
```

## 2.1 연습문제

1. 다음의 데이터를 R의 객체로 만들어 보아라.

- (1) 합격, 불합격, 불합격, 합격, 불합격
- (2) 1등급, 2등급, 3등급, 2등급, 5등급, 3등급, 4등급, 4등급, 3등급, 3등급
- (3) 12cm, 23cm, 32cm, 22cm, 19cm, 21cm, 20cm

2. 다음의 데이터 객체를 외부 파일로 출력해 보아라.

- (1) R의 `datas` 패키지에 포함되어 있는 `women` 데이터 프레임을 `women.csv` 파일로 출력해 보아라.
- (2) (1)의 파일을 다시 읽어 들여서 `women2`라는 이름의 데이터 프레임을 만들어라.

만약에 연습문제를 별도의 페이지에서 시각하고 싶다면 `\clearpage` 명령어를 사용하면 됩니다.

## 2.5 bitPublish 활용을 위한 팁

bitPublish는 LaTeX 기반의 솔루션이기 때문에, 좀더 유용하게 사용하기 위해서 LaTeX에 대한 지식을 요구합니다. R 환경에서는 `knitr` 패키지를 이용해서 Quarto 문서를 마크다운 문서로 렌더링하기 때문에 `knitr` 체크에 대한 지식도 필요합니다.

그러나 너무 걱정하지 마십시오. 심도있는 지식이 아닌, 기초적인 지식으로도 충분히 책을 저작할 수 있습니다. 이 절에서는 요긴하게 사용된 LaTeX 팁을 소개합니다.

### 2.5.1 이미지 삽입

#### 2.5.1.1 이미지 파일

이미지 파일을 삽입하기 위해서는 다음처럼 `knitr` 패키지의 `include_graphics()` 함수를 사용하는 것을 추천합니다. 이 때, `knitr` 체크는 다음과 같이 사용합니다.

```
```{r}
#| label: fig-label
#| echo: false
#| out-width: "90%"
#| fig-align: "center"
#| fig-cap: "이미지 캡션"
```

--- bitPublish를 이용하여

```
#| fig-pos: "htb!"  
  
knitr::include_graphics("이미지 파일이름.png")  
...  
...  
...
```

이 예제는 다음과 같은 작업을 수행합니다.

- ‘이미지 파일이름.png’ 파일을 본문에 삽입하는데,
 - 이미지의 크기는 본문 너비의 90%에 해당하게 적용하며, 이미지는 가운데 정렬로 배치함
- 이미지의 캡션은 “이미지 캡션”으로 정의하며,
- 크로스-레퍼런스를 위한 라벨의 이름은 ‘fig-label’로 설정하는데, 라벨의 접두어 ‘fig-’를 사용한 ‘fig-라벨이름’의 포맷을 권장함
- **이미지의 페이지 배치는**
 1. 이미지를 정확히 해당 위치(here)에 일단 배치,
 2. 여의치 않으면, 그 다음 페이지 가장 윗부분에 배치,
 3. 만약 이것도 여의치 않으면, 그 다음 페이지 가장 아랫 부분에 배치.

이미지의 페이지 배치는 다음과 같은 fig-pos[6] 옵션으로 지정합니다.

플롯 지정자	의미
h	here의 약자로, 본문에 위치한 그 자리에 이미지 배치
t	top의 약자로, 이미지가 위치한 페이지의 가장 윗부분에 배치. 크기가 맞지 않으면 다음 페이지의 가장 윗쪽에 배치
b	bottom의 약자로, 이미지가 위치한 페이지 가장 아래에 이미지 배치. 크기가 맞지 않으면 다음 페이지의 가장 아래에 배치
p	이미지가 텍스트 없는 단독 페이지를 따로 만들어 배치
!	적절한 위치 결정을 위해, 적용한 지정자 재정의. 지정자가 나열된 순서에 따라 적절한 위치에 이미지 삽입

플롯 지정자 예제에서 사용한 “htb!”는 일단 이미지를 정확히 그 위치(h)에 배치하려고 하고, 여의치 않으면 다음은 위(t)이기 때문에 그 다음 페이지 가장 윗부분에 배치합니다. 만약 그것도 여의치 않으면 그 다음은 아래(b)이기 때문에 그 다음 페이지 가장 아랫 부분에 배치합니다.

그런데 bitPublish에서 fig-pos의 기본값⁵이 “htb!”로 설정되었기 때문에 이 설정을 생략해도 적당한 페이지에 이미지를 배치합니다. fig-align의 기본값도 “center”입니다.

fig-pos과 label

웹 페이지로 배포하는 문서는 페이지로 구분되지 않기 때문에 이미지는 항상 그(here) 자리에 출력됩니다.



그래서 본문에서 “다음 그림처럼”과 같은 표현이 가능하지만, 페이지가 구분되는 책에서는 이 표현은 적절하지 않습니다. 왜냐하면 페이지 상황에 따라 이미지가 다음 페이지로 넘어갈 수 있는 등 저작 과정에서는 예측이 안됩니다.

그래서 'fig-pos'의 값을 “htb!”으로 설정하고 'label'을 지정하여 크로스-레퍼런스를 이용해서, “그림 2.1처럼”과 같이 표현하는 것이 좋습니다.

2.5.1.2 R 그래픽스

R 스크립트로 그리는 R 그래픽스는 knitr 청크(chunk)를 이용해서, R 스크립트를 실행하여 삽입합니다.

```
`` `{r}
#| label: fig-hist
#| echo: false
#| fig-width: 6
#| fig-height: 4
#| out-width: "70%"
#| fig-cap: "정규분포 히스토그램"
set.seed(123)
hist(rnorm(500), col = "lightblue", main = "정규분포의 히스토그램")
````
```

그런데 R 플롯에 한글이 포함될 경우에 그림 2.2처럼 한글이 출력되지 않는 문제가 발생합니다.

한글을 정상적으로 출력하기 위해서는 문서의 맨 앞에 다음 스크립트를 추가합니다.

<sup>5</sup>“htb!”을 기본값으로 설정하기 위해서는, 뒷부분에서 설명할 플롯에서의 한글 폰트 적용을 위한 `init_environments.R` 파일을 로드하는 설정을 전제로 합니다.

--- bitPublish를 이용하여

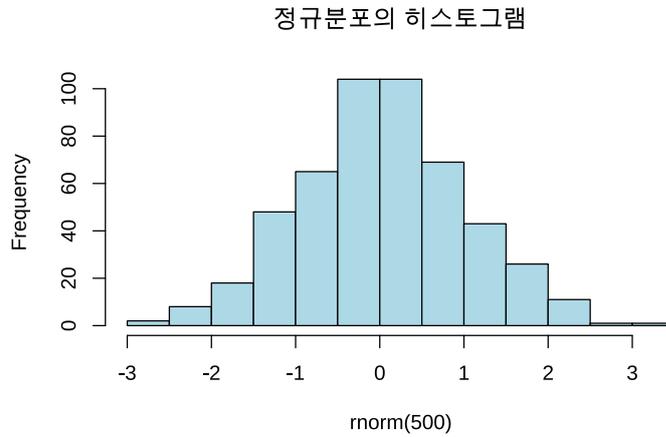


그림 2.2: 정규분포 히스토그램

```
```{r init, include=FALSE}
source(here::here("_extensions", "bit2r", "bitPublish", "init_environments.R"))
```
```

그런 후, R 플롯을 그리는 knitr 청크에 `nanum` 옵션의 값을 `true`로 추가합니다. 이 설정은 나눔스퀘어 폰트를 사용해서 R 플롯을 시각화한다는 것을 의미합니다. 다음 스크립트는 그림 2.3과 같은 R 그래픽 플롯을 출력합니다.

```
```{r}
#| label: fig-hist2
#| echo: false
#| fig-width: 6
#| fig-height: 4
#| out-width: "70%"
#| fig-cap: "정규분포 히스토그램"
#| nanum: true
set.seed(123)

hist(rnorm(500), col = "lightblue", main = "정규분포의 히스토그램")
```
```

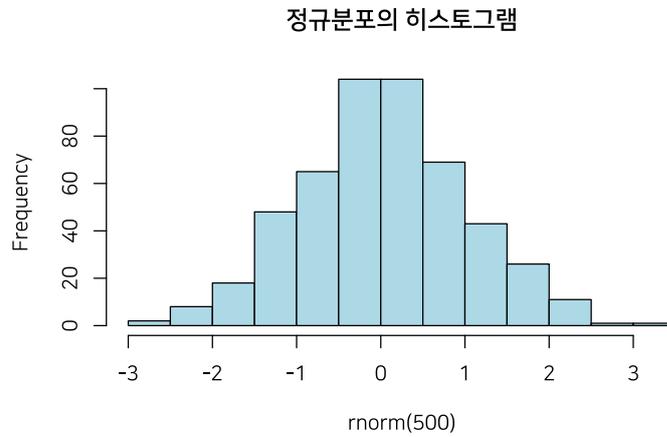


그림 2.3: 정규분포 히스토그램

만약 한글이 포함되지 않은 플롯에서 라틴 문자에 사용할 폰트로 Nimbus Sans L 폰트를 사용하려면, `nimbus` 옵션의 값을 `true`로 추가하면 됩니다.

### 2.5.1.3 ggplot2 그래픽스

`ggplot2` 그래픽스도 별도의 설정없이 한글을 출력합니다. 테마는 `hrbrthemes` 패키지의 `theme_ipsum()` 함수를 사용하며, 폰트는 R 그래픽스와 동일한 방법으로 나눔스퀘어 폰트를 사용할 수 있습니다.

다음 스크립트는 그림 2.4와 같은 `ggplot2` 그래픽스 플롯을 출력합니다.

```
```\r}
#| label: fig-hist3
#| echo: false
#| fig-width: 6
#| fig-height: 4
#| out-width: "70%"
#| fig-cap: "ggplot2 정규분포 히스토그램"
#| nanum: true
set.seed(123)
random_norm <- data.frame(rnd = rnorm(500))
```

--- bitPublish를 이용하여

```
ggplot(random_norm, aes(x = rnd)) +  
  geom_histogram(color = "darkblue", fill = "lightblue") +  
  labs(title = "ggplot2의 히스토그램",  
        subtitle = "500개 정규분포 난수")  
  ...
```



그림 2.4: ggplot2 정규분포 히스토그램

2.5.2 표 삽입

본문에 표를 삽입하는 방법에 다음처럼 3가지 방법을 사용할 수 있습니다.

- 마크다운을 이용한 표
 - 캡션의 핸들링과 라벨의 지정에 취약해서 권하지 않음
 - 그러나, 캡션과 라벨을 사용하지 않을 때에는 유용함
- LaTeX을 이용한 표
 - 가장 세밀하게 조정할 수 있으나, LaTeX에 대한 지식이 필요함
- R을 이용한 표
 - R에 익숙하면 쉽게 사용할 수 있음

2.5.2.1 마크다운을 이용한 표

마크다운을 이용한 표 생성의 경우에는 캡션이 표 내부에 위치하고, 라벨을 지정할 수 없어서 표를 삽입하는 방법으로 적절하지 않습니다.

다음과 같이 표를 정의합니다. 캡션과 라벨을 사용하지 않을 때에는 유용합니다.

```
| 기본값 | 왼쪽 정렬 | 가운데 정렬 | 오른쪽 정렬 |
|-----|:-----|:-----:|:-----:|
|내용 1 |내용 2   |내용 3   |내용 4   |
|내용 5 |내용 6   |내용 7   |내용 8   |
|내용 9 |내용 10  |내용 11  |내용 12  |
```

Table: 마크다운 기반 표

기본값	왼쪽 정렬	가운데 정렬	오른쪽 정렬
내용 1	내용 2	내용 3	내용 4
내용 5	내용 6	내용 7	내용 8
내용 9	내용 10	내용 11	내용 12

표 2.3: 마크다운 기반 표

2.5.2.2 LaTeX을 이용한 표

LaTeX을 이용한 표를 삽입하는 방법은 LaTeX 스크립트를 기술하는 것입니다. 다음처럼 테이블을 생성하는 LaTeX 스크립트를 기술합니다.

```
\begin{table}[htb!]
\centering
\begin{tabular}[]{\lrcr}
\toprule
기본값 & 왼쪽 정렬 & 가운데 정렬 & 오른쪽 정렬 \\
\midrule
내용 1 & 내용 2 & 내용 3 & 내용 4 \\
내용 5 & 내용 6 & 내용 7 & 내용 8 \\
내용 9 & 내용 10 & 내용 11 & 내용 12 \\
\bottomrule
\end{tabular}
\caption{\label{tab-latex}LaTeX 기반 표}
```

```
\end{table}
```

기본값	왼쪽 정렬	가운데 정렬	오른쪽 정렬
내용 1	내용 2	내용 3	내용 4
내용 5	내용 6	내용 7	내용 8
내용 9	내용 10	내용 11	내용 12

표 2.4: LaTeX 기반 표

2.5.2.3 R을 이용한 표

R을 이용해서 표를 만드는 방법에 대표적인 것이 knitr 패키지의 `kable()` 함수와 `kableExtra` 패키지의 `kable_styling()` 함수를 사용하는 방법입니다. 그러나 이 방법은 캡션의 위치가 표의 위에 위치하는 것과 아직 Quarto에서 크로스-레퍼런스가 지원되지 않는 관계로 `gt` 패키지의 `gt()` 함수를 사용합니다. 이 함수로 크로스-레퍼런스가 지원되지 않았으나, `as_latex_with_caption()` 함수가 이를 지원합니다. 이 함수는 bitPublish에서 제공하는 사용자 정의 함수입니다.

다음은 표 2.5를 생성합니다.

```
```{r}
#| echo: false
tab <- tibble::tribble(
 ~기본값, ~왼쪽 정렬`, ~가운데 정렬`, ~오른쪽 정렬`,
 "내용 1", "내용 2", "내용 3", "내용 4",
 "내용 5", "내용 6", "내용 7", "내용 8",
 "내용 9", "내용 10", "내용 11", "내용 12"
)

tab |>
 gt::gt() |>
 gt::cols_align(align = "left", columns = `왼쪽 정렬`) |>
 gt::cols_align(align = "center", columns = `가운데 정렬`) |>
 gt::cols_align(align = "right", columns = `오른쪽 정렬`) |>
 as_latex_with_caption(label = "tab-r", caption = "R 기반 표")
```

...

기본값	왼쪽 정렬	가운데 정렬	오른쪽 정렬
내용 1	내용 2	내용 3	내용 4
내용 5	내용 6	내용 7	내용 8
내용 9	내용 10	내용 11	내용 12

표 2.5: R 기반 표

### 2.5.3 수식 입력

수식 입력은 Latex의 수학 공식을 입력하는 TeX 문법을 따릅니다.

인라인 수식의 경우에는  $$ 와  $$  사이에서 수식을 정의하고, 한 라인 전체에 수식을 사용하기 위해서는  $$ 와  $$  사이에서 수식을 정의합니다.

데이터가  $x_1, x_2, x_3, \dots, x_n$ 일 때, 도수분포표는 다음과 같이 만듭니다.

데이터가  $x_1, x_2, x_3, \dots, x_n$ 일 때, 도수분포표는 다음과 같이 만듭니다.

누적상대도수:

$$

$CF_k = \sum_{i=1}^k f_i^R = \frac{F_k}{n}$

$$

누적상대도수:

$$CF_k = \sum_{i=1}^k f_i^R = \frac{F_k}{n}$$

### 2.5.4 찾아보기 등록

LaTeX의 찾아보기(인덱싱) 기능을 활성화하기 위해, 이미 bitPublish는 프리엠블에 다음처럼 makeidx 패키지를 로드하고 있습니다.

`\usepackage{makeidx}`

또한 프리엠블에서는 찾아보기 기능을 활성화하기 위해서 다음의 명령을 포함했습니다.

--- bitPublish를 이용하여

```
\makeindex
```

그리고 문서에 찾아보기 색인을 표시하기 위한 다음의 명령어를 사용해야 하는데,

```
\printindex
```

이 명령어는 `_extensions/bit2r/bitPublish/_extension.yml` 파일에 다음과 같이 설정되어 있습니다.

```
include-after-body:
- text: |
 \printindex
```

그러므로 찾아보기에 색인 등록을 위해서는 여러분이 작성할 본문에 다음의 인덱스 명령을 사용하기만 하면 됩니다.

```
\index{key}
```

다음 예는 찾아보기 색인에 척도를 등록합니다. 그리고 LaTeX이 알아서 해당 문장이 위치한 페이지를 찾아보기 색인과 연결해 줍니다.

그 현상에 숫자를 부여한 것을 척도 `\index{척도}`라 한다.

#### 찾아보기에서의 언더라인의 표현



LaTeX에서는 언더라인()은 특수문자로 인식합니다. 그래서 문자로서의 언더라인을 입력하려면 역슬래시(\)를 붙여야 합니다. 그러므로 `\index{key}` 명령의 key 값에 언더라인이 있다면 다음처럼 역슬래시를 붙여 표현해야 합니다.

```
\index{include_graphics()}
```

### 2.5.5 참고문헌 등록

참고문헌(bibliography) 등록은 LaTeX의 `biblatex` 패키지를 이용합니다. 이 패키지를 사용할 수 있는 설정은 이미 되어 있으니, 'bitPublish를 이용하여 한글 책 조판하기' 예제를 위한 다음의 `_quarto.yml`을 참고하여 참고문헌 등록을 적용합니다.

```

book:
 title: "bitPublish를 이용하여 한글 책 조판하기"
 chapters:
 - index.qmd
 - chap_exams.qmd
 - chap_intro_bitpublish.qmd
 - references.qmd
 bibliography: references.bib
 link-citations: false

```

상기 YAML 헤더에서 chapters 옵션은 책에 장을 추가하는데 ‘references.qmd’ 파일이 참고 문헌 등록에 해당하는 파일입니다. 그 내용은 다음과 같습니다. 이 설정은 참고문헌을 나타내는 시작 페이지의 제목을 “참고문헌”으로 표기합니다.

```

참고문헌 {.unnumbered}

::: {#refs}
:::

```

그리고 참고문헌에 등록하기 위해서는 YAML 헤더에 bibliography 옵션에 기술된 ‘references.bib’ 파일에 참고문헌을 정의합니다.

다음은 ‘references.bib’ 파일에 정의된 일부분으로 참고한 도서와 웹 사이트의 예입니다.

```

@book{rstat,
 author = {유충현, 이상호},
 title = {R을 이용한 통계학 이해},
 year = {2013},
 publisher = {자유아카데미}
}

@misc{quarto,
 author = {posit},
 title = {Welcome to Quarto},

```

--- bitPublish를 이용하여

```
howpublished = {\url{https://quarto.org/}},
note = {Accessed: 2023-05-15}
}
```

‘references.bib’ 파일에 참고문헌을 정의했다고 ‘참고문헌’이라는 제목의 참고문헌에 등록되지 않습니다. 참고문헌에 등록하기 위해서는, 참고한 본문 내용에 인용(citation) 명령을 적용해야 합니다.

다음과 같은 LaTeX 인용 명령을 사용합니다.

```
\cite{key}
```

다음의 `\cite` 명령은 ‘참고 문헌’ 페이지에 참고한 문헌에 대한 정보를 기술하고, 인용된 본문에 링크를 걸어줍니다.

```
Quarto 홈페이지의 Quarto에 대한 소개\cite{quarto}는 다음과 같다. 이러쿵 저러쿵...
```

마크다운에도 인용을 위한 다음과 같은 명령이 있습니다.

```
[@key]
```

그러므로 다음은 앞에서의 LaTeX 명령과 동일한 작업을 수행합니다.

```
Quarto 홈페이지의 Quarto에 대한 소개[@quarto]는 다음과 같다. 이러쿵 저러쿵...
```

**\\** 인용의 경우에는 LaTeX 명령보다 마크다운 명령을 사용할 것을 권장합니다. 왜냐하면 bitPublish가 현재는 LaTeX 기반의 PDF 파일 포맷만 지원하지만, 향후 HTML, ePub 포맷과 같은 전자책 저작도 지원할 계획입니다. 호환을 위한 LaTeX 명령을 자제하는 것이 좋습니다. **||**

## 2.5.6 자동조사 처리

예제, 그림, 표 등에 라벨을 지정해서 크로스-레퍼런스를 구현하면 저작자는 번호의 증감에 신경 쓸 필요가 없습니다. 그림과 그림 사이에 또 하나의 그림을 삽입하거나, 또는 기존 그림을 제거함으로써 바뀌게 될 참조 번호를 수정할 필요가 없습니다. LaTeX이 알아서 번호를 수정해주기 때문입니다.

다음과 같이 출력되는 본문이 있다고 가정합니다.

그림 2.3과 같이 정규분포를 따르는 히스토그램을 그린다.

그런데 앞 부분에 크로스-레퍼런스를 적용한 그림 하나를 추가하면 이 문장은 다음과 같이 바뀝니다.

그림 2.4과 같이 정규분포를 따르는 히스토그램을 그린다.

그림에 크로스-레퍼런스를 적용해서 그림의 번호 3이 4로 정상적으로 변경되었는데, 조사가 어색합니다. ‘그림 2.4과’가 아니라 ‘그림 2.4와’로 출력되어야 하기 때문입니다. 왜냐하면 우리말의 조사에는 다음과 같은 규칙이 있기 때문입니다. 앞단어의 끝소리의 자소에 따라 다음처럼 조사가 결정되기 때문입니다.[5]

앞단어 끝소리	와 / 과	을 / 를	이 / 가	은 / 는	(이)라	(으)로
리을(ㄹ)	과	을	이	은	이라	로
ㄹ 아닌 중성	과	을	이	은	이라	으로
중성	와	를	가	는	라	로

LaTeX이 크로스-레퍼런스에서 계산되는 아라비아 숫자의 한글 독음 끝소리 자소에 따라 자동으로 자동조사 처리를 하기 위해서는, 크로스-레퍼런스를 위한 `\ref{}` 명령에 붙는 조사에 역 슬래시(\)붙여주면 됩니다.

다음은 크로스-레퍼런스에서 번호가 계산되었다는 전제로 자동조사 처리를 테스트한 것입니다.

그림 2.3\와 그림 2.4\은 같이 정규분포를 따르는 히스토그램을 그린다.

그림 2.3\과 그림 2.4\는 같이 정규분포를 따르는 히스토그램을 그린다.

그림 2.3과 그림 2.4는 같이 정규분포를 따르는 히스토그램을 그린다.

그림 2.3과 그림 2.4는 같이 정규분포를 따르는 히스토그램을 그린다.

--- **bitPublish**를 이용하여

## 제 3 장

# Troubleshooting

이 장에서는 ‘bitPublish’의 사용시에 발생하는 문제와 해결 방법에 대해서 다룹니다.

### 학습 목표

bitPublish 시 발생하는 문제의 원인을 파악하고 해결할 수 있다.

### 3.1 Quarto 오류

#### 3.1.1 TeXlive 버전 이슈

다음처럼 미리보기 명령이나,

```
quarto preview --to bitPublish-pdf
```

랜더링 명령에서,

```
quarto render --to bitPublish-pdf
```

아래와 같은 에러가 발생하는 경우가 있습니다.

```
updating existing packages
finding package for bclogo.sty
```

--- bitPublish를 이용하여

```
ERROR: Your TeXLive version is not updated enough to connect to the remote
repository and download packages. Please update your installation of TeXLive
or TinyTex.
```

```
Underlying message: Local TeX Live (2022) is older than remote repository (2023).
```

```
Cross release updates are only supported with
```

```
update-tlmgr-latest(.sh/.exe) --update
```

```
See https://tug.org/texlive/upgrade.html for details.
```

매년 4월 TeXLive는 원격저장소를 TeX의 새해 버전으로 업데이트합니다. 이 경우 TeX의 이전 년도 설치본은 원격 저장소에서 패키지를 다운로드/설치할 수 없습니다. 사용자가 TeXLive 또는 TeXLive는 원격저장소를 이용하는 TinyTex을 업데이트 해야 합니다.

TeXLive를 사용한다면, 다음 명령으로 TeXLive를 업데이트하세요.

```
update-tlmgr-latest --update
```

TinyTex을 사용한다면, 다음 명령으로 TinyTex을 업데이트하세요.

```
quarto update tinytex
```

## 3.2 LaTeX 오류

### 3.2.1 찾아보기 오류

다음과 같이 찾아보기 명령 `\index{}`를 사용할 때,

```
\index{include_graphics{}}
```

텍스트 내에 언더라인( )이 있을 경우 다음과 같은 에러가 발생합니다.

```
ompilation failed- Error compiling latex
```

```
Missing $ inserted.
```

```
<inserted text>
```

```
$
```

1.54 `\item include_`

`graphics(), \hyperpage{27}`

문자로서의 언더라인을 기술하기 위해서는 다음과 같이 언더라인에 역 슬래시를 붙여서 기술합니다.

`\index{include\_graphics()}`

--- **bitPublish**를 이용하여

## 제 4 장

# Version

이 장에서는 'bitPublish'의 버전에 대한 정보를 제공합니다.

### 4.1 version 0.1

#### 4.1.1 version 0.1.2

- 2023-05-27
- 예제 문서에 버전 정보를 담은 chapter 추가
  - chap\_version.qmd

### 4.2 version 0.2

#### 4.2.1 version 0.2.0

- 2023-05-29
- pdf와 html에서 공통으로 작동하는 Custom blocks 정의
  - infobox
    - \* information, caution, warning, tip
  - LaTeX or HTML output

\\ 자신은 할 수 없다고 생각하는 동안 사실은 그것을 하기 싫다고 다짐하고 있는 것이다. ||

스피노자

\\ 자신은 할 수 없다고 생각하는 동안 사실은 그것을 하기 싫다고 다짐하고 있는 것이다. ||

# 참고문헌

- [1] Allison Horst. *ÇETINKAYA-RUNDEL & LOWNDES QUARTO KEYNOTE*. <https://allisonhorst.com/cetinkaya-rundel-lowndes-quarto-keynote>. Accessed: 2023-05-17.
- [2] posit. *Creating Extensions*. <https://quarto.org/docs/extensions/creating.html>. Accessed: 2023-05-15.
- [3] posit. *Welcome to Quarto*. <https://quarto.org/>. Accessed: 2023-05-15.
- [4] Emily Riederer Yihui Xie Christophe Dervieux. *R Markdown Cookbook*. <https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html>. Accessed: 2023-05-29. 2022.
- [5] 김강수. “한글 라텍스에서 조사 이형태 교체의 자동화에 관하여”. in *The Asian Journal of TEX: 1.2* (2007), pages 153-166. URL: <http://ajt.ktug.org/2007/0102karnes.pdf>.
- [6] 미니도넛. *이미지 특정 위치에 고정하기*. <https://forestunit.tistory.com/117>. Accessed: 2023-05-20.
- [7] 이상호 유충현. *R을 이용한 통계학 이해*. 자유아카데미, 2013.

--- bitPublish를 이용하여

# 찾아보기

%>%, 9

|>, 10

as.numeric(), 6

bibtex, 40

bibliography, 40

bitPublish, 17

categorical data, 5

citation, 42

cross-reference, 29

custom blocks, 21

data, 3

dplyr, 9

EL, 20

environment, 21

factor, 6

factor(), 6

fancyhdr, 20

fancyhead, 20

footer, 20

frequency, 7

frequency table, 7

geometry, 17

ggplot2, 9, 35

gt, 38

header, 20

Helvetica, 18

hrbrthemes, 35

include\_graphics(), 31

jupyter, 16

kable(), 38

kable\_styling(), 38

kableExtra, 38

knitr, 31, 34, 38

level, 5

library(), 10

line spacing, 20

magrittr, 9

makeidx, 39

Markdown, 16

mono space, 19

multiple output formats, 21

Open Type Font, 18

OR, 20

OTF, 18

Pandoc, 16

--- bitPublish를 이용하여

Quarto, 15  
    extensions, 17  
  
sans-serif, 18  
serif, 18  
  
table(), 6  
tasks, 30  
TeXlive, 46  
theme\_ipsum(), 35  
tidyverse, 9  
Times New Roman, 18  
TinyTex, 46  
True Type Font, 18  
TTF, 18  
  
구간척도, 4  
글줄 사이, 20  
  
누적도수, 8  
데이터, 3  
도수, 7  
도수분포표, 7  
돋움체, 18  
마크다운, 16, 36  
머리글, 20  
멀티 아웃 포맷, 21  
명목척도, 4  
바닥글, 20  
바탕체, 18  
범주형 데이터, 5  
비율척도, 4  
  
사용자 정의 블록, 21  
상대도수, 8  
서열척도, 4

세리프, 18  
수준, 5  
인용, 42  
정보 블록, 22  
    경고 블록, 26  
    정보 블록, 23  
    주의 블록, 25  
    팁 블록, 26  
참고문헌, 40  
척도, 4  
측정, 4  
  
크로스-레퍼런스, 29  
파이프 연산자, 10  
판형, 17  
팬독, 16  
페이지 레이아웃, 17  
환경, 21