

Quarto를 사용한 한글책 제작의 기초

서울시립대학교 통계학과 이용희

2023-09-29

목차

서문	1
개요	1
1 책의 구성요소	3
1.1 개요	3
1.2 HTML	3
1.3 PDF	3
1.4 MS-WORD	3
2 수학 기호	5
2.1 개요	5
2.2 한글 수식	5
2.3 수식	5
2.3.1 일차연립방정식	5
2.4 Theorem	6
2.5 HTML	6
2.5.1 새로운 명령어	6
2.6 PDF	7
3 Quarto 의 출판 절차	9
3.1 사전 작업	9
3.2 Git 의 로컬 및 원격 저장소 생성	9
3.3 출판	10
3.4 quarto publish	11
3.5 github action	12
References	13

그림 목록

3.1 Github 의 원격 저장소의 Page 설정 11

표 목 록

서문

이 책은 Quarto 를 이용하여 한글책을 만들고 출판하는 간략한 방법과 팀을 설명합니다.

개요

Quarto는 출판물을 생성하기 위한 오픈 소스 도구이며, Markdown 기반으로 작성된 문서를 PDF, MS WORD 화일, 웹사이트 등 다양한 형식으로 변환할 수 있습니다.

Quarto는 다양한 프로그래밍 언어(R, python, julia 등)를 사용하여 작성된 문서를 처리하는 기능과 코드의 실행 및 결과 통합을 지원하여 데이터 분석 및 보고서 작성에 특히 유용합니다.

1 책의 구성요소

1.1 개요

1.2 HTML

1.3 PDF

1.4 MS-WORD

2 수학 기호

2.1 개요

2.2 한글 수식

부부_{ij}² = \int 엄마_{ij}² + 아빠_{ij}²

$$\text{부부}_{ij}^2 = \int \text{엄마}_{ij}^2 + \text{아빠}_{ij}^2$$

2.3 수식

2.3.1 일차연립방정식

다음과 같이 n 개의 변수 x_1, x_2, \dots, x_n 에 대한 m 개의 방정식이 있다면 이를 일차연립방정식(a system of linear equations)이라고 한다.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= y_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= y_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= y_m \end{aligned} \tag{2.1}$$

위에서 일차연립방정식 식 2.1 에 나타나는 변수 x_1, x_2, \dots, x_n 와 계수 a_{ij}, y_i 으로 좀더 보기 좋고 효율적으로 표현하기 위하여 행렬 \mathbf{A} 과 벡터 \mathbf{x}, \mathbf{y} 의 다음과 같이 표기하고

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

일차연립방정식 식 2.1 을 다음과 같이 표현할 수 있다.

$$\mathbf{Ax} = \mathbf{y}, \quad \text{즉} \quad \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \tag{2.2}$$

2 수학 기호

식 식 2.2 은 y_i 의 값이 변수 x_1, x_2, \dots, x_n 와 계수 $a_{i1}, a_{i2}, \dots, a_{in}$ 를 다음과 같은 식으로 계산한다는 의미이다. 즉 일차연립 방정식 식 2.1 을 행렬과 벡터로 표현한 것이다.

$$\sum_{j=1}^n a_{ij} x_j = y_i, \quad i = 1, 2, \dots, m$$

2.4 Theorem

사용법은 Quarto 사용설명서 를 참조하자.

정리 2.1 (Line). *The equation of any straight line, called a linear equation, can be written as:*

$$y = mx + b$$

See 정리 2.1

2.5 HTML

mathjax 에서 매크로 이용은 사이트 참조

2.5.1 새로운 명령어

include-in-header:

- text: |

```
<script type="text/javascript">
```

```
window.MathJax = {
```

```
  tex: {
```

```
    macros: {
```

```
      RR: '\{\bf R\}', // a simple string replacement
```

```
      hatmat: '\{\pmb X (\{\pmb X\}^t \{\pmb X\})^{-1} \{\pmb X\}^t\}',
```

```
      bold: ['\boldsymbol{#1}', 1], // this macro has one parameter
```

```
      pardiff: ['\frac{\partial#2}{\partial#1}', 2, 'x'], // this macro has an optional para
```

```
      pardifftwo: ['\frac{\partial#1}{\partial#2}', 2]
```

```
    }
```

```
  }
```

```
};
```

```
</script>
```

$$\mathbf{X} + \mathbf{\Sigma} + \mathbf{\Sigma} = \mathbf{R} = \frac{\partial f}{\partial \mathbf{x}} = \frac{\partial f}{\partial \theta} = \frac{\partial [\theta] f}{\partial \mathbf{x}} = \boldsymbol{\theta} = \mathbf{X}(\mathbf{X}^t \mathbf{X})^{-1} \mathbf{X}^t = \boldsymbol{\theta}$$

\$\$ \pmb X + \{\pmb \Sigma\} + \Sigma = \text{RR} = \text{pardiff}\{f\} = \text{pardifftwo}\{f\}\{\theta\} = \text{pardiff}[\theta]\{f\} =

$$\left\{\frac{x}{y}\right\} + \frac{x}{y}$$

2.6 PDF

3 Quarto 의 출판 절차

3.1 사전 작업

1. 먼저 Quarto 를 설치한다. 설치하는 Quarto 시작페이지를 참고하자.
2. Rstudio 또는 Vscode 를 이용하여 로컬 폴더에 Quarto book 프로젝트를 생성한다.
 - 참고로 Git 프로그램은 설치되어 있다고 가정하며 설치가 안됐으면 터미널에서 Git 를 사용할 수 있도록 설치한다.

💡 힌트

Github 계정이 있다고 가정한다. Github 계정이 없다면 github.com에서 계정을 만든다.
로컬 저장소를 github 에 push 하는 경우 **personal access token**을 만들어서 password 에 입력해야 한다. 자세한 poersonal access token 생성 과정은 여기 를 참조하세요.

3.2 Git 의 로컬 및 원격 저장소 생성

💡 힌트

Git 저장소를 구성하기 전에 Git 가 관리할 필요가 없는 파일들을 정하고 .gitignore 파일에 넣어 주어야 한다. 이 작업은 Git 초기화 작업 이전에 수행하는 것이 좋다.
자신의 프로젝트에서 사용할 언어에 맞게 .gitignore 파일을 먼저 구성하여 프로젝트 폴더에 미리 저장하는 것이 좋다. 다양한 언어에 대한 예제 .gitignore 파일들은 여기 에서 찾을 수 있다.

이제 git 환경을 구성하고 github 의 원격저장소에 올린다

1. 터미널을 열고 프로젝트 파일들이 있는 폴더로 이동한다.
2. 다음과 같은 명령으로 git 저장소를 초기화한다.

만약 Git 버전 2.28.0 또는 그 이상을 사용한다면:

```
git init -b main
```

만약 Git 버전 2.27.1 또는 그 이하를 사용한다면:

```
git init && git symbolic-ref HEAD refs/heads/main
```

3. 다음에 다음과 같은 명령으로 저장소의 변경 내용을 추가하고 커밋한다.

```
git add .
git commit -m "First commit"
```

4. 이제 Github 계정에서 원격 저장소를 만든다. 만드는 방법은 Github 안내서를 참조하여 만들면 된다. 이 경우 README 파일은 만들지 않으며 로컬 저장소에서 만들어서 나중에 원격저장소에 올린다.
5. Github에서 원격 저장소를 만든 후에 저장소의 주소를 복사한다.
6. 터미널로 돌아가서 로컬 저장소가 있는 폴더에서 다음 명령을 입력한다.

```
git remote add origin <REMOTE_URL>
```

7. 프로젝트 폴더에서 커밋한 파일을 github의 원격 저장소에 push 하자.

```
git push -u origin main
```

이제 github의 원격 저장소에 가면 커밋한 파일들이 원격 저장소에 올려진 것을 확인할 수 있다.

- 위에서 설명한 내용들은 Github 안내서를 참조하였다.
- 참고로 github 에 먼저 원격 저장소를 만들고 로컬 저장소에 클론(clone)하는 방법도 있다. 로컬에서 프로젝트 파일들을 어느 정도 만든 뒤에 원격 저장소에 올라는 것이 프로젝트를 일반적인 절차이므로 위의 방법을 제사하였다. 만약 github 에 먼저 원격 저장소를 만들고 로컬 저장소에 클론하려면 Github 안내서를 참조하면 된다.

💡 힌트

Git 로 파일을 관리하는 도중에 관리할 필요가 없는 새로운 파일이 생겨서 .gitignore 파일에 넣어 주고 적용할 경우가 생긴다.

먼저 .gitignore 파일에 관리할 필요가 없는 파일들을 추가해 주고 다음 명령어를 실행한다.

```
git rm -r --cached .
```

위의 명령이 실행되면 다시 파일들의 변경을 추가하고 커밋한다.

```
git add .
git commit -m "apply change in .gitignore"
```

3.3 출판

Quarto 로 제작한 웹사이트나 책을 Github Page 에 웹사이트 형태로 출판하는 방법은 아래와 같이 3가지가 있다.

- 사이트를 로컬에서 만들고 원격 저장소에 옮기고 github 에서 출판하는 방법
- quarto publish 명령문을 사용하는 방법
- Github Action 을 이용하는 방법

이 책에서는 quarto publish 와 github action 을 이용하는 방법에 대하여 설명한다.

3.4 quarto publish

다음의 설명하는 절차는 Quarto 설명서 를 참조하였다.

1. 먼저 해야할 일

- `_quarto.yml` 의 `format: html:` 옵션에서 `embed-resources: false` 로 지정하는 것이 좋다.
- `main branch` 에서 모든 것을 `commit` 한다.
- `build` 의 결과를 저장하는 폴더를 `.gitignore` 에 추가

```
/.quarto/
/_site/
```

2. gh-pages branch 를 로컬 저장소에 만든다. 다음과 같은 명령어로 gh-pages branch 를 만든다.

```
git checkout --orphan gh-pages
git reset --hard # make sure you've committed changes before running this!
git commit --allow-empty -m "Initialising gh-pages branch"
git push origin gh-pages
```

3. Github 의 원격 저장소로 이동하여 Settings -> Page 메뉴로 이동하여 다른 그림과 같이 Source 는 Deploy from a branch, Branch 는 gh-page 와 /(root) 로 지정되어 있는지 확인한다.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://ilovedata.github.io/quartoTutorial/>

Last deployed by  ilovedata last month

[Visit site](#)

...


Build and deployment


Source

Deploy from a branch ▾

Branch

Your GitHub Pages site is currently being built from the gh-pages branch. [Learn more about configuring the publishing source for your site.](#)

 gh-pages ▾

 / (root) ▾

Save

Learn how to [add a Jekyll theme](#) to your site.

Your site was last deployed to the [github-pages](#) environment by the [pages build and deployment](#) workflow.

[Learn more about deploying to GitHub Pages using custom workflows](#)

그림 3.1: Github 의 원격 저장소의 Page 설정

3 Quarto 의 출판 절차

4. 다시 main 저장소로 돌아간다.

```
git checkout main
```

5. Quarto 의 publish 명령을 이용하여 사이트를 만든다

```
quarto publish gh-pages
```

- 주의 : -quarto.yaml 에 있는 모든 format 을 publish 하기 때문에 모든 format 이 오류없이 실행되게 하거나 html 만 실행하도록 만든다.

6. 이후에는 변경된 파일을 커밋하고 다시 publish 하면 됨

3.5 github action

4. 먼저 프로젝트 _quarto.yaml 에 다음과 같이 추가하여 로컬에서만 프로그램이 실행되도록 만든다.

```
execute:
```

```
  freeze: auto
```

5. 다시 rendering 을 한다.

```
quarto render
```

References

$$1 + 1$$

