

Rohan Alexander

데이터로 이야기하기

R과 Python을 활용한 애플리케이션과 함께

For Mum and Dad

목차

서문

i 채프먼 앤 헐/CRC는 이 책을 2023년 7월에 출판했습니다. 여기^a에서 구매하실 수 있습니다. 이 온라인 버전은 인쇄된 내용에 일부 업데이트가 있습니다. 인쇄 버전과 일치하는 온라인 버전은 여기^b에서 확인할 수 있습니다.

^a<https://www.routledge.com/Telling-Stories-with-Data-With-Applications-in-R/Alexander/p/book/9781032134772>

^bhttps://rohanalexander.github.io/telling_stories-published/

추천사

이 깔끔하고 재미있는 책은 통계 커뮤니케이션, 프로그래밍, 모델링에 대한 광범위한 주제를 다루며, 어떤 통계 과정이나 독학 프로그램에도 유용한 보충 자료가 될 것입니다. 저는 이 책을 정말 좋아합니다!

앤드류 겔만(Andrew Gelman), 컬럼비아 대학교, 회귀 및 기타 이야기¹ 저자

훌륭한 책입니다. 통계학에서 커뮤니케이션과 재현성은 점점 더 중요해지고 있으며, 이 책은 이러한 주제들을 실용적이고 매력적이며 진정으로 독특한 방식으로 다룹니다.

다니엘라 위튼(Daniela Witten), 워싱턴 대학교, 통계 학습 입문² 저자

많은 데이터 과학 서적들이 형식적인 계산을 수행하는 방법을 알려줍니다. 대신, 데이터로 이야기하기는 분석의 사고방식과 과정에 참여하는 방법을 알려줍니다. 이 책은 학생들이 의미를 파악하고 이야기를 전달하는 데 필요한 계산, 통계 및 철학적 기술을 갖추게 함으로써 독특하게 실용적이고 힘을 실어주는 책으로 돋보입니다.

에밀리 리더러(Emily Riederer), 캐피탈 원, R 마크다운 쿡북³ 저자

데이터로 이야기하기는 데이터를 사용하여 배우고 긍정적인 변화를 이끌어내는 사려 깊은

¹<https://avehtari.github.io/ROS-Examples/>

²<https://www.statlearning.com>

³<https://bookdown.org/yihui/rmarkdown-cookbook/>

가이드입니다. 이 책은 프로세스의 각 단계를 포함하며, 많은 데이터 과학자와 미래의 데이터 스토리텔러에게 오래 지속되는 동반자가 될 수 있습니다.

크리스토퍼 피터스(Christopher Peters), 재피어

이것은 또 다른 통계 책이 아닙니다. 그보다 훨씬 낫습니다. 양적 연구, 과학적 정당화, 품질 관리, 커뮤니케이션 및 인식론적 겸손에 대한 책입니다. 어떤 방법론 커리큘럼에도 귀중한 보충 자료이며, 독학하는 사람들에게도 유용합니다.

리처드 맥엘리스(Richard McElreath), 막스 플랑크 진화 인류학 연구소, 통계적 재고⁴ 저자

영리한 직업 선택은 성장하는 자원과 자신의 기술이 상호 보완적인 분야를 선택하는 것입니다. 앞으로 수십 년 동안 데이터를 분석하는 데 능숙한 사람들은 번성할 것입니다. 이는 통계를 분석하고 설득력 있는 이야기를 전달하는 것을 의미합니다. 로한 알렉산더의 책은 이 두 가지를 모두 할 수 있도록 도와줄 것입니다.

앤드류 리(Andrew Leigh), 호주 의회 의원, 랜덤니스타스: 급진적인 연구자들이 우리 세상을 바꾸는 방법⁵ 저자

모든 데이터 분석가는 데이터로 이야기를 해야 하지만, 전통적인 교과서는 통계적 방법론에만 초점을 맞춥니다. 데이터로 이야기하기는 데이터 수집, 커뮤니케이션, 재현성을 포함한 전체 데이터 과학 워크플로우를 가르칩니다. 이 독특한 책을 강력히 추천합니다!

코스케 이마이(Kosuke Imai), 하버드 대학교, 양적 사회 과학: 입문⁶ 저자

이 책은 데이터 과학을 시작하는 모든 사람에게 현명한 조언으로 가득 찬 특별하고 훌륭한 책입니다. 개념과 코드를 혼합하여 아이디어를 즉시 구체화하고, 재현 가능한 워크플로우에 대한 강조는 빠르게 발전하는 분야에 엄격함을 더합니다.

데이비드 스피겔halter 경(Sir David Spiegelhalter), 케임브리지 대학교, 통계의 예술⁷ 저자

⁴<https://xcelab.net/rm/statistical-rethinking/>

⁵<https://yalebooks.yale.edu/book/9780300236125/randomistas/>

⁶<https://press.princeton.edu/books/quantitative-social-science>

⁷<https://dspiegel29.github.io/ArtofStatistics/>

(진실된) 데이터로 이야기하기는 화려한 통계 모델과 빅데이터 이상의 것을 요구합니다. 로한 알렉산더는 일련의 매혹적인 사례 연구를 통해 좋은 질문을 하고, 데이터를 획득하고, 모델을 추정하고, 결과를 전달하는 방법을 가르쳐줍니다. 이 전체론적 접근 방식은 명확하고 매력적인 산문으로 설명됩니다. 이 책의 페이지는 투명성과 재현성의 중요성을 강조하는 상세한 R 예제로 가득합니다. 저는 이 책을 정말 좋아하며 모든 학생들에게 추천합니다.

빈센트 아렐-번독(Vincent Arel-Bundock), 몬트리올 대학교, 인과 분석 및 양적 방법⁸ 저자

이 책은 데이터를 통해 이야기를 전달하는 데 도움을 줄 것입니다. 이 책은 여러분이 관찰한 데이터를 기반으로 관심 있는 세상의 한 측면에 대한 지식을 구축하고 공유할 수 있는 토대를 마련합니다. 불 주위에 작은 그룹으로 이야기를 나누는 것은 인간과 사회의 발전에 중요한 역할을 했습니다 (Wiessner 2014). 오늘날 데이터에 기반한 우리의 이야기는 수백만 명에게 영향을 미칠 수 있습니다.

이 책에서 우리는 데이터를 탐색하고, 조사하고, 밀어붙이고, 조작하고, 반죽하고, 궁극적으로는 그 함의를 이해하려고 노력할 것입니다. 다양한 특징들이 이 책의 선택을 이끌어냅니다.

제가 박사 학위를 받은 대학의 모토는 *naturam primum cognoscere rerum* 또는 대략 “사물의 본질을 먼저 배우는 것”입니다. 그러나 원래 인용문은 *temporis aeterni quoniam* 또는 대략 “영원한 시간을 위해”로 이어집니다. 우리는 이 두 가지를 모두 할 것입니다. 저는 지속적이고 재현 가능한 지식을 구축할 수 있도록 하는 도구, 접근 방식 및 워크플로우에 중점을 둡니다.

이 책에서 데이터에 대해 이야기할 때, 일반적으로 인간과 관련될 것입니다. 인간은 우리 이야기의 대부분의 중심에 있을 것이며, 우리는 사회적, 문화적, 경제적 이야기를 할 것입니다. 특히, 이 책 전체에서 저는 사회 현상과 데이터 모두에서 불평등에 주목할 것입니다. 대부분의 데이터 분석은 세상을 있는 그대로 반영합니다. 가장 취약한 사람들 중 많은 이들이 이 점에서 이중 부담에 직면합니다. 즉, 불이익을 받을 뿐만 아니라 그 정도를 측정하기가 더 어렵습니다. 우리 데이터셋에 있는 사람들의 데이터를 존중하는 것이 주요 관심사이며, 우리 데이터셋에 체계적으로 포함되지 않은 사람들을 생각하는 것도 마찬가지입니다.

데이터는 종종 다양한 맥락과 분야에 특화되어 있지만, 데이터를 이해하는 데 사용되는 접근 방식은 유사한 경향이 있습니다. 데이터는 또한 점점 더 전 세계적으로 다양한 출처에서 자원과 기회를 얻을 수 있습니다. 따라서 저는 많은 분야와 지역의 예시를 활용합니다.

지식이 되려면 우리의 발견은 다른 사람들에게 전달되고, 이해되고, 신뢰되어야 합니다. 과학적, 경제적 진보는 다른 사람들의 작업을 기반으로 할 때만 이루어질 수 있습니다. 그리고 이것은 우리가 그들이 무엇을 했는지 이해할 수 있을 때만 가능합니다. 마찬가지로, 우리가 세상에 대한 지식을 창출하려면 다른 사람들이 우리가 무엇을 했는지, 무엇을 발견했는지, 그리고 우리의 작업을 어떻게 수행했는지 정확하게 이해할 수 있도록 해야 합니다. 따라서 이 책에서는 커뮤니케이션과 재현성에 대해 특히 규범적으로 다룰 것입니다.

양적 작업의 품질을 향상시키는 것은 엄청난 도전이지만, 그것은 우리 시대의 도전입니다. 데이터는 우리 주변에 있지만, 지속적인 지식은 거의 창출되지 않고 있습니다. 이 책은 작은 방식으로나마 그것을 바꾸는 데 기여하기를 바랍니다.

독자 및 예상 배경 지식

이 책을 읽는 일반적인 독자는 1학년 학부 통계학에 대한 약간의 지식이 있습니다. 예를 들어 회귀 분석을 실행해 본 경험이 있을 것입니다. 그러나 특정 수준을 대상으로 하지 않고, 거의 모든 양적 과정에 관련된 측면을 제공합니다. 저는 이 책을 학부, 대학원 및 전문 수준에서 가르쳤습니다. 모든 사람은 고유한 요구사항이 있지만, 이 책의 어떤 측면이 여러분에게 도움이 되기를 바랍니다.

⁸<https://www.leslibraires.ca/livres/analyse-causale-et-methodes-quantitatives-une-vincent-arel-bundock-9782760643215.html>

열정과 관심은 사람들을 멀리 이끌었습니다. 그것들이 있다면 다른 것에 대해 너무 걱정하지 마십시오. 가장 성공적인 학생들 중 일부는 양적 또는 코딩 배경이 없는 학생들이었습니다.

이 책은 많은 내용을 다루지만, 특정 측면에 대해 깊이 있게 다루지는 않습니다. 따라서 데이터 과학: 첫 번째 소개 (Timbers, Campbell, 와/과 Lee 2022), 데이터 과학을 위한 R (Wickham, Çetinkaya-Rundel, 와/과 Grolemund [2016년] 2023), 통계 학습 입문 (James 기타 [2013년] 2021), 통계적 재고 (McElreath [2015년] 2020)와 같은 더 자세한 책들을 특히 보완합니다. 이 책들에 관심이 있다면, 이 책이 좋은 시작점이 될 수 있습니다.

구성 및 내용

이 책은 여섯 부분으로 구성되어 있습니다: I) 기초, II) 커뮤니케이션, III) 획득, IV) 준비, V) 모델링, VI) 응용.

파트 I—기초—는 이 책으로 무엇을 달성하려는지, 왜 이 책을 읽어야 하는지에 대한 개요를 제공하는 ??으로 시작합니다. 장 ??는 세 가지 예제를 통해 설명합니다. 이 예제들의 의도는 이 책에서 권장하는 전체 워크플로우를 무엇이 일어나는지에 대한 세부 사항에 너무 신경 쓰지 않고 경험할 수 있도록 하는 것입니다. 그 워크플로우는 계획, 시뮬레이션, 획득, 모델링, 그리고 커뮤니케이션입니다. 이 장의 모든 것을 처음부터 따르지 못하는 것은 정상적이지만, 직접 코드를 입력하고 실행하면서 진행해야 합니다. 이 책에서 한 장만 읽을 시간이 있다면, 저는 이장을 추천합니다. 장 ??는 제가 옹호하는 워크플로우에서 사용되는 재현성을 위한 몇 가지 핵심 도구를 소개합니다. 여기에는 Quarto, R 프로젝트, Git 및 GitHub, 그리고 R을 실제로 사용하는 것과 같은 측면이 포함됩니다.

파트 II—커뮤니케이션—은 서면 및 정적 커뮤니케이션을 다룹니다. 장 ??은 양적 글쓰기가 가져야 할 특징과 명확한 양적 연구 논문을 작성하는 방법을 자세히 설명합니다. 장 ??의 정적 커뮤니케이션은 그래프, 표, 지도와 같은 기능을 소개합니다.

파트 III—획득—은 우리 세상을 데이터로 바꾸는 데 중점을 둡니다. 장 ??는 측정으로 시작하여 데이터에 대한 우리의 접근 방식을 지배하는 샘플링의 필수 개념을 단계별로 설명합니다. 그런 다음 인구 조사 및 기타 정부 통계와 같이 데이터로 사용하기 위해 명시적으로 제공되는 데이터셋을 고려합니다. 이들은 일반적으로 깨끗하고 잘 문서화되어 있으며 미리 패키지화된 데이터셋입니다. 장 ??는 API(응용 프로그래밍 인터페이스) 사용, 데이터 스크래핑, PDF에서 데이터 가져오기, OCR(광학 문자 인식)과 같은 측면을 다룹니다. 데이터는 사용 가능하지만 반드시 데이터셋으로 설계된 것은 아니며, 우리가 직접 가져와야 한다는 생각입니다. 마지막으로 장 ??는 우리에게 더 많은 것이 요구되는 측면을 다룹니다. 예를 들어, 실험을 수행하거나 A/B 테스트를 실행하거나 설문 조사를 해야 할 수도 있습니다.

파트 IV—준비—는 원본, 편집되지 않은 데이터를 탐색하고 공유할 수 있는 형태로 존중하며 변환하는 방법을 다룹니다. 장 ??는 데이터 정리 및 준비 작업에 접근할 때 따라야 할 몇 가지 원칙을 자세히 설명합니다. 다음, 취해야 할 특정 단계와 구현할 검사를 설명합니다. 장 ??는 R 데이터 패키지 및 패키트 사용을 포함하여 이러한 데이터셋을 저장하고 검색하는 방법에 중점을 둡니다. 그런 다음 데이터셋이 기반으로 하는 사람들을 존중하면서 가능한 한 광범위하게 데이터셋을 배포하고자 할 때 고려해야 할 사항과 취해야 할 단계를 계속 설명합니다.

파트 V—모델링—은 장 ??의 탐색적 데이터 분석으로 시작합니다. 이것은 데이터셋을 이해하는 중요한 과정이지만, 일반적으로 최종 제품에 포함되지 않는 것입니다. 이 과정 자체로 목적이 있습니다. 장 ??에서는 데이터를 탐색하기 위한 선형 모델 사용이 소개됩니다. 그리고 장 ??에서는 로지스틱, 포아송, 음이항 회귀를 포함한 일반화 선형 모델을 다룹니다. 또한 다단계 모델링도 소개합니다.

파트 VI—응용—는 모델링의 세 가지 응용을 제공합니다. 장 ??는 관찰 데이터에서 인과 관계 주장을 하는 데 중점을 두며, 차이-차이, 회귀 불연속성, 도구 변수와 같은 접근 방식을 다룹니다. 장 ??은 사후 계층화 다단계 회귀를 소개하는데, 이는 통계 모델을 사용하여 알려진 편향에 대해 샘플을 조정하는 것입니다. 장 ??는 텍스트-데이터에 중점을 둡니다.

장 ??는 몇 가지 결론적인 언급을 제공하고, 몇 가지 미해결 문제를 자세히 설명하며, 다음 단계를 제안합니다.

교육학 및 주요 특징

여러분은 직접 작업을 해야 합니다. 자료와 코드를 직접 적극적으로 살펴보아야 합니다. (stephenking 은?) “아마추어는 영감을 기다리지만, 우리는 일어나서 일하러 간다”고 말합니다. 이 책을 수동적으로 읽지 마십시오. 저의 역할은 Hamming ([1997년] 2020, p. 2-3)에 가장 잘 설명되어 있습니다:

저는 말하자면 코치일 뿐입니다. 저는 여러분을 위해 마일을 달릴 수 없습니다. 기껏해야 스타일을 논하고 여러분의 스타일을 비판할 수 있을 뿐입니다. 여러분은 육상 코스가 여러분에게 도움이 되려면 마일을 달려야 한다는 것을 알고 있습니다. 따라서 이 책에서 듣고 읽는 내용을 신중하게 생각해야 합니다. 그래야만 여러분을 변화시키는 데 효과적일 수 있습니다. 이것이 분명히 목적이어야 합니다…

이 책은 밀도 높은 12주 입문 과정을 중심으로 구성되어 있습니다. 고급 독자들이 도전할 수 있을 만큼 충분한 자료를 제공하면서도, 모든 독자들이 숙달해야 할 핵심을 확립합니다. 일반적인 과정은 장 ??까지의 대부분의 자료를 다루고, 그 다음 특별히 관심 있는 다른 장을 선택합니다. 그러나 이는 학생들의 배경과 관심사에 따라 다릅니다.

장 ??부터 여러분은 데이터로 설득력 있는 이야기를 전달할 수 있는 워크플로우(계획, 시뮬레이션, 획득, 모델링, 커뮤니케이션)를 갖게 될 것입니다. 각 후속 장에서는 이 워크플로우에 깊이를 더할 것입니다. 이를 통해 여러분은 점점 더 정교하고 신뢰할 수 있는 방식으로 이야기할 수 있게 될 것입니다. 이 워크플로우는 학계와 산업계 모두에서 일반적으로 요구되는 기술을 포함합니다. 여기에는 커뮤니케이션, 윤리, 재현성, 연구 질문 개발, 데이터 수집, 데이터 정리, 데이터 보호 및 배포, 탐색적 데이터 분석, 통계 모델링 및 확장이 포함됩니다.

이 책의 특징 중 하나는 윤리 및 불평등 문제가 한 장에 몰려 쉽게 무시될 수 있는 것이 아니라, 전체에 통합되어 있다는 것입니다. 이것들은 중요하지만, 그 가치를 즉시 파악하기 어려울 수 있으므로 긴밀하게 통합되어 있습니다.

이 책은 또한 잠재적 고용주에게 보여줄 수 있는 작업 포트폴리오를 구축할 수 있도록 설계되었습니다. 산업계에서 일자리를 원한다면, 이것이 아마도 가장 중요한 일일 것입니다. E. Robinson 와/과 Nolis (2020, p. 55)는 포트폴리오가 여러분이 할 수 있는 것을 보여주는 프로젝트 모음이며, 구직에 성공하는 데 도움이 될 수 있다고 설명합니다.

소설 마지막 사무라이 (DeWitt 2000 p. 326)에서 한 등장인물은 다음과 같이 말합니다:

[학자는] 구절의 어떤 단어를 보더라도 즉시 그 단어가 나타났던 다른 구절을 떠올릴 수 있어야 한다. … [그래서] 텍스트는 빙산 덩어리 같아서 각 단어는 눈 덮인 봉우리이고 그 표면 아래에는 거대한 교차 참조의 얼어붙은 덩어리가 있다.

유사하게, 이 책은 자체 포함된 텍스트와 지침을 제공할 뿐만 아니라 전문 지식이 구축되는 중요한 지식 덩어리를 개발하는 데 도움을 줍니다. 어떤 장도 최종적인 결론을 내리지 않고, 다른 작업과 관련하여 작성됩니다.

이 책을 수업에서 사용하는 방법은 다양합니다. 전통적인 판서 강의도 효과적이지만, 학생들이 수업 전에 장을 읽는 데 전념할 수 있다면(주간 퀴즈나 중간고사를 통해 동기 부여), 그룹 기반 프로젝트와 토론을 위해 수업을 활용하는 것이 즐겁습니다. 매주 2~4명의 학생으로 구성된 소그룹을 만들고(학생들이 새로운 사

람들과 함께 작업할 기회를 주기 위해 매주 무작위로 새로운 그룹을 만듭니다). 그런 다음 일반적으로 “생각-작-공유” 연습 (Lyman 1981)을 따라 대부분의 연습 문제를 먼저 스스로 해결하고, 그룹과 비교한 다음, 마지막으로 선택된 답변을 수업과 공유하도록 합니다.

시기와 범위 측면에서, 파트 I “기초”가 다루어지는 한, 나머지 장들은 상당히 독립적입니다. 첫 번째 논문은 특히 중요하며, 학생들이 미래의 논문에 대한 교훈을 통합할 수 있도록 학생들에게 신속하게 돌려주어야 합니다.

각 장에는 다음과 같은 특징이 있습니다:

- 해당장을 읽기 전에 살펴보아야 할 필수 자료 목록. 명확히 말하면, 먼저 해당 자료를 읽은 다음 이 책으로 돌아와야 합니다. 각 장에는 광범위한 참고 자료도 포함되어 있습니다. 해당 주제에 특히 관심이 있다면, 이를 추가 탐색을 위한 시작점으로 사용해야 합니다.
- 해당장에서 개발되는 핵심 개념 및 기술 요약. 기술장에는 추가적으로 해당장에서 사용되는 소프트웨어 및 패키지 목록이 포함되어 있습니다. 이러한 기능들의 조합은 학습 체크리스트 역할을 하며, 장을 완료한 후 다시 확인해야 합니다.
- “연습”은 작은 시나리오를 제공하고 이 책에서 옹호하는 워크플로우를 통해 작업하도록 요청합니다. 이 작업은 아마 15-30분 정도 걸릴 것입니다. 미국 바이올리니스트 힐러리 한은 거의 매일 바이올린 연습(종종 스케일 또는 유사한 연습)을 공개적으로 기록합니다. 여러분도 비슷한 것을 하도록 권장하며, 이것들은 그것을 가능하게 하도록 설계되었습니다.
- 필수 자료를 살펴본 후, 그러나장을 진행하기 전에 지식을 테스트하기 위해 완료해야 하는 몇 가지 “퀴즈” 질문.장을 완료한 후에는 각 측면을 이해했는지 확인하기 위해 질문을 다시 살펴보아야 합니다. 답변 가이드는 요청 시 제공됩니다.
- 자료에 적극적으로 참여하도록 더욱 장려하는 “과제”. 이 질문에 대한 답변을 논의하기 위해 소그룹을 구성하는 것을 고려할 수 있습니다.

일부 장에는 추가적으로 다음이 포함됩니다:

- “오, 우리가 그 데이터에 대해 좋은 데이터를 가지고 있다고 생각하는군요!”라는 섹션은 흥잡을 데 없고 명확한 데이터가 있다고 종종 가정되지만 현실은 그와는 거리가 먼 특정 상황에 초점을 맞춥니다.
- “거인의 어깨”라는 섹션은 우리가 구축하는 지적 기반을 만든 사람들 중 일부에 초점을 맞춥니다.

소프트웨어 정보 및 규칙

이 책에서 시작하는 소프트웨어는 R (R Core Team 2024)입니다. 이 언어는 오픈 소스이고, 널리 사용되며, 전체 워크플로우를 다룰 만큼 충분히 일반적이지만, 잘 개발된 기능이 많을 만큼 충분히 구체적이기 때문에 선택되었습니다. 저는 여러분이 이전에 R을 사용해 본 적이 있다고 가정하지 않으며, 이 책에 R을 선택한 또 다른 이유는 R 사용자 커뮤니티 때문입니다. 이 커뮤니티는 초보자에게 특히 환영하며, 보완적인 초보자 친화적인 자료가 많이 있습니다.

프로그래밍 언어가 없다면 R은 시작하기에 좋은 언어입니다. “R 필수”⁹를 꼭 살펴보세요.

R에 익숙해졌다면, 또 다른 오픈 소스 프로그래밍 언어인 Python도 배우는 것을 권장합니다. Python은 R보다 초보자가 시작하기에 약간 덜 쉽지만, 산업계에서 널리 사용됩니다. 저는 Python을 사용하는 것이 합리적인 경우에 사용할 것이며, 이는 여러분에게 R과 Python 모두에 대한 친숙함을 제공할 것입니다.

1. R과 RStudio를 자신의 컴퓨터에 다운로드하세요. R은 여기¹⁰에서 무료로 다운로드할 수 있으며, RStudio Desktop은 여기¹¹에서 무료로 다운로드할 수 있습니다. 그리고 Quarto는 여기¹²에서 다운로드하세요.
2. Posit Cloud 여기¹³에서 계정을 만드세요. 이를 통해 클라우드에서 R을 실행할 수 있습니다.

⁹https://tellingstorieswithdata.com/20-r_essentials.html

¹⁰<http://cran.utstat.utoronto.ca/>

¹¹<https://rstudio.com/products/rstudio/download/#download>

¹²<https://quarto.org/docs/get-started/>

¹³<https://posit.cloud>

3. VS Code를 여기¹⁴에서 무료로 다운로드하세요.
4. Google Colab 여기¹⁵에서 무료 계정을 만드세요.

패키지는 tidyverse와 같이 타자기 텍스트로 표시되며, 함수는 filter()와 같이 타자기 텍스트로 표시되지만 팔호가 포함됩니다.

저자 소개

저는 토론토 대학교의 정보학부와 통계학과에 공동 임용된 조교수입니다. 또한 캐나다 통계 과학 연구소(CANSSI) 온타리오의 부소장, 매시 칼리지의 선임 연구원, 슈워츠 라이스만 기술 및 사회 연구소의 교수 연구원, 데이터 과학 연구소 재현성 주제 프로그램의 공동 책임자입니다. 저는 호주 국립 대학교에서 경제학 박사 학위를 받았으며, 경제사에 중점을 두었고 존 탕(의장), 마르틴 마리오티, 팀 해튼, 자크 워드의 지도를 받았습니다.

제 연구는 데이터 과학의 신뢰성을 향상시키는 워크플로우를 개발하는 방법을 탐구합니다. 저는 특히 데이터 과학에서 테스트의 역할에 관심이 있습니다.

저는 가르치는 것을 즐기며, 다양한 배경을 가진 학생들이 데이터를 사용하여 설득력 있는 이야기를 전달하는 방법을 배우도록 돋는 것을 목표로 합니다. 저는 다양한 분야에서 통계적 방법을 사용하는 데 능숙할 뿐만 아니라 그 한계를 이해하고 작업의 더 넓은 맥락에 대해 깊이 생각하는 학생들을 양성하려고 노력합니다. 저는 토론토 대학교의 정보학부와 통계학과에서 학부 및 대학원 수준 모두에서 가르칩니다. 저는 RStudio 공인 Tidyverse 트레이너입니다.

저는 모니카 알렉산더와 결혼했으며 두 자녀가 있습니다. 저는 아마도 책에 너무 많은 돈을 쓰고, 도서관에서 너무 많은 시간을 보낼 것입니다. 여러분만의 책 추천이 있다면 듣고 싶습니다.

감사

많은 분들이 이 책을 개발하는 데 도움이 된 코드, 데이터, 예제, 지침, 기회, 생각, 시간을 아낌없이 주셨습니다.

데이비드 그립스, 커티스 힐, 로빈 로이드-스타크스, 그리고 테일러 앤 프랜시스 팀에게 이 책을 편집하고 출판하며 귀중한 지침과 지원을 제공해 주셔서 감사합니다. 이 책을 철저히 편집해 준 에리카 올로프에게 감사합니다. 이 책의 초기 초안을 철저히 검토하고 개선에 도움이 되는 상세한 피드백을 제공해 준 이사벨라 게멘트에게 감사합니다.

이 책의 모든 단어를 검토하고 많은 부분을 개선하며, 이 책에 다루어진 많은 내용에 대한 저의 생각을 날카롭게 하는 데 도움을 준 애니 콜린스에게 감사합니다. 가르치는 즐거움 중 하나는 애니와 같이 재능 있는 사람들과 그들의 경력을 시작할 때 함께 일할 기회를 얻는 것입니다.

이 책의 초기 계획에 대해 상세한 의견을 제공해 준 에밀리 리더러에게 감사합니다. 그녀는 초안이 작성된 후 원고로 돌아와 세부적으로 검토했습니다. 그녀의 사려 깊은 의견은 이 책을 크게 개선했습니다. 더 나아가 그녀의 작업은 이 책의 많은 내용에 대한 저의 생각을 바꾸었습니다.

저는 전체 장을 읽어준 많은 검토자들을 만날 수 있었던 행운을 누렸습니다. 때로는 두세 번, 심지어 그 이상 읽어주기도 했습니다. 그들은 기대 이상으로 훌륭한 제안을 제공하여 이 책을 개선하는 데 큰 도움이 되었습니다. 이에 대해 저는 알버트 랩, 알렉스 헤이즈, 알렉스 루스콤(경찰 폭력 “오, 당신은……” 항목도 제안), 아리엘 문도, 벤자민 하이브-케인스, 댄 라이언, 에릭 드라이스데일, 플로렌스 발레-뒤부아, 잭 베일리, 재 해트릭-심퍼스, 존 칸, 조나단 킨(파케트 전문 지식을 아낌없이 공유), 로렌 케네디(MRP에 대한 저의 생각을 발전시키기 위해 코드, 데이터, 전문 지식을 아낌없이 공유), 리암 웰시, 리자 볼튼(이 책을 가르

¹⁴<https://code.visualstudio.com>

¹⁵<https://colab.google>

치는 방법에 대한 저의 아이디어를 발전시키는 데 도움), 루이스 코레이아, 맷 라토, 마티아스 베르거, 마이클 문, 로베르토 렌티니, 라이언 브릭스, 그리고 테일러 라이트에게 감사드립니다.

많은 분들이 구체적인 제안을 해주셔서 많은 것이 개선되었습니다. 이 모든 분들은 이 책이 기반으로 하는 오픈 소스 프로그래밍 언어 커뮤니티를 특징짓는 관대함의 정신에 기여합니다. 이 모든 분들께 감사드립니다. 아 마푸즈는 포아송 회귀를 다루는 것이 중요하다고 깨닫게 해주었습니다. 아론 밀러는 FINER 프레임워크를 제안했습니다. 앤리슨 프레스메인즈 힐은 워드뱅크를 제안했습니다. 크리스 워쇼는 민주주의 기금 유권자 연구 그룹 설문조사 데이터를 제안했습니다. 크리스티나 웨이는 많은 코드 오류를 지적했습니다. 클레어 배터실은 글쓰기에 대한 많은 책을 추천해 주었습니다. 엘라 케이는 Quarto로 전환할 것을 제안하고 정당하게 주장했습니다. 파리아 칸다커는 “R 필수” 장이 된 것을 제안했습니다. 하림 나비드는 자신의 산업 경험을 아낌없이 공유했습니다. 히스 프리슨은 토론토 노숙자 데이터에 대한 도움을 주었습니다. 제시카 그론스벨은 통계 실습에 대한 귀중한 제안을 해주었습니다. 켈리 치우는 텍스트-데이터의 중요성을 강조했습니다. 레슬리 루트는 “오, 우리가 그 데이터에 대해 좋은 데이터를 가지고 있다고 생각하는군요!”라는 아이디어를 내놓았습니다. 마이클 청은 EDA에 대한 저의 접근 방식을 형성했습니다. 마이클 도넬리, 피터 헵번, 레오 레이몬드-벨질은 제가 몰랐던 정치학, 사회학, 통계학의 고전 논문에 대한 정보를 제공했습니다. 닉 호튼은 ?@sec-exploratory-data-analysis의 해설리 위컴 비디오를 제안했습니다. 폴 호지츠는 R 패키지를 만드는 방법을 가르쳐 주었고 이 책의 표지 그림을 만들었습니다. 라두 크라이우는 샘플링이 적절한 위치를 차지하도록 했습니다. 샤를라 갤판드는 R을 사용하는 방법에 대한 저의 접근 방식을 옹호했습니다. 토마스 윌리엄 로젠탈은 Shiny의 잠재력을 깨닫게 해주었습니다. 톰 카르도소와 제인 슈워츠는 언론인들이 수집한 훌륭한 데이터 소스였습니다. 얀보 탕은 낸시 리드의 “거인의 어깨” 항목을 도왔습니다. 마지막으로 크리스 매디슨과 마이아 발린트는 마지막 시를 제안했습니다.

저의 박사 학위 지도 교수님인 존 탕, 마르틴 마리오티, 팀 해튼, 자크 워드에게 감사드립니다. 그들은 제가 관심 있는 지적 공간을 탐색할 자유를 주었고, 그러한 관심사를 추구할 수 있도록 지원했으며, 모든 것이 구체적인 결과로 이어지도록 지도해 주었습니다. 그 시절에 배운 것이 이 책의 토대가 되었습니다.

이 책은 크리스 베일, 스콧 커닝햄, 앤드류 헤이스(이 책이 나오기 훨씬 전에 이 책과 같은 이름의 강의를 독립적으로 가르쳤음), 리사 렌드웨이, 그랜트 맥더모트, 네이선 마티아스, 데이비드 밍노, 에드 루빈을 포함하여 온라인에서 무료로 제공되는 다른 사람들의 노트와 교육 자료로부터 큰 도움을 받았습니다. 이 모든 분들께 감사드립니다. 학자들이 자료를 온라인에서 무료로 제공하는 변화된 규범은 훌륭한 것이며, 여기¹⁶에서 제공되는 이 책의 무료 온라인 버전이 기여하기를 바랍니다.

일부 평가 항목을 개발하는 데 도움을 준 사만다-조 카에타노에게 감사합니다. 그리고 루브릭의 일부 측면을 적용할 수 있도록 허락해 준 리사 롬키와 앤런 청에게도 감사합니다. 장 ?? 튜토리얼의 일부 측면의 촉매제는 McPhee (2017, p. 186)와 챌시 팔렛-펠레리티였습니다. “대화형 커뮤니케이션” 튜토리얼의 아이디어는 마우리시오 바르가스 세풀베다(“파차”)와 앤드류 휘트비의 작업이었습니다.

다음 분들의 수정에 감사드립니다: 에이미 패로우, 아르쉬 라칸팔, 세자르 빌라레알 구즈만, 클로이 티어스 타인, 핀 코롤-오드와이어, 플라비아 로페즈, 그레고리 파워, 흥 시, 제이든 정, 존 헤이즈, 조이스 쉬안, 로라 클라인, 로레나 알마라즈 데 라 가르자, 매튜 로버트슨, 미카엘라 드루일라드, 모우니카 타남, 림 알라사디, 롭 짐머만, 타예자 치쿰비리케, 위즈단 타리크, 양 우, 그리고 한예원.

켈리 라이온스는 지원, 지도, 멘토링, 그리고 우정을 제공했습니다. 그녀는 매일 학자가 어떤 모습이어야 하는지, 그리고 더 나아가 한 사람으로서 어떤 사람이 되기를 열망해야 하는지를 보여줍니다.

그렉 월슨은 가르침에 대해 생각할 구조를 제공하고 “스케일” 스타일 연습을 제안했습니다. 그는 이 책의 촉매제였으며, 초안에 대한 유용한 의견을 제공했습니다. 그는 매일 지적 공동체에 기여하는 방법을 보여줍니다.

팬데믹 기간 동안 첫째, 그리고 둘째 아이를 돌보며 이 책을 쓸 수 있도록 해준 엘르 코테에게 감사합니다.

2021년 크리스마스 현재 이 책은 부분적으로 완성된 노트들의 흩어진 모음이었습니다. 모든 것을 내려놓고 두 달 동안 지구 반대편에서 건너와 모든 것을 다시 쓰고 응집력 있는 초안을 만들 기회를 준 엄마와 아빠에게 감사합니다.

마리야 타플라가와 ANU 호주 정치 연구 센터(정치 및 국제 관계 학부)에 캔버라에서 2주간의 “글쓰기 휴양” 자금을 지원해 주셔서 감사합니다.

마지막으로 모니카 알렉산더에게 감사합니다. 당신이 없었다면 저는 책을 쓰지 못했을 것입니다. 심지어 가능하다고 생각조차 하지 못했을 것입니다. 이 책의 최고의 아이디어 중 많은 부분이 당신의 것이며, 그렇

¹⁶<https://tellingstorieswithdata.com/>

지 않은 것들도 당신이 여러 번 읽어주면서 더 좋게 만들었습니다. 이 책을 쓰는 데 헤아릴 수 없는 도움을 주시고, 이 책이 기반으로 하는 토대를 제공해 주시고(도서관에서 R에서 특정 행을 가져오는 방법을 여러 번 보여주셨던 것을 기억합니다!), 제가 글을 쓰는데 필요한 시간을 주시고, 책을 쓰는 것이 전날 완벽했던 것을 끝없이 다시 쓰는 것을 의미한다는 것이 밝혀졌을 때 격려해 주시고, 이 책의 모든 것을 여러 번 읽어 주시고, 적절하게 커피나 카테일을 만들어 주시고, 아이들을 돌봐주시고, 그 외에도 많은 것을 해주셔서 감사합니다.

저에게 연락하실 수 있습니다: rohan.alexander@utoronto.ca.

로한 알렉산더 캐나다 토론토 2023년 5월

오류 및 업데이트

i 채프먼 앤 헐/CRC는 이 책을 2023년 7월에 출판했습니다. 여기[a](#)에서 구매하실 수 있습니다. 이 온라인 버전은 인쇄된 내용에 일부 업데이트가 있습니다. 인쇄 버전과 일치하는 온라인 버전은 여기[b](#)에서 확인할 수 있습니다.

^a<https://www.routledge.com/Telling-Stories-with-Data-With-Applications-in-R/Alexander/p/book/9781032134772>

^bhttps://rohanalexander.github.io/telling_stories-published/

최종 업데이트: 2024년 11월 21일.

이 책은 The American Statistician의 Piotr Fryzlewicz (Fryzlewicz 2024)와 Amazon¹⁷의 Nick Cox 가 검토했습니다. 그들이 검토뿐만 아니라 수정 및 제안을 제공하기 위해 많은 시간을 할애해 주셔서 감사합니다.

2023년 7월 이 책이 출판된 이후 세상에는 다양한 변화가 있었습니다. 생성형 AI의 등장은 사람들이 코딩하는 방식을 바꾸었고, Quarto 덕분에 Python이 R과 더 쉽게 통합되었으며, 패키지는 계속 업데이트되고 있습니다(새로운 학생 코호트가 책을 읽기 시작한 것은 말할 것도 없습니다). 온라인 버전의 장점 중 하나는 제가 개선할 수 있다는 것입니다.

다음 분들의 수정 및 제안에 감사드립니다: 앤드류 블랙, 클레이 포드, 크리스탈 루이스, 데이비드 잔코스키, 도나 멀컨, 에미 타나카, 에밀리 수, 이네사 드 안젤리스, 제임스 웨이드, 줄리아 김, 크리시브 자인, 시머스 로스, 티노 칸기서, 자크 바티.

오류

다음 오류는 인쇄 버전에 존재하지만, 온라인 버전에서는 업데이트되었습니다. 아래에 언급되지 않은 오류를 발견하면 이슈[18](#)를 제출하거나 rohan.alexander@utoronto.ca로 이메일을 보내주십시오.

- p. xxi: 감사의 글에 Alex Hayes를 추가하십시오.
- p. 20: “use the tidyverse and janitor packages.”에 “packages”를 추가하십시오.
- p. 34: “daily-shelter-overnight-service-occupancy-capacity-2021”은 “daily-shelter-overnight-service-occupancy-capacity-2021.csv”로 변경되어야 합니다 (“.csv”가 추가됨).
- p. 34: 첫 번째 코드 청크를 두 번째 코드 청크로 대체하십시오:

```
toronto_shelters_clean <-  
  clean_names(toronto_shelters) |>  
  select(occupancy_date, id, occupied_beds)  
  
head(toronto_shelters_clean)
```

¹⁷https://www.amazon.com/gp/customer-reviews/R3S602G9RUDOF/ref=cm_cr_dp_d_rvw_ttl?ie=UTF8&ASIN=1032134771

¹⁸https://github.com/RohanAlexander/telling_stories/issues

```
toronto_shelters_clean <-
  clean_names(toronto_shelters) |>
  mutate(occupancy_date = ymd(occupancy_date)) |>
  select(occupancy_date, occupied_beds)

head(toronto_shelters_clean)
```

- p. 38: “이 시점에서 2019년 캐나다 연방 선거에서 각 정당이 얻은 의석 수에 대한 멋진 그래프를 만들 수 있습니다.”는 2021년 선거를 참조해야 합니다.
- p. 41: 불필요한 :::를 제거하십시오.
- p. 66: “New Project\$dots”는 “New Project…”로 변경되어야 합니다.
- p. 138: scale_color_brewer(palette = "Set1")는 불필요하며 제거되어야 합니다.
- p. 138: 그림 캡션은 실업률이 아닌 인플레이션을 참조해야 합니다.
- p. 154: Q9는 코드 청크 뒤와 “if” 앞에 “work”가 누락되었습니다.
- p. 188: “Leonhard Euler”는 “Carl Friedrich Gauss”로 변경되어야 합니다.
- p. 279: “detonated”는 “denoted”로 변경되어야 합니다.
- p. 342: Q5 옵션 b가 옵션 c에 반복됩니다.
- p. 347: R for Data Science의 “탐색적 데이터 분석” 장은 12가 아닌 11입니다.
- p. 353: “the the”를 수정하십시오.
- p. 355: “…결과 5,814로 추정되며, 둘 다 너무 낮습니다.”는 “…결과 11,197로 추정되며, 전자는 너무 낮고 후자는 너무 높습니다.”로 변경되어야 합니다.
- p. 371: 그림 11.11a를 참조하는 문장이 혼란스러웠으며 그림을 더 명확하게 참조해야 합니다.
- p. 587: 링크는 <https://fivethirtyeight.com/features/police-misconduct-costs-cities-millions-every-year-but-thats-where-the-accountability-ends/>여야 합니다.

제 I 편

기초

1

데이터로 이야기하기

i 채프먼 앤 홀/CRC는 이 책을 2023년 7월에 출판했습니다. 여기^a에서 구매하실 수 있습니다. 이 온라인 버전은 인쇄된 내용에 일부 업데이트가 있습니다.

^a<https://www.routledge.com/Telling-Stories-with-Data-With-Applications-in-R/Alexander/p/book/9781032134772>

선행 조건

- 셀 수 없는 것을 세기 읽기, (Keyes 2019)
 - 이 기사는 세상을 데이터로 바꾸는 것의 어려움을 논의합니다.
- 주방 카운터 천문대 읽기, (Healy 2020)
 - 데이터가 무엇을 숨기고 드러내는지에 대한 논의.
- 6분 만에 데이터 과학 윤리 시청, (Register 2020b)
 - 이 비디오는 윤리와 데이터 과학을 건설적인 방식으로 통합합니다.
- 코드는 무엇인가? 읽기, (Ford 2015)
 - 이 기사는 코드의 역할에 대한 개요를 제공하며, 처음 세 섹션에 집중해야 합니다.

1.1 이야기하기에 대하여

많은 부모들이 자녀가 태어나면 정기적으로 하는 첫 번째 일 중 하나는 자녀에게 이야기를 읽어주는 것입니다. 그렇게 함으로써 그들은 수천 년 동안 이어져 온 전통을 이어갑니다. 신화, 우화, 동화는 우리 주변 어디에서나 볼 수 있고 들을 수 있습니다. 그것들은 재미있을 뿐만 아니라 세상을 배우는 데 도움을 줍니다. 에릭 칼의 배고픈 애벌레가 데이터를 다루는 세상과는 거리가 멀어 보일 수 있지만, 유사점이 있습니다. 둘 다 이야기를 전달하고 지식을 전달하는 것을 목표로 합니다.

데이터를 사용할 때 우리는 설득력 있는 이야기를 전달하려고 노력합니다. 선거를 예측하는 것처럼 흥미로울 수도 있고, 인터넷 광고 클릭률을 높이는 것처럼 평범할 수도 있고, 질병의 원인을 찾는 것처럼 심각할 수도 있고, 농구 경기를 예측하는 것처럼 재미있을 수도 있습니다. 어떤 경우든 핵심 요소는 동일합니다. 20세기 초 영국 작가 E. M. 포스터는 모든 소설에 공통적인 측면을 이야기, 인물, 플롯, 환상, 예언, 패턴, 리듬으로 묘사했습니다 (Forster 1927). 마찬가지로, 설정에 관계없이 데이터를 사용하여 이야기를 전달할 때 공통적인 관심사가 있습니다:

1. 데이터셋은 무엇인가? 누가 왜 데이터셋을 생성했는가?
2. 데이터셋을 뒷받침하는 프로세스는 무엇인가? 그 프로세스를 고려할 때, 데이터셋에서 누락된 것이나 제대로 측정되지 않은 것은 무엇인가? 다른 데이터셋을 생성할 수 있었을까? 그렇다면 우리가 가진 데이터셋과 얼마나 다를 수 있었을까?
3. 데이터셋이 무엇을 말하려고 하며, 어떻게 그것이 말하도록 할 수 있는가? 또 무엇을 말할 수 있는가? 이를 중에서 어떻게 결정하는가?
4. 이 데이터셋에서 다른 사람들이 무엇을 보기를 바라며, 어떻게 그들을 설득할 수 있는가? 그들을 설득하기 위해 얼마나 많은 노력을 해야 하는가?
5. 이 데이터셋과 관련된 프로세스 및 결과에 누가 영향을 받는가? 데이터셋에 어느 정도 표현되어 있으며, 분석에 참여했는가?

과거에는 데이터를 사용하여 이야기를 전달하는 특정 요소가 더 쉬웠습니다. 예를 들어, 실험 설계는 농업

및 의학 과학, 물리학, 화학 분야에서 길고 견고한 전통을 가지고 있습니다. 학생의 t-분포는 1900년대 초기네스 맥주 제조업체에서 일했던 화학자 윌리엄 실리 고셋에 의해 확인되었습니다 (Boland 1984). 그가 맥주를 무작위로 샘플링하고 한 번에 한 가지 측면을 변경하는 것은 비교적 간단했을 것입니다.

오늘날 우리가 사용하는 통계 방법론의 많은 기본 원리는 그러한 환경에서 개발되었습니다. 그러한 상황에서는 일반적으로 통제 그룹을 설정하고 무작위화하는 것이 가능했으며, 윤리적 문제는 적었습니다. 결과 데이터로 전달되는 이야기는 상당히 설득력이 있었을 것입니다.

불행히도, 통계 방법론이 적용되는 다양한 환경을 고려할 때, 오늘날에는 이러한 것들이 거의 적용되지 않습니다. 반면에 우리는 많은 이점을 가지고 있습니다. 예를 들어, 잘 개발된 통계 기법, 대규모 데이터셋에 대한 더 쉬운 접근, 그리고 R 및 Python과 같은 오픈 소스 언어가 있습니다. 그러나 전통적인 실험을 수행하는 것의 어려움은 설득력 있는 이야기를 전달하기 위해 다른 측면에도 의존해야 함을 의미합니다.

1.2 워크플로우 구성 요소

데이터로 이야기를 전달하는 데 필요한 워크플로우에는 다섯 가지 핵심 구성 요소가 있습니다:

1. 계획하고 최종 지점을 스케치합니다.
2. 시뮬레이션하고 시뮬레이션된 데이터를 고려합니다.
3. 실제 데이터를 획득하고 준비합니다.
4. 실제 데이터를 탐색하고 이해합니다.
5. 수행한 작업과 발견한 내용을 공유합니다.

우리는 최종 지점을 계획하고 스케치하는 것으로 시작합니다. 이는 우리가 어디로 가고 싶은지 신중하게 생각하도록 보장하기 때문입니다. 이는 우리의 상황을 깊이 고려하도록 강제하고, 집중하고 효율적으로 유지하는 데 도움이 되며, 범위 확장을 줄이는 데 도움이 됩니다. 루이스 캐럴의 이상한 나라의 앤리스에서 앤리스는 체셔 고양이에게 어디로 가야 할지 묻습니다. 체셔 고양이는 앤리스가 어디로 가고 싶은지 물어봅니다. 그리고 앤리스가 어디든 상관없다고 대답하자, 체셔 고양이는 “충분히 오래 걸으면” 항상 어딘가에 도착할 것이기 때문에 방향은 중요하지 않다고 말합니다. 우리의 경우 문제는 우리가 오랫동안 목적 없이 걸을 여유가 없다는 것입니다. 최종 지점이 변경되어야 할 수도 있지만, 이것이 의도적이고 합리적인 결정이라는 것이 중요합니다. 그리고 그것은 초기 목표가 있을 때만 가능합니다. 많은 가치를 얻기 위해 너무 많은 시간을 할애할 필요는 없습니다. 종종 종이와 펜으로 10분이면 충분합니다.

다음 단계는 데이터를 시뮬레이션하는 것입니다. 이는 우리를 세부 사항으로 몰아넣기 때문입니다. 데이터셋의 클래스와 예상되는 값의 분포에 집중하게 하여 데이터셋을 정리하고 준비하는 데 도움이 됩니다. 예를 들어, 연령대가 정치적 선호도에 미치는 영향에 관심이 있다면, 연령대 변수가 “18-29”, “30-44”, “45-59”, “60+”의 네 가지 가능한 값을 가진 요인일 것으로 예상할 수 있습니다. 시뮬레이션 프로세스는 실제 데이터셋이 충족해야 하는 명확한 기능을 제공합니다. 이러한 기능을 사용하여 데이터 정리 및 준비를 안내할 테스트를 정의할 수 있습니다. 예를 들어, 실제 데이터셋에서 이 네 가지 값 중 하나가 아닌 연령대를 확인할 수 있습니다. 이러한 테스트가 통과하면 연령대 변수에 예상되는 값만 포함되어 있다고 확신할 수 있습니다.

데이터 시뮬레이션은 통계 모델링으로 전환할 때도 중요합니다. 그 단계에서는 모델이 데이터셋에 있는 것을 반영하는지 여부에 관심이 있습니다. 문제는 실제 데이터셋을 바로 모델링하면 모델에 문제가 있는지 알 수 없다는 것입니다. 우리는 처음에 데이터를 시뮬레이션하여 기본 데이터 생성 프로세스를 정확히 알 수 있습니다. 그런 다음 시뮬레이션된 데이터셋에 모델을 적용합니다. 우리가 입력한 것을 얻으면 모델이 적절하게 작동하고 있음을 알 수 있으며, 실제 데이터셋으로 전환할 수 있습니다. 시뮬레이션된 데이터에 대한 초기 적용 없이는 모델에 대한 확신을 갖기가 더 어려울 것입니다.

시뮬레이션은 종종 저렴합니다. 현대 컴퓨팅 자원과 프로그래밍 언어를 고려할 때 거의 무료입니다. 그리고 빠릅니다. 이는 “상황에 대한 친밀한 느낌”을 제공합니다 (Hamming [1997년] 2020, p. 239). 필수적인 것만 포함하는 시뮬레이션으로 시작하여 작동하게 한 다음 복잡하게 만드십시오.

우리가 관심 있는 데이터를 획득하고 준비하는 것은 종종 간과되는 워크플로우 단계입니다. 이는 가장 어려운 단계 중 하나일 수 있고 많은 결정을 내려야 하기 때문에 놀라운 일입니다. 이는 점점 더 연구

1.2 워크플로우 구성 요소

7

의 대상이 되고 있으며, 이 단계에서 내려진 결정이 통계 결과에 영향을 미칠 수 있음이 밝혀졌습니다 (Huntington-Klein 기타 2021; Dolatsara 기타 2021; Gould 기타 2023).

이 워크플로우 단계에서는 약간 압도당하는 느낌을 받는 것이 일반적입니다. 일반적으로 우리가 얻을 수 있는 데이터는 우리를 약간 두렵게 만듭니다. 데이터가 너무 적을 수도 있고, 이 경우 통계 기계를 어떻게 작동시킬지 걱정할 수 있습니다. 또는 그 반대 문제에 직면하여 그렇게 많은 양의 데이터를 어떻게 처리하기 시작할지 걱정할 수도 있습니다.

아마도 우리 삶의 모든 용들은 공주님일 것입니다. 단 한 번이라도 아름다움과 용기로 행동하는 것을 기다리고 있는. 아마도 우리를 두렵게 하는 모든 것은 가장 깊은 본질에서 우리의 사랑을 원하는 무력한 것입니다.

Rilke ([1929년] 2014)

이 워크플로우 단계에서 편안함을 느끼는 것은 나머지 단계를 해제합니다. 설득력 있는 이야기를 전달하는 데 필요한 데이터셋이 그 안에 있습니다. 그러나 조각가처럼, 우리는 필요 없는 모든 데이터를 반복적으로 제거하고, 필요한 데이터를 형성해야 합니다.

데이터셋을 얻은 후에는 해당 데이터셋의 특정 관계를 탐색하고 이해하고 싶을 것입니다. 우리는 일반적으로 기술 통계로 프로세스를 시작한 다음 통계 모델로 이동합니다. 데이터의 함의를 이해하기 위해 통계 모델을 사용하는 것은 편향에서 자유롭지 않으며, “진실”도 아닙니다. 모델은 우리가 지시하는 대로 작동합니다. 데이터를 사용하여 이야기를 전달할 때, 통계 모델은 그래프와 표를 사용하는 것과 마찬가지로 데이터셋을 탐색하는 데 사용하는 도구 및 접근 방식입니다. 그것들은 우리에게 결정적인 결과를 제공하지는 않지만, 특정 방식으로 데이터셋을 더 명확하게 이해할 수 있도록 해줄 것입니다.

워크플로우의 이 단계에 도달할 때쯤에는 모델은 어떤 종류의 기본 데이터 생성 프로세스를 반영하는 것만 큼이나 초기 단계, 특히 획득 및 정리에서 내려진 결정을 반영할 것입니다. 정교한 모델러는 자신의 통계 모델이 수면 위의 빙산 조각과 같다는 것을 알고 있습니다. 즉, 데이터라는 아래의 대부분 덕분에 구축되고 가능합니다. 그러나 전체 데이터 과학 워크플로우의 전문가는 모델링을 사용할 때 얻은 결과가 누구의 데이터가 중요한지, 데이터를 측정하고 기록하는 방법에 대한 결정, 그리고 데이터가 특정 워크플로우에서 사용 가능하기 훨씬 전에 세상을 반영하는 다른 측면과 같은 선택으로 인해 추가적으로 발생한다는 것을 인식합니다.

마지막으로, 우리가 수행한 작업과 발견한 내용을 가능한 한 높은 충실도로 공유해야 합니다. 자신만이 가지고 있는 지식에 대해 이야기하는 것은 여러분을 지식인으로 만들지 않으며, 여기에는 “과거의 당신”만이 가지고 있는 지식도 포함됩니다. 소통할 때, 우리가 내린 결정, 그 결정을 내린 이유, 우리의 발견, 그리고 우리 접근 방식의 약점에 대해 명확하게 설명해야 합니다. 우리는 중요한 것을 밝혀내려고 노력하고 있으므로, 처음에는 모든 것을 기록해야 하지만, 이 서면 커뮤니케이션은 나중에 다른 형태의 커뮤니케이션으로 보완될 수 있습니다. 이 워크플로우에서 내려야 할 결정이 너무 많기 때문에 우리는 전체 과정에 대해 개방적이어야 합니다. 통계 모델링과 그래프 및 표 생성뿐만 아니라 모든 것입니다. 이것이 없으면 데이터에 기반한 이야기는 신뢰성을 잃습니다.

세상은 모든 것이 신중하고 현명하게 평가되는 합리적인 능력주의가 아닙니다. 대신, 우리는 경험을 바탕으로 지름길, 해킹, 휴리스틱을 사용합니다. 불분명한 의사소통은 아무리 훌륭한 작업이라도 제대로 이해되지 못하게 하여 무의미하게 만들 것입니다. 의사소통에는 최소한의 기준이 있지만, 얼마나 인상적일 수 있는지에 대한 상한선은 없습니다. 잘 생각된 워크플로우의 정점일 때, 심지어 일종의 스프레차트라 또는 의도적인 무심함을 얻을 수도 있습니다. 이러한 숙달을 달성하려면 수년간의 노력이 필요합니다.

1.3 데이터로 이야기하기

데이터에 기반한 설득력 있는 이야기는 대략 10~20페이지 정도로 전달될 수 있습니다. 이보다 적으면 세부 사항이 너무 부족할 가능성이 있습니다. 그리고 훨씬 더 많이 작성하는 것은 쉽지만, 종종 약간의 성찰을 통해 간결성을 높이거나 여러 이야기를 분리할 수 있습니다.

전통적인 실험을 수행할 수 없는 경우에도 설득력 있는 이야기를 전달할 수 있습니다. 이러한 접근 방식은 “빅 데이터”에 의존하지 않고 (이는 만병통치약이 아닙니다 (Meng 2018; Bradley 기타 2021)), 대신 사용 가능한 데이터를 더 잘 활용하는 데 중점을 둡니다. 연구 및 독립 학습, 이론과 응용의 혼합, 이 모든 것이 실용적인 기술, 정교한 워크플로우, 그리고 자신이 모르는 것에 대한 이해와 결합되면 종종 지속적인 지식을 창출하기에 충분합니다.

데이터에 기반한 최고의 이야기는 다학제적인 경향이 있습니다. 필요한 분야에서 무엇이든 가져오지만, 거의 항상 통계, 소프트웨어 공학, 경제학, 그리고 공학 (몇 가지만 언급하자면)을 활용합니다. 따라서 엔드투엔드 워크플로우는 이러한 분야의 기술을 혼합해야 합니다. 이러한 기술을 배우는 가장 좋은 방법은 실제 데이터를 사용하여 다음과 같은 연구 프로젝트를 수행하는 것입니다:

- 연구 질문 개발;
- 관련 데이터셋 획득 및 정리;
- 해당 질문에 답하기 위해 데이터 탐색; 그리고
- 의미 있는 방식으로 소통.

데이터로 설득력 있는 이야기를 전달하는 핵심 요소는 다음과 같습니다:

1. 커뮤니케이션.
2. 재현성.
3. 윤리.
4. 질문.
5. 측정.
6. 데이터 수집.
7. 데이터 정리.
8. 탐색적 데이터 분석.
9. 모델링.
10. 스케일링.

이러한 요소는 좋은 연구 수행(윤리 및 질문), 신뢰할 수 있는 답변 도출(측정, 수집, 정리, 탐색적 데이터 분석 및 모델링), 그리고 설득력 있는 설명 생성(커뮤니케이션, 재현성 및 스케일링)을 포함한 몇 가지 다른 범주로 고려될 수 있습니다. 이러한 요소는 워크플로우가 구축되는 기반입니다(그림 ??).

이것은 마스터하기에 많은 것이지만, 커뮤니케이션이 가장 중요합니다. 간단한 분석을 잘 전달하는 것이 복잡한 분석을 제대로 전달하지 못하는 것보다 더 가치 있습니다. 후자는 다른 사람들이 이해하거나 신뢰할 수 없기 때문입니다. 명확한 의사소통의 부족은 때때로 연구자가 무엇이 진행되고 있는지, 또는 심지어 무엇을 하고 있는지 이해하지 못하는 실패를 반영합니다. 따라서 분석 수준은 데이터셋, 도구, 작업 및 기술 세트에 맞춰야 하지만, 명확성과 복잡성 사이에서 절충이 필요할 때에는 명확성을 우선하는 것이 합리적일 수 있습니다.

명확한 커뮤니케이션은 표, 그래프, 모델의 도움을 받아 평이한 언어로 작성하여 청중을 함께 이끌어가는 것을 의미합니다. 이는 무엇을 했고 왜 했는지, 그리고 무엇을 발견했는지 명확히 설명하는 것을 의미합니다. 최소한의 기준은 다른 사람이 여러분이 한 일을 독립적으로 다시 수행하고 여러분이 발견한 것을 찾을 수 있을 정도로 수행하는 것입니다. 한 가지 과제는 데이터에 몰입할수록 처음 접했을 때 어땠는지 기억하기 어렵다는 것입니다. 그러나 대부분의 청중은 그 지점에서 시작할 것입니다. 적절한 수준의 뉘앙스와 세부 사항을 제공하는 방법을 배우는 것은 어려울 수 있지만, 청중의 이익을 위해 글을 쓰는 데 집중하면 더 쉬워집니다.

재현성은 세상에 대한 지속적인 지식을 창출하는 데 필요합니다. 이는 수행된 모든 작업, 즉 처음부터 끝까지 모든 작업이 독립적으로 다시 수행될 수 있음을 의미합니다. 이상적으로는 자율적인 엔드투엔드 재현성이 가능합니다. 즉, 누구나 코드, 데이터 및 환경을 얻어 수행된 모든 것을 확인할 수 있습니다 (Heil 기

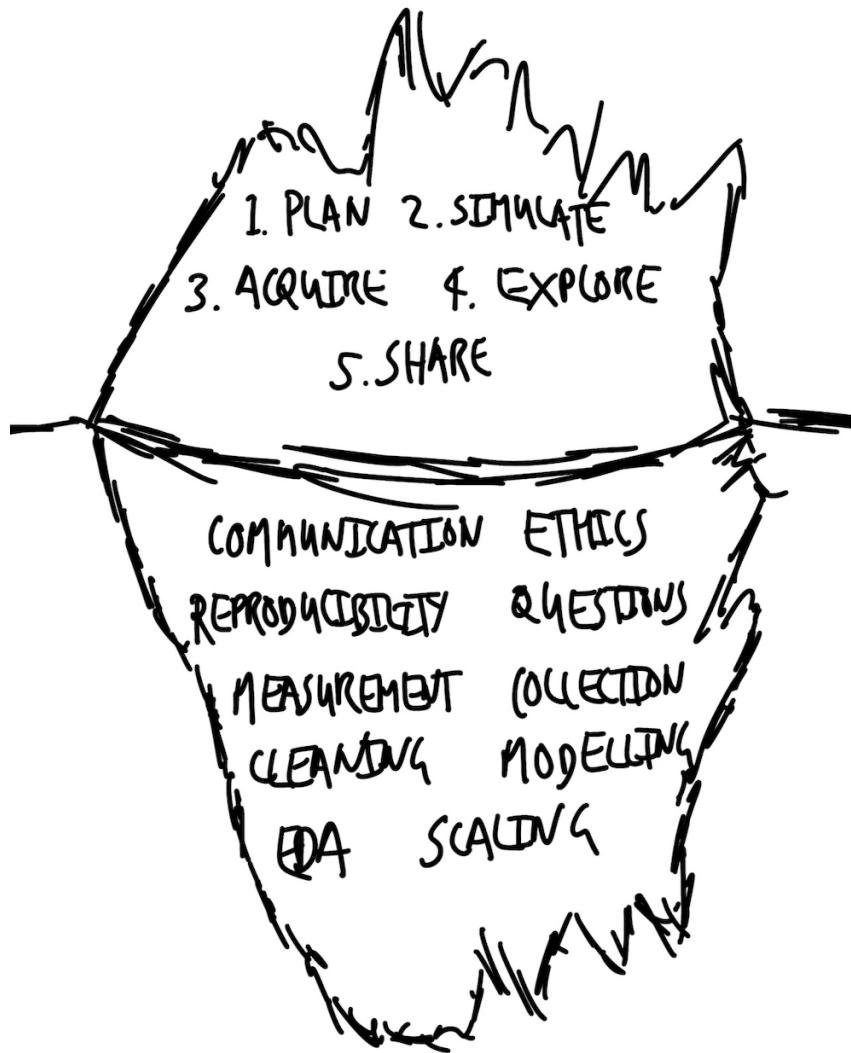


그림 1.1: 워크플로우는 다양한 요소를 기반으로 구축됩니다

타 2021). 코드에 대한 무제한 접근은 거의 항상 가능합니다. 데이터에 대한 기본 기대치도 마찬가지이지만, 항상 합리적인 것은 아닙니다. 예를 들어, 심리학 연구는 개인 식별이 가능한 작은 샘플을 가질 수 있습니다. 한 가지 방법은 유사한 속성을 가진 시뮬레이션된 데이터를 공개적으로 공유하고, 적절한 bona fides가 주어지면 실제 데이터에 액세스할 수 있는 프로세스를 정의하는 것입니다. 통계 모델은 일반적으로 광범위한 수동 검사를 받습니다. 재현성의 또 다른 측면은 광범위한 자동화된 테스트를 유사하게 포함해야 한다는 것입니다.

데이터셋이 인간과 관련될 가능성이 높기 때문에 윤리에 대한 적극적인 고려가 필요합니다. 이는 데이터셋에 누가 포함되어 있는지, 누가 누락되었는지, 그리고 그 이유는 무엇인지와 같은 사항을 고려하는 것을 의미합니다. 우리의 이야기가 과거를 어느 정도 영속시킬 것인가? 그리고 이것이 일어나야 할 일인가? 데이터셋이 인간과 관련되지 않더라도, 이야기는 인간에 의해 구성될 가능성이 높으며, 우리는 거의 모든 다른 것에 영향을 미칩니다. 이는 우리가 환경 영향과 불평등에 대한 우려를 가지고 데이터를 윤리적으로 사용해야 할 책임이 있음을 의미합니다.

윤리에 대한 많은 정의가 있지만, 데이터를 사용하여 이야기를 전달하는 데 있어서 최소한 데이터셋의 전체 맥락을 고려하는 것을 의미합니다 (D'Ignazio 와/과 Klein 2020). 법학에서 법에 대한 텍스트적 접근 방식은 인쇄된 법의 단어를 문자 그대로 고려하는 것을 의미하는 반면, 목적론적 접근 방식은 법이 더 넓은 맥락에서 해석되는 것을 의미합니다. 데이터를 사용하여 이야기를 전달하는 윤리적 접근 방식은 후자의 접근

근 방식을 채택하고, 우리 세상과 따라서 우리의 데이터를 형성하는 사회적, 문화적, 역사적, 정치적 힘을 고려하는 것을 의미합니다 (Crawford 2021).

호기심은 데이터셋과 관련 프로세스를 적절한 범위까지 탐색하려는 내적 동기를 제공합니다. 질문은 질문을 낳는 경향이 있으며, 데이터셋을 이해하는 과정이 계속됨에 따라 일반적으로 개선되고 정교해집니다. 종종 가르치는 가설 검정의 포퍼주의적 접근 방식과는 대조적으로, 질문은 일반적으로 지속적이고 진화하는 프로세스를 통해 개발됩니다 (Franklin 2005). 초기 질문을 찾는 것은 어려울 수 있습니다. 연구 질문을 합리적으로 사용 가능한 측정 가능한 변수로 조작하는 것은 특히 어렵습니다. 관심 분야를 선택하는 것이 도움이 될 수 있으며, 광범위한 주장을 스케치하여 특정 질문으로 발전시키고, 마지막으로 두 가지 다른 영역을 결합하는 것도 도움이 될 수 있습니다.

실제 데이터의 혼란스러움에 익숙해지고 편안함을 느끼는 것은 데이터가 업데이트될 때마다 새로운 질문을 할 수 있다는 것을 의미합니다. 그리고 데이터셋을 자세히 아는 것은 예상치 못한 그룹화 또는 값을 표면화하는 경향이 있으며, 이를 주제 영역 전문가와 협력하여 이해할 수 있습니다. 다양한 영역에 걸쳐 지식 기반을 개발하여 일종의 “하이브리드”가 되는 것은 특히 가치 있으며, 처음에는 어리석은 질문을 할 가능성에 익숙해지는 것도 마찬가지입니다.

측정과 데이터 수집은 우리 세상이 어떻게 데이터가 될 것인지를 결정하는 것입니다. 그것들은 어렵습니다. 세상은 너무나 활기차서 일관되게 측정하고 수집할 수 있는 것으로 축소하기 어렵습니다. 예를 들어, 누군가의 키를 생각해 봅시다. 우리는 아마도 키를 측정하기 전에 신발을 벗어야 한다는 데 동의할 것입니다. 그러나 우리의 키는 하루 종일 변합니다. 그리고 줄자로 키를 측정하는 것은 레이저를 사용하는 것과는 다른 결과를 줄 것입니다. 따라서 사람 간 또는 시간 경과에 따른 키를 비교하는 경우, 매일 같은 시간에 같은 방법으로 측정하는 것이 중요합니다. 그러나 이는 빠르게 불가능해집니다. 그리고 이는 이러한 데이터의 데이터베이스 표현과 관련된 문제를 제쳐두는 것입니다 (Kent 1993).

우리가 관심 있는 대부분의 질문은 키보다 더 복잡한 데이터를 사용할 것입니다. 누군가가 얼마나 슬픈지 어떻게 측정합니까? 고통을 어떻게 측정합니까? 무엇을 측정하고 어떻게 측정할지 누가 결정합니까? 세상을 값으로 축소하고 이를 비교할 수 있다고 생각하는 데는 일정한 오만이 필요합니다. 궁극적으로 우리는 그렇게 해야 하지만, 측정할 대상을 일관되게 정의하는 것은 어렵습니다. 이 과정은 가치 중립적이지 않습니다. 이 잔인한 축소를 합리적으로 받아들이는 유일한 방법은 우리가 측정하고 수집하는 것을 깊이 이해하고 존중하는 것입니다. 핵심 본질은 무엇이며, 무엇을 제거할 수 있습니까?

20세기 스페인 화가 파블로 피카소는 한 줄만 사용하여 동물의 윤곽을 묘사한 일련의 그림을 그렸습니다 (그림 ??). 단순함에도 불구하고 우리는 어떤 동물이 묘사되고 있는지 알아봅니다. 그림은 동물이 고양이 아닌 개임을 알려주기에 충분합니다. 이것을 사용하여 개가 아픈지 여부를 판단할 수 있을까요? 아마 아닐 것입니다. 우리는 다른 묘사를 원할 것입니다. 무엇을 측정해야 하는지, 그리고 우리가 고려하기로 결정한 것들 중에서 어떤 특징을 측정하고 수집해야 하는지, 그리고 어떤 것을 무시해야 하는지에 대한 결정은 맥락과 목적에 따라 달라집니다.



그림 1.2: 파블로 피카소의 이 그림은 한 줄로만 그려졌음에도 불구하고 분명히 개입니다

데이터 정리 및 준비는 데이터를 사용하는 데 중요한 부분입니다. 우리는 사용할 수 있는 데이터를 사용할 수 있는 데이터셋으로 정리해야 합니다. 이는 많은 결정을 내려야 합니다. 데이터 정리 및 준비 단계는 중요하며, 다른 어떤 단계만큼이나 많은 관심과 주의를 기울여야 합니다.

Kennedy 기타 (2022) 을 따라 잠재적으로 민감한 주제인 성별에 대한 정보를 수집한 설문조사를 고려해 봅시다. 네 가지 옵션: “남성”, “여성”, “말하고 싶지 않음”, 그리고 “기타”를 사용했으며, “기타”는 열린 텍

스트 상자로 해제되었습니다. 이 데이터셋을 접하면 대부분의 응답이 “남성” 또는 “여성”임을 알 수 있을 것입니다. “말하고 싶지 않음”에 대해 어떻게 해야 할지 결정해야 합니다. 데이터셋에서 이를 삭제하면 이 응답자들을 적극적으로 무시하는 것입니다. 삭제하지 않으면 분석이 더 복잡해집니다. 마찬가지로, 열린 테스트 응답을 어떻게 처리할지 결정해야 합니다. 다시 말하지만, 이러한 응답을 삭제할 수 있지만, 이는 일부 응답자의 경험을 무시하는 것입니다. 또 다른 옵션은 이를 “말하고 싶지 않음”과 병합하는 것이지만, 이는 응답자들이 해당 옵션을 명시적으로 선택하지 않았기 때문에 응답자들을 무시하는 것입니다.

많은 데이터 정리 및 준비 상황에서 쉽거나 항상 올바른 선택은 없습니다. 이는 맥락과 목적에 따라 달라집니다. 데이터 정리 및 준비에는 이와 같은 많은 선택이 포함되며, 다른 사람들이 무엇을 했고 왜 했는지 이해할 수 있도록 모든 단계를 기록하는 것이 중요합니다. 데이터는 결코 스스로 말하지 않습니다. 데이터는 데이터를 정리하고 준비한 복화술사의 꼭두각시입니다.

데이터셋의 모양과 느낌을 이해하는 과정은 탐색적 데이터 분석(EDA)이라고 합니다. 이것은 개방형 프로세스입니다. 공식적으로 모델링하기 전에 데이터셋의 형태를 이해해야 합니다. EDA 프로세스는 요약 통계, 그래프, 표, 때로는 모델링을 생성하는 반복적인 프로세스입니다. 공식적으로 끝나지 않는 프로세스이며 다양한 기술을 필요로 합니다.

EDA가 끝나고 공식적인 통계 모델링이 시작되는 지점을 명확히 구분하기는 어렵습니다. 특히 신념과 이해가 어떻게 발전하는지 고려할 때 더욱 그렇습니다 (Hullman 와/과 Gelman 2021). 그러나 핵심적으로는 데이터에서 시작하며, 데이터에 몰입하는 것을 포함합니다 (Cook, Reid, 와/과 Tanaka 2021). EDA는 일반적으로 최종 스토리에 명시적으로 포함되지 않습니다. 그러나 우리가 전달하는 스토리를 이해하는 데 중심적인 역할을 합니다. EDA 중에 취한 모든 단계를 기록하고 공유하는 것이 중요합니다.

통계 모델링은 길고 겹고한 역사를 가지고 있습니다. 통계에 대한 우리의 지식은 수백 년에 걸쳐 구축되었습니다. 통계는 일련의 건조한 정리와 증명이 아니라 세상을 탐색하는 방법입니다. 이는 “외국어 또는 대수학 지식과 유사합니다. 언제 어떤 상황에서도 유용할 수 있습니다” (Bowley 1901, p. 4). 통계 모델은 “이것이라면 저것”과 같은 방식으로 순진하게 따라야 할 레시피가 아니라 데이터를 이해하는 방법입니다 (James 기타 [2013년] 2021). 모델링은 일반적으로 데이터에서 통계적 패턴을 추론하는 데 필요합니다. 더 공식적으로, 통계적 추론은 “데이터를 사용하여 데이터를 생성한 분포를 추론하는 과정”입니다 (Wasserman 2005 p. 87).

통계적 유의성은 과학적 유의성과 동일하지 않으며, 우리는 지배적인 패러다임의 대가를 깨닫고 있습니다. 데이터에 대한 임의의 합격/불합격 통계 테스트를 사용하는 것은 거의 적절하지 않습니다. 대신, 통계 모델링의 적절한 사용은 일종의 반향 위치 측정과 같습니다. 우리는 모델에서 우리에게 돌아오는 것을 듣고 세상의 형태에 대해 배우는 데 도움을 받으면서, 그것이 세상의 한 가지 표현일 뿐임을 인식합니다.

R 및 Python과 같은 프로그래밍 언어의 사용은 작업을 빠르게 확장할 수 있도록 합니다. 이는 입력과 출력 모두를 의미합니다. 10개의 관측치를 고려하는 것이 1,000개, 심지어 1,000,000개를 고려하는 것만큼이나 쉽습니다. 이를 통해 우리의 이야기가 어느 정도 적용되는지 더 빨리 확인할 수 있습니다. 또한 우리의 출력을 한 사람이 소비하는 것만큼이나 10명, 또는 100명이 쉽게 소비할 수 있습니다. API(응용 프로그래밍 인터페이스)를 사용하면 우리의 이야기가 초당 수천 번 고려될 수도 있습니다.

1.4 우리 세상은 어떻게 데이터가 되는가?

에딩턴의 유명한 이야기가 있습니다. 어떤 사람들이 그물로 바다에서 물고기를 잡았습니다. 잡은 물고기의 크기를 조사한 후, 그들은 바다에 있는 물고기의 최소 크기가 있다고 결정했습니다! 그들의 결론은 사용된 도구에서 비롯된 것이지 현실에서 비롯된 것이 아닙니다.

Hamming ([1997년] 2020, p. 177)

어느 정도 우리는 시간을 낭비하고 있습니다. 우리는 세상의 완벽한 모델을 가지고 있습니다. 그것은 세상

입니다! 그러나 너무 복잡합니다. 모든 것이 그것에 영향을 미치는 셀 수 없는 요인에 의해 완벽하게 영향을 받는다는 것을 완벽하게 안다면, 우리는 동전 던지기, 주사위 굴리기, 그리고 다른 모든 무작위 과정들을 매번 완벽하게 예측할 수 있을 것입니다. 그러나 우리는 할 수 없습니다. 대신, 우리는 합리적으로 측정 가능한 것으로 단순화해야 하며, 그것이 우리가 데이터라고 정의하는 것입니다. 우리의 데이터는 파생된 지저분하고 복잡한 세상의 단순화입니다.

“합리적으로 측정 가능”에 대한 다양한 근사치가 있습니다. 따라서 데이터셋은 항상 선택의 결과입니다. 우리는 현재 작업에 대해 여전히 합리적인지 여부를 결정해야 합니다. 우리는 통계 모델을 사용하여 데이터를 깊이 생각하고, 탐색하고, 더 잘 이해하는 데 도움을 받습니다.

많은 통계학은 우리가 가진 데이터를 철저히 고려하는 데 중점을 둡니다. 이는 우리의 데이터가 농업, 천문학 또는 물리 과학에서 비롯되었을 때 적절했습니다. 이것은 비인간적 맥락에서 체계적인 편향이 존재하거나 영향을 미칠 수 없다는 것을 의미하는 것이 아니라, 데이터 과학의 부상과 함께, 부분적으로는 인간이 생성한 데이터셋에 대한 적용 가치 때문에, 우리는 데이터셋에 없는 것을 적극적으로 고려해야 합니다. 우리 데이터셋에서 체계적으로 누락된 사람은 누구입니까? 우리 접근 방식에 잘 맞지 않아 부적절하게 단순화되는 데이터는 누구의 데이터입니까? 세상이 데이터가 되는 과정이 추상화와 단순화를 필요로 한다면, 우리는 언제 합리적으로 단순화할 수 있고 언제 부적절할지 명확히 해야 합니다.

우리 세상이 데이터가 되는 과정은 필연적으로 측정을 포함합니다. 역설적으로, 종종 측정을 수행하고 세부 사항에 깊이 몰입하는 사람들은 그것에서 벗어난 사람들보다 데이터에 대한 신뢰가 적습니다. 거리 측정, 경계 정의, 인구 계산과 같은 겉보기에 명확한 작업조차도 실제로는 놀랍도록 어렵습니다. 우리 세상을 데이터로 바꾸는 것은 많은 결정을 필요로 하고 많은 오류를 발생시킵니다. 다른 많은 고려 사항 중에서, 우리는 무엇을 측정할지, 얼마나 정확하게 측정할지, 그리고 누가 측정을 수행할지 결정해야 합니다.

i 오, 우리가 그 데이터에 대해 좋은 데이터를 가지고 있다고 생각하는군요!

겉보기에 간단해 보이는 것이 빠르게 어려워지는 중요한 예시는 산모 관련 사망입니다. 이는 임신 중이거나 낙태 직후, 임신 또는 그 관리와 관련된 원인으로 사망하는 여성의 수를 의미합니다 (World Health Organization 2019). 이러한 사망의 비극을 원인별 데이터로 바꾸는 것은 어렵지만 중요합니다. 이는 미래의 사망을 완화하는 데 도움이 되기 때문입니다. 일부 국가에는 모든 사망에 대한 데이터를 수집하는 잘 개발된 민사 등록 및 생체 통계(CRVS) 시스템이 있습니다. 그러나 많은 국가에는 CRVS가 없어 기록되지 않은 사망이 발생합니다. 사망이 기록되더라도, 특히 자격을 갖춘 의료 인력이나 장비가 부족할 때 사망 원인을 정의하는 것이 어려울 수 있습니다. 산모 사망은 일반적으로 많은 원인이 있기 때문에 특히 어렵습니다. 일부 CRVS 시스템에는 사망 등록 양식에 사망을 산모 사망으로 간주해야 하는지 여부를 지정하는 확인란이 있습니다 (Dattani 2024). 그러나 일부 선진국조차도 최근에야 이를 채택했습니다. 예를 들어, 미국에서는 2003년에야 도입되었으며, 2015년에도 엘라배마, 캘리포니아, 웨스트버지니아는 표준 질문을 채택하지 않았습니다 (MacDorman 와/과 Declercq 2018). 이는 산모 사망이 과소 보고되거나 잘못 분류될 위험이 있음을 의미합니다.

우리는 일반적으로 다양한 도구를 사용하여 세상을 데이터로 바꿉니다. 천문학에서 더 나은 망원경, 그리고 결국 위성과 탐사선의 개발은 다른 세계에 대한 새로운 이해를 가능하게 했습니다. 마찬가지로, 우리 자신의 세상을 데이터로 바꾸는 새로운 도구들이 매일 개발되고 있습니다. 한때 인구 조사가 세대를 정의하는 사건이었지만, 이제는 정기적인 설문조사, 초 단위로 사용 가능한 거래 데이터, 그리고 인터넷의 거의 모든 상호 작용이 어떤 종류의 데이터가 됩니다. 이러한 도구의 개발은 흥미로운 새로운 이야기를 가능하게 했습니다.

우리 세상은 불완전하게 데이터가 됩니다. 그럼에도 불구하고 데이터를 사용하여 세상을 배우려면, 우리는 데이터의 불완전성과 그 불완전성의 함의를 적극적으로 이해하려고 노력해야 합니다.

1.5 데이터 과학이란 무엇이며, 세상을 배우기 위해 어떻게 사용해야 하는가?

데이터 과학에 대한 합의된 정의는 없습니다. Wickham, Çetinkaya-Rundel, 와/과 Grolemund ([2016년] 2023)는 “…원시 데이터를 이해, 통찰력, 지식으로 바꿀 수 있도록 합니다.”라고 말합니다. 마찬가지로, Leek 와/과 Peng (2020)은 “…데이터로 답할 수 있는 양적 질문을 공식화하고, 데이터를 수집하고 정

리하고, 데이터를 분석하고, 질문에 대한 답을 관련 청중에게 전달하는 과정”이라고 주장합니다. Baumer, Kaplan, 와/과 Horton (2021) 는 이를 “…데이터에서 의미 있는 정보를 추출하는 과학”으로 간주합니다. 그리고 Timbers, Campbell, 와/과 Lee (2022) 는 이를 “재현 가능하고 감사 가능한 프로세스를 통해 데이터에서 통찰력을 생성하는 과정”으로 정의합니다. 더 이전 시대에 Foster (1968) 는 우리가 지금 데이터 과학이라고 부르는 것을 명확하게 지적합니다. 그는 “(통계는) 방대한 데이터를 처리하고 분석하며, 데이터에서 정보를 추출하는 수학적 방법을 개발하는 것과 관련이 있습니다. 이 모든 활동을 컴퓨터 방법과 결합하면 각 부분의 합보다 더 큰 것을 얻을 수 있습니다.”라고 말합니다.

Craiu (2019) 는 데이터 과학이 무엇인지에 대한 불확실성이 중요하지 않을 수 있다고 주장합니다. 왜냐하면 “…누가 시인이나 과학자를 만드는지 정말로 말할 수 있겠습니까?” 그는 데이터 과학자를 “…데이터 기반 연구 의제를 가지고 있고, 통계 방법의 원칙적인 구현을 준수하거나 열망하며, 효율적인 계산 기술을 사용하는 사람”이라고 광범위하게 말합니다.

어떤 경우든, 구체적인 기술적 정의와 함께, 약간의 특수성을 잊더라도 간단한 정의를 갖는 것이 가치가 있습니다. 확률은 종종 비공식적으로 “사물을 세는 것”으로 정의됩니다 (McElreath [2015년] 2020, p. 10). 유사한 비공식적 의미에서 데이터 과학은 다음과 같이 정의될 수 있습니다: 인간이 사물을 측정하고, 일반적으로 다른 인간과 관련되며, 정교한 평균을 사용하여 설명하고 예측하는 것. 우리는 더 자세한 정의를 제공하기 위해 ?@sec-concluding-remarks에서 이를 다시 다룹니다.

그것은 약간 귀엽게 들릴 수 있지만, 19세기 통계학자이자 경제학자인 프랜시스 에지워스는 통계를 “사회 현상에 의해 제시되는 수단”의 과학으로 간주했으므로 좋은 동반자를 찾았습니다 (Edgeworth 1885). 어떤 경우든, 이 정의의 한 가지 특징은 데이터를 테라 놀리우스 또는 무주지로 취급하지 않는다는 것입니다. 통계학자들은 데이터를 우리가 결코 알 수 없는 어떤 과정의 결과로 보지만, 데이터를 사용하여 이해하려고 노력합니다. 많은 통계학자들은 데이터와 측정에 깊이 관심을 기울이지만, 통계학에는 데이터가 그냥 나타나는 많은 경우가 있습니다. 데이터는 누구의 소유도 아닙니다. 그러나 실제로는 결코 그렇지 않습니다.

데이터는 생성되고, 수집되고, 정리되고, 준비되어야 하며, 이러한 결정은 중요합니다. 모든 데이터셋은 sui generis, 즉 그 자체로 하나의 클래스이므로, 하나의 데이터셋을 잘 알게 되면 모든 데이터셋이 아닌 하나의 데이터셋만 알게 됩니다.

많은 데이터 과학은 “과학”에 초점을 맞추지만, “데이터”에도 초점을 맞추는 것이 중요합니다. 그리고 그것이 데이터 과학의 귀여운 정의의 또 다른 특징입니다. 일부 데이터 과학자는 광범위한 문제에 관심이 있는 제너럴리스트입니다. 종종 이들을 하나로 묶는 것은 지저분한 데이터를 수집하고 정리하고 준비해야 하는 필요성입니다. 그리고 종종 가장 많은 시간을 필요로 하고, 가장 자주 업데이트되며, 가장 많은 관심을 기울일 가치가 있는 것은 바로 그 데이터의 세부 사항입니다.

Jordan (2019) 은 의료 사무실에서 산전 초기 검사를 기반으로 자신의 자녀(당시 태아)가 다운 증후군을 앓을 확률을 받았다고 설명합니다. 배경 설명을 하자면, 확실히 알기 위한 검사를 할 수 있지만, 그 검사는 태아가 생존하지 못할 위험이 있으므로, 이 초기 검사를 수행한 다음 부모는 일반적으로 초기 검사에서 얻은 다운 증후군 확률을 사용하여 결정적인 검사를 할지 여부를 결정합니다. Jordan (2019) 은 초기 검사에서 제공된 확률이 10년 전에 영국에서 수행된 연구를 기반으로 결정되고 있음을 발견했습니다. 문제는 그 후 10년 동안 영상 기술이 향상되어 초기 검사가 그렇게 고해상도 이미지를 예상하지 못했고, 초기 검사에서 다운 증후군 진단이 (잘못된) 증가가 있었다는 것입니다. 데이터가 문제였습니다.

i 개인의 어깨

マイ클 조던 박사는 캘리포니아 대학교 버클리의 폐홍 첸 석좌 교수입니다. 1985년 캘리포니아 대학교 샌디에이고에서 인지 과학 박사 학위를 취득한 후, MIT 조교수로 임용되었고 1997년 정교수로 승진했으며, 1998년 버클리로 옮겼습니다. 그의 연구 분야 중 하나는 통계적 머신러닝입니다. 예를 들어, 특히 중요한 논문 중 하나는 Blei, Ng, 와/과 Jordan (2003) 이며, 이는 텍스트를 그룹화하여 주제를 정의하는 방법을 정의했으며, 우리는 장 ?? 에서 이를 다룹니다.

어려운 것은 “과학” 부분만이 아니라 “데이터” 부분도 마찬가지입니다. 예를 들어, 연구원들은 컴퓨터 과학에서 가장 인기 있는 텍스트 데이터셋 중 하나를 다시 조사했고, 데이터의 약 30%가 부적절하게 중복되었음을 발견했습니다 (Bandy 와/과 Vincent 2021). 이러한 유형의 데이터셋을 전문으로 하는 전체 분야인 언어학이 있으며, 데이터의 부적절한 사용은 어떤 분야가 해제모니를 가질 때의 위험 중 하나입니다. 데이터 과학의 강점은 다양한 배경과 훈련을 가진 사람들을 데이터셋에 대해 배우는 작업에 함께 모은다는 것입니다. 과거에 수행된 것에 의해 제한되지 않습니다. 이는 우리가 우리 자신의 전통에서 오지 않았지만, 우리만큼 데이터셋에 관심이 있는 사람들을 존중하기 위해 노력해야 함을 의미합니다. 데이터 과학은 다학

제적이며 점점 더 중요해지고 있습니다. 따라서 우리 세상을 반영해야 합니다. 데이터 과학에는 다양한 배경, 접근 방식 및 분야가 필요합니다.

우리 세상은 지저분하며, 우리의 데이터도 마찬가지입니다. 데이터로 이야기를 성공적으로 전달하려면 프로세스가 어려울 것이라는 사실에 익숙해져야 합니다. 영국 수학자 한나 프라이어는 문제를 해결하기 위해 코드를 다시 작성하는 데 6개월을 보냈다고 설명합니다 (Thornhill 2021). 여러분은 그것을 고수하는 법을 배워야 합니다. 또한 때로는 실패를 받아들여야 하며, 이는 회복력을 개발하고 내재적 동기를 가짐으로써 가능합니다. 데이터의 세계는 가능성과 확률을 고려하고, 그들 사이에서 결충하는 방법을 배우는 것입니다. 우리가 확실히 아는 것은 거의 없으며, 완벽한 분석은 없습니다.

궁극적으로, 우리는 모두 데이터를 사용하여 이야기를 전달할 뿐이지만, 이러한 이야기는 점점 더 세상에서 가장 중요한 이야기 중 하나가 되고 있습니다.

1.6 연습 문제

퀴즈

1. 데이터 과학이란 무엇입니까 (자신의 말로)?
2. Register (2020b)에 따르면, 데이터 결정은 (하나를 선택하십시오)?
 - a. 실제 사람들에게 영향을 미칩니다.
 - b. 아무에게도 영향을 미치지 않습니다.
 - c. 훈련 세트에 있는 사람들에게 영향을 미칩니다.
 - d. 테스트 세트에 있는 사람들에게 영향을 미칩니다.
3. Keyes (2019)에 따르면, 데이터 과학이란 무엇입니까 (하나를 선택하십시오)?
 - a. 데이터 과학은 과학적 방법, 프로세스, 알고리즘 및 시스템을 사용하여 많은 구조화된 및 비구조화된 데이터에서 지식과 통찰력을 추출하는 학제 간 분야입니다.
 - b. 의사 결정을 위한 대량의 데이터에 대한 양적 분석.
 - c. 인간성을 셀 수 있는 것으로 비인간적으로 축소하는 것.
4. Keyes (2019)에 따르면, 표준화된 범주를 요구하는 데이터 시스템의 한 가지 결과는 무엇입니까 (하나를 선택하십시오)?
 - a. 사용자 경험 저하.
 - b. 보안 조치 손상.
 - c. 기술 혁신 증가.
 - d. 개인의 정체성과 경험의 말소.
5. Healy (2020)에 따르면, 데이터를 다루는 것에 대한 일반적인 비판은 무엇입니까 (하나를 선택하십시오)?
 - a. 너무 시간이 많이 걸리고 비효율적이라는 것.
 - b. 숫자 뒤에 있는 인간 삶의 현실과 거리를 두게 한다는 것.
 - c. 분석을 위해 값비싼 소프트웨어와 광범위한 훈련이 필요하다는 것.
6. Healy (2020)에 따르면, 그 비판에 대한 한 가지 답변은 무엇입니까 (하나를 선택하십시오)?
 - a. 데이터를 다루는 것은 의미에 대한 질문과 대면하게 합니다.
 - b. 데이터 분석은 수행되어서는 안 됩니다.
 - c. 데이터는 자동화된 프로세스에 의해서만 분석되어야 합니다.
 - d. 질적 접근 방식이 지배적인 접근 방식이어야 합니다.
7. Keyes (2019)와 Healy (2020)을 어떻게 조화시킬 수 있습니까?
8. 윤리가 데이터 과학의 핵심 요소인 이유는 무엇입니까 (하나를 선택하십시오)?
 - a. 데이터 과학은 항상 민감한 개인 정보를 포함하기 때문입니다.
 - b. 윤리적 고려 사항이 분석을 더 쉽게 만들기 때문입니다.
 - c. 데이터셋은 인간과 관련될 가능성이 높으며 맥락을 고려해야 하기 때문입니다.
 - d. 규제가 모든 데이터 분석에 윤리 승인을 요구하기 때문입니다.
9. 이 장에서 설명된 Crawford (2021)에 따르면, 다음 중 우리 세상과 따라서 우리의 데이터를 형성하는 힘은 무엇입니까 (모두 선택하십시오)?
 - a. 정치적.
 - b. 물리적.

- c. 역사적.
 - d. 문화적.
 - e. 사회적.
10. Ford (2015)에 따르면, 컴퓨터란 무엇입니까 (하나를 선택하십시오)?
- a. 파일에 입력한 기호를 하위 수준 명령으로 변환하는 소프트웨어.
 - b. 누군가가 입력하거나 복사하거나 다른 곳에서 붙여넣은 일련의 기호 (일반적인 키보드 문자를 사용하여 어떤 종류의 파일로 저장됨).
 - c. 이점이 있는 시계.
 - d. 편지 카드에 구멍을 뚫고, 상자에 넣고, 로드한 다음, 컴퓨터가 카드를 넘겨 구멍이 있는 곳을 식별하고 메모리 일부를 업데이트하는 것.
11. 성별에 대한 설문조사 결과가 다음과 같다고 가정해 봅시다: “남성: 879”, “여성: 912”, “논바이너리: 10”, “말하고 싶지 않음: 3”, “기타: 1”. “말하고 싶지 않음”을 고려하는 적절한 방법은 무엇입니까 (하나를 선택하십시오)?
- a. 삭제합니다.
 - b. 상황에 따라 다릅니다.
 - c. 포함합니다.
 - d. “기타”로 병합합니다.
12. 인종 및/또는 성별을 예측 변수로 포함하면 모델의 성능이 향상되는 직업을 가지고 있다고 가정해 봅시다. 분석에 이러한 변수를 포함할지 여부를 결정할 때 어떤 요소를 고려할 것입니까 (자신의 말로)?
13. 데이터 과학에서 재현성이란 무엇을 의미합니까 (하나를 선택하십시오)?
- a. 다른 데이터셋으로 유사한 결과를 생성할 수 있는 것.
 - b. 분석의 모든 단계를 다른 사람이 독립적으로 다시 수행할 수 있도록 보장하는 것.
 - c. 동료 심사 저널에 결과를 게시하는 것.
 - d. 데이터를 보호하기 위해 독점 소프트웨어를 사용하는 것.
14. 측정과 관련된 과제는 무엇입니까 (하나를 선택하십시오)?
- a. 일반적으로 간단하고 거의 주의를 기울일 필요가 없습니다.
 - b. 무엇을 어떻게 측정할지 결정하는 것은 복잡하고 맥락에 따라 다릅니다.
 - c. 데이터 수집은 객관적이고 편향이 없습니다.
 - d. 측정은 항상 정확하고 시간이 지나도 일관적입니다.
15. 조각가 비유에서 조각하는 행위는 데이터 워크플로우에서 무엇을 나타냅니까 (하나를 선택하십시오)?
- a. 데이터에 맞는 복잡한 모델을 생성하는 것.
 - b. 원시 데이터를 획득하는 것.
 - c. 필요한 데이터셋을 드러내기 위해 데이터를 정리하고 준비하는 것.
 - d. 결과를 시각화하는 것.
16. 탐색적 데이터 분석(EDA)이 개방형 프로세스인 이유는 무엇입니까 (하나를 선택하십시오)?
- a. 따라야 할 고정된 단계가 있기 때문입니다.
 - b. 데이터의 형태와 패턴을 이해하기 위해 지속적인 반복이 필요하기 때문입니다.
 - c. 구조화된 방식으로 가설을 테스트하는 것을 포함하기 때문입니다.
 - d. 자동화할 수 있기 때문입니다.
17. 통계 모델을 신중하게 사용해야 하는 이유는 무엇입니까 (하나를 선택하십시오)?
- a. 항상 결정적인 결과를 제공하기 때문입니다.
 - b. 초기 단계에서 내려진 결정을 반영할 수 있기 때문입니다.
 - c. 대부분의 청중에게 너무 복잡하기 때문입니다.
 - d. 데이터가 잘 제시되면 불필요하기 때문입니다.
18. 키 측정의 어려움에 대해 생각하는 것에서 얻을 수 있는 한 가지 교훈은 무엇입니까 (하나를 선택하십시오)?
- a. 키는 변동성이 거의 없는 간단한 측정입니다.
 - b. 모든 측정은 올바른 도구로 수행되면 정확합니다.
 - c. 간단한 측정조차도 데이터 품질에 영향을 미치는 복잡성을 가질 수 있습니다.
 - d. 키는 데이터 분석에서 유용한 변수가 아닙니다.
19. 데이터셋에서 누락된 사람을 고려하지 않는 것의 위험은 무엇입니까 (하나를 선택하십시오)?
- a. 분석에 큰 영향을 미치지 않습니다.
 - b. 데이터 양을 줄여 분석을 단순화합니다.
 - c. 전체 맥락을 나타내지 않는 결론으로 이어질 수 있습니다.

20. 통계 모델링의 목적은 무엇입니까 (하나를 선택하십시오)?
 - a. 데이터를 탐색하고 이해하는 데 도움이 되는 도구.
 - b. 가설을 증명하는 것.
 - c. 탐색적 데이터 분석을 대체하는 것.
21. “우리 데이터는 지저분하고 복잡한 세상의 단순화이다”라는 말은 무엇을 의미합니까 (하나를 선택하십시오)?
 - a. 데이터는 현실의 모든 측면을 완벽하게 포착합니다.
 - b. 데이터는 분석을 가능하게 하기 위해 현실을 단순화하지만, 모든 세부 사항을 포착할 수는 없습니다.
 - c. 데이터는 항상 부정확하고 쓸모없습니다.

수업 활동

- 강사는 수업 사진을 찍은 다음 화면에 사진을 표시해야 합니다. 소그룹으로 학생들은 사진이 보여주는 세 가지 측면과 보여주지 않는 세 가지 측면을 식별해야 합니다. 이것이 데이터 과학과 어떻게 관련되는지 논의하십시오.
- 강사는 각 그룹에 측정에 사용할 다른 항목을 제공해야 합니다. 일부는 다른 것보다 더 유용합니다. 예를 들어, 줄자, 종이, 자, 마커, 저울 등. 그런 다음 학생들은 해당 항목을 사용하여 다음 질문에 답해야 합니다: “머리카락 길이는 얼마입니까?”. 숫자를 스프레드시트에 추가하십시오. 스프레드시트만 있다면 머리카락 길이에 대해 무엇을 이해하고 무엇을 이해하지 못할 것입니까? 이를 더 넓은 데이터 과학과 연결하십시오.

과제

이 과제의 목적은 걸보기에 간단해 보이는 것조차 측정의 어려움을 명확히 하고, 따라서 더 복잡한 영역에서 측정 문제의 가능성성을 명확히 하는 것입니다.

무, 겨자잎, 루꼴라와 같이 빠르게 자라는 식물의 씨앗을 구하십시오. 씨앗을 심고 사용한 흙의 양을 측정하십시오. 물을 주고 사용한 물의 양을 측정하십시오. 매일 변화를 기록하십시오. 더 일반적으로, 가능한 많이 측정하고 기록하십시오. 측정의 어려움에 대한 생각을 기록하십시오. 결국 씨앗이 싹을 틔울 것이고, 어떻게 자라는지 측정해야 합니다.

2

소방 호스에서 마시기

i 채프먼 앤 헐/CRC는 이 책을 2023년 7월에 출판했습니다. 여기^a에서 구매하실 수 있습니다. 이 온라인 버전은 인쇄된 내용에 일부 업데이트가 있습니다.

^a<https://www.routledge.com/Telling-Stories-with-Data-With-Applications-in-R/Alexander/p/book/9781032134772>

선행 조건

- 탁월함의 평범함: 계층화와 올림픽 수영 선수에 대한 민족지학적 보고서 읽기, (Chambliss 1989)
 - 이 논문은 탁월함이 특별한 재능이나 선물이 아니라 기술, 훈련, 태도 때문임을 밝힙니다.
- 원자 습관으로서의 데이터 과학 읽기, (Barrett 2021a)
 - 이 블로그 게시물은 작고 일관된 행동을 포함하는 데이터 과학 학습 접근 방식을 설명합니다.
- AI 편향이 실제로 발생하는 방식과 수정하기 어려운 이유 읽기, (Hao 2019)
 - 이 기사는 모델이 편향을 영속화할 수 있는 몇 가지 방법을 강조합니다.

핵심 개념 및 기술

- 통계 프로그래밍 언어 R은 데이터를 사용하여 흥미로운 이야기를 전달할 수 있도록 합니다. 다른 언어와 마찬가지로 숙달의 길은 느릴 수 있습니다.
- 프로젝트에 접근하는 데 사용하는 워크플로우는 계획, 시뮬레이션, 획득, 탐색, 공유입니다.
- R을 배우는 방법은 작은 프로젝트로 시작하여 목표 달성을 필요한 것을 작은 단계로 나누고, 다른 사람의 코드를 보고, 각 단계를 달성하기 위해 그것을 활용하는 것입니다. 해당 프로젝트를 완료하고 다음 프로젝트로 넘어갑니다. 각 프로젝트마다 조금씩 더 나아질 것입니다.

소프트웨어 및 패키지

- 기본 R (R Core Team 2024)
- 핵심 tidyverse (Wickham 기타 2019)
 - dplyr (Wickham, François, 기타 2022)
 - ggplot2 (Wickham 2016)
 - tidyR (Wickham, Vaughan, 와/과 Girlich 2023)
 - stringr (Wickham 2022c)
 - readr (Wickham, Hester, 와/과 Bryan 2022)
- janitor (Firke 2023)
- lubridate (Golemund 와/과 Wickham 2011)
- opendatatoronto (Gelfand 2022b)
- tinytable (Arel-Bundock 2024)

```
library(janitor)
library(lubridate)
#library(opendatatoronto)
library(tidyverse)
library(tinytable)
```

2.1 안녕하세요, 세상!

시작하는 방법은 시작하는 것입니다. 이 장에서는 이 책에서 옹호하는 데이터 과학 워크플로우의 세 가지 완전한 예제를 살펴봅니다. 이는 다음을 의미합니다:

계획 → 시뮬레이션 → 획득 → 탐색 → 공유

R에 익숙하지 않다면 일부 코드가 다소 생소할 수 있습니다. 통계에 익숙하지 않다면 일부 개념이 생소할 수 있습니다. 걱정하지 마십시오. 곧 모든 것이 익숙해질 것입니다.

이야기를 전달하는 방법을 배우는 유일한 방법은 직접 이야기를 전달하기 시작하는 것입니다. 이는 이러한 예제를 작동시키려고 노력해야 함을 의미합니다. 직접 스케치를 하고, 모든 것을 직접 입력하고(R에 익숙하지 않고 로컬에 설치되어 있지 않다면 Posit Cloud를 사용하십시오), 모든 것을 실행하십시오. 처음에는 어려울 것이라는 점을 깨닫는 것이 중요합니다. 이것은 정상입니다.

새로운 도구를 배울 때마다 오랫동안 형편없을 것입니다… 하지만 좋은 소식은 그것이 일반적이라는 것입니다. 그것은 모든 사람에게 일어나는 일이며, 일시적일 뿐입니다.

Hadley Wickham, Barrett (2021a) 인용.

여기서 철저하게 안내받을 것입니다. 데이터를 사용하여 이야기를 전달하는 흥분을 경험함으로써, 여러분은 계속해서 노력할 수 있는 힘을 얻기를 바랍니다.

워크플로우의 첫 번째 단계는 계획하는 것입니다. 우리는 상황에 대해 더 많이 알게 되면서 나중에 업데이트해야 할지라도 최종 지점을 설정해야 하기 때문에 이를 수행합니다. 그런 다음 시뮬레이션합니다. 이는 계획의 세부 사항에 집중하게 하기 때문입니다. 일부 프로젝트에서는 데이터 획득이 데이터셋을 다운로드하는 것처럼 간단할 수 있지만, 다른 프로젝트에서는 예를 들어 설문조사를 수행하는 경우 데이터 획득이 주요 초점이 될 수 있습니다. 다양한 양적 방법을 사용하여 데이터를 탐색하여 이해합니다. 마지막으로, 청중의 요구에 초점을 맞춰 우리의 이해를 공유합니다.

시작하려면 Posit Cloud¹로 이동하여 계정을 만드십시오. 무료 버전으로도 충분합니다. 처음에는 데스크톱 대신 이를 사용합니다. 이는 모든 사람이 동일하게 시작할 수 있도록 하기 위함이지만, 비용을 지불하지 않으려면 나중에 로컬 설치로 변경해야 합니다. 계정을 만들고 로그인하면 ?@fig-02-rstudio_cloud-1과 비슷하게 보일 것입니다.

“내 프로젝트”에 있을 것입니다. 여기에서 새 프로젝트를 시작해야 합니다: “새 프로젝트” → “새 RStudio 프로젝트”(그림 ??). “제목 없는 프로젝트”를 클릭하고 이름을 바꿔 프로젝트 이름을 지정할 수 있습니다.

이제 세 가지 예제를 살펴볼 것입니다: 호주 선거, 토론토 쉼터 사용량, 신생아 사망률. 이 예제들은 점점 더 복잡해지지만, 첫 번째 예제부터 데이터를 사용하여 이야기를 전달할 것입니다. 여기서 많은 측면을 간략하게 설명하지만, 거의 모든 것이 책의 나머지 부분에서 훨씬 더 자세히 설명됩니다.

2.2 호주 선거

호주는 하원 151석을 가진 의회 민주주의 국가이며, 하원에서 정부가 구성됩니다. 두 개의 주요 정당(“자유당”과 “노동당”), 두 개의 소수 정당(“국민당”과 “녹색당”), 그리고 많은 소규모 정당과 무소속이 있습니다. 이 예제에서는 2022년 연방 선거에서 각 정당이 얻은 의석 수를 그래프로 만들 것입니다.

¹<https://posit.cloud>

2.2 호주 선거

19

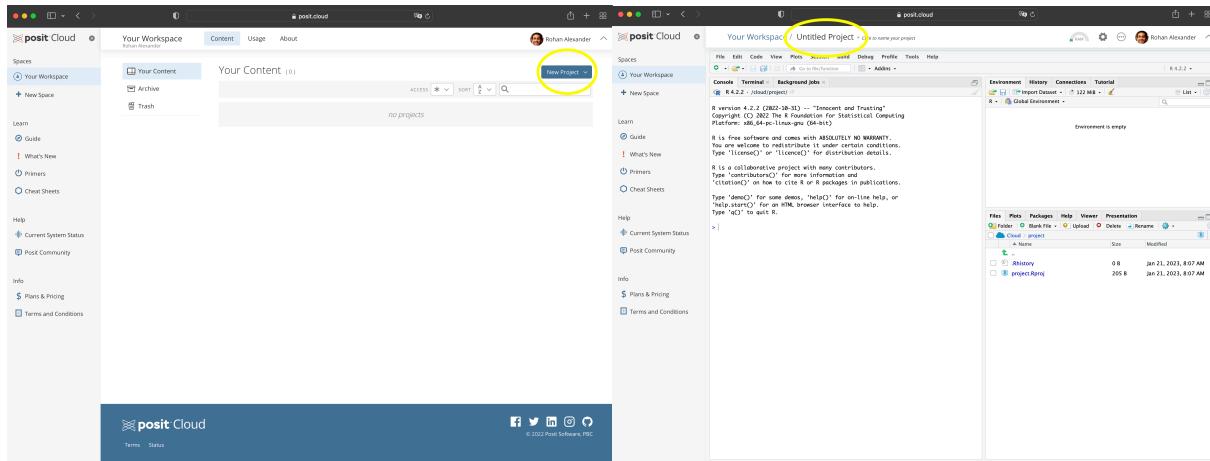


그림 2.1: Posit Cloud 및 새 프로젝트 시작하기

2.2.1 계획

이 예제에서는 두 가지 측면을 계획해야 합니다. 첫 번째는 필요한 데이터셋이 어떻게 생겼는지, 두 번째는 최종 그래프가 어떻게 생겼는지입니다.

데이터셋의 기본 요구 사항은 의석 이름(호주에서는 때때로 “선거구”라고 불림)과 당선된 사람의 정당을 포함해야 한다는 것입니다. 필요한 데이터셋의 빠른 스케치는 그림 ??입니다.



그림 2.2: 호주 선거 관련 잠재적 데이터셋 및 그래프 스케치

우리는 또한 관심 있는 그래프를 계획해야 합니다. 각 정당이 얻은 의석 수를 표시하고 싶으므로, 우리가 목표로 할 수 있는 빠른 스케치는 그림 ??입니다.

2.2.2 시뮬레이션

이제 스케치에 구체성을 부여하기 위해 일부 데이터를 시뮬레이션합니다.

시작하려면 Posit Cloud 내에서 새 Quarto 문서를 만드십시오: “파일” → “새 파일” → “Quarto 문서...”. “2022년 호주 선거 탐색”과 같은 제목을 지정하고, 저자로 자신의 이름을 추가하고, “시각적 마크다운 편집기 사용”을 클릭 해제하십시오 (그림 ??). 다른 옵션은 기본값으로 두고 “생성”을 클릭하십시오.

“패키지 rmarkdown 필요…”와 같은 알림이 표시될 수 있습니다 (그림 ??). 그런 경우 “설치”를 클릭하십시오. 이 예제에서는 모든 것을 이 하나의 Quarto 문서에 넣을 것입니다. “australian_elections.qmd”로 저장해야 합니다: “파일” → “다른 이름으로 저장…”.

기본 내용을 거의 모두 제거한 다음, 제목 자료 아래에 새 R 코드 청크를 만드십시오: “코드” → “청크 삽입”. 그런 다음 다음을 설명하는 서문 문서를 추가하십시오:

2 소방 호스에서 마시기

(a) 새 Quarto 문서 만들기

(b) 필요한 경우 rmarkdown 설치

(c) 초기 설정 및 서문 포함

(d) 청크를 실행하기 위해 녹색 화살표 강조 표시

(e) 메시지를 제거하기 위해 십자가 강조 표시

(f) 렌더링 버튼 강조 표시

그림 2.3: Quarto 문서 시작하기

- 문서의 목적;
- 저자 및 연락처 정보;
- 파일이 작성되었거나 마지막으로 업데이트된 시점; 그리고
- 파일이 의존하는 실행 조건.

```
##### ❷ #####  
# ❷: 2022년 ❷월 ❷일 오후 ❷시 ❷분  
# ❷월 ❷일 ❷시 ❷분에 작성되었습니다.  
# ❷작성자: Rohan Alexander  
# ❷이메일: rohan.alexander@utoronto.ca  
# ❷날짜: 2023년 1월 1일  
# ❷문서: ❷월 ❷일 오후 ❷시 ❷분에 작성되었습니다.
```

R에서 “#”으로 시작하는 줄은 주석입니다. 이는 R에 의해 코드로 실행되지 않고, 대신 사람이 읽도록 설계되었음을 의미합니다. 이 서문의 각 줄은 “#”으로 시작해야 합니다. 또한 “#####”로 둘러싸서 이것이 서문 섹션임을 명확히 하십시오. 결과는 ?@fig-quarto-australian-elections-3과 같아야 합니다.

이 후에는 작업 공간을 설정해야 합니다. 여기에는 필요한 패키지를 설치하고 로드하는 것이 포함됩니다. 패키지는 각 컴퓨터에 한 번만 설치하면 되지만, 사용할 때마다 로드해야 합니다. 이 경우 tidyverse와 janitor 패키지를 사용할 것입니다. 이전에 설치되지 않았다면 설치해야 하며, 그런 다음 각각 로드해야 합니다.

개인의 어깨

Hadley Wickham은 RStudio의 수석 과학자입니다. 2008년 아이오와 주립 대학교에서 통계학 박사 학위를 취득한 후 라이스 대학교 조교수로 임용되었고, 2013년 RStudio(현재 Posit)의 수석 과학자가 되었습니다. 그는 tidyverse 패키지 컬렉션을 개발했으며, R for Data Science (Wickham, Çetinkaya-Rundel, 와/과 Grolemund [2016년] 2023) 및 Advanced R (Wickham 2019)를 포함한 많은 책을 출판했습니다. 그는 2019년 COPSS 회장상을 수상했습니다.

패키지 설치 예시는 다음과 같습니다. R 코드 청크와 관련된 작은 녹색 화살표를 클릭하여 이 코드를 실행하십시오 (그림 ??).

```
##### ❷ #####  
install.packages("tidyverse")  
install.packages("janitor")
```

이제 패키지가 설치되었으므로 로드해야 합니다. 패키지 설치 단계는 컴퓨터당 한 번만 수행하면 되므로, 해당 코드는 실수로 실행되지 않도록 주석 처리하거나 제거해야 합니다. 또한 패키지를 설치할 때 인쇄된 메시지를 제거할 수 있습니다 (그림 ??).

```
##### ❷ #####  
# install.packages("tidyverse")  
# install.packages("janitor")  
  
library(tidyverse)  
library(janitor)
```

“렌더링”을 클릭하여 전체 문서를 렌더링할 수 있습니다 (그림 ??). 이렇게 하면 일부 패키지를 설치하라는 메시지가 표시될 수 있습니다. 그런 경우 동의해야 합니다. 그러면 HTML 문서가 생성됩니다.

방금 설치된 패키지에 대한 소개를 위해 각 패키지에는 패키지 및 해당 함수에 대한 정보를 제공하는 도움말 파일이 포함되어 있습니다. 패키지 이름 앞에 물음표를 붙인 다음 콘솔에서 해당 코드를 실행하여 액세스할 수 있습니다. 예를 들어 ?tidyverse.

데이터를 시뮬레이션하려면, “Division”과 “Party”라는 두 변수와 각각에 대한 일부 값을 가진 데이터셋을 만들어야 합니다. “Division”의 경우 합리적인 값은 151개 호주 선거구 중 하나의 이름일 것입니다. “Party”的 경우 합리적인 값은 “Liberal”, “Labor”, “National”, “Green”, “Other” 중 하나일 것입니다. 다시, 이 코드는 R 코드 청크와 관련된 작은 녹색 화살표를 클릭하여 실행할 수 있습니다.

```
##### 亂數亂 #####
set.seed(853)

simulated_data <-
tibble(
  # 151개 호주 선거구 중 151개의 값
  "Division" = 1:151,
  # 151개 호주 선거구 중 하나의 이름
  "Party" = sample(
    x = c("Liberal", "Labor", "National", "Green", "Other"),
    size = 151,
    replace = TRUE
  )
)

simulated_data
```

```
# A tibble: 151 x 2
  Division Party
  <int> <chr>
1       1 Liberal
2       2 Labor
3       3 Other
4       4 Liberal
5       5 Other
6       6 Other
7       7 Labor
8       8 Green
9       9 National
10      10 National
# i 141 more rows
```

어느 시점에서 코드가 실행되지 않고 도움을 요청하고 싶을 것입니다. 코드의 작은 스니펫을 스크린샷하여 누군가가 그것을 기반으로 도움을 줄 수 있을 것이라고 기대하지 마십시오. 그들은 거의 확실히 그럴 수 없습니다. 대신, 그들이 실행할 수 있는 방식으로 전체 스크립트를 제공해야 합니다. GitHub가 무엇인지는 장 ??에서 더 자세히 설명할 것이지만, 지금 당장 도움이 필요하다면 스크린샷을 찍는 것보다 더 유용한 방식으로 코드를 공유할 수 있도록 GitHub Gist를 순진하게 생성해야 합니다. 첫 번째 단계는 GitHub²에서 무료 계정을 만드는 것입니다 (그림 ??). 적절한 사용자 이름을 생각하는 것이 중요합니다. 이는 여러분의 전문 프로필의 일부가 될 것이기 때문입니다. 전문적이고, 어떤 과정과도 독립적이며, 이상적으로는 실제 이름과 관련된 사용자 이름을 사용하는 것이 좋습니다. 그런 다음 오른쪽 상단에서 “+”를 찾고 “새 Gist”를 선택하십시오 (그림 ??).

여기에서 오류가 발생하는 마지막 부분뿐만 아니라 모든 코드를 Gist에 추가해야 합니다. 그리고 끝에 “.R”이 포함된 의미 있는 파일 이름을 지정하십시오. 예를 들어 “australian_elections.R”. ?@fig-githubgisttwo에서는 library(Tidyverse) 대신 library(tidyverse)와 같이 잘못된 대소문자가 사용되었습니다.

“공개 Gist 생성”을 클릭하십시오. 그런 다음 이 Gist의 URL을 도움을 요청하는 사람과 공유하고, 문제가 무엇인지, 무엇을 달성하려고 하는지 설명할 수 있습니다. 모든 코드를 사용할 수 있으므로 그들이 더 쉽게 도움을 줄 수 있을 것입니다.

²<https://github.com>

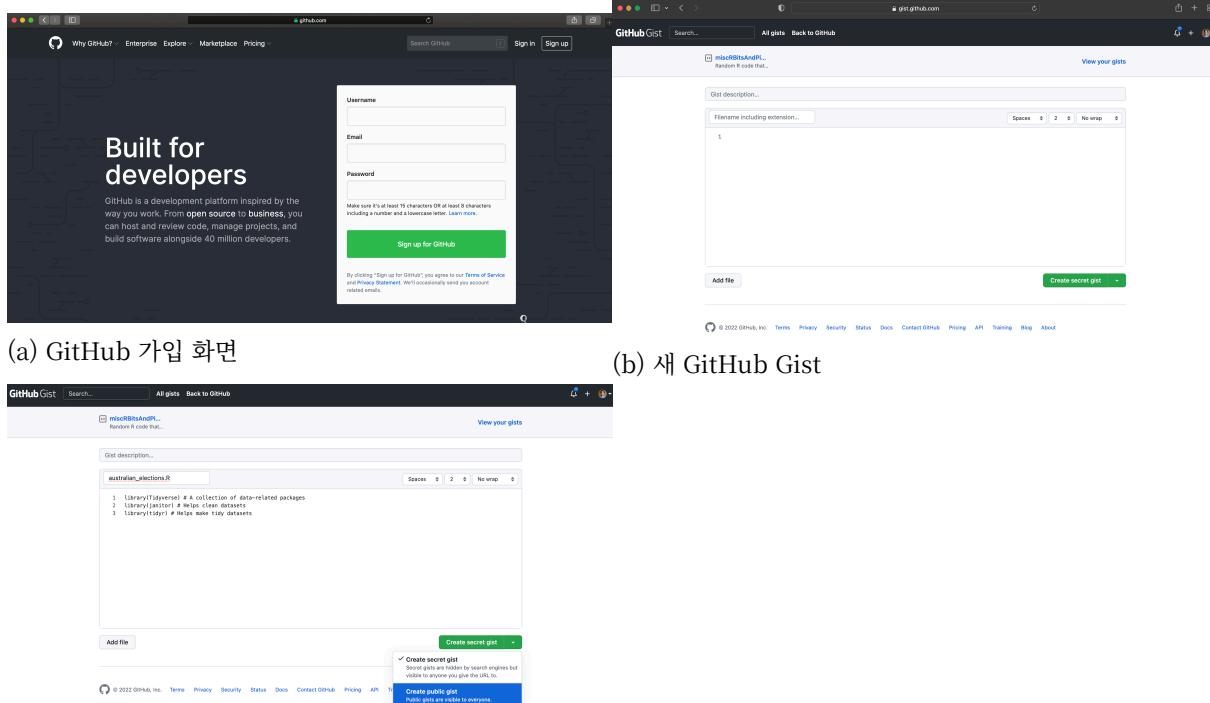


그림 2.4: 도움을 요청할 때 코드를 공유하기 위해 Gist 만들기

2.2.3 획득

이제 실제 데이터를 얻고 싶습니다. 필요한 데이터는 호주 연방 선거를 조직하는 비당파 기관인 호주 선거 관리 위원회(AEC)에서 가져옵니다. `readr`의 `read_csv()`에 웹사이트 페이지를 전달할 수 있습니다. `readr`는 `tidyverse`의 일부이므로 명시적으로 로드할 필요가 없습니다. `<-` 또는 “할당 연산자”는 `read_csv()`의 출력을 “`raw_elections_data`”라는 객체에 할당합니다.

```
#### 헤더 데이터 ####
raw_elections_data <-
  read_csv(
    file =
      paste0("https://results.aec.gov.au/27966/website/Downloads/",
            "HouseMembersElectedDownload-27966.csv"),
    show_col_types = FALSE,
    skip = 1
  )

# 이 데이터에는 행이 있습니다. 이 행은 다른 행과 함께
write_csv(
  x = raw_elections_data,
  file = "australian_voting.csv"
)
```

`head()`를 사용하여 데이터셋을 빠르게 살펴볼 수 있습니다. `head()`는 처음 여섯 행을 보여주고, `tail()`은 마지막 여섯 행을 보여줍니다.

```
head(raw_elections_data)
```

```
# A tibble: 6 x 8
  DivisionID DivisionNm StateAb CandidateID GivenNm   Surname PartyNm PartyAb
  <dbl> <chr>      <chr>       <dbl> <chr>      <chr>    <chr>    <chr>
1     179 Adelaide  SA          36973 Steve    GEORGANAS Austral~ ALP
2     197 Aston     VIC         36704 Alan     TUDGE     Liberal   LP
3     198 Ballarat  VIC         36409 Catherine KING     Austral~ ALP
4     103 Banks    NSW         37018 David    COLEMAN   Liberal   LP
5     180 Barker   SA          37083 Tony     PASIN     Liberal   LP
6     104 Barton   NSW         36820 Linda    BURNEY   Austral~ ALP
```

```
tail(raw_elections_data)
```

```
# A tibble: 6 x 8
  DivisionID DivisionNm StateAb CandidateID GivenNm   Surname PartyNm PartyAb
  <dbl> <chr>      <chr>       <dbl> <chr>      <chr>    <chr>    <chr>
1     152 Wentworth NSW         37451 Allegra SPENDER Independ~ IND
2     153 Werrawa   NSW         36810 Anne Maree STANLEY Austral~ ALP
3     150 Whitlam   NSW         36811 Stephen JONES     Austral~ ALP
4     178 Wide Bay  QLD        37506 Llew O'BRIEN   Liberal~ LNP
5     234 Wills     VIC         36452 Peter  KHALIL    Austral~ ALP
6     316 Wright    QLD        37500 Scott   BUCHHOLZ Liberal~ LNP
```

데이터를 사용하려면 정리해야 합니다. 계획 단계에서 원했던 데이터셋과 유사하게 만들려고 합니다. 계획에서 벗어나는 것은 괜찮지만, 이는 의도적이고 합리적인 결정이어야 합니다. 저장한 데이터셋을 읽어들인 후 가장 먼저 할 일은 변수 이름을 조정하는 것입니다. janitor의 `clean_names()`를 사용하여 이를 수행할 것입니다.

```
##### ## ## #####
raw_elections_data <-
  read_csv(
    file = "australian_voting.csv",
    show_col_types = FALSE
  )
```

```
# 헤더는 잘 되었지만 변수 이름은 문제입니다.
cleaned_elections_data <-
  clean_names(raw_elections_data)

# 이제 헤더는 잘 되었습니다.
head(cleaned_elections_data)
```

```
# A tibble: 6 x 8
  division_id division_nm state_ab candidate_id given_nm   surname party_nm party_ab
  <dbl> <chr>      <chr>       <dbl> <chr>      <chr>    <chr>
1     179 Adelaide  SA          36973 Steve    GEORGANAS Australian ~
2     197 Aston     VIC         36704 Alan     TUDGE     Liberal
3     198 Ballarat  VIC         36409 Catherine KING     Australian ~
4     103 Banks    NSW         37018 David    COLEMAN   Liberal
5     180 Barker   SA          37083 Tony     PASIN     Liberal
6     104 Barton   NSW         36820 Linda    BURNEY   Australian ~
# i 1 more variable: party_ab <chr>
```

이름이 더 빨리 입력되는 이유는 RStudio가 자동으로 완성하기 때문입니다. 이를 위해 변수 이름을 입력하기 시작한 다음 “tab” 키를 사용하여 완성합니다.

데이터셋에는 많은 변수가 있으며, 우리는 주로 “division_nm”과 “party_nm” 두 가지에 관심이 있습니다. tidyverse의 일부로 로드한 dplyr의 select()를 사용하여 관심 있는 특정 변수를 선택할 수 있습니다. “파이프 연산자”, |>,는 한 줄의 출력을 다음 줄의 함수의 첫 번째 입력으로 전달합니다.

```
cleaned_elections_data <-  
  cleaned_elections_data |>  
  select(  
    division_nm,  
    party_nm  
)  
  
head(cleaned_elections_data)
```

```
# A tibble: 6 x 2  
  division_nm party_nm  
  <chr>       <chr>  
1 Adelaide    Australian Labor Party  
2 Aston       Liberal  
3 Ballarat    Australian Labor Party  
4 Banks       Liberal  
5 Barker      Liberal  
6 Barton      Australian Labor Party
```

일부 변수 이름은 여전히 약어이기 때문에 명확하지 않습니다. names()를 사용하여 이 데이터셋의 열 이름을 살펴볼 수 있습니다. 그리고 dplyr의 rename()을 사용하여 이름을 변경할 수 있습니다.

```
names(cleaned_elections_data)
```

```
[1] "division_nm" "party_nm"
```

```
cleaned_elections_data <-  
  cleaned_elections_data |>  
  rename(  
    division = division_nm,  
    elected_party = party_nm  
)
```

```
head(cleaned_elections_data)
```

```
# A tibble: 6 x 2  
  division elected_party  
  <chr>     <chr>  
1 Adelaide Australian Labor Party  
2 Aston     Liberal  
3 Ballarat  Australian Labor Party  
4 Banks     Liberal  
5 Barker    Liberal  
6 Barton    Australian Labor Party
```

이제 unique()를 사용하여 “elected_party” 열의 고유 값을 살펴볼 수 있습니다.

```
cleaned_elections_data$elected_party |>  
  unique()
```

```
[1] "Australian Labor Party"
[2] "Liberal"
[3] "Liberal National Party of Queensland"
[4] "The Greens"
[5] "The Nationals"
[6] "Independent"
[7] "Katter's Australian Party (KAP)"
[8] "Centre Alliance"
```

우리가 원했던 것보다 더 많은 세부 정보가 있으므로, `dplyr`의 `case_match()`를 사용하여 정당 이름을 시뮬레이션한 것과 일치하도록 단순화할 수 있습니다.

```
cleaned_elections_data <-
  cleaned_elections_data |>
  mutate(
    elected_party =
      case_match(
        elected_party,
        "Australian Labor Party" ~ "Labor",
        "Liberal National Party of Queensland" ~ "Liberal",
        "Liberal" ~ "Liberal",
        "The Nationals" ~ "Nationals",
        "The Greens" ~ "Greens",
        "Independent" ~ "Other",
        "Katter's Australian Party (KAP)" ~ "Other",
        "Centre Alliance" ~ "Other"
      )
  )

head(cleaned_elections_data)
```

```
# A tibble: 6 x 2
  division elected_party
  <chr>     <chr>
1 Adelaide  Labor
2 Aston     Liberal
3 Ballarat  Labor
4 Banks     Liberal
5 Barker    Liberal
6 Barton    Labor
```

이제 데이터가 계획과 일치합니다 (그림 ??). 모든 선거구에 대해 당선된 사람의 정당을 가지고 있습니다.

이제 데이터셋을 깔끔하게 정리했으므로 저장해야 합니다. 다음 단계에서 정리된 데이터셋으로 시작할 수 있도록 말입니다. 원시 데이터를 대체하지 않고 나중에 정리된 데이터셋을 쉽게 식별할 수 있도록 새 파일 이름으로 저장해야 합니다.

```
write_csv(
  x = cleaned_elections_data,
  file = "cleaned_elections_data.csv"
)
```

2.2.4 탐색

우리가 만든 데이터셋을 탐색하고 싶을 수 있습니다. 데이터셋을 더 잘 이해하는 한 가지 방법은 그래프를 만드는 것입니다. 특히, 여기서는 ?@fig-australiaexample-graph에서 계획한 그래프를 만들고 싶습니다.

먼저, 방금 만든 데이터셋을 읽어들입니다.

```
#### #### ####
cleaned_elections_data <-
  read_csv(
    file = "cleaned_elections_data.csv",
    show_col_types = FALSE
  )
```

dplyr의 count()를 사용하여 각 정당이 얻은 의석 수를 빠르게 셀 수 있습니다.

```
cleaned_elections_data |>
  count(elected_party)
```

elected_party	n
<chr>	<int>
1 Greens	4
2 Labor	77
3 Liberal	48
4 Nationals	10
5 Other	12

관심 있는 그래프를 만들기 위해 tidyverse의 일부인 ggplot2를 사용합니다. 이 패키지의 핵심은 “+”를 사용하여 레이어를 추가하여 그래프를 만든다는 것입니다. 이를 “더하기 연산자”라고 부릅니다. 특히 ggplot2의 geom_bar()를 사용하여 막대 차트를 만들 것입니다 (그림 ??).

```
cleaned_elections_data |>
  ggplot(aes(x = elected_party)) + # aes는 "에스"로 읽는 단축.
  geom_bar()

cleaned_elections_data |>
  ggplot(aes(x = elected_party)) +
  geom_bar() +
  theme_minimal() + # 주제는 단축.
  labs(x = "정당", y = "의석 수") # 주제는 단축.
```

?@fig-canadanice-1은 우리가 목표로 한 것을 달성합니다. 그러나 기본 옵션을 수정하고 레이블을 개선하여 더 보기 좋게 만들 수 있습니다 (그림 ??).

2.2.5 공유

이 시점까지 우리는 데이터를 다운로드하고, 정리하고, 그래프를 만들었습니다. 일반적으로 우리가 한 일을 어느 정도 자세히 전달해야 합니다. 이 경우, 우리가 한 일, 왜 했는지, 그리고 무엇을 발견했는지에 대해 몇 단락을 작성하여 워크플로우를 마무리할 수 있습니다. 예시는 다음과 같습니다.

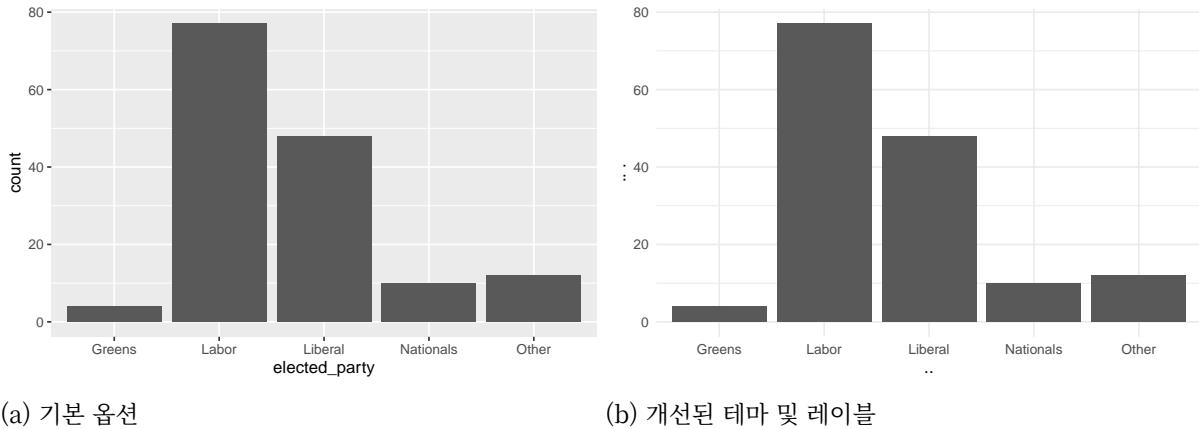


그림 2.5: 2022년 호주 연방 선거에서 정당별 획득 의석 수

요 정당(“자유당”과 “노동당”), 두 개의 소수 정당(“국민당”과 “녹색당”), 그리고 많은 소규모 정당이 있습니다. 2022년 연방 선거는 5월 21일에 치러졌으며, 약 1,500만 표가 투표되었습니다. 우리는 각 정당이 얻은 의석 수에 관심이 있었습니다.

우리는 호주 선거 관리 위원회 웹사이트에서 의석별 결과를 다운로드했습니다. 통계 프로그래밍 언어 R (R Core Team 2024)과 tidyverse (Wickham 기타 2019) 및 janitor (Firke 2023)를 사용하여 데이터셋을 정리하고 정돈했습니다. 그런 다음 각 정당이 얻은 의석 수 그래프를 만들었습니다 (그림 ??).

노동당이 77석을 얻었고, 자유당이 48석으로 그 뒤를 이었습니다. 소수 정당은 국민당이 10석, 녹색당이 4석을 얻었습니다. 마지막으로 10명의 무소속 의원과 소규모 정당 후보들이 당선되었습니다.

의석 분포는 두 주요 정당에 편향되어 있으며, 이는 호주 유권자들의 비교적 안정적인 선호도를 반영하거나, 전국적인 네트워크나 자금과 같은 주요 정당으로서의 이점 때문에 관성 때문일 수 있습니다. 이러한 분포의 이유에 대한 더 나은 이해는 향후 연구에서 흥미로운 주제입니다. 데이터셋은 투표한 모든 사람으로 구성되지만, 호주에서는 일부가 체계적으로 투표에서 제외되며, 일부는 다른 사람보다 투표하기가 훨씬 더 어렵다는 점에 유의해야 합니다.

이 예제는 몇 단락에 불과하지만, 초록을 형성하도록 축소하거나, 각 단락을 섹션으로 확장하여 전체 보고서를 형성하도록 늘릴 수 있습니다. 첫 번째 단락은 일반적인 개요, 두 번째 단락은 데이터에 초점, 세 번째 단락은 결과에 초점, 네 번째 단락은 토론입니다. Hao (2019)의 예시를 따라, 네 번째 단락은 편향이 스며들 수 있는 영역을 고려하기에 좋은 곳입니다.

2.3 토론토의 노숙자 인구

토론토에는 많은 노숙자 인구가 있습니다 (City of Toronto 2021). 혹독한 겨울은 쉼터에 충분한 공간이 있는 것이 중요함을 의미합니다. 이 예제에서는 2021년 쉼터 사용량 표를 만들어 각 월의 평균 사용량을 비교할 것입니다. 우리의 예상은 12월과 같은 추운 달에 7월과 같은 따뜻한 달보다 사용량이 더 많다는 것입니다.

2.3.1 계획

관심 있는 데이터셋은 날짜, 쉼터, 그리고 그날 밤 점유된 침대 수를 포함해야 합니다. 작동할 데이터셋의 빠른 스케치는 ?@fig-torontohomeless-data입니다. 우리는 매월 평균 일일 점유 침대 수를 포함하는 표를 만드는 데 관심이 있습니다. 표는 ?@fig-torontohomeless-table과 비슷하게 보일 것입니다.

date	shelter	occupancy
2021-07-01	—	27
2021-07-01	—	35
⋮	⋮	⋮

MONTH	AVERAGE OCCUPANCY
JULY	500
AUGUST	475
⋮	⋮
DECEMBER	1000

(a) 데이터셋의 빠른 스케치

(b) 매월 평균 점유 침대 수 테이블의 빠른 스케치

그림 2.6: 토론토 쉼터 사용량 관련 데이터셋 및 테이블 스케치

2.3.2 시뮬레이션

다음 단계는 데이터셋과 유사한 데이터를 시뮬레이션하는 것입니다. 시뮬레이션은 데이터 생성 프로세스에 대해 깊이 생각할 기회를 제공합니다. 분석으로 전환할 때, 이는 우리에게 가이드가 될 것입니다. 시뮬레이션을 먼저 사용하지 않고 분석을 수행하는 것은 목표 없이 화살을 쏘는 것과 같다고 생각할 수 있습니다. 즉, 확실히 무언가를 하고 있지만, 잘하고 있는지는 명확하지 않습니다.

Posit Cloud에서 새 Quarto 문서를 만들고 저장한 다음, 새 R 코드 청크를 만들고 서문 문서를 추가하십시오. 그런 다음 필요한 패키지를 설치 및/또는 로드하십시오. 다시 tidyverse와 janitor를 사용할 것입니다. 이들은 이전에 설치되었으므로 다시 설치할 필요가 없습니다. 또한 lubridate도 사용할 것입니다. 이것은 tidyverse의 일부이므로 독립적으로 설치할 필요는 없지만 로드해야 합니다. 또한 opendatatoronto와 knitr도 사용할 것이며, 이들은 설치하고 로드해야 합니다.

```
##### 🔒 #####
# 📝: 2021년 07월 01일 오후 10시 00분.
# 📝: Rohan Alexander
# 📩: rohan.alexander@utoronto.ca
# 📅: 2022년 7월 1일
# 📜 📜: -

##### 🔒 🔒 🔒 #####
install.packages("opendatatoronto")
install.packages("knitr")

library(knitr)
library(janitor)
library(lubridate)
library(opendatatoronto)
library(tidyverse)
```

이전 예제에 세부 정보를 추가하자면, 패키지에는 다른 사람들이 작성한 코드가 포함되어 있습니다. 이 책에서 정기적으로 보게 될 몇 가지 일반적인 패키지가 있습니다. 특히 tidyverse. 패키지를 사용하려면 먼저 설치해야 하고, 그런 다음 로드해야 합니다. 패키지는 컴퓨터당 한 번만 설치하면 되지만, 사용할 때마다 로드해야 합니다. 이는 이전에 설치한 패키지를 여기에서 다시 설치할 필요가 없음을 의미합니다.

i 거인의 어깨

로버트 젠틀맨 박사는 R의 공동 창시자입니다. 1988년 위싱턴 대학교에서 통계학 박사 학위를 취득한 후 오클랜드 대학교로 옮겼습니다. 그 후 23andMe를 포함한 다양한 역할을 수행했으며, 현재 하버드 의과대학 계산 생의학 센터의 전무이사입니다.

i 거인의 어깨

로스 이아카 박사는 R의 공동 창시자입니다. 그는 1985년 캘리포니아 대학교 버클리에서 통계학 박사 학위를 취득했습니다. 그는 마오리족 지진의 신인 “Ruaumoko”라는 제목의 논문을 작성했습니다. 그 후 오클랜드 대학교로 옮겨 평생 그곳에 머물렀습니다. 그는 2008년 뉴질랜드 왕립 학회 테 아파랑기에서 피커링 메달을 수상했습니다.

사람들이 R과 우리가 사용하는 패키지를 만들기 위해 시간을 기부한다는 점을 고려할 때, 그들을 인용하는 것이 중요합니다. 필요한 정보를 얻기 위해 `citation()`을 사용합니다. 인수를 사용하지 않고 실행하면 R 자체에 대한 인용 정보를 제공하고, 패키지 이름인 인수를 사용하여 실행하면 해당 패키지에 대한 인용 정보를 제공합니다.

```
citation() # R의 저작권 및 저작자
```

To cite R in publications use:

R Core Team (2025). _R: A Language and Environment for Statistical Computing_. R Foundation for Statistical Computing, Vienna, Austria.
<https://www.R-project.org/>.

A BibTeX entry for LaTeX users is

```
@Manual{,
  title = {R: A Language and Environment for Statistical Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2025},
  url = {https://www.R-project.org/},
}
```

We have invested a lot of time and effort in creating R, please cite it when using it for data analysis. See also 'citation("pkgname")' for citing R packages.

```
citation("ggplot2") # ggplot2의 저작권 및 저작자
```

To cite ggplot2 in publications, please use

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

A BibTeX entry for LaTeX users is

```
@Book{,
  author = {Hadley Wickham},
  title = {ggplot2: Elegant Graphics for Data Analysis},
```

```

publisher = {Springer-Verlag New York},
year = {2016},
isbn = {978-3-319-24277-4},
url = {https://ggplot2.tidyverse.org},
}

```

시뮬레이션으로 돌아가서, 우리는 “date”, “shelter”, “occupancy” 세 가지 변수가 필요합니다. 이 예제는 `set.seed()`를 사용하여 시드를 추가하여 이전 예제를 기반으로 구축됩니다. 시드는 동일한 코드를 실행할 때마다 항상 동일한 무작위 데이터를 생성할 수 있도록 합니다. 어떤 정수든 시드로 사용할 수 있습니다. 이 경우 시드는 853입니다. 이 시드를 사용하면 이 예제와 동일한 무작위 숫자를 얻을 수 있습니다. 다른 시드를 사용하면 다른 무작위 숫자를 예상해야 합니다. 마지막으로, `rep()`을 사용하여 특정 횟수만큼 무언가를 반복합니다. 예를 들어, “Shelter 1”을 365번 반복합니다. 이는 약 1년치에 해당합니다.

```

##### 例題 #####
set.seed(853)

simulated_occupancy_data <-
tibble(
  date = rep(x = as.Date("2021-01-01") + c(0:364), times = 3),
  # Eddelbuettel の: https://stackoverflow.com/a/21502386
  shelter = c(
    rep(x = "Shelter 1", times = 365),
    rep(x = "Shelter 2", times = 365),
    rep(x = "Shelter 3", times = 365)
  ),
  number_occupied =
  rpois(
    n = 365 * 3,
    lambda = 30
  ) # ので 例題 1,095 の
)

```

```
head(simulated_occupancy_data)
```

```

# A tibble: 6 x 3
  date      shelter  number_occupied
  <date>    <chr>          <int>
1 2021-01-01 Shelter 1            28
2 2021-01-02 Shelter 1            29
3 2021-01-03 Shelter 1            35
4 2021-01-04 Shelter 1            25
5 2021-01-05 Shelter 1            21
6 2021-01-06 Shelter 1            30

```

이 시뮬레이션에서 우리는 먼저 2021년의 모든 날짜 목록을 만듭니다. 그 목록을 세 번 반복합니다. 우리는 1년 내내 매일 세 개의 쉼터에 대한 데이터를 가정합니다. 매일 밤 점유된 침대 수를 시뮬레이션하기 위해, 우리는 포아송 분포에서 추출하며, 평균 30개의 침대가 쉼터당 점유된다고 가정합니다. 이는 임의의 선택입니다. 배경 설명을 하자면, 포아송 분포는 종종 카운트 데이터가 있을 때 사용되며, `?@sec-its-just-a-generalized-linear-model`에서 다시 다룹니다.

2.3.3 획득

토론토 시에서 제공하는 토론토 쉼터 사용량 데이터를 사용합니다. 쉼터 사용량은 매일 새벽 4시에 점유된 침대 수를 세어 측정합니다. 데이터에 액세스하려면 `opendatatoronto`를 사용한 다음 자체 사본을 저장합니다.

```
#### # # ####
toronto_shelters <-
# 이 데이터 Open Data Toronto의 키워드 "쉼터"로 된 ID와 일치합니다.
list_package_resources("21c83b32-d5a8-4106-a54f-010dbe49f6f2") |>
# 해당 키워드 2021년 데이터로 필터링합니다.
filter(name ==
  "daily-shelter-overnight-service-occupancy-capacity-2021.csv") |>
# 데이터는 날짜로 정렬된 형태로 있습니다.
get_resource()

write_csv(
  x = toronto_shelters,
  file = "toronto_shelters.csv"
)

head(toronto_shelters)
```

이것을 우리가 관심 있었던 데이터셋(그림 ??)과 유사하게 만들기 위해 많은 작업이 필요하지 않습니다. `clean_names()`를 사용하여 이름을 더 쉽게 입력할 수 있도록 변경하고, `select()`를 사용하여 관련 열만 남겨야 합니다.

```
toronto_shelters_clean <-
  clean_names(toronto_shelters) |>
  mutate(occupancy_date = ymd(occupancy_date)) |>
  select(occupancy_date, occupied_beds)

head(toronto_shelters_clean)
```

```
# A tibble: 6 x 2
  occupancy_date occupied_beds
  <date>           <dbl>
1 2021-01-01        NA
2 2021-01-01        NA
3 2021-01-01        NA
4 2021-01-01        NA
5 2021-01-01        NA
6 2021-01-01         6
```

남은 것은 정리된 데이터셋을 저장하는 것입니다.

```
write_csv(
  x = toronto_shelters_clean,
  file = "cleaned_toronto_shelters.csv"
)
```

2.3.4 탐색

먼저, 방금 만든 데이터셋을 로드합니다.

```
#### # # ####
toronto_shelters_clean <-
  read_csv(
    "cleaned_toronto_shelters.csv",
```

표 2.1: 2021년 토론토 쉼터 사용량

```
toronto_shelters_clean |>
  mutate(occupancy_month = month(
    occupancy_date,
    label = TRUE,
    abbr = FALSE
  )) |>
  arrange(month(occupancy_date)) |>
  drop_na(occupied_beds) |>
  summarise(number_occupied = mean(occupied_beds),
            .by = occupancy_month) |>
  tt()
```

표 2.2: 2021년 토론토 쉼터 사용량

```
toronto_shelters_clean |>
  mutate(occupancy_month = month(
    occupancy_date,
    label = TRUE,
    abbr = FALSE
  )) |>
  arrange(month(occupancy_date)) |>
  drop_na(occupied_beds) |>
  summarise(number_occupied = mean(occupied_beds),
            .by = occupancy_month) |>
  tt(
    digits = 1
  ) |>
  style_tt(j = 2, align = "r") |>
  setNames(c("月", "月の 月の 月の 月の 月"))
```

```
show_col_types = FALSE
)
```

데이터셋에는 각 쉼터에 대한 일일 기록이 포함되어 있습니다. 우리는 각 월의 평균 사용량을 이해하는 데 관심이 있습니다. 이를 위해 lubridate의 month()를 사용하여 월 열을 추가해야 합니다. 기본적으로 month()는 월의 숫자를 제공하므로, 월의 전체 이름을 얻기 위해 “label”과 “abbr” 두 가지 인수를 포함합니다. tidyverse의 일부인 tidyr의 drop_na()를 사용하여 침대 수에 대한 데이터가 없는 행을 제거합니다. 우리는 시작하는 데 중점을 두기 때문에 여기서는 생각 없이 이 작업을 수행할 것이지만, 이는 중요한 결정이며 장 ??와 장 ??에서 누락된 데이터에 대해 더 자세히 이야기합니다. 그런 다음 dplyr의 summarise()를 사용하여 월별 그룹을 기반으로 요약 통계를 생성합니다. tinytable의 tt()를 사용하여 표 ??를 생성합니다.

이전과 마찬가지로, 이것은 괜찮아 보이며 우리가 목표로 한 것을 달성합니다. 그러나 기본값을 약간 조정하여 더 보기 좋게 만들 수 있습니다 (표 ??). 특히 열 이름을 더 읽기 쉽게 만들고, 적절한 소수점 이하 자릿 수만 표시하고, 정렬을 변경합니다 (j는 관심 있는 열 번호를 지정하는 데 사용되고 r은 정렬 유형, 즉 오른 쪽 정렬입니다).

2.3.5 공유

우리는 우리가 한 일, 왜 했는지, 그리고 무엇을 발견했는지에 대해 몇 단락을 작성하여 작업을 요약해야 합니다. 예시는 다음과 같습니다.

토론토에는 많은 노숙자 인구가 있습니다. 혹독한 겨울은 쉼터에 충분한 공간이 있는 것이 중요함을 의미합니다. 우리는 추운 달과 따뜻한 달에 쉼터 사용량이 어떻게 변하는지 이해하는데 관심이 있습니다.

우리는 토론토 시에서 제공하는 토론토 쉼터 침대 점유율 데이터를 사용합니다. 특히, 매일 새벽 4시에 점유된 침대 수를 세어 측정합니다. 우리는 이를 월별 평균으로 계산하는 데 관심이 있습니다. 우리는 통계 프로그래밍 언어 R (R Core Team 2024)과 tidyverse (Wickham 2017), janitor (Firke 2023), opendatatoronto (Gelfand 2022b), lubridate (Grolemund 와/과 Wickham 2011), knitr (Xie 2023)를 사용하여 데이터셋을 정리하고 정돈하고 분석했습니다. 그런 다음 각 월의 매일 밤 평균 점유 침대 수 표를 만들었습니다 (표 ??).

2021년 12월의 일일 평균 점유 침대 수는 7월의 30개에 비해 34개로 더 높았습니다 (표 ??). 더 일반적으로, 7월부터 12월까지 일일 평균 점유 침대 수가 꾸준히 증가했으며, 매월 약간의 전체적인 증가가 있었습니다.

데이터셋은 쉼터를 기반으로 하므로, 우리의 결과는 특히 크거나 작은 쉼터에 특정한 변화에 의해 왜곡될 수 있습니다. 특정 쉼터가 추운 달에 특히 매력적일 수 있습니다. 또한, 우리는 점유된 침대 수에 관심이 있었지만, 계절에 따라 침대 공급이 변한다면, 추가적인 관심 통계는 점유율이 될 것입니다.

이 예제는 몇 단락에 불과하지만, 초록을 형성하도록 축소하거나, 각 단락을 섹션으로 확장하여 전체 보고서를 형성할 수 있습니다. 첫 번째 단락은 일반적인 개요, 두 번째 단락은 데이터에 초점, 세 번째 단락은 결과에 초점, 네 번째 단락은 토론입니다. Hao (2019) 의 예시를 따라, 네 번째 단락은 편향이 스며들 수 있는 영역을 고려하기에 좋은 곳입니다.

2.4 신생아 사망률

신생아 사망률은 생후 첫 달 이내에 발생하는 사망을 의미합니다. 신생아 사망률(NMR)은 1,000명당 신생아 사망자 수입니다 (UN IGME 2021). 제3차 지속 가능한 개발 목표(SDG)는 NMR을 12로 줄이는 것을 목표로 합니다. 이 예제에서는 지난 50년간 아르헨티나, 호주, 캐나다, 케냐의 추정 NMR 그래프를 만들 것입니다.

2.4.1 계획

이 예제에서는 데이터셋이 어떻게 생겼는지, 그리고 그래프가 어떻게 생겼는지 생각해야 합니다.

데이터셋에는 국가와 연도를 지정하는 변수가 있어야 합니다. 또한 해당 국가의 해당 연도에 대한 NMR 추정치가 포함된 변수도 있어야 합니다. 대략적으로 그림 ?? 와 같아야 합니다.

우리는 x축에 연도, y축에 추정 NMR이 있는 그래프를 만들고 싶습니다. 각 국가는 자체 시리즈를 가져야 합니다. 우리가 찾고 있는 것의 빠른 스케치는 그림 ?? 입니다.

2.4.2 시뮬레이션

우리는 계획과 일치하는 데이터를 시뮬레이션하고 싶습니다. 이 경우 국가, 연도, NMR 세 가지 열이 필요합니다.

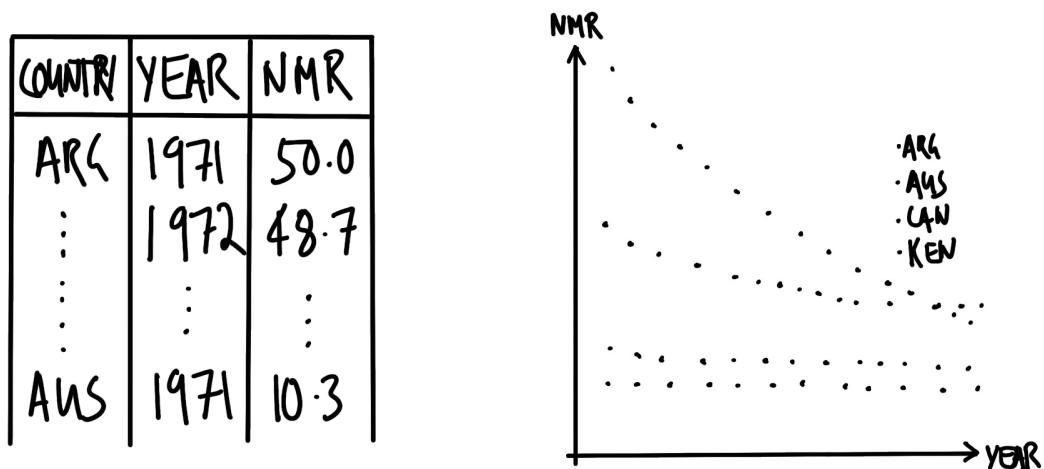


그림 2.7: 신생아 사망률(NMR)에 대한 데이터셋 및 그래프 스케치

Posit Cloud 내에서 새 Quarto 문서를 만들고 저장하십시오. 서문 문서를 추가하고 작업 공간을 설정하십시오. `tidyverse`, `janitor`, `lubridate`를 사용할 것입니다.

```
##### ❶ #####
# ❷: 이 50년간 4년마다 국가별 신생아 사망률은 어떤 패턴을 보였나?
# ❸: Rohan Alexander
# ❹: rohan.alexander@utoronto.ca
# ❺: 2022년 7월 1일
# ❻: -
```

```
##### ❽ ❾ ❿ #####
library(janitor)
library(lubridate)
library(tidyverse)
```

패키지에 포함된 코드는 저자가 업데이트하고 새 버전을 출시함에 따라 때때로 변경될 수 있습니다. `packageVersion()`을 사용하여 사용 중인 패키지 버전을 확인할 수 있습니다. 예를 들어, `tidyverse` 버전 2.0.0과 `janitor` 버전 2.2.0을 사용하고 있습니다.

```
packageVersion("tidyverse")
```

```
[1] '2.0.0'
```

```
packageVersion("janitor")
```

```
[1] '2.2.1'
```

설치된 모든 패키지의 버전을 업데이트하려면 `update.packages()`를 사용합니다. `tidyverse_update()`를 사용하여 `tidyverse` 패키지만 설치할 수 있습니다. 이것은 매일 실행할 필요는 없지만, 때때로 패키지를 업데이트하는 것이 좋습니다. 많은 패키지가 하위 호환성을 보장하기 위해 노력하지만, 어느 시점부터는 불가

능해집니다. 패키지를 업데이트하면 이전 코드를 다시 작성해야 할 수 있습니다. 시작할 때는 큰 문제가 아니며, 어쨌든 장 ??에서 다루는 특정 버전을 로드하는 도구가 있습니다.

시뮬레이션으로 돌아가서, `rep()`를 사용하여 각 국가의 이름을 50번 반복하고, 50년의 기간을 전달할 수 있도록 합니다. 마지막으로, `runif()`를 사용하여 균일 분포에서 추출하여 해당 국가의 해당 연도에 대한 특정 NMR 값을 시뮬레이션합니다.

```
#### 테스트 코드 ####
set.seed(853)

simulated_nmr_data <-
tibble(
  country =
  c(rep("Argentina", 50), rep("Australia", 50),
    rep("Canada", 50), rep("Kenya", 50)),
  year =
  rep(c(1971:2020), 4),
  nmr =
  runif(n = 200, min = 0, max = 100)
)

head(simulated_nmr_data)
```

```
# A tibble: 6 x 3
  country     year     nmr
  <chr>      <int>   <dbl>
1 Argentina  1971 35.9
2 Argentina  1972 12.0
3 Argentina  1973 48.4
4 Argentina  1974 31.6
5 Argentina  1975 3.74
6 Argentina  1976 40.4
```

이 시뮬레이션은 작동하지만, 50년 대신 60년을 시뮬레이션하는 데 관심이 있다면 시간이 많이 걸리고 오류가 발생하기 쉽습니다. 이 코드를 개선하는 한 가지 방법은 50의 모든 인스턴스를 변수로 대체하는 것입니다.

```
#### 테스트 코드 ####
set.seed(853)

number_of_years <- 50

simulated_nmr_data <-
tibble(
  country =
  c(rep("Argentina", number_of_years), rep("Australia", number_of_years),
    rep("Canada", number_of_years), rep("Kenya", number_of_years)),
  year =
  rep(c(1:number_of_years + 1970), 4),
  nmr =
  runif(n = number_of_years * 4, min = 0, max = 100)
)

head(simulated_nmr_data)
```

```
# A tibble: 6 x 3
```

```
country      year      nmr
<chr>      <dbl> <dbl>
1 Argentina 1971 35.9
2 Argentina 1972 12.0
3 Argentina 1973 48.4
4 Argentina 1974 31.6
5 Argentina 1975  3.74
6 Argentina 1976 40.4
```

결과는 동일하지만, 이제 50년에서 60년으로 변경하려면 한 곳에서만 변경하면 됩니다.

이 시뮬레이션된 데이터셋은 비교적 간단하고 우리가 코드를 작성했기 때문에 신뢰할 수 있습니다. 그러나 실제 데이터셋으로 전환할 때, 그것이 주장하는 바와 일치하는지 확인하기가 더 어렵습니다. 데이터를 신뢰하더라도, 그 신뢰를 다른 사람들과 공유할 수 있어야 합니다. 한 가지 방법은 데이터가 제대로 되어 있는지 테스트를 설정하는 것입니다. 예를 들어, 우리는 다음을 예상합니다:

1. “country”는 이 네 가지 중 하나여야 합니다: “Argentina”, “Australia”, “Canada”, “Kenya”.
2. 반대로, “country”는 이 네 가지 국가를 모두 포함해야 합니다.
3. “year”는 1971보다 작거나 2020보다 크지 않아야 하며, 정수여야 합니다. 문자나 소수점이 있는 숫자가 아니어야 합니다.
4. “nmr”은 0에서 1,000 사이의 값이어야 하며, 숫자여야 합니다.

이러한 기능을 기반으로 데이터셋이 통과할 것으로 예상되는 일련의 테스트를 작성할 수 있습니다.

```
simulated_nmr_data$country |>
  unique() == c("Argentina", "Australia", "Canada", "Kenya")

simulated_nmr_data$country |>
  unique() |>
  length() == 4

simulated_nmr_data$year |> min() == 1971
simulated_nmr_data$year |> max() == 2020
simulated_nmr_data$nmr |> min() >= 0
simulated_nmr_data$nmr |> max() <= 1000
simulated_nmr_data$nmr |> class() == "numeric"
```

이러한 테스트를 통과했으므로 시뮬레이션된 데이터셋을 신뢰할 수 있습니다. 더 중요한 것은 이러한 테스트를 실제 데이터셋에 적용할 수 있다는 것입니다. 이를 통해 해당 데이터셋에 대한 신뢰를 높이고 그 신뢰를 다른 사람들과 공유할 수 있습니다.

2.4.3 획득

유엔 아동 사망률 추정 기관 간 그룹(IGME)은 다운로드하여 저장할 수 있는 NMR 추정치를 제공³합니다.

```
##### 例 例 #####
raw_igme_data <-
  read_csv(
    file =
      "https://childmortality.org/wp-content/uploads/2021/09/UNIGME-2021.csv",
    show_col_types = FALSE
  )

write_csv(x = raw_igme_data, file = "igme.csv")
```

³<https://childmortality.org/>

이와 같이 확립된 데이터의 경우, 데이터에 대한 지원 자료를 읽는 것이 유용할 수 있습니다. 이 경우 코드북은 여기⁴에서 사용할 수 있습니다. 이 후에는 데이터셋을 더 잘 이해하기 위해 빠르게 살펴볼 수 있습니다. `head()`로 데이터셋이 어떻게 생겼는지, `tail()`로 마지막 여섯 행을, `names()`로 열 이름을 확인할 수 있습니다.

```
names(raw_igme_data)
```

[1] "Geographic area"	"Indicator"	"Sex"
[4] "Wealth Quintile"	"Series Name"	"Series Year"
[7] "Regional group"	"TIME_PERIOD"	"OBS_VALUE"
[10] "COUNTRY_NOTES"	"CONNECTION"	"DEATH_CATEGORY"
[13] "CATEGORY"	"Observation Status"	"Unit of measure"
[16] "Series Category"	"Series Type"	"STD_ERR"
[19] "REF_DATE"	"Age Group of Women"	"Time Since First Birth"
[22] "DEFINITION"	"INTERVAL"	"Series Method"
[25] "LOWER_BOUND"	"UPPER_BOUND"	"STATUS"
[28] "YEAR_TO_ACHIEVE"	"Model Used"	

이름을 정리하고 관심 있는 행과 열만 유지하고 싶습니다. 계획에 따라 “Sex”가 “Total”, “Series Name”이 “UN IGME estimate”, “Geographic area”가 “Argentina”, “Australia”, “Canada”, “Kenya” 중 하나, “Indicator”가 “Neonatal mortality rate”인 행에 관심이 있습니다. 이 후에는 “geographic_area”, “time_period”, “obs_value” 몇 개의 열에만 관심이 있습니다.

```
cleaned_igme_data <-
  clean_names(raw_igme_data) |>
  filter(
    sex == "Total",
    series_name == "UN IGME estimate",
    geographic_area %in% c("Argentina", "Australia", "Canada", "Kenya"),
    indicator == "Neonatal mortality rate"
  ) |>
  select(geographic_area, time_period, obs_value)

head(cleaned_igme_data)
```

```
# A tibble: 6 x 3
  geographic_area time_period obs_value
  <chr>          <chr>        <dbl>
1 Argentina      1970-06     24.9
2 Argentina      1971-06     24.7
3 Argentina      1972-06     24.6
4 Argentina      1973-06     24.6
5 Argentina      1974-06     24.5
6 Argentina      1975-06     24.1
```

두 가지 다른 측면을 수정해야 합니다: “time_period”的 클래스는 문자이지만 연도여야 하고, “obs_value”的 이름은 더 유익하도록 “nmr”로 변경해야 합니다.

```
cleaned_igme_data <-
  cleaned_igme_data |>
  mutate(
    time_period = str_remove(time_period, "-06"),
    time_period = as.integer(time_period)
```

⁴https://childmortality.org/wp-content/uploads/2021/03/CME-Info_codebook_for_downloads.xlsx

```
) |>
filter(time_period >= 1971) |>
rename(nmr = obs_value, year = time_period, country = geographic_area)

head(cleaned_igme_data)
```

```
# A tibble: 6 x 3
  country     year    nmr
  <chr>      <int>  <dbl>
1 Argentina  1971  24.7
2 Argentina  1972  24.6
3 Argentina  1973  24.6
4 Argentina  1974  24.5
5 Argentina  1975  24.1
6 Argentina  1976  23.3
```

마지막으로, 시뮬레이션된 데이터셋을 기반으로 개발한 테스트를 데이터셋이 통과하는지 확인할 수 있습니다.

```
cleaned_igme_data$country |>
unique() == c("Argentina", "Australia", "Canada", "Kenya")
```

```
[1] TRUE TRUE TRUE TRUE
```

```
cleaned_igme_data$country |>
unique() |>
length() == 4
```

```
[1] TRUE
```

```
cleaned_igme_data$year |> min() == 1971
```

```
[1] TRUE
```

```
cleaned_igme_data$year |> max() == 2020
```

```
[1] TRUE
```

```
cleaned_igme_data$nmr |> min() >= 0
```

```
[1] TRUE
```

```
cleaned_igme_data$nmr |> max() <= 1000
```

```
[1] TRUE
```

```
cleaned_igme_data$nmr |> class() == "numeric"
```

```
[1] TRUE
```

남은 것은 깔끔하게 정리된 데이터셋을 저장하는 것입니다.

```
write_csv(x = cleaned_igme_data, file = "cleaned_igme_data.csv")
```

2.4.4 탐색

정리된 데이터셋을 사용하여 추정 NMR 그래프를 만들고 싶습니다. 먼저, 데이터셋을 읽어들입니다.

```
##### ❸ #####
cleaned_igme_data <-
  read_csv(
    file = "cleaned_igme_data.csv",
    show_col_types = FALSE
  )
```

이제 NMR이 시간 경과에 따라 어떻게 변했는지, 그리고 국가 간의 차이를 그래프로 만들 수 있습니다 (그림 ??).

```
cleaned_igme_data |>
  ggplot(aes(x = year, y = nmr, color = country)) +
  geom_point() +
  theme_minimal() +
  labs(x = "연도", y = "신생아 사망률(NMR)", color = "나라") +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "bottom")
```

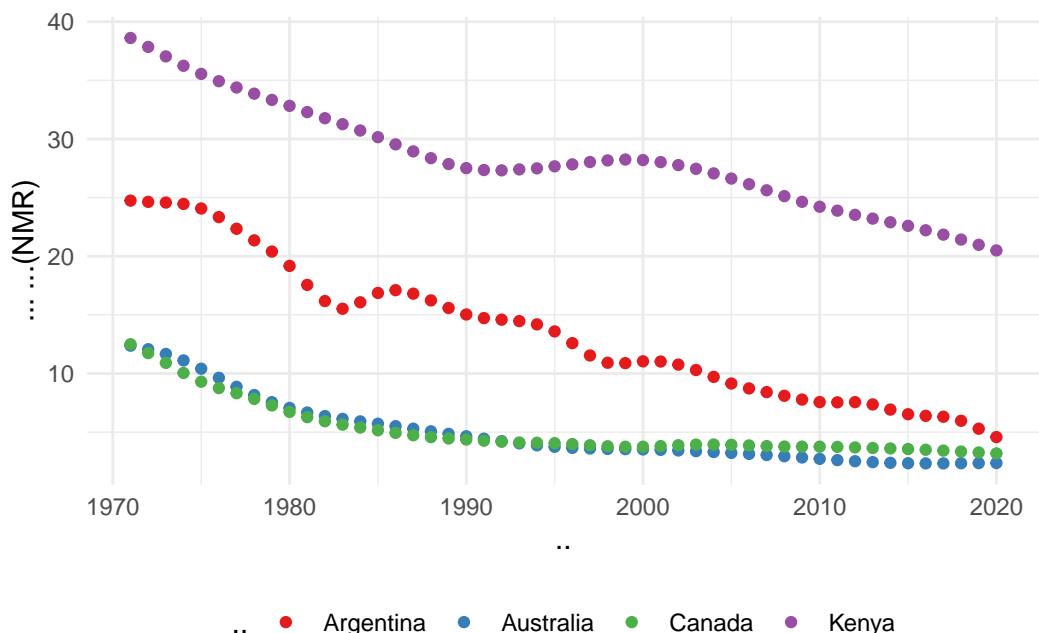


그림 2.8: 아르헨티나, 호주, 캐나다, 케냐의 신생아 사망률(NMR) (1971-2020)

2.4.5 공유

이 시점까지 우리는 데이터를 다운로드하고, 정리하고, 테스트를 작성하고, 그래프를 만들었습니다. 일반적으로 우리가 한 일을 어느 정도 자세히 전달해야 합니다. 이 경우, 우리가 한 일, 왜 했는지, 그리고 무엇을 발견했는지에 대해 몇 단락을 작성할 것입니다.

신생아 사망률은 생후 첫 달 이내에 발생하는 사망을 의미합니다. 특히, 신생아 사망률(NMR)은 1,000명당 신생아 사망자 수입니다. 우리는 지난 50년간 아르헨티나, 호주, 캐나다, 케냐 네 개국의 NMR 추정치를 얻습니다.

유엔 아동 사망률 추정 기관 간 그룹(IGME)은 웹사이트 <https://childmortality.org/>에서 NMR 추정치를 제공합니다. 우리는 그들의 추정치를 다운로드한 다음 통계 프로그래밍 언어 R (R Core Team 2024)을 사용하여 데이터셋을 정리하고 정돈했습니다.

우리는 시간 경과에 따른 추정 NMR과 관심 있는 네 개국 간의 상당한 변화를 발견했습니다 (그림 ??). 1970년대에는 추정 NMR이 감소하는 경향이 있었습니다. 호주와 캐나다는 그 시점에서 낮은 NMR을 보였고 2020년까지 그 수준을 유지했으며, 약간 더 감소했습니다. 아르헨티나와 케냐의 추정치는 2020년까지 계속해서 상당한 감소를 보였습니다.

우리의 결과는 시간 경과에 따른 추정 NMR의 상당한 개선을 시사합니다. NMR 추정치는 추정치를 뒷받침하는 통계 모델과 기본 데이터에 기반합니다. 데이터의 이중 부담은 종종 결과가 좋지 않은 그룹, 이 경우 국가에 대해 고품질 데이터를 쉽게 사용할 수 없다는 것입니다. 우리의 결론은 추정치를 뒷받침하는 모델과 기본 데이터의 품질에 따라 달라지며, 우리는 이를 중 어느 것도 독립적으로 검증하지 않았습니다.

2.5 결론

이 장에서 많은 내용을 다루었으며, 모든 것을 이해하지 못하는 것은 정상입니다. 가장 좋은 방법은 세 가지 사례 연구를 각자 시간을 내어 살펴보는 것입니다. 복사-붙여넣기 대신 모든 코드를 직접 입력하고, 완전히 이해하지 못하더라도 조금씩 실행하십시오. 그런 다음 자신만의 주석을 추가해 보십시오.

또한 이 시점에서 이 장의 모든 것을 완전히 이해할 필요는 없습니다. 일부 학생들은 이 책의 다음 몇장을 계속해서 읽고 나중에 이 장으로 돌아오는 것이 가장 좋다고 생각합니다. 흥미롭게도 우리는 한두 시간의 작업만으로 데이터를 사용하여 세상에 대해 무언가를 배울 수 있음을 보여주었습니다. 이러한 기술을 개발하면서, 우리는 또한 우리 작업의 더 넓은 영향에 대해 점점 더 정교하게 고려해야 합니다.

“우리는 우리 작업의 사회적 영향에 대해 생각할 필요가 없습니다. 왜냐하면 그것은 어렵고 다른 사람들이 우리를 위해 할 수 있기 때문입니다.”는 정말 나쁜 주장입니다. 저는 CV [컴퓨터 비전] 연구를 중단했습니다. 왜냐하면 제 작업이 미치는 영향을 보았기 때문입니다. 저는 그 작업을 좋아했지만, 군사적 응용과 개인 정보 보호 문제는 결국 무시할 수 없게 되었습니다. 그러나 기본적으로 모든 얼굴 인식 작업은 우리가 더 넓은 영향 쟝션을 진지하게 받아들인다면 출판되지 않을 것입니다. 거의 이점이 없고 엄청난 단점 위험이 있습니다. 공정하게 말하자면, 저는 여기서 많은 겸손을 가져야 합니다. 대학원 대부분 동안 저는 과학이 비정치적이고 연구는 주제가 무엇이든 객관적으로 도덕적이고 좋다는 신화를 믿었습니다.

Joe Redmon, 2020년 2월 20일

“데이터 과학”이라는 용어는 학계, 산업계, 심지어 더 일반적으로 널리 사용되지만, 우리가 보았듯이 정의하기 어렵습니다. 데이터 과학에 대한 의도적으로 적대적인 정의는 “인간성을 셀 수 있는 것으로 비인간적으로 축소하는 것”입니다 (Keyes 2019). 의도적으로 논란의 여지가 있지만, 이 정의는 지난 10년 동안 데이터 과학 및 양적 방법론에 대한 수요가 증가한 한 가지 이유를 강조합니다. 즉, 개인과 그들의 행동이 이제 그 중심에 있다는 것입니다. 많은 기술은 수십 년 동안 존재했지만, 지금 인기를 끄는 것은 이러한 인간 중심적인 접근 방식입니다.

불행히도, 많은 작업이 개인에게 초점을 맞추고 있음에도 불구하고, 개인 정보 보호 및 동의 문제, 그리고 더 넓은 윤리적 문제는 거의 최우선으로 고려되지 않는 것 같습니다. 일부 예외는 있지만, 일반적으로 AI, 머신러닝, 데이터 과학이 사회를 혁신할 것이라고 주장하면서도, 이러한 유형의 문제는 혁명을 받아들이기 전에 생각해야 할 것이 아니라, 있으면 좋은 것으로 취급되는 경향이 있습니다.

대부분의 경우, 이러한 유형의 문제는 새로운 것이 아닙니다. 과학 분야에서는 CRISPR 기술과 유전자 편집에 대한 광범위한 윤리적 고려가 있었습니다 (Brokowski 와/과 Adli 2019; Marchese 2022). 그리고 이전 시대에는 유사한 대화가 있었습니다. 예를 들어, 베르너 폰 브라운이 나치 독일을 위해 로켓을 만들었음에도 불구하고 미국을 위해 로켓을 만들도록 허용된 것에 대한 논의가 있었습니다 (M. Neufeld 2002; Wilford 1977). 의학 분야에서는 이러한 우려가 오랫동안 최우선으로 고려되었습니다 (American Medical Association and New York Academy of Medicine 1848). 데이터 과학은 다른 분야의 경험을 바탕으로 이러한 문제를 생각하고 사전에 해결하기보다는 자체적인 터스키기 순간을 맞이할 운명인 것 같습니다.

그렇긴 하지만, 일부 데이터 과학자들이 실습을 둘러싼 윤리에 대해 더 많은 관심을 갖기 시작했다는 증거가 있습니다. 예를 들어, 권위 있는 머신러닝 컨퍼런스인 NeurIPS는 2020년부터 모든 제출물에 윤리적 측면에 대한 진술을 요구하고 있습니다.

균형 잡힌 관점을 제공하기 위해 저자는 윤리적 측면 및 미래 사회적 결과를 포함하여 작업의 잠재적인 더 넓은 영향에 대한 진술을 포함해야 합니다. 저자는 긍정적인 결과와 부정적인 결과를 모두 논의하도록 주의해야 합니다.

NeurIPS 2020 컨퍼런스 논문 모집

데이터 과학의 윤리적 고려와 더 넓은 영향에 대한 우려의 목적은 어떤 것을 규범적으로 허용하거나 금지하는 것이 아니라, 가장 중요하게 다루어야 할 몇 가지 문제를 제기할 기회를 제공하는 것입니다. 데이터 과학 응용의 다양성, 분야의 상대적인 깊음, 그리고 변화의 속도는 그러한 고려 사항이 때때로 의도적으로 제쳐두어지며, 이는 해당 분야의 나머지 부분에서 허용된다는 것을 의미합니다. 이는 과학, 의학, 공학, 회계와 같은 분야와 대조됩니다. 아마도 이러한 분야는 더 자각적일 것입니다 (그림 ??).

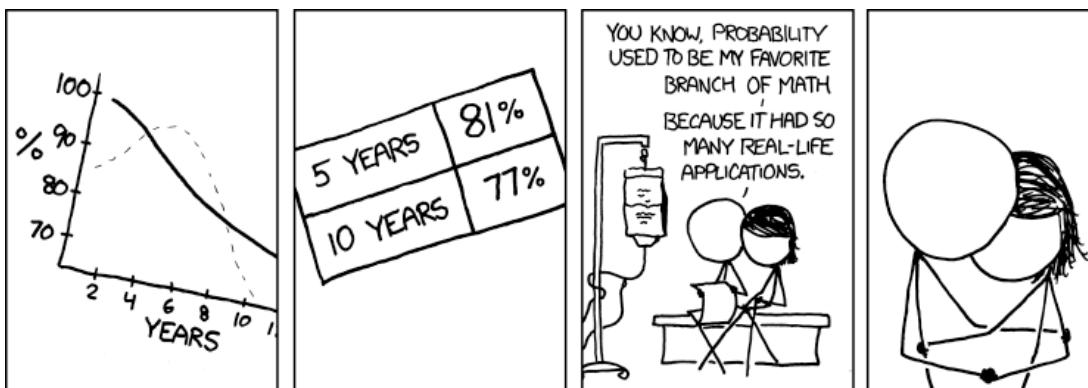


그림 2.9: 랜달 먼로의 “확률”에서 보여주듯이 숫자는 맥락에서 벗어날 수 없습니다: <https://xkcd.com/881/>.

2.6 연습 문제

퀴즈

1. 데이터 과학이란 무엇입니까 (자신의 말로)?
2. Register (2020b)에 따르면, 데이터 결정은 (하나를 선택하십시오)?
 - a. 실제 사람들에게 영향을 미칩니다.
 - b. 아무에게도 영향을 미치지 않습니다.
 - c. 훈련 세트에 있는 사람들에게 영향을 미칩니다.
 - d. 테스트 세트에 있는 사람들에게 영향을 미칩니다.
3. Keyes (2019)에 따르면, 데이터 과학이란 무엇입니까 (하나를 선택하십시오)?
 - a. 데이터 과학은 과학적 방법, 프로세스, 알고리즘 및 시스템을 사용하여 많은 구조화된 및 비구조화된 데이터에서 지식과 통찰력을 추출하는 학제 간 분야입니다.
 - b. 의사 결정을 위한 대량의 데이터에 대한 양적 분석.
 - c. 인간성을 셀 수 있는 것으로 비인간적으로 축소하는 것.
4. Keyes (2019)에 따르면, 표준화된 범주를 요구하는 데이터 시스템의 한 가지 결과는 무엇입니까 (하나를 선택하십시오)?
 - a. 사용자 경험 저하.
 - b. 보안 조치 손상.
 - c. 기술 혁신 증가.
 - d. 개인의 정체성과 경험의 말소.
5. Healy (2020)에 따르면, 데이터를 다루는 것에 대한 일반적인 비판은 무엇입니까 (하나를 선택하십시오)?
 - a. 너무 시간이 많이 걸리고 비효율적이라는 것.
 - b. 숫자 뒤에 있는 인간 삶의 현실과 거리를 두게 한다는 것.
 - c. 분석을 위해 값비싼 소프트웨어와 광범위한 훈련이 필요하다는 것.
6. Healy (2020)에 따르면, 그 비판에 대한 한 가지 답변은 무엇입니까 (하나를 선택하십시오)?
 - a. 데이터를 다루는 것은 의미에 대한 질문과 대면하게 합니다.
 - b. 데이터 분석은 수행되어서는 안 됩니다.
 - c. 데이터는 자동화된 프로세스에 의해서만 분석되어야 합니다.
 - d. 질적 접근 방식이 지배적인 접근 방식이어야 합니다.
7. Keyes (2019)와 Healy (2020)을 어떻게 조화시킬 수 있습니까?
8. 윤리가 데이터 과학의 핵심 요소인 이유는 무엇입니까 (하나를 선택하십시오)?
 - a. 데이터 과학은 항상 민감한 개인 정보를 포함하기 때문입니다.
 - b. 윤리적 고려 사항이 분석을 더 쉽게 만들기 때문입니다.
 - c. 데이터셋은 인간과 관련될 가능성이 높으며 맥락을 고려해야 하기 때문입니다.
 - d. 규제가 모든 데이터 분석에 윤리 승인을 요구하기 때문입니다.
9. 이 장에서 설명된 Crawford (2021)에 따르면, 다음 중 우리 세상과 따라서 우리의 데이터를 형성하는 힘은 무엇입니까 (모두 선택하십시오)?
 - a. 정치적.
 - b. 물리적.
 - c. 역사적.
 - d. 문화적.
 - e. 사회적.
10. Ford (2015)에 따르면, 컴파일러란 무엇입니까 (하나를 선택하십시오)?
 - a. 파일에 입력한 기호를 하위 수준 명령으로 변환하는 소프트웨어.
 - b. 누군가가 입력하거나 복사하거나 다른 곳에서 붙여넣은 일련의 기호 (일반적인 키보드 문자를 사용하여 어떤 종류의 파일로 저장됨).
 - c. 이점이 있는 시계.
 - d. 편치 카드에 구멍을 뚫고, 상자에 넣고, 로드한 다음, 컴퓨터가 카드를 넘겨 구멍이 있는 곳을 식별하고 메모리 일부를 업데이트하는 것.
11. 성별에 대한 설문조사 결과가 다음과 같다고 가정해 봅시다: “남성: 879”, “여성: 912”, “논바이너리: 10”, “말하고 싶지 않음: 3”, “기타: 1”. “말하고 싶지 않음”을 고려하는 적절한 방법은 무엇입니까 (하나를 선택하십시오)?

- a. 삭제합니다.
 - b. 상황에 따라 다릅니다.
 - c. 포함합니다.
 - d. “기타”로 병합합니다.
12. 인종 및/또는 성별을 예측 변수로 포함하면 모델의 성능이 향상되는 직업을 가지고 있다고 가정해 봅시다. 분석에 이러한 변수를 포함할지 여부를 결정할 때 어떤 요소를 고려할 것입니까 (자신의 말로)?
13. 데이터 과학에서 재현성이란 무엇을 의미합니까 (하나를 선택하십시오)?
- a. 다른 데이터셋으로 유사한 결과를 생성할 수 있는 것.
 - b. 분석의 모든 단계를 다른 사람이 독립적으로 다시 수행할 수 있도록 보장하는 것.
 - c. 동료 심사 저널에 결과를 게시하는 것.
 - d. 데이터를 보호하기 위해 독점 소프트웨어를 사용하는 것.
14. 측정과 관련된 과제는 무엇입니까 (하나를 선택하십시오)?
- a. 일반적으로 간단하고 거의 주의를 기울일 필요가 없습니다.
 - b. 무엇을 어떻게 측정할지 결정하는 것은 복잡하고 맥락에 따라 다릅니다.
 - c. 데이터 수집은 객관적이고 편향이 없습니다.
 - d. 측정은 항상 정확하고 시간이 지나도 일관적입니다.
15. 조각가 비유에서 조각하는 행위는 데이터 워크플로우에서 무엇을 나타냅니까 (하나를 선택하십시오)?
- a. 데이터에 맞는 복잡한 모델을 생성하는 것.
 - b. 원시 데이터를 획득하는 것.
 - c. 필요한 데이터셋을 드러내기 위해 데이터를 정리하고 준비하는 것.
 - d. 결과를 시각화하는 것.
16. 탐색적 데이터 분석(EDA)이 개방형 프로세스인 이유는 무엇입니까 (하나를 선택하십시오)?
- a. 따라야 할 고정된 단계가 있기 때문입니다.
 - b. 데이터의 형태와 패턴을 이해하기 위해 지속적인 반복이 필요하기 때문입니다.
 - c. 구조화된 방식으로 가설을 테스트하는 것을 포함하기 때문입니다.
 - d. 자동화할 수 있기 때문입니다.
17. 통계 모델을 신중하게 사용해야 하는 이유는 무엇입니까 (하나를 선택하십시오)?
- a. 항상 결정적인 결과를 제공하기 때문입니다.
 - b. 초기 단계에서 내려진 결정을 반영할 수 있기 때문입니다.
 - c. 대부분의 청중에게 너무 복잡하기 때문입니다.
 - d. 데이터가 잘 제시되면 불필요하기 때문입니다.
18. 키 측정의 어려움에 대해 생각하는 것에서 얻을 수 있는 한 가지 교훈은 무엇입니까 (하나를 선택하십시오)?
- a. 키는 변동성이 거의 없는 간단한 측정입니다.
 - b. 모든 측정은 올바른 도구로 수행되면 정확합니다.
 - c. 간단한 측정조차도 데이터 품질에 영향을 미치는 복잡성을 가질 수 있습니다.
 - d. 키는 데이터 분석에서 유용한 변수가 아닙니다.
19. 데이터셋에서 누락된 사람을 고려하지 않는 것의 위험은 무엇입니까 (하나를 선택하십시오)?
- a. 분석에 큰 영향을 미치지 않습니다.
 - b. 데이터 양을 줄여 분석을 단순화합니다.
 - c. 전체 맥락을 나타내지 않는 결론으로 이어질 수 있습니다.
20. 통계 모델링의 목적은 무엇입니까 (하나를 선택하십시오)?
- a. 데이터를 탐색하고 이해하는 데 도움이 되는 도구.
 - b. 가설을 증명하는 것.
 - c. 탐색적 데이터 분석을 대체하는 것.
21. “우리 데이터는 지저분하고 복잡한 세상의 단순화이다”라는 말은 무엇을 의미합니까 (하나를 선택하십시오)?
- a. 데이터는 현실의 모든 측면을 완벽하게 포착합니다.
 - b. 데이터는 분석을 가능하게 하기 위해 현실을 단순화하지만, 모든 세부 사항을 포착할 수는 없습니다.
 - c. 데이터는 항상 부정확하고 쓸모없습니다.

수업 활동

- 강사는 수업 사진을 찍은 다음 화면에 사진을 표시해야 합니다. 소그룹으로 학생들은 사진이 보여주는 세 가지 측면과 보여주지 않는 세 가지 측면을 식별해야 합니다. 이것이 데이터 과학과 어떻게 관련되는지 논의하십시오.
- 강사는 각 그룹에 측정에 사용할 다른 항목을 제공해야 합니다. 일부는 다른 것보다 더 유용합니다. 예를 들어, 줄자, 종이, 자, 마커, 저울 등. 그런 다음 학생들은 해당 항목을 사용하여 다음 질문에 답해야 합니다: “머리카락 길이는 얼마입니까?”. 숫자를 스프레드시트에 추가하십시오. 스프레드시트만 있다면 머리카락 길이에 대해 무엇을 이해하고 무엇을 이해하지 못할 것입니까? 이를 더 넓은 데이터 과학과 연결하십시오.

과제

이 과제의 목적은 곁보기에 간단해 보이는 것조차 측정의 어려움을 명확히 하고, 따라서 더 복잡한 영역에서 측정 문제의 가능성을 명확히 하는 것입니다.

무, 겨자잎, 루꼴라와 같이 빠르게 자라는 식물의 씨앗을 구하십시오. 씨앗을 심고 사용한 흙의 양을 측정하십시오. 물을 주고 사용한 물의 양을 측정하십시오. 매일 변화를 기록하십시오. 더 일반적으로, 가능한 많이 측정하고 기록하십시오. 측정의 어려움에 대한 생각을 기록하십시오. 결국 씨앗이 싹을 틔울 것이고, 어떻게 자라는지 측정해야 합니다.

1. 계획:

- 데이터셋: 각 관측치는 선거구 이름과 당선된 후보의 정당을 포함해야 합니다.
- 그래프: 각 정당이 얻은 선거구 수를 보여주는 그래프를 만들어야 합니다.

2. 시뮬레이션:

- Quarto 문서를 생성합니다.
- 필요한 패키지(tidyverse, janitor)를 로드합니다.
- 선거구에 정당을 무작위로 할당하여 선거 결과를 시뮬레이션합니다: 선거구 번호를 추가한 다음, 6가지 옵션 중 하나를 복원 추출 방식으로 338번 무작위로 선택하기 위해 sample() 함수를 사용합니다.

3. 획득:

- 캐나다 선거관리위원회에서 CSV 파일을 여기⁵에서 다운로드합니다.
- 이름을 정리한 다음, 관심 있는 두 열(“electoral_district_name_nom_de_circonscription” 및 “elected_candidate_candidat_elu”)을 선택합니다. 마지막으로, 프랑스어 부분을 제거하고 이름을 단순화하여 열 이름을 변경합니다.
- 필요한 열은 당선된 후보에 대한 것입니다. 이 열에는 당선된 후보의 성, 이름, 공백, 그리고 슬래시로 구분된 영어와 프랑스어 정당 이름이 포함되어 있습니다. tidyverse의 separate()를 사용하여 이 열을 조각으로 분리한 다음, select()를 사용하여 정당 정보만 유지합니다 (아래에 일부 도우미 코드가 있습니다).
- 마지막으로, 시뮬레이션한 내용과 일치하도록 정당 이름을 프랑스어에서 영어로 다시 코딩합니다.

... { .cell }

```
cleaned_elections_data <-  
  cleaned_elections_data |>  
  separate(  
    col = elected_candidate,  
    into = c("Other", "party"),  
    sep = "/"  
  ) |>  
  select(-Other)
```

⁵https://www.elections.ca/res/rep/off/ovr2021app/53/data_donnees/table_tableau11.csv

:::

4. 탐색:

- 2021년 캐나다 연방 선거에서 각 정당이 얻은 선거구 수를 멋진 그래프로 만듭니다.

5. 공유:

- 무엇을 했고, 왜 했으며, 무엇을 발견했는지에 대해 몇 단락으로 작성합니다. GitHub Gist 링크를 제출합니다.

3

재현 가능한 워크플로우

- i** Chapman and Hall/CRC는 이 책을 2023년 7월에 출판했습니다. 여기^a에서 구매할 수 있습니다. 이 온라인 버전은 인쇄된 내용에 일부 업데이트가 있습니다.

^a<https://www.routledge.com/Telling-Stories-with-Data-With-Applications-in-R/Alexander/p/book/9781032134772>

선행 조건

- What has happened down here is the winds have changed 읽기, (Gelman 2016)
 - 복제 위기(replication crisis)와 사회 과학이 이에 대응하여 어떻게 변화했는지에 대한 개요를 제공하는 블로그 게시물입니다.
- Good enough practices in scientific computing 읽기, (Wilson 기타 2017)
 - 컴퓨터 사용 방식에 초점을 맞춰 데이터 과학을 수행하는 방법에 대한 명확하고 쉽게 채택할 수 있는 권장 사항을 제공하는 논문입니다.
- How to improve your relationship with your future self 읽기, (Bowers 와/과 Voors 2016)
 - 분석에 초점을 맞춰 데이터 과학을 수행하는 방법에 대한 명확하고 쉽게 채택할 수 있는 권장 사항을 제공하는 논문입니다.
- Overcoming barriers to sharing code 시청, (M. Alexander 2021)
 - 이 비디오는 코드 공유에 익숙해지는 것에 대한 개인적인 성찰입니다.
- Make a reprex\$… Please 시청, (Gelfand 2021)
 - 이 비디오는 도움을 요청할 때 재현 가능한 예제를 만드는 것이 왜 그렇게 중요한지 자세히 설명합니다.
- The tidyverse style guide 읽기, (Wickham 2021c)
 - R에서 코딩할 때 권장되는 모범 사례를 문서화한 웹사이트입니다.
- Code smells and feels 시청, (Bryan 2018b)
 - 코딩할 때 피해야 할 사항을 자세히 설명하는 비디오입니다.

주요 개념 및 기술

- 재현성은 일반적으로 누군가가 당신에게 부과하는 것으로 시작됩니다. 그것은 번거롭고 짜증날 수 있습니다. 이것은 일반적으로 짧은 휴식 후에 프로젝트를 다시 방문해야 할 때까지 지속됩니다. 그 시점에서 당신은 재현성이 데이터 과학의 요구 사항일 뿐만 아니라 진정한 발전을 이룰 수 있는 유일한 방법이기 때문에 우리 자신을 돋는 데 도움이 된다는 것을 깨닫게 됩니다.
- 재현성은 데이터, 코드 및 환경 공유를 의미합니다. 이는 Quarto, R 프로젝트, Git 및 GitHub를 사용하여 향상됩니다. Quarto는 일반 텍스트와 R 코드를 통합하는 문서를 구축합니다. R 프로젝트는 사용자의 개인 디렉토리 설정에 의존하지 않는 파일 구조를 가능하게 합니다. Git 및 GitHub는 코드와 데이터를 더 쉽게 공유할 수 있도록 합니다.
- 이것은 흡잡을 데 없는 워크플로우는 아니지만, 충분히 좋고 많은 이점을 제공합니다. 다양한 도구를 통해 다양한 측면을 개선할 것이지만, 코드 구조와 주석을 개선하는 것이 큰 도움이 됩니다.
- 항상 오류가 발생하며, 디버깅은 연습을 통해 향상되는 기술임을 인식하는 것이 중요합니다. 그러나 도움을 받을 수 있는 핵심 측면 중 하나는 다른 사람들이 사용할 수 있는 재현 가능한 예제를 만들 수 있다는 것입니다.

소프트웨어 및 패키지

- Base R (R Core Team 2024)
- AER (Kleiber 와/과 Zeileis 2008)
- future (Bengtsson 2021)

- `gitcreds` (Csárdi 2022)
- `lintr` (Hester 기타 2022)
- `renv` (Ushey 2022)
- `reprex` (Bryan 기타 2022)
- `styler` (Müller 와/과 Walthert 2022)
- `tidyverse` (Wickham 기타 2019)
- `tinytable` (Arel-Bundock 2024)
- `tinytex` (Xie 2019)
- `usethis` (Wickham, Bryan, 와/과 Barrett 2022)

```
#| message: false
#| warning: false
```

```
library(AER)
library(future)
library(gitcreds)
library(lintr)
library(renv)
library(reprex)
library(styler)
library(tidyverse)
library(tinytable)
library(tinytex)
library(usethis)
```

3.1 서론

기계 학습에 대해 명심해야 할 가장 중요한 점은 성능이 한 데이터 세트의 샘플로 평가되지만, 모델은 반드시 동일한 특성을 따르지 않을 수 있는 샘플에서 사용된다는 것입니다. 따라서 “90% 정확도로 평가된 모델을 사용할 것인가, 아니면 80% 정확도로 평가된 인간을 사용할 것인가”라는 질문에 대한 답은 데이터가 평가 프로세스에 따라 일반적인지 여부에 달려 있습니다. 인간은 적응력이 있지만, 모델은 그렇지 않습니다. 상당한 불확실성이 있다면 인간을 선택하십시오. 그들은 (엄청난 양의 데이터로 훈련된 모델에 비해) 패턴 인식 능력이 떨어질 수 있지만, 그들은 자신이 하는 일을 이해하고, 그것에 대해 추론할 수 있으며, 새로운 것에 직면했을 때 즉흥적으로 대처할 수 있습니다.

프랑수아 솔레, 2020년 2월 20일.

과학이 테스트 가능한 설명과 예측의 관점에서 지식을 체계적으로 구축하고 조직하는 것이라면, 데이터 과학은 이를 데이터에 집중합니다. 이는 지식을 구축, 조직 및 공유하는 것이 중요한 측면임을 의미합니다. 당신만이 할 수 있는 방식으로 지식을 한 번 만드는 것은 이 기준을 충족하지 못합니다. 따라서 재현 가능한 데이터 과학 워크플로우가 필요합니다.

M. Alexander (2019a)는 재현 가능한 연구를 사용된 모든 자료가 주어졌을 때 정확히 다시 수행할 수 있는 연구로 정의합니다. 이는 코드, 데이터 및 환경을 제공하는 것의 중요성을 강조합니다. 최소한의 기대치는 다른 사람이 당신의 코드, 데이터 및 환경을 독립적으로 사용하여 그림과 표를 포함한 결과를 얻을 수 있다는 것입니다. 아이러니하게도 학문 분야마다 재현성에 대한 정의가 다릅니다. Barba (2018)는 다양한 학문 분야를 조사하고 지배적인 언어 사용이 다음 정의를 의미한다고 결론 내립니다.

- 재현 가능한 연구는 “[저자가 분석을 다시 실행하고 결과를 재현하기 위해 필요한 모든 데이터와 컴퓨터 코드를 제공하는 경우]”입니다.
- 복제는 “새로운 데이터를 수집하고 (다른 방법을 사용하여) 새로운 분석을 완료하여 다른 연구와 동일한 과학적 발견에 도달하는” 연구입니다.

우리의 목적을 위해 National Academies of Sciences, Engineering, and Medicine (2019, p. 46)의 정의를 사용합니다. “재현성은 동일한 입력 데이터, 계산 단계, 방법 및 코드, 분석 조건을 사용하여 일관된 결과를 얻는 것입니다.” 구체적으로 무엇이라고 불리든, (Gelman 2016은?) 다양한 사회 과학에서 재현성 부족이 얼마나 큰 문제인지 밝힙니다. 재현할 수 없는 작업은 세상에 대한 우리의 지식에 기여하지 않습니다. 이것은 낭비적이며 잠재적으로 비윤리적일 수도 있습니다. Gelman (2016) 이후 많은 사회 과학 분야에서 많은 작업이 이루어졌고 상황이 약간 개선되었지만, 여전히 많은 작업이 남아 있습니다. 이는 생명 과학 (Heil 기타 2021), 암 연구 (Begley 와/과 Ellis 2012; Mullard 2021), 컴퓨터 과학 (Pineau 기타 2021)에서도 마찬가지입니다.

(Gelman 2016이?) 언급하는 몇 가지 예시는 전체적인 관점에서 그리 중요하지 않습니다. 그러나 동시에 우리는 큰 영향을 미치는 분야에서 유사한 접근 방식이 사용되는 것을 보았고 계속해서 보고 있습니다. 예를 들어, 많은 정부는 일부 주장이 신뢰성이 부족하다는 증거가 있음에도 불구하고 공공 정책을 구현하는 “넛지” 부서를 만들었습니다 (Sunstein 와/과 Reisch 2017; Maier 기타 2022; Szaszki 기타 2022). 정부는 점점 더 공개하지 않는 알고리즘을 사용하고 있습니다 (Chouldechova 기타 2018). 그리고 Herndon, Ash, 와/과 Pollin (2014)는 2007-2008년 금융 위기 이후 긴축 정책을 정당화하기 위해 정부가 사용한 경제학 연구가 재현 불가능하다는 것을 문서화합니다.

최소한, 그리고 몇 가지 예외를 제외하고, 우리는 코드, 데이터 세트 및 환경을 공개해야 합니다. 이것들이 없으면, 어떤 발견이 무엇을 의미하는지 알기 어렵습니다 (Miyakawa 2020). 더 평범하게는, 실수나 부주의하게 간과된 측면이 있는지 알 수 없습니다 (Merali 2010; Hillel 2017; Silver 2020). 점점 더 (buckheit1995wavelab에?) 따라, 우리는 논문을 광고로 간주하고, 관련 코드, 데이터 및 환경을 실제 작업으로 간주합니다. Apple의 공동 창립자인 스티브 잡스는 자신의 기술에 가장 능숙한 사람들이 아무도 보지 못할 작업의 측면까지도 공개적인 측면만큼 잘 마무리되고 고품질임을 보장하는 방법에 대해 이야기했습니다 (Isaacson 2011). 데이터 과학에서도 마찬가지입니다. 고품질 작업의 특징 중 하나는 README와 코드 주석이 관련 논문의 초록만큼이나 잘 다듬어져 있다는 것입니다.

워크플로우는 문화적, 사회적 맥락 내에 존재하며, 이는 재현성 필요성에 대한 추가적인 윤리적 이유를 부과합니다. 예를 들어, Y. Wang 와/과 Kosinski (2018)은 게이 남성과 이성애자 남성의 얼굴을 구별하기 위해 신경망을 훈련합니다. ((murphy2017는?) 논문, 관련 문제 및 저자의 의견에 대한 요약을 제공합니다.) 이를 위해 Y. Wang 와/과 Kosinski (2018, p. 248)은 “성인, 백인, 완전히 보이는, 사용자 프로필에 보고된 성별과 일치하는” 사람들의 사진 데이터 세트가 필요했습니다. 그들은 Amazon Mechanical Turk를 사용하여 이를 확인했습니다. Amazon Mechanical Turk는 작업자에게 특정 작업을 완료하기 위해 소액을 지불하는 온라인 플랫폼입니다. 이 작업에 대한 Mechanical Turk 작업자에게 제공된 지침은 백인 어머니와 흑인 아버지를 둔 제44대 미국 대통령 버락 오바마를 “흑인”으로 분류해야 하며, 라틴계는 인종이 아니라 민족이라는 것을 명시합니다 (Mattson 2017). 분류 작업은 객관적으로 보일 수 있지만, 아마도 무심코 특정 계층과 배경을 가진 미국인의 견해를 반영합니다.

이것은 Y. Wang 와/과 Kosinski (2018) 워크플로우의 한 부분에 대한 특정 우려 사항일 뿐입니다. Gelman, Mattson, 와/과 Simpson (2018)을 포함한 다른 사람들도 더 광범위한 우려를 제기합니다. 주요 문제는 통계 모델이 훈련된 데이터에 특화되어 있다는 것입니다. 그리고 Y. Wang 와/과 Kosinski (2018) 모델에서 발생할 수 있는 문제를 식별할 수 있는 유일한 이유는 그들이 사용한 특정 데이터 세트를 공개하지 않았음에도 불구하고 그들의 절차에 대해 공개적이었기 때문입니다. 우리의 작업이 신뢰성을 가지려면 다른 사람들도 재현할 수 있어야 합니다.

우리의 작업을 더 재현 가능하게 만들기 위해 취할 수 있는 몇 가지 단계는 다음과 같습니다.

1. 전체 워크플로우가 문서화되었는지 확인합니다. 여기에는 다음과 같은 질문에 대한 답변이 포함될 수 있습니다.
 - 원본, 편집되지 않은 데이터 세트는 어떻게 얻었으며, 다른 사람들도 지속적으로 액세스할 수 있습니까?
 - 원본, 편집되지 않은 데이터를 분석된 데이터로 변환하기 위해 어떤 특정 단계가 수행되었으며, 이를 다른 사람들에게 어떻게 제공할 수 있습니까?
 - 어떤 분석이 수행되었으며, 이를 얼마나 명확하게 공유할 수 있습니까?

- 최종 논문 또는 보고서가 어떻게 작성되었으며, 다른 사람들도 그 과정을 얼마나 따라갈 수 있습니까?
2. 처음에는 완벽한 재현성에 대해 걱정하지 않고, 대신 각 후속 프로젝트에서 개선하려고 노력합니다. 예를 들어, 다음 요구 사항은 점점 더 부담스러워지며, 첫 번째를 할 수 있을 때까지 마지막을 할 수 없다고 걱정할 필요가 없습니다.
- 전체 워크플로우를 다시 실행할 수 있습니까?
 - 다른 사람이 전체 워크플로우를 다시 실행할 수 있습니까?
 - “미래의 당신”이 전체 워크플로우를 다시 실행할 수 있습니까?
 - “미래의 다른 사람”이 전체 워크플로우를 다시 실행할 수 있습니까?
3. 최종 논문 또는 보고서에 데이터 세트 및 접근 방식의 한계에 대한 자세한 논의를 포함합니다.

이 책에서 우리가 옹호하는 워크플로우는 다음과 같습니다.

계획 → 시뮬레이션 → 획득 → 탐색 → 공유

그러나 “엄청나게 많이 생각하고, 주로 읽고 쓰고, 때로는 코딩한다”고도 생각할 수 있습니다.

이 워크플로우의 재현성을 향상시킬 수 있는 다양한 도구가 있습니다. 여기에는 Quarto, R 프로젝트, Git 및 GitHub가 포함됩니다.

3.2 Quarto

3.2.1 시작하기

Quarto는 코드와 자연어를 “문학적 프로그래밍”이라고 불리는 방식으로 통합합니다 (Knuth 1984). 이는 R 코드 청크를 포함하도록 특별히 설계된 Markdown의 변형인 R Markdown의 후속작입니다. Quarto는 Microsoft Word와 같은 “What You See Is What You Get” (WYSIWYG) 언어와 비교하여 HyperText Markup Language (HTML) 또는 LaTeX와 유사한 마크업 언어를 사용합니다. 이는 모든 측면이 일관적이라는 것을 의미합니다. 예를 들어, 모든 최상위 제목은 동일하게 보일 것입니다. 그러나 특정 측면이 어떻게 나타나기를 원하는지 지정하거나 “마크업”해야 합니다. 그리고 문서를 렌더링할 때만 어떻게 보이는지 확인할 수 있습니다. 시각 편집기 옵션도 사용할 수 있으며, 이는 사용자가 직접 마크업할 필요를 줍니다.

앞으로 Quarto를 사용하는 것이 합리적이지만, R Markdown용으로 작성된 많은 자료가 있습니다. 이러한 이유로 온라인 부록 “R Markdown”¹에서 R Markdown 동등물을 제공합니다.

i 개인의 어깨

페르난도 페레즈는 캘리포니아 대학교 버클리의 통계학 부교수이자 로렌스 버클리 국립 연구소의 데이터 과학 및 기술 부서의 교수 과학자입니다. 그는 콜로라도 대학교 볼더에서 입자 물리학 박사 학위를 받았습니다. 박사 학위 과정에서 그는 Python을 대화식으로 사용할 수 있게 하는 iPython을 만들었으며, 이는 현재 Project Jupyter의 기반이 되었고, R Markdown 및 현재 Quarto와 같은 유사한 노트북 접근 방식에 영감을 주었습니다. Somers (2018) 는 오픈 소스 노트북 접근 방식이 어떻게 선순환 피드백 루프를 생성하여 과학 컴퓨팅을 극적으로 개선하는지 설명합니다. 그리고 Romer (2018) 는 Jupyter와 같은 오픈 소스 접근 방식의 기능을 과학적 합의와 발전을 가능하게 하는 기능과 일치시킵니다. 2017년에 페레즈는 ACM (Association for Computing Machinery) 소프트웨어 시스템상을 수상했습니다.

문학적 프로그래밍의 한 가지 장점은 코드가 실행되고 문서의 일부를 형성하는 “라이브” 문서를 얻는다는 것입니다. Quarto의 또 다른 장점은 유사한 코드가 HTML 및 PDF를 포함한 다양한 문서로 컴파일될 수 있다는 것입니다. Quarto는 또한 제목, 저자 및 날짜를 포함하는 기본 옵션을 제공합니다. 한 가지 단점은 코드가 실행되어야 하므로 문서가 컴파일되는 데 시간이 걸릴 수 있다는 것입니다.

¹<https://tellingstorieswithdata.com/22-rmarkdown.html>

Quarto는 여기²에서 다운로드해야 합니다. (Posit Cloud를 사용하는 경우 이미 설치되어 있으므로 이 단계를 건너뛰십시오.) 그런 다음 RStudio 내에서 새 Quarto 문서를 만들 수 있습니다. “파일” → “새 파일” → “Quarto 문서...”.

새 Quarto 문서를 열고 “소스” 보기 를 선택하면, 세 개의 대시 쌍 안에 포함된 기본 상단 내용과 몇 가지 마크다운 필수 명령 및 R 청크를 보여주는 텍스트 예제를 볼 수 있으며, 각각은 다음 섹션에서 더 자세히 설명됩니다.

3.2.2 상단 내용

상단 내용은 제목, 저자, 날짜와 같은 측면을 정의하는 것으로 구성됩니다. Quarto 문서 상단의 세 개의 대시 안에 포함됩니다. 예를 들어, 다음은 제목, 문서가 렌더링된 날짜로 자동 업데이트되는 날짜, 그리고 저자를 지정합니다.

```
---
title: "ㅁ ㅁ"
author: "ㅁ ㅁ"
date: format(Sys.time(), "%d %B %Y")
format: html
---
```

초록은 논문의 짧은 요약이며, 이를 상단 내용에 추가할 수 있습니다.

```
---
title: "ㅁ ㅁ"
author: "ㅁ ㅁ"
date: format(Sys.time(), "%d %B %Y")
abstract: "ㅁ ㅁ ㅁ ㅁ."
format: html
---
```

기본적으로 Quarto는 HTML 문서를 생성하지만, 출력 형식을 변경하여 PDF를 생성할 수 있습니다. 이는 백그라운드에서 LaTeX를 사용하며 지원 패키지 설치가 필요합니다. 이를 위해 `tinytex`를 설치합니다. 그러나 백그라운드에서 사용되므로 로드할 필요는 없습니다.

```
---
title: "ㅁ ㅁ"
author: "ㅁ ㅁ"
date: format(Sys.time(), "%d %B %Y")
abstract: "ㅁ ㅁ ㅁ ㅁ."
format: pdf
---
```

3.2.3 참조

BibTeX 파일을 상단 내용에 지정한 다음 필요에 따라 텍스트 내에서 호출하여 참조를 포함할 수 있습니다.

```
---
title: "ㅁ ㅁ"
author: "ㅁ ㅁ"
date: format(Sys.time(), "%d %B %Y")
format: pdf
abstract: "ㅁ ㅁ ㅁ ㅁ."
bibliography: bibliography.bib
---
```

“`bibliography.bib`”라는 별도의 파일을 만들고 Quarto 파일 옆에 저장해야 합니다. BibTeX 파일에는 참조 할 항목에 대한 항목이 필요합니다. 예를 들어, R에 대한 인용은 `citation()`으로 얻을 수 있으며 이를

²<https://quarto.org/docs/get-started/>

“bibliography.bib” 파일에 추가할 수 있습니다. 패키지에 대한 인용은 패키지 이름을 포함하여 찾을 수 있습니다. 예를 들어 `citation("tidyverse")`를 사용하여 출력을 “.bib” 파일에 다시 추가합니다. 책이나 기사에 대한 인용을 얻으려면 Google Scholar³ 또는 doi2bib⁴를 사용하는 것이 도움이 될 수 있습니다.

텍스트에서 이 항목을 참조하는 데 사용할 고유 키를 만들어야 합니다. 이는 고유하다면 무엇이든 될 수 있지만, “citeR”과 같이 의미 있는 키는 기억하기 더 쉬울 수 있습니다.

```
@Manual{citeR,
  title = {R: 純粹統計計算},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {維也納, 奧},
  year = {2021},
  url = {https://www.R-project.org/},
}

@book{tellingstories,
  title = {統計學 故事},
  author = {尹 智浩},
  year = {2023},
  publisher = {統計出版社/CRC},
  url = {https://tellingstorieswithdata.com}
}
```

Quarto 문서에서 R을 인용하려면 `@citeR`을 포함하면 연도 주위에 괄호가 붙습니다: R Core Team (2024), 또는 `[@citeR]`을 포함하면 전체 주위에 괄호가 붙습니다: (R Core Team 2024).

논문 끝의 참조 목록은 BibTeX 파일을 호출하고 논문에 참조를 포함하여 자동으로 생성됩니다. Quarto 문서 끝에 “# 참조”라는 제목을 포함하면 실제 인용이 그 뒤에 포함됩니다. Quarto 파일이 렌더링되면 Quarto는 내용에서 이를 보고, 필요한 참조 세부 정보를 얻기 위해 BibTeX 파일로 이동하여 참조 목록을 작성한 다음 렌더링된 문서 끝에 추가합니다.

BibTeX는 항목의 대문자 사용을 조정하려고 시도합니다. 이는 도움이 될 수 있지만, 때로는 특정 대문자 사용을 고집하는 것이 더 좋습니다. BibTeX가 특정 대문자 사용을 강제하도록 하려면 항목 주위에 단일 중괄호 대신 이중 중괄호를 사용하십시오. 예를 들어, 위 예시에서 `{R Core Team}`은 정확히 그 대문자 사용으로 인쇄되지만, `{Telling Stories with Data}`는 BibTeX의 변덕에 따라 달라집니다. 특정 대문자 사용을 고집하는 것은 특정 대문자 사용이 있을 수 있는 R 패키지를 인용할 때와 조직을 저자로 인용할 때 중요합니다. 예를 들어, `usesthis`를 인용할 때 `title = {{usesthis: Automate Package and Project Setup}}`, 을 사용해야 하며, `title = {usesthis: Automate Package and Project Setup}`, 을 사용해서는 안 됩니다. 그리고 예를 들어, 데이터가 토큰화 시에서 제공되었다면, 해당 데이터 세트를 인용할 저자를 지정할 때 `author = {{City of Toronto}}`, 를 사용해야 하며, `author = {City of Toronto}`, 를 사용해서는 안 됩니다. 후자는 잘못된 참조 목록 항목인 “Toronto, City of”를 초래하는 반면, 전자는 올바른 참조 목록 항목인 “City of Toronto”를 초래합니다.

3.2.4 필수 명령

Quarto는 Markdown의 변형을 기본 구문으로 사용합니다. 필수 Markdown 명령에는 강조, 헤더, 목록, 링크 및 이미지가 포함됩니다. RStudio에는 “도움말” → “Markdown 빠른 참조”에 이러한 내용이 포함되어 있습니다. 시각 편집기를 사용할지 소스 편집기를 사용할지는 당신의 선택입니다. 그러나 어느 쪽이든, 이러한 필수 사항을 이해하는 것이 좋습니다. 왜냐하면 시각 편집기를 항상 사용할 수 있는 것은 아니기 때문입니다 (예를 들어 GitHub에서 Quarto 문서를 빠르게 볼 때). 경험이 쌓이면 Sublime Text와 같은 텍스트 편집기 또는 VS Code와 같은 대체 통합 개발 환경을 사용하는 것이 유용할 수 있습니다.

- 강조: `*纯粹统计计算*`, `**纯粹统计计算**`
- 헤더 (각각 한 줄에 앞뒤로 빈 줄이 있어야 합니다):

```
# 空 行 空 行 空
```

³<https://scholar.google.com>

⁴<https://www.doi2bib.org>

```
## ❶ ❷ ❸ ❹
```

```
#### ❺ ❻ ❽ ❾
```

- 하위 목록이 있는 순서 없는 목록:

```
* ❻ ❵ ❶
* ❻ ❵ ❷
  + ❻ ❷a
  + ❻ ❷b
```

- 하위 목록이 있는 순서 있는 목록:

```
1. ❻ ❵ ❶
2. ❻ ❵ ❷
3. ❻ ❵ ❸
  + ❻ ❷a
  + ❻ ❷b
```

- URL을 추가할 수 있습니다: [❻ ❽](<https://www.tellingstorieswithdata.com>)은 이 책⁵으로 나타납니다.
- 단락은 빈 줄을 남겨서 생성됩니다.

❻ ❷ ❷ ❷, ❷ ❷ ❷ ❷ ❷ ❷ ❷.

❷ ❷ ❷ ❷ ❷ ❷, ❷ ❷ ❷ ❷ ❷ ❷.

일부 측면을 추가한 후 실제 문서를 보고 싶을 수 있습니다. 문서를 빌드하려면 “렌더링”을 클릭하십시오.

3.2.5 R 청크

Quarto 문서 내의 코드 청크에 R 및 기타 여러 언어에 대한 코드를 포함할 수 있습니다. 문서를 렌더링하면 코드가 실행되어 문서에 포함됩니다.

R 청크를 만들려면 세 개의 백틱으로 시작한 다음 중괄호 안에 Quarto에 이것이 R 청크임을 알려줍니다. 이 청크 안의 모든 것은 R 코드로 간주되어 실행됩니다. 우리는 (citeaer의?) 데이터를 사용합니다. 이들은 Applied Econometrics with R 책에 동반되는 R 패키지 AER를 제공합니다. tidyverse를 로드하고 AER를 설치 및 로드한 다음 설문 응답자가 지난 2주 동안 의사를 방문한 횟수에 대한 그래프를 만들 수 있습니다.

```
#| message: false
#| warning: false

library(AER)
library(tidyverse)

data("DoctorVisits", package = "AER")

DoctorVisits |>
  ggplot(aes(x = illness)) +
  geom_histogram(stat = "count")
```

해당 코드의 출력은 ?@fig-doctervisits입니다.

```
#| label: fig-doctervisits
#| echo: false
#| eval: true
#| warning: false
#| message: false
#| fig-cap: "1977-1978년 미국 의사방문 횟수 분포도"
```

⁵<https://www.tellingstorieswithdata.com>

```
data("DoctorVisits", package = "AER")
```

```
DoctorVisits |>
  ggplot(aes(x = illness)) +
  geom_histogram(stat = "count")
```

청크에서 사용할 수 있는 다양한 평가 옵션이 있습니다. 각 줄에 청크별 주석 구분 기호 “#|”로 시작하여 옵션을 포함합니다. 유용한 옵션은 다음과 같습니다.

- echo: 코드 자체가 문서에 포함될지 여부를 제어합니다. 예를 들어, #| echo: false는 코드가 실행되고 출력이 표시되지만, 코드 자체는 문서에 포함되지 않음을 의미합니다.
- include: 코드의 출력이 문서에 포함될지 여부를 제어합니다. 예를 들어, #| include: false는 코드를 실행하지만, 출력이 생성되지 않으며, 코드 자체도 문서에 포함되지 않음을 의미합니다.
- eval: 코드가 문서에 포함되어야 하는지 여부를 제어합니다. 예를 들어, #| eval: false는 코드가 실행되지 않으므로 포함할 출력이 없지만, 코드 자체는 문서에 포함됨을 의미합니다.
- warning: 경고가 문서에 포함되어야 하는지 여부를 제어합니다. 예를 들어, #| warning: false는 경고가 포함되지 않음을 의미합니다.
- message: 메시지가 문서에 포함되어야 하는지 여부를 제어합니다. 예를 들어, #| message: false는 메시지가 문서에 포함되지 않음을 의미합니다.

예를 들어, 출력을 포함하되 코드는 포함하지 않고 경고를 억제할 수 있습니다.

```
#| echo: false
#| warning: false
```

```
library(AER)
library(tidyverse)
```

```
data("DoctorVisits", package = "AER")
```

```
DoctorVisits |>
  ggplot(aes(x = illness)) +
  geom_histogram(stat = "count")
```

R 청크의 양쪽에 빈 줄을 두십시오. 그렇지 않으면 제대로 실행되지 않을 수 있습니다. 그리고 논리 값에는 소문자를 사용하십시오. 즉, “false”가 아닌 “FALSE”입니다.

```
#| echo: false
#| warning: false
```

```
library(AER)
library(tidyverse)
```

```
data("DoctorVisits", package = "AER")
```

```
DoctorVisits |>
  ggplot(aes(x = illness)) +
  geom_histogram(stat = "count")
```

Quarto 문서 자체는 필요한 모든 데이터 세트를 로드해야 합니다. 환경에 있는 것만으로는 충분하지 않습니다. 이는 Quarto 문서가 렌더링될 때 문서의 코드를 평가하기 때문이며, 반드시 환경을 평가하는 것은 아닙니다.

코드를 작성할 때 여러 줄에 걸쳐 동일한 변경을 하거나 특정 항목의 모든 인스턴스를 변경하고 싶을 수 있습니다. 이를 여러 커서로 달성합니다. 여러 연속된 줄에 걸쳐 커서를 사용하려면 Mac에서는 “option”을, PC에서는 “Alt”를 누른 상태에서 관련 줄 위로 커서를 드래그하십시오. 특정 항목의 모든 인스턴스를 선택하려면 변수 이름과 같은 하나의 인스턴스를 강조 표시한 다음 찾기/바꾸기 (Mac에서는 Command + F, PC에서는 CTRL + F)를 사용하여 “모두”를 선택하십시오. 그러면 다른 모든 인스턴스에 커서가 활성화됩니다.

3.2.6 방정식

LaTeX를 사용하여 방정식을 포함할 수 있습니다. LaTeX는 TeX 프로그래밍 언어를 기반으로 합니다. 두 개의 달러 기호를 시작 및 끝 태그로 사용하여 LaTeX에서 수학 모드를 호출합니다. 그러면 그 안에 있는 모든 것이 LaTeX 마크업으로 평가됩니다. 예를 들어, 다음을 사용하여 복리 공식을 생성할 수 있습니다.

```
$$
A = P \left(1 + \frac{r}{n}\right)^{nt}
$$
```

$$A = P \left(1 + \frac{r}{n}\right)^{nt}$$

LaTeX는 포괄적인 마크업 언어이지만, 우리는 주로 관심 있는 모델을 지정하는 데 사용할 것입니다. 장 ??에서 시작하여 우리가 활용할 중요한 측면을 포함하는 몇 가지 예시를 여기에 포함합니다.

```
$$
y_i | \mu_i, \sigma \sim \text{Normal}(\mu_i, \sigma)
$$
```

$$y_i | \mu_i, \sigma \sim \text{Normal}(\mu_i, \sigma)$$

밑줄은 아래 첨자를 얻는 데 사용됩니다: y_i 의 경우 y_{i} . 그리고 중괄호로 묶어 하나 이상의 항목의 아래 첨자를 얻을 수 있습니다: $y_{i,c}$ 의 경우 $y_{\{i,c\}}$. 이 경우 줄 안에 수학 모드를 원했으므로 시작 및 끝 태그로 하나의 달러 기호만으로 묶습니다.

그리스 문자는 일반적으로 백슬래시가 앞에 붙습니다. 일반적인 그리스 문자에는 다음이 포함됩니다: α 의 경우 α , β 의 경우 β , δ 의 경우 δ , ϵ 의 경우 ϵ , γ 의 경우 γ , λ 의 경우 λ , μ 의 경우 μ , ϕ 의 경우 ϕ , π 의 경우 π , Π 의 경우 Π , ρ 의 경우 ρ , σ 의 경우 σ , Σ 의 경우 Σ , τ 의 경우 τ , θ 의 경우 θ .

LaTeX 수학 모드는 문자를 변수로 가정하여 이탤릭체로 만듭니다. 그러나 때로는 “Normal”과 같이 변수가 아니기 때문에 단어가 일반 글꼴로 나타나기를 원할 때가 있습니다. 이 경우 $\text{mbox}()$ 로 묶습니다. 예를 들어, Normal의 경우 $\text{mbox}\{\text{Normal}\}$.

`\begin{aligned}`와 `\end{aligned}`를 사용하여 여러 줄에 걸쳐 방정식을 정렬합니다. 그러면 정렬할 항목은 앤페샌드로 표시됩니다. 다음은 장 ??에서 추정할 모델입니다.

```
$$
\begin{aligned}
y_i | \pi_i & \sim \text{Bern}(\pi_i) \\
\text{mbox}\{\text{logit}\}(\pi_i) & = \beta_0 + \alpha_{\text{gender}}[i]^{\text{mbox}\{\text{gender}\}} + \alpha_{\text{age}}[i]^{\text{mbox}\{\text{age}\}} + \\
& \quad \alpha_{\text{state}}[i]^{\text{mbox}\{\text{state}\}} + \alpha_{\text{edu}}[i]^{\text{mbox}\{\text{edu}\}} \\
\beta_0 & \sim \text{Normal}(0, 2.5) \\
\alpha_{\text{gender}} & \sim \text{Normal}(0, 2.5) \\
\alpha_{\text{age}} & \sim \text{Normal}(0, 2.5) \\
\alpha_{\text{state}} & \sim \text{Normal}(0, 2.5) \\
\alpha_{\text{edu}} & \sim \text{Normal}(0, 2.5) \\
\sigma_{\text{gender}} & \sim \text{Exponential}(1) \\
\sigma_{\text{state}} & \sim \text{Exponential}(1) \\
\sigma_{\text{edu}} & \sim \text{Exponential}(1)
\end{aligned}
$$
```

\$\$

$$\begin{aligned}
 y_i | \pi_i &\sim \text{Bern}(\pi_i) \\
 \text{logit}(\pi_i) &= \beta_0 + \alpha_{g[i]}^{\text{gender}} + \alpha_{a[i]}^{\text{age}} + \alpha_{s[i]}^{\text{state}} + \alpha_{e[i]}^{\text{edu}} \\
 \beta_0 &\sim \text{Normal}(0, 2.5) \\
 \alpha_g^{\text{gender}} &\sim \text{Normal}(0, 2.5) \text{ for } g = 1, 2 \\
 \alpha_a^{\text{age}} &\sim \text{Normal}(0, \sigma_{\text{age}}^2) \text{ for } a = 1, 2, \dots, A \\
 \alpha_s^{\text{state}} &\sim \text{Normal}(0, \sigma_{\text{state}}^2) \text{ for } s = 1, 2, \dots, S \\
 \alpha_e^{\text{edu}} &\sim \text{Normal}(0, \sigma_{\text{edu}}^2) \text{ for } e = 1, 2, \dots, E \\
 \sigma_{\text{gender}} &\sim \text{Exponential}(1) \\
 \sigma_{\text{state}} &\sim \text{Exponential}(1) \\
 \sigma_{\text{edu}} &\sim \text{Exponential}(1)
 \end{aligned}$$

마지막으로, 특정 함수는 LaTeX에 내장되어 있습니다. 예를 들어, `\log`를 사용하여 “log”를 적절하게 조판할 수 있습니다.

3.2.7 상호 참조

그림, 표, 방정식을 상호 참조하는 것이 유용할 수 있습니다. 이렇게 하면 텍스트에서 참조하기가 더 쉬워집니다. 그림의 경우 그림을 생성하거나 포함하는 R 청크의 이름을 참조합니다. 예를 들어, 다음 코드를 고려하십시오.

```

#| echo: fenced
#| label: fig-theuniquefilename
#| fig-cap: ❷ ❸ ❹ ❺ ❻ ❽
#| warning: false

data("DoctorVisits", package = "AER")

```

```

DoctorVisits |>
  ggplot(aes(x = illness)) +
  geom_histogram(stat = "count")

```

그러면 `(@fig-theuniquefilename)`은 `(?@fig-theuniquefilename)`을 생성합니다. R 청크의 레이블이 `fig-theuniquefilename`이기 때문입니다. Quarto가 이것이 그림임을 알 수 있도록 청크 이름 시작 부분에 “fig”를 추가해야 합니다. 그런 다음 캡션을 지정하는 “fig-cap:”을 R 청크에 포함합니다.

Quarto 문서 내의 R 청크에 `#| layout-ncol: 2`를 추가하여 두 개의 그래프를 나란히 표시할 수 있습니다 (`(?@fig-doctorgraphssidebyside)`). 여기에서 `?@fig-doctorgraphssidebyside-1`은 최소 테마를 사용하고, `?@fig-doctorgraphssidebyside-2`는 고전 테마를 사용합니다. 이 둘은 R 청크의 동일한 레이블 `#| label: fig-doctorgraphssidebyside`를 상호 참조하며, R 청크에 추가 옵션 `#| fig-subcap: ["❷ ❸ ❹ ❺ ❻ ❽", "❻ ❺ ❷ ❹ ❸ ❽"]`이 추가되어 하위 캡션을 제공합니다. 텍스트 내에 문자를 추가하는 것은 텍스트에서 사용될 때 레이블 끝에 “-1”과 “-2”를 추가하여 수행됩니다: `([@fig-doctorgraphssidebyside]), [@fig-doctorgraphssidebyside-1], [@fig-doctorgraphssidebyside-2]`는 각각 `(?@fig-doctorgraphssidebyside)`, `?@fig-doctorgraphssidebyside-1`, `?@fig-doctorgraphssidebyside-2`를 생성합니다.

```

#| eval: true
#| warning: false
#| label: fig-doctorgraphssidebyside
#| echo: fenced

```

```
#| fig-cap: "도CTOR VISITS"
#| fig-subcap: ["도CTOR VISITS", "도CTOR VISITS"]
#| layout-ncol: 2

DoctorVisits |>
  ggplot(aes(x = illness)) +
  geom_histogram(stat = "count") +
  theme_minimal()
```

```
DoctorVisits |>
  ggplot(aes(x = visits)) +
  geom_histogram(stat = "count") +
  theme_classic()
```

표를 상호 참조하는 데 유사한 접근 방식을 취할 수 있습니다. 예를 들어, `(@tbl-docvisitable)`은 (`?@tbl-docvisitable`)을 생성합니다. 이 경우 Quarto가 테이블임을 알 수 있도록 레이블 시작 부분에 “tbl”을 지정합니다. 그리고 “tbl-cap:”으로 테이블의 캡션을 지정합니다.

```
#| echo: fenced
#| label: tbl-docvisitable
#| tbl-cap: "도CTOR VISITS"

DoctorVisits |>
  count(visits) |>
  tt() |>
  style_tt(j = 2, align = "r") |>
  setNames(c("도CTOR VISITS", "도CTOR VISITS"))
```

마지막으로, 방정식도 상호 참조할 수 있습니다. 이를 위해 `{#eq-macroidentity}`와 같은 태그를 추가한 다음 참조합니다.

```
$$
Y = C + I + G + (X - M)
$$ {#eq-gdpidentity}
```

예를 들어, `@eq-gdpidentity`를 사용하여 방정식 ??를 생성합니다.

$$Y = C + I + G + (X - M) \quad (3.1)$$

상호 참조를 사용할 때 레이블은 비교적 간단해야 합니다. 일반적으로 이름을 간단하지만 고유하게 유지하고, 구두점을 피하고, 문자 및 하이픈을 사용하십시오. 밑줄은 오류를 유발할 수 있으므로 사용하지 마십시오.

3.3 R 프로젝트 및 파일 구조

프로젝트는 소프트웨어 개발에서 널리 사용되며, 특정 프로젝트와 관련된 모든 파일(데이터, 분석, 보고서 등)을 함께 유지하고 서로 관련시키기 위해 존재합니다. (소프트웨어 개발 의미의 “프로젝트”는 프로젝트 관리 의미의 “프로젝트”와 다릅니다.) R 프로젝트는 RStudio에서 만들 수 있습니다. “파일” → “새 프로젝트”를 클릭한 다음 “빈 프로젝트”를 선택하고 R 프로젝트 이름을 지정하고 저장할 위치를 결정합니다. 예를 들어, 모성 사망률에 초점을 맞춘 R 프로젝트는 “maternalmortality”라고 불릴 수 있습니다. R 프로젝트를 사용하면 “[다른 컴퓨터나 사용자 간에 그리고 시간이 지남에 따라 신뢰할 수 있고 정중한 동작]을 가능하게 합니다” (Bryan 와/과 Hester 2020). 이는 폴더의 컨텍스트를 더 넓은 존재에서 제거하기 때문입니다. 파일은 컴퓨터의 기반이 아니라 R 프로젝트의 기반과 관련하여 존재합니다.

프로젝트가 생성되면 해당 폴더에 “.RProj” 확장자를 가진 새 파일이 나타납니다. R 프로젝트, Quarto 문서 및 적절한 파일 구조를 가진 폴더의 예는 여기⁶에서 확인할 수 있습니다. “코드” → “ZIP 다운로드”를 통해 다운로드할 수 있습니다.

R 프로젝트를 사용하는 주된 장점은 파일 내에서 자체 포함된 방식으로 파일을 참조할 수 있다는 것입니다. 즉, 다른 사람들이 우리의 작업을 재현하려고 할 때, 모든 파일 참조와 구조를 변경할 필요가 없습니다. 모든 것이 “.Rproj” 파일과 관련하여 참조되기 때문입니다. 예를 들어, ~/Documents/projects/book/data/에서 CSV를 읽는 대신 book/data/에서 읽을 수 있습니다. 다른 사람이 projects 폴더를 가지고 있지 않을 수 있으므로 전자는 작동하지 않지만 후자는 작동합니다.

프로젝트 사용은 신뢰할 수 있는 작업에 기대되는 최소한의 재현성 수준을 충족하는 데 필요합니다. setwd()와 같은 함수 및 컴퓨터별 파일 경로를 사용하는 것은 작업을 특정 컴퓨터에 부적절하게 바인딩합니다.

폴더를 설정하는 다양한 방법이 있습니다. Wilson 기타 (2017) 의 변형 중 하나는 위에 링크된 예제 파일 구조에 표시된 것처럼 시작할 때 유용한 경우가 많습니다.

```
example_project/
├── .gitignore
└── LICENSE.md
├── README.md
└── example_project.Rproj
├── data
│   ├── 00-simulated_data
│   │   └── simulated_data.csv
│   ├── 01-raw_data
│   │   └── raw_data.csv
│   ├── 02-analysis_data
│   │   └── analysis_data.csv
│   └── ...
├── model
│   └── first_model.rds
└── other
    ├── datasheet
    │   └── ...
    ├── literature
    │   └── ...
    ├── llm_usage
    │   └── ...
    ├── sketches
    │   └── ...
└── paper
    ├── paper.pdf
    ├── paper.qmd
    └── references.bib
    └── ...
└── scripts
    ├── 00-simulate_data.R
    ├── 01-test_simulated_data.R
    ├── 02-download_data.R
    ├── 03-clean_data.R
    ├── 04-test_analysis_data.R
    ├── 05-eda.R
    ├── 06-model_data.R
    └── 07-replication.R
```

⁶https://github.com/RohanAlexander/starter_folder



여기에는 시뮬레이션된 데이터, 덮어쓰지 않아야 하는 편집되지 않은 데이터 (Wilson 기타 2017), 그리고 함께 정리된 분석 데이터가 포함된 `data` 폴더가 있습니다. `model` 폴더에는 저장된 모델 추정치가 포함되어 있습니다. `other` 폴더에는 데이터시트, 문헌, LLM 사용 및 스케치와 같은 측면이 포함되어 있으며, 이는 상황에 따라 유용합니다. `paper` 폴더에는 Quarto 문서와 BibTeX 파일이 포함되어 있습니다. 마지막으로 `scripts`에는 데이터를 시뮬레이션, 다운로드, 테스트 및 분석하는 코드가 포함되어 있습니다.

유용한 다른 측면으로는 프로젝트에 대한 개요 세부 정보를 지정하는 `README.md`와 라이선스가 있습니다. `README`에 무엇을 넣을지에 대한 예시는 여기⁷에서 확인할 수 있습니다. 이 프로젝트 골격의 또 다른 유용한 변형은 Mineault 와/과 The Good Research Code Handbook Community (2021)에서 제공합니다.

3.4 버전 관리

이 책에서는 Git과 GitHub의 조합을 통해 버전 관리를 구현합니다. 여기에는 다음과 같은 다양한 이유가 있습니다.

1. 코드와 데이터를 더 쉽게 공유하여 작업의 재현성을 향상시킵니다.
2. 작업을 더 쉽게 공유합니다.
3. 체계적인 접근 방식을 장려하여 워크플로우를 개선합니다.
4. 팀 작업이 더 쉬워집니다.

Git은 흥미로운 역사를 가진 버전 관리 시스템입니다 (Brown 2018). 버전 관리를 시작하는 일반적인 방법은 “first_go.R”, “first_go-fixed.R”, “first_go-fixed-with-mons-edits.R”과 같이 하나의 파일에 여러 복사본을 두는 것입니다. 그러나 이것은 곧 번거로워집니다. 종종 곧 날짜를 사용하게 됩니다. 예를 들어, “2022-01-01-analysis.R”, “2022-01-02-analysis.R”, “2022-01-03-analysis.R” 등입니다. 이것은 기록을 유지하지만, 어떤 변경이 언제 이루어졌는지 기억하기 어렵기 때문에 다시 돌아가야 할 때 검색하기 어려울 수 있습니다. 어쨌든, 정기적으로 작업하는 프로젝트에는 빠르게 다루기 힘들어집니다.

대신 Git을 사용하여 파일의 한 버전을 가질 수 있습니다. Git은 해당 파일에 대한 변경 기록과 특정 시점의 해당 파일 스냅샷을 유지합니다. Git이 스냅샷을 찍는 시점을 결정합니다. 또한 이 스냅샷과 마지막 스냅샷 사이에 무엇이 변경되었는지 설명하는 메시지를 포함합니다. 이런 식으로 파일의 버전은 항상 하나뿐이며, 기록을 더 쉽게 검색할 수 있습니다.

한 가지 복잡한 점은 Git이 소프트웨어 개발자 팀을 위해 설계되었다는 것입니다. 따라서 작동하더라도 비개발자에게는 다소 다루기 힘들 수 있습니다. 그럼에도 불구하고 Git은 데이터 과학에 유용하게 적용되었으며, 유일한 협력자가 미래의 자신일지라도 마찬가지입니다 (Bryan 2018a).

GitHub, GitLab 및 기타 여러 회사는 Git을 기반으로 하는 사용하기 쉬운 서비스를 제공합니다. 장단점이 있지만, 여기서는 GitHub를 소개합니다. 왜냐하면 GitHub가 지배적인 플랫폼이기 때문입니다 (Eghbal 2020, p. 21). Git과 GitHub는 Posit Cloud에 내장되어 있어 로컬 설치에 문제가 있는 경우 좋은 옵션을 제공합니다. Git의 초기 어려운 측면 중 하나는 용어입니다. 폴더는 “repo”라고 불립니다. 스냅샷을 만드는 것은 “commit”이라고 불립니다. 결국 익숙해지겠지만, 처음에는 혼란스러운 것이 정상입니다. Bryan (2020)은 Git 및 GitHub 설정 및 사용에 특히 유용합니다.

3.4.1 Git

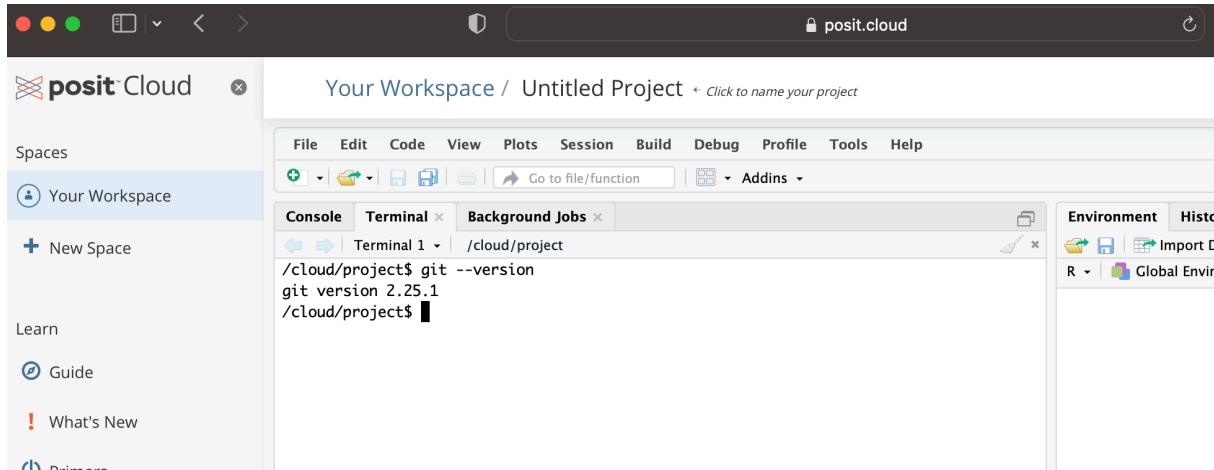
먼저 Git이 설치되어 있는지 확인해야 합니다. RStudio를 열고 터미널로 이동하여 다음을 입력한 다음 Enter/Return을 누르십시오.

⁷https://social-science-data-editors.github.io/template_README/

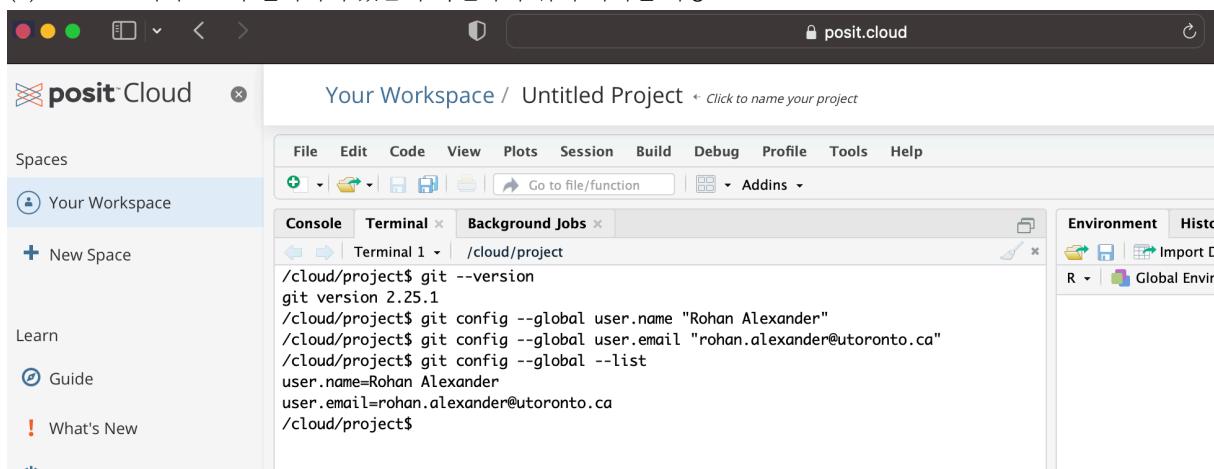
```
#| eval: false
#| echo: true

git --version
```

버전 번호가 나오면 완료된 것입니다 (그림 ??).



(a) RStudio에서 Git이 설치되어 있는지 확인하기 위해 터미널 사용



(b) RStudio에서 Git에 사용자 이름과 이메일 주소 추가

그림 3.1: Git 설정에 관련된 단계 개요

Git은 Posit Cloud에 사전 설치되어 있으며, Mac에는 사전 설치되어 있어야 하고, Windows에는 사전 설치되어 있을 수 있습니다. 응답으로 버전 번호가 나오지 않으면 설치해야 합니다. 이를 위해 Bryan (2020, 5장)의 운영 체제별 지침을 따르십시오.

Git이 설치되면 사용자 이름과 이메일을 알려야 합니다. Git은 스냅샷을 찍을 때마다, 또는 Git의 언어로 커밋을 할 때마다 이 정보를 추가하기 때문에 이 작업을 수행해야 합니다.

다시 터미널에서 다음을 입력하고 세부 정보를 자신의 것으로 바꾼 다음 각 줄 뒤에 “enter/return”을 누르십시오.

```
#| eval: false
#| echo: true
```

```
git config --global user.name "로한 알렉산더"
git config --global user.email "rohan.alexander@utoronto.ca"
git config --global --list
```

이 설정이 제대로 완료되면 “user.name” 및 “user.email”에 입력한 값이 마지막 줄 뒤에 반환됩니다 (그림 ??).

이러한 세부 정보(사용자 이름 및 이메일 주소)는 공개됩니다. 필요한 경우 이메일 주소를 숨기는 다양한 방법이 있으며, GitHub는 이에 대한 지침을 제공합니다. Bryan (2020, 7장)은 이 단계에 대한 더 자세한 지침과 문제 해결 가이드를 제공합니다.

3.4.2 GitHub

이제 Git이 설정되었으므로 GitHub를 설정해야 합니다. 장 ??에서 GitHub 계정을 만들었으며, 여기에서 다시 사용합니다. github.com에 로그인한 후 먼저 새 폴더를 만들어야 합니다. Git에서는 이를 “repo”라고 부릅니다. 오른쪽 상단에서 “+”를 찾은 다음 “새 저장소”를 선택합니다 (그림 ??).

이 시점에서 저장소에 적절한 이름을 추가할 수 있습니다. 지금은 “공개”로 두십시오. 나중에 언제든지 삭제할 수 있기 때문입니다. 그리고 “README로 이 저장소 초기화” 상자를 선택하십시오. “.gitignore 추가”를 R로 변경하십시오. 그런 다음 “저장소 생성”을 클릭하십시오.

그러면 상당히 비어 있는 화면으로 이동하지만, 필요한 세부 정보(URL)는 녹색 “복제 또는 다운로드” 버튼에 있으며, 클립보드를 클릭하여 복사할 수 있습니다 (그림 ??).

이제 RStudio로 돌아와 Posit Cloud에서 “Git 저장소에서 새 프로젝트”를 사용하여 새 프로젝트를 만듭니다. 방금 복사한 URL을 요청할 것입니다 (그림 ??). 로컬 컴퓨터를 사용하는 경우 이 단계는 메뉴를 통해 수행됩니다: “파일” → “새 프로젝트...” → “버전 제어” → “Git”을 선택한 다음 URL을 붙여넣고 폴더에 의미 있는 이름을 지정하고 “새 세션에서 열기”를 선택한 다음 “프로젝트 생성”을 클릭합니다.

이 시점에서 사용할 수 있는 새 폴더가 생성되었습니다. GitHub로 다시 푸시할 수 있어야 하며, 이를 위해 RStudio 작업 공간을 GitHub 계정과 연결하려면 개인 액세스 토큰(PAT)을 사용해야 합니다. 이를 위해 `usethis`와 `gitcreds`를 사용합니다. 이들은 각각 반복 작업을 자동화하는 패키지와 GitHub로 인증하는 패키지입니다. PAT를 만들려면 브라우저에서 GitHub에 로그인한 상태에서 `usethis`를 설치하고 로드한 후 R 세션에서 `create_github_token()`을 실행합니다. GitHub는 다양한 옵션이 채워진 브라우저에서 열립니다 (그림 ??). “Note”를 “PAT for RStudio”와 같이 정보가 담긴 이름으로 바꾸는 것이 유용할 수 있습니다. 그런 다음 “토큰 생성”을 클릭합니다.

이 토큰을 복사할 기회는 한 번뿐이며, 실수를 하면 새 토큰을 생성해야 합니다. PAT를 R 스크립트나 Quarto 문서에 포함하지 마십시오. 대신 `gitcreds`를 설치하고 로드한 후 `gitcreds_set()`을 실행하면 콘솔에서 PAT를 추가하라는 메시지가 나타납니다.

활발하게 작업 중인 프로젝트에 GitHub를 사용하려면 다음 절차를 따릅니다.

1. 가장 먼저 해야 할 일은 거의 항상 “pull”을 사용하여 변경 사항을 가져오는 것입니다. 이렇게 하려면 RStudio의 Git 창을 열고 파란색 아래쪽 화살표를 클릭합니다. 이렇게 하면 GitHub에 있는 폴더의 변경 사항이 우리 자신의 폴더 버전으로 가져와집니다.
2. 그런 다음 폴더의 복사본에 변경 사항을 적용할 수 있습니다. 예를 들어, README를 업데이트 한 다음 평소처럼 저장할 수 있습니다.
3. 이 작업이 완료되면 추가, 커밋 및 푸시해야 합니다. RStudio의 Git 창에서 추가할 파일을 선택합니다. 이렇게 하면 파일이 스테이징 영역에 추가됩니다. 그런 다음 “커밋”을 클릭합니다 (그림 ??). 새 창이 열립니다. 변경 사항에 대한 정보가 담긴 메시지를 추가한 다음 해당 새 창에서 “커밋”을 클릭합니다 (그림 ??). 마지막으로 “푸시”를 클릭하여 변경 사항을 GitHub로 보냅니다.

Git과 GitHub에는 몇 가지 일반적인 문제점이 있습니다. 버전 관리에 익숙하지 않은 경우 정기적으로 커밋하고 푸시하는 것이 좋습니다. 이렇게 하면 필요한 경우 다시 돌아갈 수 있는 스냅샷 수가 늘어납니다. 모든 커밋에는 정보가 담긴 커밋 메시지가 있어야 합니다. 버전 관리에 익숙하지 않은 경우 좋은 커밋 메시지

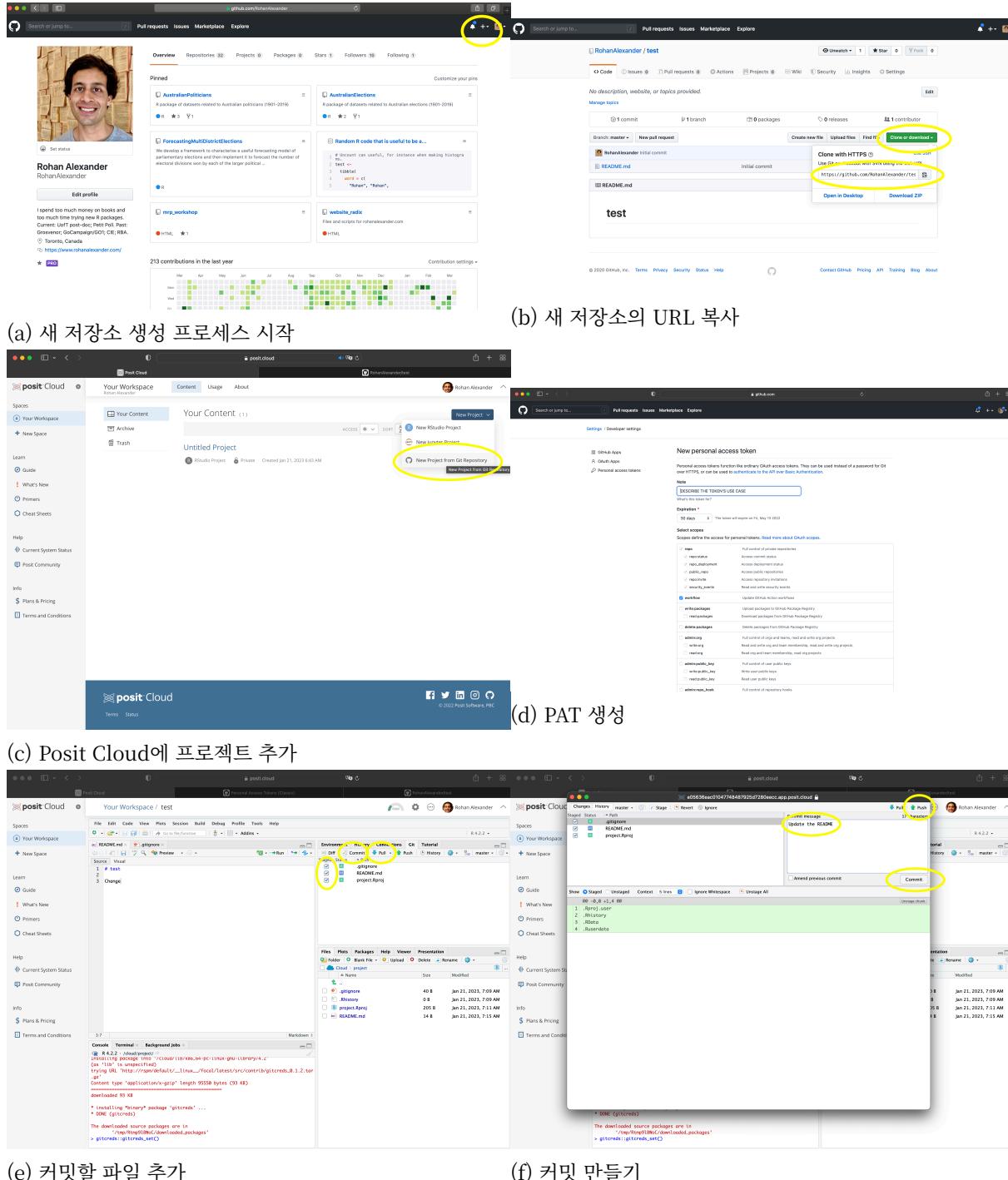


그림 3.2: GitHub 설정에 관련된 단계 개요

는 변경 사항에 대한 짧은 요약, 빈 줄, 그리고 변경 사항이 무엇이며 왜 이루어졌는지에 대한 설명을 포함해야 합니다. 예를 들어, 커밋이 논문에 그래프를 추가하는 경우 커밋 메시지는 다음과 같을 수 있습니다.

1

1 1.0.0-alpha.1 릴리스 허브 허브 커밋입니다.

전반적인 품질과 커밋 동작 사이에 관계가 있다는 증거가 있습니다 (Sprint 와/과 Conci 2019). 경험에 쌓이면 커밋 메시지가 프로젝트의 일종의 일지 역할을 할 것입니다. 그러나 가장 중요한 것은 정기적으로 커밋하는 것입니다.

Git과 GitHub는 데이터 과학자보다는 소프트웨어 개발자를 위해 설계되었습니다. GitHub는 고려할 파일의 크기를 100MB로 제한하며, 50MB도 경고를 표시할 수 있습니다. 데이터 과학 프로젝트에는 이보다 큰 데이터 세트가 정기적으로 포함됩니다. 장 ?? 에서는 프로젝트가 완료되었을 때 특히 유용한 데이터 예제 사용에 대해 논의하지만, 프로젝트를 활발하게 작업할 때는 Git과 GitHub에 관한 한 큰 데이터 파일을 무시하는 것이 유용할 수 있습니다. 이를 위해 “.gitignore” 파일을 사용하여 Git을 사용하여 추적하지 않은 모든 파일을 나열합니다. 예제 폴더⁸에는 “.gitignore” 파일이 포함되어 있습니다. 그리고 usethis에서 git_vaccinate()를 실행하는 것이 도움이 될 수 있습니다. 이렇게 하면 프로젝트별로 수행하는 것을 잊어버린 경우 다양한 파일을 전역 “.gitignore” 파일에 추가합니다. Mac 사용자는 “.DS_Store” 파일이 무시된다는 점이 유용할 것입니다.

RStudio의 Git 창을 사용하여 터미널을 사용할 필요가 없었지만, GitHub로 이동하여 새 프로젝트를 설정할 필요는 없었습니다. Git과 GitHub를 설정했으므로 usethis를 사용하여 워크플로우의 이 측면을 더욱 개선할 수 있습니다.

먼저 usethis의 git_sitrep()으로 Git이 설정되었는지 확인합니다. 그러면 사용자 이름과 이메일에 대한 정보가 인쇄되어야 합니다. 필요한 경우 use_git_config()를 사용하여 이러한 세부 정보를 업데이트할 수 있습니다.

```
#| eval: false
#| include: true

use_git_config(
  user.name = "로한 알렉산더",
  user.email = "rohan.alexander@utoronto.ca"
)
```

GitHub에서 새 프로젝트를 시작한 다음 로컬로 추가하는 대신, 이제 use_git()을 사용하여 프로젝트를 시작하고 파일을 커밋할 수 있습니다. 커밋한 후 use_github()를 사용하여 GitHub로 푸시하면 GitHub에도 폴더가 생성됩니다.

Git과 GitHub에 위협을 느끼는 것은 정상입니다. 많은 데이터 과학자들은 그것을 사용하는 방법을 조금만 알고 있으며, 그것으로 충분합니다. 필요한 경우 최근 스냅샷을 가질 수 있도록 정기적으로 푸시하십시오.

3.4.3 Git 충돌

파우스트 박사는 크리스토퍼 말로우의 16세기 희곡입니다. 흥미롭게도 두 가지 버전이 있으며, 말로우가 실제로 “그” 버전으로 의도한 것이 어떤 버전인지 아무도 모릅니다. 정확히 세는 방식에 따라 Marlowe (1604) 에는 약 2,048줄이 있고, Marlowe (1616) 에는 약 2,852줄이 있습니다. 줄 안에서도 변경 사항이 있습니다 (그림 ??). 저자가 오래 전에 사망했기 때문에 인류는 단순히 두 가지 버전이 존재하는 이상한 상황에 놓여 있습니다. 말로우가 Git을 가지고 있었다면 이런 일은 일어나지 않았을 것입니다!

Git과 GitHub를 사용할 때 우리는 주기적으로 버전을 체크인합니다. 그러나 동일한 저장소에서 작업하는 두 사람이 동일한 줄에 변경 사항을 적용하는 코드를 체크인하려고 하면 어떻게 될까요? Git은 병합 충돌을 설정하고, 충돌을 발생시킨 사람, 즉 두 번째 커밋을 한 사람이 이를 해결해야 합니다.

Git은 파일에 두 가지를 모두 표시하고 시작 부분에 <<<<< HEAD, 충돌하는 변경 사항을 구분하는 =====, 그리고 끝과 어떤 브랜치가 충돌을 생성하는지 보여주는 >>>>> new_branch와 같은 식별 마커를 추가하여 충돌하는 줄을 식별합니다.

⁸https://github.com/RohanAlexander/starter_folder

1	CHORUS. Not marching now in fields of Thrasymene,	1	CHORUS. Not marching in the fields of Thrasymene,
2	Where Mars did mate ¹ the Carthaginians;	2	Where Mars did mate the warlike Carthaginians; 1
3	Nor sporting in the dalliance of love,	3	Nor sporting in the dalliance of love,
4	In courts of kings where state is overturn'd;	4	In courts of kings where state is overturn'd;
5	Nor in the pomp of proud audacious deeds,	5	Nor in the pomp of proud audacious deeds,
6	Intends our Muse to vaunt ² her ³ heavenly verse:	6	Intends our Muse to vaunt her 2 heavenly verse:
7	Only this, gentlemen,-we must perform	7	Only this, gentles,-we must now perform
8	The form of Faustus' fortunes, good or bad:	8	The form of Faustus' fortunes, good or bad:
9	To patient judgments we appeal our plaud,	9	And now to patient judgments we appeal,
10	And speak for Faustus in his infancy.	10	And speak for Faustus in his infancy.
11	Now is he born, his parents base of stock,	11	Now is he born of parents base of stock,
12	In Germany, within a town call'd Rhodes:	12	In Germany, within a town call'd Rhodes:
13	Of riper years, to Wittenberg he went,	13	At riper years, to Wittenberg he went,
14	Whereas ⁴ his kinsmen chiefly brought him up.	14	Whereas his kinsmen chiefly brought him up.
15	So soon he profits in divinity,	15	So much he profits in divinity,
16	The fruitful plot of scholarism grac'd,	16	That shortly he was grac'd with doctor's name,
17	That shortly he was grac'd with doctor's name,	17	Excelling all, and sweetly can dispute
18	Excelling all whose sweet delight disputes	18	In th' heavenly matters of theology;
19	In heavenly matters of theology;	19	Till swo ⁿ with cunning, of 3 a self-conceit,
20	Till swo ⁿ with cunning, ⁵ of a self-conceit,	20	His waxen wings did mount above his reach,
21	His waxen wings did mount above his reach,	21	And, melting, heavens conspir'd his overthrow;
22	And, melting, heavens conspir'd his overthrow;	22	For, falling to a devilish exercise,
23	For, falling to a devilish exercise,	23	And glutted now with learning's golden gifts,
24	And glutted now ⁶ with learning's golden gifts,	24	He surfeits upon 4 cursed necromancy;
25	He surfeits upon cursed necromancy;	25	Nothing so sweet as magic is to him,
26	Nothing so sweet as magic is to him,	26	Which he prefers before his chiefest bliss:
27	Which he prefers before his chiefest bliss:	27	And this the man that in his study sits.
28	And this the man that in his study sits.	28	[Exit.]
29	[Exit.]	29	

(a) 1604년 버전의 처음 몇 줄

(b) 1616년 버전의 처음 몇 줄

그림 3.3: 파우스트 박사의 1604년 버전과 1616년 버전 간의 차이점 표시

```
#| eval: false
#| echo: true

<<<<< HEAD
@@ @@
=====
@@ @@ @@
>>>> new_branch
```

충돌을 해결하려는 사람의 임무는 어떤 내용을 유지할지 선택하는 것입니다. 파일을 편집하고 저장한 다음 일반적인 방식으로 추가하고 커밋합니다.

3.5 R 실습

3.5.1 오류 처리

프로그래밍을 하다 보면 결국 코드가 깨질 것입니다. 결국이라고 말하는 것은 아마 하루에 10번 또는 20번 정도를 의미합니다.

Gelfand (2021)

R을 사용하거나 어떤 프로그래밍 언어를 사용하든 모든 사람은 어느 시점에서 문제를 겪습니다. 이것은 정상입니다. 프로그래밍은 어렵습니다. 어느 시점에서 코드가 실행되지 않거나 오류가 발생할 것입니다. 이

것은 모든 사람에게 일어나는 일입니다. 좌절하는 것은 흔한 일이지만, 앞으로 나아가기 위해 문제를 해결하기 위한 전략을 개발합니다.

1. 오류 메시지가 나타나면 때로는 유용할 것입니다. 유용한 내용이 있는지 주의 깊게 읽어보십시오.
2. 오류 메시지를 검색해 보십시오. 결과를 더 적절하게 만들려면 검색에 “tidyverse” 또는 “in R”을 포함하는 것이 유용할 수 있습니다. 때로는 Stack Overflow 결과가 유용할 수 있습니다.
3. 함수 앞에 “?”를 붙여 함수의 도움말 파일을 살펴보십시오. 예를 들어, `?pivot_wider()`와 같이 말입니다. 일반적인 문제는 약간 잘못된 인수 이름이나 형식을 사용하는 것입니다. 예를 들어, 실수로 객체 이름 대신 문자열을 포함하는 경우입니다.
4. 오류가 발생하는 위치를 확인하고 오류가 해결될 때까지 코드를 제거하거나 주석 처리한 다음 천천히 코드를 다시 추가하십시오.
5. `class()`를 사용하여 객체의 클래스를 확인하십시오. 예를 들어, `class(data_set$data_column)`과 같이 말입니다. 예상한 것인지 확인하십시오.
6. R을 다시 시작하십시오: “세션” → “R 다시 시작 및 출력 지우기”. 그런 다음 모든 것을 다시 로드하십시오.
7. 컴퓨터를 다시 시작하십시오.
8. 오류 메시지 대신 수행하려는 작업을 검색하십시오. 결과를 더 적절하게 만들려면 검색에 “tidyverse” 또는 “in R”을 포함해야 합니다. 예를 들어, “ggplot을 사용하여 R에서 그래프 PDF 저장”과 같이 말입니다. 때로는 관련 블로그 게시물이나 Stack Overflow 답변이 도움이 될 것입니다.
9. 문제를 격리하고 다른 사람들이 도움을 줄 수 있도록 작고 자체 포함된 재현 가능한 예제 “reprex”를 만드십시오.
10. Quarto 문서에서 작업하는 경우 청크 옵션에 레이블을 포함하여 실수가 발생할 수 있는 위치를 더 쉽게 찾을 수 있도록 하십시오.

더 일반적으로, 항상 가능하지는 않지만, 휴식을 취하고 다음 날 다시 돌아오는 것이 거의 항상 도움이 됩니다.

3.5.2 재현 가능한 예제

아무도 당신에게 조언하거나 도울 수 없습니다. 아무도. 당신이 해야 할 일은 단 하나뿐입니다. 자신 안으로 들어가십시오.

(릴케?)

도움을 요청하는 것은 다른 기술과 마찬가지로 기술입니다. 연습을 통해 더 잘하게 됩니다. “이것은 작동하지 않습니다”, “모든 것을 시도했습니다”, “당신의 코드는 작동하지 않습니다”, 또는 “여기 오류 메시지가 있습니다. 어떻게 해야 합니까?”라고 말하지 않도록 노력하는 것이 중요합니다. 일반적으로 이러한 의견을 기반으로 도움을 줄 수는 없습니다. 너무 많은 가능한 문제가 있기 때문입니다. 다른 사람들이 당신을 돋기 쉽게 만들어야 합니다. 여기에는 몇 가지 단계가 포함됩니다.

1. 데이터와 코드의 작고 자체 포함된 예제를 제공하고, 무엇이 잘못되었는지 자세히 설명하십시오.
2. 지금까지 시도한 내용을 문서화하십시오. 어떤 Stack Overflow 및 Posit Forum 게시물을 보았는지, 그리고 왜 원하는 것이 아닌지 포함하십시오.
3. 원하는 결과에 대해 명확하게 설명하십시오.

최소한의 재현 가능한 예제(“reprex”)를 만드는 것으로 시작하십시오. 이것은 오류를 재현하는 데 필요한 것만 포함하는 코드입니다. 즉, 코드는 오류를 재현하지만 더 작고 간단한 버전일 가능성이 높습니다.

때로는 이 과정에서 문제를 해결할 수 있습니다. 그렇지 않다면, 다른 사람이 도움을 줄 수 있는 기회를 제공합니다. 다른 사람이 이전에 해결하지 못한 문제가 당신에게 있을 가능성은 거의 없습니다. 주요 어려움은 당신이 하고 싶은 것과 일어나고 있는 일을 다른 사람들이 모두 인식할 수 있는 방식으로 전달하는 것입니다. 끈기를 기르는 것이 중요합니다.

재현 가능한 예제를 개발하는 데 `reprex`가 특히 유용합니다. 설치 후 다음을 수행합니다.

1. `reprex` 패키지를 로드합니다: `library(reprex)`.
2. 문제가 있는 코드를 강조 표시하고 복사합니다.
3. 콘솔에서 `reprex()`를 실행합니다.

코드가 자체 포함되어 있으면 뷰어에 미리 보기와 표시됩니다. 그렇지 않으면 오류가 발생하며, 코드를 자체 포함되도록 다시 작성해야 합니다.

오류를 재현하기 위해 데이터가 필요한 경우 R에 내장된 데이터를 사용해야 합니다. `library(help = "datasets")`를 사용하여 R에 내장된 많은 데이터 세트를 볼 수 있습니다. 그러나 가능하다면 `mtcars` 또는 `faithful`과 같은 일반적인 옵션을 사용해야 합니다. `?@sec-fire-hose`에서 소개된 GitHub Gist와 `reprex`를 결합하면 다른 사람이 당신을 도울 수 있는 가능성이 높아집니다.

3.5.3 사고방식

당신이 어떤 IDE에서 개발하든, 어떤 도구를 사용하여 작업을 수행하든, 당신은 실제적이고 유효하며 유능한 사용자이자 프로그래머입니다.

문을 부수고 들어갑시다. 모두를 위한 충분한 공간이 있습니다.

Sharla Gelfand, 2020년 3월 10일.

코드를 작성한다면, 당신은 프로그래머입니다. 어떻게 하든, 무엇을 위해 사용하든, 누구든 상관없이 말입니다. 그러나 위대한 프로그래머들이 공통적으로 가지고 있는 몇 가지 특성이 있습니다.

- **집중:** 종종 “R 배우기”와 같은 목표는 문제가 되는 경향이 있습니다. 왜냐하면 그것에는 실제 끝점이 없기 때문입니다. “ggplot2”를 사용하여 2022년 호주 선거에 대한 히스토그램을 만드는 것과 같이 더 작고 구체적인 목표를 갖는 것이 더 효율적인 경향이 있습니다. 이것은 몇 시간 안에 집중하고 달성할 수 있는 것입니다. “R을 배우고 싶다”와 같이 모호한 목표의 문제는 곁가지로 빠지기 쉽고 도움을 받기 어렵다는 것입니다. 이것은 사기를 저하시키고 사람들이 너무 일찍 포기하게 만들 수 있습니다.
- **호기심:** “시도해 보는 것”은 거의 항상 유용합니다. 즉, 확실하지 않다면 그냥 시도해 보십시오. 일반적으로 최악의 경우 시간을 낭비하는 것입니다. 돌이킬 수 없을 정도로 무언가를 망가뜨리는 경우는 거의 없습니다. 예를 들어, 데이터프레임 대신 벡터를 `ggplot()`에 전달하면 어떻게 되는지 알고 싶다면 시도해 보십시오.
- **실용적:** 동시에 합리적인 범위 내에서 유지하고 한 번에 작은 변경만 하는 것이 유용할 수 있습니다. 예를 들어, 회귀 분석을 실행하고 싶고 `lm()` 대신 `rstanarm`을 사용할 가능성에 대해 궁금하다고 가정해 봅시다. 진행하는 실용적인 방법은 처음에 `rstanarm`의 한 측면을 사용한 다음 다음에 다른 변경을 하는 것입니다.
- **끈기:** 다시 말하지만, 이것은 균형 잡힌 행동입니다. 모든 프로젝트에는 예상치 못한 문제와 이슈가 발생합니다. 한편으로는 이러한 문제에도 불구하고 인내하는 것이 좋은 경향입니다. 그러나 다른 한편으로는 돌파구가 불가능해 보인다면 때로는 무언가를 포기할 준비가 되어 있어야 합니다. 멘토는 합리적인 것을 더 잘 판단하는 경향이 있으므로 유용할 수 있습니다.
- **계획:** 수행할 작업을 과도하게 계획하는 것이 거의 항상 유용합니다. 예를 들어, 일부 데이터의 히스토그램을 만들고 싶을 수 있습니다. 필요한 단계를 계획하고 각 단계가 어떻게 구현될 수 있는지 스케치해야 합니다. 예를 들어, 첫 번째 단계는 데이터를 가져오는 것입니다. 어떤 패키지가 유용할 수 있습니까? 데이터는 어디에 있을 수 있습니까? 데이터가 거기에 없으면 백업 계획은 무엇입니까?

- 완벽보다 완료: 우리 모두는 다양한 완벽주의적 경향을 가지고 있지만, 처음에는 어느 정도 그것들을 끄는 것이 유용할 수 있습니다. 처음에는 작동하는 코드를 작성하는 것에만 신경 쓰십시오. 언제든지 돌아와서 측면을 개선할 수 있습니다. 그러나 실제로 출시하는 것이 중요합니다. 작업을 완료하는 데 생긴 코드가 결코 완성되지 않는 아름다운 코드보다 낫습니다.

3.5.4 코드 주석 및 스타일

코드는 주석 처리되어야 합니다. 주석은 특정 코드가 작성된 이유와 일반적인 대안이 선택되지 않은 이유에 중점을 두어야 합니다. 실제로 코드를 작성하기 전에 주석을 작성하여 무엇을 하고 싶은지, 왜하고 싶은지 설명한 다음 코드를 작성하는 것이 좋습니다 (Fowler 와/과 Beck 2018, p. 59).

특히 R에서는 코드를 작성하는 한 가지 방법만 있는 것은 아닙니다. 그러나 혼자 작업하더라도 더 쉽게 작업할 수 있는 몇 가지 일반적인 지침이 있습니다. 대부분의 프로젝트는 시간이 지남에 따라 진화하며, 코드 주석의 한 가지 목적은 미래의 당신이 수행된 작업과 특정 결정이 내려진 이유를 다시 추적할 수 있도록 하는 것입니다 (Bowers 와/과 Voors 2016).

R 스크립트의 주석은 # 기호를 포함하여 추가할 수 있습니다. (#의 동작은 Quarto 문서의 R 청크 내부 줄과 R 청크 외부 줄에서 헤더 수준을 설정하는 것과 다릅니다.) 줄 시작 부분에 주석을 넣을 필요는 없으며, 중간에 넣을 수도 있습니다. 일반적으로 코드의 모든 측면이 무엇을 하는지 주석을 달 필요는 없지만, 명확하지 않은 부분은 주석을 달아야 합니다. 예를 들어, 어떤 값을 읽어들이는 경우 어디에서 왔는지 주석을 달고 싶을 수 있습니다.

무엇을 하고 있는지 주석을 달아야 합니다 (Wickham 2021c). 무엇을 달성하려고 합니까? 이상한 점을 설명하기 위해 주석을 달아야 합니다. 예를 들어, 특정 행, 예를 들어 27번 행을 제거하는 경우 왜 그 행을 제거하는지 설명해야 합니다. 그 순간에는 명확해 보일 수 있지만, 미래의 당신은 기억하지 못할 것입니다.

코드를 섹션으로 나누어야 합니다. 예를 들어, 작업 공간 설정, 데이터 세트 읽기, 데이터 세트 조작 및 정리, 데이터 세트 분석, 마지막으로 표 및 그림 생성 등이 있습니다. 각 섹션은 무엇이 진행되고 있는지 설명하는 주석으로 구분되어야 하며, 길이에 따라 별도의 파일로 나눌 수도 있습니다.

또한 각 파일의 상단에는 파일의 목적, 전제 조건 또는 종속성, 날짜, 저자 및 연락처 정보, 마지막으로 위험 요소 또는 할 일과 같은 기본 정보를 기록하는 것이 중요합니다.

R 스크립트에는 서문과 명확한 섹션 구분이 있어야 합니다.

```
##### ## #####
# 주석: 이 주석은 코드 내부 줄에 있는 주석입니다
# 주석: 주석 줄
# 주석: 주석 줄
# 주석: 주석 줄
# 주석: 주석 줄
# 주석: 주석 줄 및 주석 줄입니다.
# 주석:
# - 주석 줄 및 주석 줄입니다 주석 줄 및 주석 줄?

#####
## 주석 줄 및 주석 줄 #####
# install.packages("tidyverse")
# 주석 줄
library(tidyverse)

# 주석 줄 및 주석 줄.
raw_data <- read_csv("inputs/data/unedited_data.csv")

#####
## 주석 줄 및 주석 줄 #####
...
```

마지막으로, 코드가 작동하기 위해 사용자가 코드를 주석 처리하거나 주석 해제하는 것, 또는 디렉토리 지정과 같은 다른 수동 단계에 의존하지 않도록 노력하십시오. 이렇게 하면 자동화된 코드 검사 및 테스트를 사용할 수 없게 됩니다.

이 모든 것은 시간이 걸립니다. 대략적인 경험 법칙으로, 코드를 작성하는 데 걸린 시간만큼 주석을 달고 코드를 개선하는 데 시간을 할애해야 합니다. 잘 주석 처리된 코드의 예로는 Dolatsara 기타 (2021) 및 Burton, Cruz, 와/과 Hahn (2021) 이 있습니다.

3.5.5 테스트

테스트는 코드 전체에 작성되어야 하며, 끝에서 한꺼번에 작성하는 것이 아니라 진행하면서 작성해야 합니다. 이렇게 하면 속도가 느려질 것입니다. 그러나 생각하는 데 도움이 되고 실수를 수정하는 데 도움이 되어 코드를 더 좋게 만들고 전반적인 생산성을 향상시킬 것입니다. 테스트 없는 코드는 의심스럽게 보아야 합니다. R 패키지 (Vidoni 2021)의 테스트 관행은 물론 R 코드 전반에 걸쳐 개선의 여지가 있습니다.

다른 사람들, 그리고 이상적으로는 자동화된 프로세스가 코드를 테스트해야 하는 필요성은 우리가 재현성을 강조하는 이유 중 하나입니다. 또한 파일 경로를 하드코딩하지 않고, 프로젝트를 사용하고, 파일 이름에 공백을 두지 않는 것과 같은 작은 측면을 강조하는 이유이기도 합니다.

완전하고 일반적인 테스트 스위트를 정의하기는 어렵지만, 일반적으로 다음을 테스트하려고 합니다.

- 1) 경계 조건,
- 2) 클래스,
- 3) 누락된 데이터,
- 4) 관측치 및 변수의 수,
- 5) 중복,
- 6) 회귀 결과.

이 모든 것을 처음에는 시뮬레이션된 데이터에서 수행한 다음 실제 데이터로 이동합니다. 이는 아풀로 프로그램 중 테스트의 진화를 반영합니다. 처음에는 요구 사항에 대한 기대를 기반으로 테스트가 수행되었으며, 이러한 테스트는 나중에 실제 발사 측정값을 고려하도록 업데이트되었습니다 (Simpkinson 1971, p. 21). 무한한 수의 테스트를 작성할 수 있지만, 많은 생각 없는 테스트보다 적은 수의 고품질 테스트가 더 좋습니다.

한 가지 유형의 테스트는 “어설션”입니다. 어설션은 코드 전체에 작성되어 무언가가 참인지 확인하고 그렇지 않으면 코드 실행을 중지합니다 (Irving 기타 2021, p. 272). 예를 들어, 변수가 숫자여야 한다고 어설션할 수 있습니다. 이 어설션에 대해 테스트되었고 문자임이 밝혀지면 테스트가 실패하고 스크립트 실행이 중지됩니다. 데이터 과학의 어설션 테스트는 일반적으로 데이터 정리 및 준비 스크립트에서 사용됩니다. 장 ?? 에서 이에 대해 더 자세히 설명합니다. 단위 테스트는 코드의 완전한 측면을 확인합니다 (Irving 기타 2021, p. 274). 모델링을 고려할 때 장 ?? 에서 더 자세히 다룰 것입니다.

3.6 효율성

일반적으로 이 책에서는 무언가를 완료하는 데만 관심이 있습니다. 가장 좋거나 가장 효율적인 방법으로 완료하는 데는 반드시 관심이 없습니다. 왜냐하면 대부분의 경우 그것에 대해 걱정하는 것은 시간 낭비이기 때문입니다. 대부분의 경우 클라우드에 푸시하고 합리적인 시간 동안 실행되도록 한 다음 파이프라인의 다른 측면에 대해 걱정하는 것이 더 좋습니다. 그러나 그것은 결국 불가능해집니다. 특정 시점에서 (그리고 이는 상황에 따라 다릅니다) 효율성이 중요해집니다. 결국 못생기거나 느린 코드, 그리고 특정 방식으로 작업을 수행하려는 독단적인 고집은 영향을 미칩니다. 그리고 그 시점에서 효율성을 보장하기 위해 새로운 접근 방식에 개방적이어야 합니다. 명백한 성능 향상을 위한 가장 일반적인 영역은 거의 없습니다. 대신 측정, 평가 및 사고 능력을 개발하는 것이 중요합니다.

코드 효율성을 향상시키는 가장 좋은 방법 중 하나는 다른 사람의 시선을 빌릴 수 있도록 준비하는 것입니다. 그들의 시간을 최대한 활용하려면 코드를 읽기 쉽게 만드는 것이 중요합니다. 그래서 “코드 린팅”과 “스타일링”으로 시작합니다. 이것은 코드 속도를 높이는 것이 아니라, 다른 사람이 코드를 보거나 우리가 다시 방문할 때 더 효율적으로 만듭니다. 이를 통해 공식적인 코드 검토 및 리팩토링이 가능해집니다. 리팩토링은 코드를 더 좋게 만들기 위해 코드를 다시 작성하는 것이지만, 코드가 하는 일을 변경하지는 않습니다 (동일한 작업을 다른 방식으로 수행합니다). 그런 다음 실행 시간 측정으로 전환하고 병렬 처리를 도입하여 컴퓨터가 여러 프로세스에 대한 코드를 동시에 실행할 수 있도록 합니다.

3.6.1 코드 환경 공유

우리는 코드 공유의 필요성에 대해 길게 논의했으며, GitHub를 사용하여 이에 대한 접근 방식을 제시했습니다. 그리고 장 ??에서는 데이터 공유에 대해 논의할 것입니다. 그러나 다른 사람들이 우리의 코드를 실행할 수 있도록 하는 또 다른 요구 사항이 있습니다. 장 ??에서 R 자체와 R 패키지가 새로운 기능이 개발되고 오류가 수정되며 기타 일반적인 개선 사항이 있을 때마다 때때로 업데이트된다는 점을 논의했습니다. “R 필수 사항” 온라인 부록⁹은 tidyverse의 한 가지 장점이 더 구체적이기 때문에 기본 R보다 더 빠르게 업데이트될 수 있다는 점을 설명합니다. 그러나 이는 우리가 사용하는 모든 코드와 데이터를 공유하더라도 사용 가능한 소프트웨어 버전이 오류를 유발할 수 있음을 의미할 수 있습니다.

이에 대한 해결책은 사용된 환경을 자세히 설명하는 것입니다. 이를 수행하는 방법은 여러 가지가 있으며 복잡성을 더할 수 있습니다. 우리는 R 및 R 패키지의 사용된 버전을 문서화하고 다른 사람들이 정확한 버전을 더 쉽게 설치할 수 있도록 하는 데 중점을 둡니다. 본질적으로 우리는 재현성에 도움이 될 것이기 때문에 사용한 설정을 격리하는 것입니다 (Perkel 2023). R에서는 `renv`를 사용하여 이를 수행할 수 있습니다.

`renv`가 설치되고 로드되면 `init()`을 사용하여 필요한 인프라를 설정합니다. 사용된 패키지와 버전을 기록할 파일을 만들 것입니다. 그런 다음 `snapshot()`을 사용하여 실제로 사용 중인 것을 문서화합니다. 이렇게 하면 정보를 기록하는 “잠금 파일”이 생성됩니다.

R 프로젝트에서 어떤 패키지를 사용하고 있는지 확인하려면 `dependencies()`를 사용할 수 있습니다. 예제 풀 더¹⁰에 대해 이 작업을 수행하면 `rmarkdown`, `bookdown`, `knitr`, `rmarkdown`, `bookdown`, `knitr`, `palmerpenguins`, `tidyverse`, `renv`, `haven`, `readr`, `tidyverse` 패키지가 사용됨을 나타냅니다.

원한다면 잠금 파일(“`renv.lock`”)을 열어 정확한 버전을 확인할 수 있습니다. 잠금 파일은 설치된 다른 모든 패키지와 다운로드된 위치도 문서화합니다. 외부에서 이 프로젝트에 접근하는 사람은 `restore()`를 사용하여 우리가 사용한 패키지의 정확한 버전을 설치할 수 있습니다.

3.6.2 코드 린팅 및 스타일링

빠른 것은 가치 있지만, 주로 코드가 빠르게 실행되는 것이 아니라 빠르게 반복할 수 있는 능력에 관한 것입니다. Backus (1981, p. 26)는 1954년에도 프로그래머 비용이 컴퓨터 비용만큼 들었으며, 요즘에는 추가적인 컴퓨팅 성능이 프로그래머보다 훨씬 저렴하다고 설명합니다. 성능이 좋은 코드는 중요하지만, 다른 사람의 시간을 효율적으로 사용하는 것도 중요합니다. 코드는 한 번만 작성되는 경우가 거의 없습니다. 대신 일반적으로 실수만 수정하더라도 다시 돌아와야 하며, 이는 코드가 사람에게 읽힐 수 있어야 함을 의미합니다 (Matsumoto 2007, p. 478). 이렇게 하지 않으면 효율성 비용이 발생할 것입니다.

린팅 및 스타일링은 주로 스타일 문제에 대해 코드를 확인하고, 코드를 읽기 쉽게 재정렬하는 과정입니다. (린팅의 또 다른 측면은 닫는 괄호를 잊는 것과 같은 프로그래밍 오류를 처리하는 것이지만, 여기서는 스타일 문제에 중점을 둡니다.) 종종 가장 큰 효율성 향상은 다른 사람이 코드를 읽기 쉽게 만드는 데서 비롯됩니다. 심지어 이것이 휴식 후 코드로 돌아오는 우리 자신일지도 말입니다. 미국 독점 거래 회사인 Jane Street는 코드 가독성을 보장하는 데 매우 강한 초점을 맞추고 있으며, 이는 위험 완화의 핵심 부분입니다 (Minsky 2011). 우리 모두가 잠재적으로 벤더스러운 코드 관리 하에 수십억 달러를 가지고 있지는 않지만, 우리 모두는 코드가 오류를 생성하지 않는 것을 선호할 것입니다.

`lintr`의 `lint()`을 사용하여 코드를 린팅합니다. 예를 들어, 다음 R 코드를 고려하십시오 (“`linting_example.R`”로 저장됨).

```
#| include: true
#| message: false
#| warning: false
#| eval: false

SIMULATED_DATA <-
tibble(
  division = c(1:150, 151),
  party = sample(
    x = c("Liberal"),
```

⁹https://tellingstorieswithdata.com/20-r_essentials.html

¹⁰https://github.com/RohanAlexander/starter_folder

```

size = 151,
replace = T
)
)

#| echo: true
#| eval: false

lint(filename = "linting_example.R")

```

결과는 “linting_example.R” 파일이 열리고 `lint()`가 찾은 문제가 “마커”에 인쇄됩니다 (그림 ??). 그런 다음 문제를 처리하는 것은 당신에게 달려 있습니다.

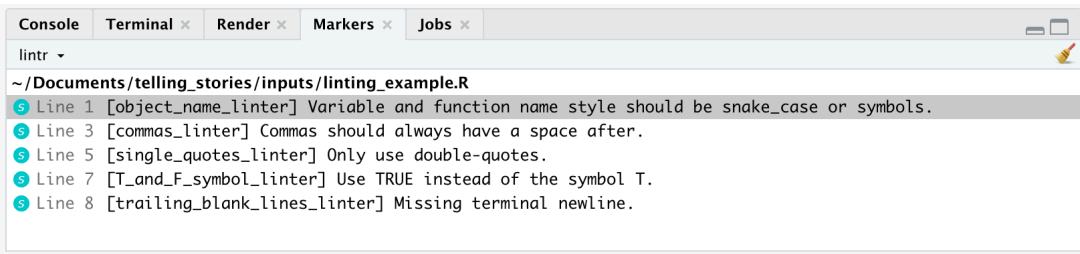


그림 3.4: 예제 R 코드의 린팅 결과

권장 변경 사항을 적용하면 Wickham (2021c) 에서 정의한 모범 사례와 일치하는 더 읽기 쉬운 코드가 생성됩니다.

```

#| include: true
#| message: false
#| warning: false
#| eval: false

simulated_data <-
tibble(
  division = c(1:150, 151),
  party = sample(
    x = c("Liberal"),
    size = 151,
    replace = TRUE
  )
)

```

처음에는 린터가 식별하는 일부 측면, 예를 들어 후행 공백과 이중 따옴표만 사용하는 것이 작고 중요하지 않게 보일 수 있습니다. 그러나 그것들은 더 큰 문제를 해결하는 데 방해가 됩니다. 또한, 작은 것을 제대로 처리할 수 없다면, 누가 우리가 큰 것을 제대로 처리할 수 있다고 믿을 수 있겠습니까? 따라서 린터가 식별하는 모든 작은 측면을 처리하는 것이 중요합니다.

`lintr` 외에도 `styler`도 사용합니다. 이것은 린터와 달리 스타일 문제를 자동으로 조정합니다. 린터는 검토할 문제 목록을 제공했습니다. 이를 실행하려면 `style_file()`을 사용합니다.

```

#| echo: true
#| eval: false

style_file(path = "linting_example.R")

```

이렇게 하면 공백 및 들여쓰기와 같은 변경 사항이 자동으로 적용됩니다. 따라서 오류가 발생하지 않았는지 변경 사항을 검토하고 확인하기 위해 프로젝트가 끝날 때 한 번만 하는 것이 아니라 정기적으로 수행해야 합니다.

3.6.3 코드 검토

이러한 스타일 측면을 모두 처리한 후 코드 검토로 넘어갈 수 있습니다. 이는 다른 사람이 코드를 검토하고 비판하는 과정입니다. 많은 전문 작가들은 편집자를 가지고 있으며, 코드 검토는 데이터 과학에서 이에 가장 가깝습니다. 코드 검토는 코드 작성의 중요한 부분이며, Irving 기타 (2021, p. 465)는 이를 “버그를 찾는 가장 효과적인 방법”이라고 설명합니다. 특히 코딩을 배울 때 피드백을 받는 것이 개선에 큰 도움이 되기 때문에 매우 유용하지만, 상당히 부담스러울 수 있습니다.

다른 사람의 코드를 검토할 때 정중하고 협력적인 태도를 취하십시오. 공백 및 구분과 같은 스타일과 관련된 작은 측면은 린터와 스타일러가 처리했어야 하지만, 그렇지 않은 경우 이에 대한 일반적인 권장 사항을 제시하십시오. 데이터 과학에서 코드 검토자로서 대부분의 시간은 다음과 같은 측면에 할애해야 합니다.

- 1) 정보가 담긴 README가 있습니까? 어떻게 개선할 수 있습니까?
- 2) 파일 이름과 변수 이름이 일관되고, 정보가 담겨 있으며, 의미가 있습니까?
- 3) 주석을 통해 무엇이 수행되고 있는지 이해할 수 있습니까?
- 4) 테스트가 적절하고 충분합니까? 고려되지 않은 예외적인 경우나 코너 솔루션이 있습니까? 마찬가지로, 불필요한 테스트가 제거될 수 있습니까?
- 5) 변수로 변경하고 설명할 수 있는 매직 넘버가 있습니까?
- 6) 변경할 수 있는 중복 코드가 있습니까?
- 7) 해결해야 할 미해결 경고가 있습니까?
- 8) 더 작은 함수로 분리할 수 있는 특히 큰 함수나 파이프가 있습니까?
- 9) 프로젝트의 구조가 적절합니까?
- 10) 코드를 데이터로 변경할 수 있습니까 (Irving 기타 2021, p. 462)?

예를 들어, 총리와 대통령의 이름을 찾는 코드를 고려해 봅시다. 이 코드를 처음 작성했을 때 관련 이름을 코드에 직접 추가했을 것입니다. 그러나 코드 검토의 일부로, 이를 변경하도록 권장할 수 있습니다. 관련 이름의 작은 데이터 세트를 만들고, 해당 데이터 세트를 조회하도록 코드를 다시 작성하도록 권장할 수 있습니다.

코드 검토는 코드가 적어도 한 명의 다른 사람에게 이해될 수 있도록 보장합니다. 이는 세상에 대한 지식을 구축하는 데 중요한 부분입니다. Google에서 코드 검토는 주로 결함을 찾는 것이 아니라, 가독성과 유지 보수성을 보장하고 교육을 제공하는 것입니다 (Sadowski 기타 2018). Jane Street에서도 마찬가지입니다. 그들은 코드 검토를 사용하여 버그를 잡고, 기관 지식을 공유하고, 교육을 지원하며, 직원이 읽을 수 있는 코드를 작성하도록 의무화합니다 (Minsky 2015).

마지막으로, 코드 검토는 번거롭고 하루 종일 걸리는 모든 코드를 읽는 과정이 될 필요도 없고 되어서도 안 됩니다. 최고의 코드 검토는 단 하나의 파일에 대한 빠른 검토이며, 몇 줄의 변경 사항을 제안하는 데 중점을 둡니다. 실제로 한 명의 개인이 아닌 소규모 팀이 검토하는 것이 더 나을 수 있습니다. 한 번에 너무 많은 코드를 검토하지 마십시오. 최대 몇 백 줄 정도이며, 이는 약 한 시간이 걸려야 합니다. 그 이상은 효율성 감소와 관련이 있는 것으로 밝혀졌기 때문입니다 (J. Cohen, Teleki, 와/과 Brown 2006, p. 79).

3.6.4 코드 리팩토링

코드를 리팩토링한다는 것은 새 코드가 이전 코드와 동일한 결과를 달성하지만, 새 코드가 더 잘 수행하도록 다시 작성하는 것을 의미합니다. 예를 들어, (refactornature는?) 중요한 영국 코로나 모델의 기반이 된 코드가 처음에 역학자들에 의해 작성되었으며, 몇 달 후 왕립 학회, Microsoft 및 GitHub 팀에 의해 명확하고 정리되었다고 논의합니다. 이는 두 버전 모두 동일한 입력이 주어졌을 때 동일한 출력을 생성했음에도 불구하고 모델에 대한 더 많은 신뢰를 제공하기 때문에 가치 있었습니다.

우리는 일반적으로 다른 사람이 작성한 코드와 관련하여 코드 리팩토링을 언급합니다. (비록 우리가 실제로 코드를 작성했지만, 그것이 오래 전의 일일 수도 있습니다.) 코드를 리팩토링하기 시작할 때, 다시 작성된 코드가 원본 코드와 동일한 결과를 달성하는지 확인하고 싶습니다. 이는 우리가 의존할 수 있는 적절한 테스트 스위트가 작성되어 있어야 함을 의미합니다. 이러한 테스트가 존재하지 않는다면, 우리는 그것들을 생성해야 할 수도 있습니다.

우리는 다른 사람들이 더 쉽게 이해할 수 있도록 코드를 다시 작성하며, 이는 우리의 결론에 대한 더 많은 신뢰를 가능하게 합니다. 그러나 그렇게 하기 전에 기존 코드가 무엇을 하는지 이해해야 합니다. 시작하는 한 가지 방법은 코드를 살펴보고 광범위한 주석을 추가하는 것입니다. 이러한 주석은 일반적인 주석과 다

릅니다. 이는 각 코드 청크가 무엇을 하려고 하는지, 그리고 어떻게 개선될 수 있는지 이해하려는 우리의 적극적인 과정입니다.

코드 리팩토링은 코드가 모범 사례를 충족하는지 확인할 수 있는 기회입니다. (Trisovic2022는?) 9,000개의 R 스크립트를 검토한 결과 다음과 같은 핵심 권장 사항을 제시합니다.

1. `setwd()` 및 모든 절대 경로를 제거하고, “`.Rproj`” 파일과 관련된 상대 경로만 사용하도록 합니다.
2. 명확한 실행 순서가 있는지 확인합니다. 처음에는 파일 이름에 숫자를 사용하여 이를 달성하도록 권장했지만, 결국에는 `targets` (Landau 2021)와 같은 더 정교한 접근 방식을 대신 사용할 수 있습니다.
3. 코드가 다른 컴퓨터에서 실행될 수 있는지 확인합니다.

예를 들어, 다음 코드를 고려하십시오.

```
#| eval: false

setwd("/Users/rohanalexander/Documents/telling_stories")

library(tidyverse)

d = read_csv("cars.csv")

mtcars =
  mtcars |>
  mutate(K_P_L = mpg / 2.352)

library(datasauRus)
```

`datasaurus_dozen`

R 프로젝트를 생성하여 `setwd()`를 제거하고, 모든 `library()` 호출을 맨 위에 그룹화하고, “`=`” 대신 “`<-`”를 사용하고, 변수 이름을 일관되게 유지하여 변경할 수 있습니다.

```
#| eval: false

library(tidyverse)
library(datasauRus)

cars_data <- read_csv("cars.csv")

mpg_to_kpl_conversion_factor <- 2.352

mtcars <-
  mtcars |>
  mutate(kpl = mpg / mpg_to_kpl_conversion_factor)
```

3.7 결론

이 장에서는 많은 것을 고려했으며, 암도되는 것은 정상입니다. 필요에 따라 Quarto 섹션으로 돌아오십시오. 많은 사람들이 Git과 GitHub에 대해 혼란스러워하며, 그저 필요한 만큼만 알고 있습니다. 그리고 효율성에 대한 많은 자료가 있었지만, 성능이 좋은 코드의 가장 중요한 측면은 다른 사람이 읽기 쉽게 만드는 것입니다. 심지어 그 사람이 휴식 후 돌아온 당신 자신일지라도 말입니다.

3.8 연습 문제

실습

- (계획) 다음 시나리오를 고려하십시오: 어떤 나라에서는 의회에서 의석을 얻을 수 있는 정당이 항상 네 개뿐입니다. 주어진 의석과 관련된 지역에서 가장 많은 표를 얻은 후보가 그 의석을 차지합니다. 의회는 총 175석으로 구성됩니다. 한 분석가는 의석별 각 정당의 득표수에 관심이 있습니다. 데이터 세트가 어떻게 생겼을지 스케치하고, 모든 관측치를 보여주기 위해 만들 수 있는 그래프를 스케치하십시오.
- (시뮬레이션) 설명된 시나리오를 더 자세히 고려하고 상황을 시뮬레이션하십시오. 아래 코드를 사용하여 적절한 상황을 신중하게 지정하십시오. 그런 다음 시뮬레이션된 데이터를 기반으로 다섯 가지 테스트를 작성하십시오.

```
#| eval: false
#| echo: true
```

```
library(tidyverse)

election_results <-
  tibble(
    seat = rep(1:175, each = 4),
    party = rep(x = 1:4, times = 175),
    votes = runif(n = 175 * 4, min = 0, max = 1000) |> floor()
  )
```

- (획득) 관심 있는 국가의 실제 투표 데이터 소스를 지정하십시오.
- (탐색) 다음 코드로 시작하여 각 정당이 얻은 의석 수 표를 만드십시오.

```
#| eval: false
#| echo: true
```

```
library(tidyverse)

election_results |>
  slice_max(votes, n = 1, by = seat) |>
  count(party) |>
  tt()
```

- (공유) 식별한 소스에서 데이터를 수집한 것처럼 (시뮬레이션된 것이 아니라) 그리고 시뮬레이션된 데이터를 사용하여 만든 표가 실제 상황을 반영한 것처럼 두 단락을 작성하십시오. 단락에 포함된 정확한 세부 정보는 사실일 필요는 없지만 합리적이어야 합니다 (즉, 실제로 데이터를 얻거나 그래프를 만들 필요는 없습니다). 코드를 R 파일과 Quarto 문서로 적절하게 분리하십시오. README가 있는 GitHub 저장소 링크를 제출하십시오.

퀴즈

- Gelman (2016)에 따르면, 연구자들이 데이터 분석의 유연성을 이용하여 유의미한 결과를 찾는 통계적 개념은 무엇입니까 (하나 선택)?
 - 무작위 표본 추출.
 - P-해킹.
 - 귀무 가설 검정.
 - 베이즈 추론.
- Gelman (2016)에 따르면, “p-해킹”은 무엇입니까 (하나 선택)?
 - p-값을 수정하는 방법.
 - 유의미하지 않은 결과가 유의미해질 때까지 데이터 또는 분석을 조작하는 것.

- c. 계산 효율성을 향상시키는 기술.
 - d. 데이터 공유에 대한 윤리적 접근 방식.
3. Gelman (2016)에 따르면, 파일 서랍 문제(file drawer problem)는 무엇입니까 (하나 선택)?
- a. 유의미한 결과만 출판하여 발생하는 편향.
 - b. 보관된 데이터에 접근하기 어려움.
 - c. 데이터 코딩 및 입력 오류.
 - d. 오래된 실험을 복제하는 데 어려움.
4. Gelman (2016)에 따르면, 긍정적인 결과만 출판하려는 경향을 나타내는 용어는 무엇입니까 (하나 선택)?
- a. 데이터 마이닝.
 - b. 출판 편향.
 - c. 확증 편향.
 - d. 표본 오차.
5. Gelman (2016)에 따르면, 연구자들이 동일한 데이터로 유의미한 결과를 얻을 수 있도록 하는 데이터 분석에서 가질 수 있는 수많은 선택을 설명하는 용어는 무엇입니까 (하나 선택)?
- a. 연구자 자유도.
 - b. 데이터 마이닝.
 - c. 표본 편향.
 - d. 효과 크기 조작.
6. Gelman (2016)에 따르면, 갈림길의 정원(garden of forking paths)은 어떤 문제를 나타냅니까 (하나 선택)?
- a. 기계 학습에서 의사 결정 트리의 복잡성.
 - b. 동일한 데이터로 수행할 수 있는 여러 잠재적 분석.
 - c. 시간이 지남에 따라 이론과 응용 작업의 분기.
 - d. 학문 분야의 분기.
7. Gelman (2016)에 따르면, 연구에서 “복제”는 무엇입니까 (하나 선택)?
- a. 새로운 데이터를 사용하여 원본 결과를 재현하는 연구.
 - b. 이전 방법론을 비판하는 연구.
 - c. 여러 연구의 메타 분석.
 - d. 원본 연구 논문의 정확한 복사본.
8. Gelman (2016)에 따르면, 사회 과학에서 재현 불가능한 결과에 기여하는 요인은 무엇입니까 (하나 선택)?
- a. 부적절한 표본 크기.
 - b. 고급 통계 소프트웨어 부족.
 - c. 선택적 보고로 이어지는 연구자 자유도.
 - d. 질적 데이터에 대한 과도한 의존.
9. Gelman (2016)에 따르면, 복제 위기(replication crisis)는 무엇을 의미합니까 (하나 선택)?
- a. 새로운 이론을 만드는 데 어려움.
 - b. 유사한 연구의 과잉 생산.
 - c. 이전 연구 결과의 복제에 어려움.
 - d. 실험 참가자 부족.
10. Gelman (2016)에 따르면, 복제 위기를 완화하는 데 도움이 되는 것은 무엇입니까 (하나 선택)?
- a. 데이터 기밀 유지.
 - b. 유의미한 결과만 출판.
 - c. 연구 및 분석 계획 사전 등록.
 - d. 독점 소프트웨어 사용 증가.
11. Gelman (2016)은 심리학의 복제 위기에 초점을 맞춥니다. 자신의 경험, 아마도 다른 수업에서 얻은 경험을 바탕으로 다른 학문 분야를 선택하고, 해당 학문 분야에서 복제 문제가 있을 수 있다고 생각하는 정도와 그 이유에 대해 작성하십시오.
12. 익숙한 학문 분야를 선택하십시오. 해당 분야에서 재현성을 향상시킬 수 있는 관행은 무엇입니까? 간략하게 설명하십시오.
13. Wilson 기타 (2017)에 따르면, 다음 중 중요한 데이터 관리 관행은 무엇입니까 (모두 선택)?
- a. 원본 데이터와 정리된 버전 모두 저장.
 - b. 데이터 처리 단계 문서화.
 - c. 데이터 저장에 비독점 파일 형식 사용.

14. Wilson 기타 (2017) 에 따르면, 프로젝트의 홈 디렉토리에 README 파일을 만드는 것이 왜 중요니까 (하나 선택)?
- 원본 데이터 파일을 저장하기 위해.
 - 프로젝트의 목적을 설명하고 개요를 제공하기 위해.
 - 프로젝트의 모든 오류 및 버그를 나열하기 위해.
 - 프로젝트 파일의 모든 버전을 추적하기 위해.
15. Wilson 기타 (2017) 에 따르면, 버전 제어를 사용하는 주요 이점은 무엇입니까 (하나 선택)?
- 연구자를 위해 코드를 자동으로 작성합니다.
 - 변경 사항을 추적하고 협업을 돋憬니다.
 - 데이터 백업의 필요성을 대체합니다.
 - 모든 데이터가 암호화되도록 보장합니다.
16. Wilson 기타 (2017) 에 따르면, 프로젝트에서 파일 이름을 지정하는 권장 관행은 무엇입니까 (하나 선택)?
- 파일 이름에 내용 또는 기능을 반영합니다.
 - result1.csv, result2.csv와 같은 순차 번호 사용.
 - 파일 이름을 고유하게 만들기 위해 특수 문자 포함.
 - 파일 이름에 공백 및 구두점 사용.
17. Wilson 기타 (2017) 에 따르면, 원본 데이터의 수정되지 않은 복사본을 저장해야 하는 이유는 무엇입니까 (하나 선택)?
- 데이터 저장 공간을 보존하기 위해.
 - 법적 규정을 준수하기 위해.
 - 확인 및 재현성을 위한 변경되지 않은 소스를 보장하기 위해.
 - 소프트웨어 업데이트와의 호환성을 유지하기 위해.
18. Wilson 기타 (2017) 에 따르면, 개방형 파일 형식을 사용하는 주요 이점은 무엇입니까 (하나 선택)?
- 처리 속도가 더 빠릅니다.
 - 독점 소프트웨어 없이 접근 가능합니다.
 - 데이터를 더 효율적으로 압축합니다.
 - 데이터 보안을 강화합니다.
19. Wilson 기타 (2017) 에 따르면, 데이터 파일을 구성할 때 권장되는 관행은 무엇입니까 (모두 선택)?
- 의미 있고 일관된 파일 이름 사용.
 - 모든 파일을 단일 폴더에 저장.
 - 파일을 명확한 디렉토리 구조로 구성.
 - 버전 추적을 위해 파일 이름에 날짜 포함.
20. Wilson 기타 (2017) 에 따르면, 데이터 처리 단계를 문서화하는 것이 왜 중요합니까 (하나 선택)?
- 데이터 분석 속도를 높입니다.
 - 데이터 암호화에 도움이 됩니다.
 - 저장 요구 사항을 줄입니다.
 - 다른 사람들이 분석을 이해하고 재현할 수 있도록 합니다.
21. 재현성의 이점은 무엇입니까?
- 결과를 독립적으로 확인할 수 있습니다.
 - 코드 실행 속도를 높입니다.
 - 데이터 시각화를 더 쉽게 만듭니다.
 - 문서화 필요성을 줄입니다.
22. M. Alexander (2019a) 에 따르면, 연구는 다음의 경우 재현 가능합니다 (하나 선택)?
- 동료 심사 저널에 출판됩니다.
 - 연구에 사용된 모든 자료가 제공됩니다.
 - 저자가 자료를 제공하지 않아도 정확히 재현될 수 있습니다.
 - 연구에 사용된 모든 자료가 주어졌을 때 정확히 재현될 수 있습니다.
23. 문학적 프로그래밍은 무엇입니까 (하나 선택)?
- 코드와 문서를 다른 파일로 분리합니다.
 - 코드의 구문 오류를 자동으로 수정합니다.
 - 코드 문서 생성을 자동화합니다.
 - 동일한 문서에 코드와 자연어를 통합합니다.

24. 재현 가능한 워크플로우에서 git의 주요 기능은 무엇입니까 (하나 선택)?
 - a. 데이터 정리를 자동화합니다.
 - b. 코드를 병렬로 실행합니다.
 - c. 데이터 시각화를 보고서에 통합합니다.
 - d. 코드에 대한 버전 제어 시스템을 제공합니다.
25. Wickham (2021c)에 따르면, “00_get_data.R” 및 “get data.R” 파일은 어떻게 분류됩니까 (하나 선택)?
 - a. 나쁨; 나쁨.
 - b. 좋음; 나쁨.
 - c. 나쁨; 좋음.
 - d. 좋음; 좋음.
26. 재현 가능한 연구를 위해 Quarto를 사용하는 이점은 무엇입니까 (하나 선택)?
 - a. 통계 분석을 자동화합니다.
 - b. 코드와 텍스트를 통합합니다.
 - c. 버전 제어의 필요성을 대체합니다.
 - d. 데이터 시각화 기능을 향상시킵니다.
27. Quarto에서 최상위 제목을 어떻게 나타냅니까 (하나 선택)?
 - a.
 - b. 제목
 - c.

3.8.1 제목

- b. 제목
- c.

4

제목

- d. • 제목
28. 다음 중 Quarto에서 굵은 텍스트를 생성하는 것은 무엇입니까 (하나 선택)?
- a. ****문자****
 - b. #**문자**#
 - c. ***문자***
 - d. #**문자**#
29. Quarto R 코드 청크에서 “echo” 옵션은 무엇을 합니까 (하나 선택)?
- a. 코드 출력을 억제하기 위해.
 - b. 코드가 문서에 표시될지 여부를 제어하기 위해.
 - c. 코드를 조건부로 평가하기 위해.
 - d. 출력에 경고를 포함하기 위해.
30. Quarto R 청크에서 경고를 숨기는 옵션은 무엇입니까 (하나 선택)?
- a. echo: false
 - b. eval: false
 - c. warning: false
 - d. message: false
31. R 코드 청크를 실행하고 결과를 표시하지만, Quarto R 청크에서 코드를 표시하지 않는 옵션은 무엇입니까 (하나 선택)?
- a. echo: false
 - b. include: false
 - c. eval: false
 - d. warning: false
 - e. message: false
32. R 프로젝트가 중요한 이유는 무엇입니까 (모두 선택)?
- a. 재현성에 도움이 됩니다.
 - b. 코드를 더 쉽게 공유할 수 있습니다.
 - c. 작업 공간을 더 체계적으로 만듭니다.
33. R 프로젝트 이름이 저장소의 내용을 반영하는 것이 왜 중요합니까 (모두 선택)?
- a. 일관성.
 - b. 전문성.
 - c. 세부 사항에 대한 주의.
34. 패키지와 데이터 세트가 로드되었다고 가정할 때, 이 코드에서 무엇이 잘못되었습니까:
DoctorVisits |> filter(visits) (하나 선택)?
- a. DoctorVisits
 - b. |>
 - c. filter
 - d. visits
35. reprex는 무엇이며, reprex를 만들 수 있는 것이 왜 중요합니까 (모두 선택)?
- a. 오류를 재현할 수 있도록 하는 재현 가능한 예제.
 - b. 다른 사람들이 당신을 돋도록 돋는 재현 가능한 예제.
 - c. 구축 과정에서 자신의 문제를 해결할 수 있는 재현 가능한 예제.
 - d. 자신을 돋기 위해 실제로 노력했음을 보여주는 재현 가능한 예제.
36. Gelfand (2021)에 따르면, “막힌 것을 해결하는 데 도움이 필요하다면, 첫 번째 단계는 reprex, 즉 재현 가능한 예제를 만드는 것입니다. reprex의 목표는 다른 사람들이 실행하고 당신의 고통을 느낄 수 있도록 문제 있는 코드를 패키징하는 것입니다. 그러면 바라건대, 그들이 해결책을 제공하고 당신의 고통을 끝낼 수 있을 것입니다.”에서 핵심 부분은 무엇입니까 (하나 선택)?
- a. 문제 있는 코드를 패키징합니다.

- b. 다른 사람들이 실행하고 당신의 고통을 느낄 수 있습니다.
 - c. 첫 번째 단계는 reprex를 만드는 것입니다.
 - d. 그들이 해결책을 제공하고 당신의 고통을 끝낼 수 있습니다.
37. Gelfand (2021)에 따르면, 도움을 구할 때 재현 가능한 예제를 만드는 것이 왜 중요합니까 (하나 선택)?
- a. 문서화 필요성을 줄입니다.
 - b. 코딩 기술을 보여줍니다.
 - c. 다른 사람들이 문제를 재현하고 해결책을 제공할 수 있도록 합니다.
 - d. 소프트웨어 라이선스를 준수합니다.
38. 다른 사람과 협업할 때 코드의 효율성을 높이는 관행은 무엇입니까 (하나 선택)?
- a. 절대 파일 경로 사용.
 - b. 명확한 주석 및 문서 작성.
 - c. 함수 사용 최소화.
 - d. 지적 재산을 보호하기 위해 코드 난독화.
39. 다음 중 버전 제어를 위해 Git을 사용하는 이점은 무엇입니까 (모두 선택)?
- a. 시간 경과에 따른 변경 사항 추적.
 - b. 여러 사용자 간의 협업 촉진.
 - c. 데이터 백업 자동화.
 - d. 코드 실행 속도 향상.
40. R 스크립트에서 `setwd()` 사용을 피하는 이유는 무엇입니까 (하나 선택)?
- a. 코드 실행 속도를 늦출 수 있습니다.
 - b. 관리자 권한이 필요합니다.
 - c. 코드를 덜 이식 가능하고 재현 가능하게 만듭니다.
 - d. 최근 R 버전에서 더 이상 사용되지 않습니다.
41. 재현성의 맥락에서 `renv` 패키지의 기능은 무엇입니까 (하나 선택)?
- a. 코드 청크를 병렬로 실행하기 위해.
 - b. 소프트웨어 환경을 문서화하고 공유하기 위해.
 - c. 코드 린팅을 자동화하기 위해.
 - d. 시뮬레이션에서 코드 효율성을 향상시키기 위해.
42. 다음 중 재현 가능한 워크플로우에 기여하지 않는 것은 무엇입니까 (하나 선택)?
- a. 작업 디렉토리를 설정하기 위해 `setwd()` 사용.
 - b. 결과뿐만 아니라 코드와 데이터 공유.
 - c. 논문에서 R 및 Python 코드를 통합하기 위해 Quarto 사용.
 - d. Git 및 GitHub를 사용한 버전 제어.
43. (`tidyversestyleguide`에?) 따르면, 다음 변수 이름 중 권장 스타일을 따르는 것은 무엇입니까 (하나 선택)?
- a. total-Sales
 - b. TotalSales
 - c. total_sales
 - d. total sales
44. R에서 `lintr` 패키지의 주요 기능은 무엇입니까 (하나 선택)?
- a. 알고리즘의 논리적 오류를 찾기 위해.
 - b. 코드를 더 빠르게 실행하기 위해.
 - c. 스타일 일관성을 위한 코드 린팅을 제공하기 위해.
 - d. 데이터 분포를 시각화하기 위해.
45. 코드 리팩토링은 무엇입니까 (하나 선택)?
- a. 오류를 수정하기 위해 코드를 디버깅하는 것.
 - b. 동작을 변경하지 않고 구조를 개선하기 위해 코드를 다시 작성하는 것.
 - c. 기존 코드에 새로운 기능을 추가하는 것.
 - d. 코드를 한 언어에서 다른 언어로 변환하는 것.
46. 코드에서 “매직 넘버”를 피해야 하는 이유는 무엇입니까 (하나 선택)?
- a. 실행 속도를 늦춥니다.
 - b. 코드 가독성 및 유지 보수성을 저하시킵니다.
 - c. 특정 소프트웨어와 호환되지 않습니다.
 - d. 구문 오류를 유발합니다.
47. 코드를 작성할 때 린터를 사용하는 주된 목적은 무엇입니까 (하나 선택)?

- a. 알고리즘의 논리적 오류를 찾기 위해.
 - b. 코드를 더 빠르게 실행하기 위해.
 - c. 코딩 스타일 지침을 적용하기 위해.
 - d. 코드를 기계어로 컴파일하기 위해.
48. 재현성의 맥락에서 “미래의 당신”은 무엇을 의미합니까 (하나 선택)?
- a. 자동화된 코드 생성.
 - b. 나중에 코드를 이해하고 재사용할 수 있는 능력.
 - c. 코드의 예측 분석.
 - d. 미래 동료와의 협업.

수업 활동

- 시작 폴더¹를 사용하여 새 저장소를 만드십시오. 수업의 공유 Google 문서에 GitHub 저장소 링크를 추가하십시오.
- Quarto를 사용하여 제목, 저자 및 초록이 있는 PDF를 만드십시오.²
- 세 섹션을 추가하고 palmerpenguins::penguins에 대해 종별 부리 길이 평균을 생성하는 코드를 추가하십시오 (코드 자체는 숨겨져 있음).
- R 및 palmerpenguins의 인용을 추가한 다음 성별 체질량 그래프를 추가하십시오.
- 그래프에 대한 텍스트 단락과 상호 참조를 추가하십시오. 또한 연도별 종 수에 대한 표를 추가하십시오.
- [강사는 (매우 천천히) 이 모든 것을 라이브 코딩하고 학생들이 코딩을 따라하도록 해야 합니다.] 로컬 컴퓨터에 git을 설정하십시오.³ GitHub 저장소를 만든 다음 로컬 복사본을 만들고 일부 변경 사항을 적용한 다음 푸시하십시오.⁴
- 파트너의 GitHub 저장소를 찾아 포크하고 변경 사항을 적용한 다음 풀 리퀘스트를 만드십시오.
- 다음 코드는 오류를 생성합니다. ?@sec-dealingwitherrors의 전략을 따라 수정하십시오.

```
#| eval: false
```

```
tibble(year = 1875:1972,
      level = as.numeric(datasets::LakeHuron)) |>
  ggplot(aes(x = year, y = level)) |>
  geom_point()
```

- 다음 코드는 오류를 생성합니다. ?@sec-dealingwitherrors의 전략을 따라 수정하십시오.

```
#| eval: false
```

```
tibble(year = 1871:1970,
      annual_nile_flow = as.character(datasets::Nile)) |>
  ggplot(aes(x = annual_nile_flow)) +
  geom_histogram()
```

- 다음 코드는 오류를 생성합니다. ?@sec-omgpleasemakeareprexplease를 따라 reprex를 만들고 (mtcars와 같은 더 일반적인 데이터 세트를 사용하도록 예제를 변경), GitHub Gist에 추가한 다음 강사에게 이메일을 보내십시오.

```
#| eval: false
```

```
tibble(year = 1875:1972,
      level = as.numeric(datasets::LakeHuron)) |>
  ggplot(aes(x = year, y = level)) |>
  geom_point()
```

¹https://github.com/RohanAlexander/starter_folder

²PDF를 로컬에서 설정하는 데 어려움을 겪는 소수의 학생들이 항상 있습니다. 최악의 경우, 다른 모든 것을 HTML로 로컬에서 수행한 다음 Posit Cloud에서 PDF를 빌드하십시오.

³GitHub 이메일을 숨겼다면 로컬에서 이메일 주소를 추가할 때 별칭을 사용해야 합니다.

⁴git이 로컬에서 작동하지 않는 소수의 학생들이 항상 있을 것입니다. 저는 가장 좋은 접근 방식은 시연하는 동안 고급 학생과 짹을 치어 분류하고, 문제가 남아 있으면 사무실 시간 동안 개별적으로 처리하는 것이라고 생각합니다.

- 다음 코드는 오류를 생성합니다. ChatGPT 또는 동등한 LLM을 사용하여 수정하십시오. 다음을 논의하십시오: 1) 프롬프트, 2) 수정된 코드.

```
#| eval: false

penguins |>
  ggplot(aes(x = bill_length_mm, y = bill_depth_mm, color = species)) |>
  geom_point()
```

과제 I

이 과제의 목적은 동료 검토를 주고받는 것입니다. 일반적으로 동료 검토, 특히 코드 검토 (Sadowski 기타 2018)는 전문가로서 일하는 데 중요한 부분입니다.

`usethis::git_vaccinate()`를 실행하여 시작하십시오. 그런 다음 장 ??의 활동에서 작업한 내용을 시작 폴더⁵를 사용하도록 업데이트하십시오. 여기에는 무엇보다도 다운로드 및 정리를 적절한 스크립트로 이동하고, README를 업데이트하고, 제목을 추가하는 등이 포함됩니다. 일반적으로 온라인 부록 ??의 Donaldson 논문 채점 기준표를 살펴보고, 추가 작업을 너무 많이 하지 않고 가능한 한 많이 준수하도록 노력해야 합니다. 그런 다음 다른 사람과 교환하십시오.

Google (2022) 및 Feldman (2024)를 읽으십시오. 그런 다음 GitHub Issues를 사용하여 저장소 내용에 대한 동료 검토를 수행하십시오. Feldman (2024)에 따라 동료 검토는 다음 구조를 사용하고 깔끔하게 서식이 지정되어야 합니다.

1. 요약 [검토 중인 원고에 대한 간략한 요약을 추가하십시오.]
2. 강력한 긍정적 측면: [간략하게 작성하십시오. 두세 개의 점을 찍으십시오.]
3. 필요한 중요한 개선 사항: [이것이 가장 중요한 섹션입니다. 이것은 논문 저자가 수정하고/하거나 해결해야 하는 문제입니다. 매우 건설적이고 정중하며, 부드럽지만 명확하게 작성하고, 저자를 돋기 위해 가능한 한 많은 정보를 제공하십시오. 여기에는 실수/오류, 누락된 정보, 간과, 오해 등이 포함될 수 있습니다. 가능하다면 왜 이러한 것들이 실수인지 설명하고, 수정 사항 또는 올바른 정보를 찾을 수 있는 링크를 제공하십시오.]
4. 개선을 위한 제안: [이것은 저자가 더 잘할 수 있도록 돋기 위한 것입니다. 당신이 확실하지 않은 것에 대해 언급하거나 의견을 전술하거나, 오탏 또는 사소한 코드 문제를 지적할 수 있지만, 이것이 제안임을 겸손하게 받아들이고 매우 긍정적이고 건설적으로 작성하십시오. 약 5~6개의 점이 있어야 합니다.]
5. 평가: [채점 기준표의 각 요소를 추가하고, 각 요소에 대한 의견과 점수를 제공하십시오. 이것은 채점에 사용되지 않으며, 저자에게 각 요소를 개선하기 위해 얼마나 많은 노력을 기울여야 하는지에 대한 아이디어를 제공하기 위한 것입니다.]
6. 예상 총점: [X] / [Y].
7. 기타 의견: [기타 의견.]

과제 II

이 과제의 목적은 다음을 편안하게 다루는 것입니다.

1. Quarto, 그리고
2. Git 및 GitHub.

웹사이트는 커뮤니케이션의 중요한 부분입니다. 예를 들어, 작업 포트폴리오를 공개적으로 사용할 수 있는 장소입니다. 웹사이트를 만드는 한 가지 방법은 Quarto의 내장 웹사이트를 사용하는 것입니다. RStudio에서 GitHub를 설정하면 약 5분 안에 웹사이트를 온라인에 게시할 수 있습니다.

새 프로젝트를 생성하여 시작하십시오 (“파일” -> “새 프로젝트” -> “새 디렉토리” -> “Quarto 웹사이트”). 이름을 지정하고 “새 세션에서 열기” -> “프로젝트 생성”을 선택하십시오 (그림 ??).

기본 웹사이트는 “빌드” -> “웹사이트 렌더링”으로 생성할 수 있습니다 (그림 ??). 기본적으로 “뷰어” 창에

⁵https://github.com/rohanalexander/starter_folder

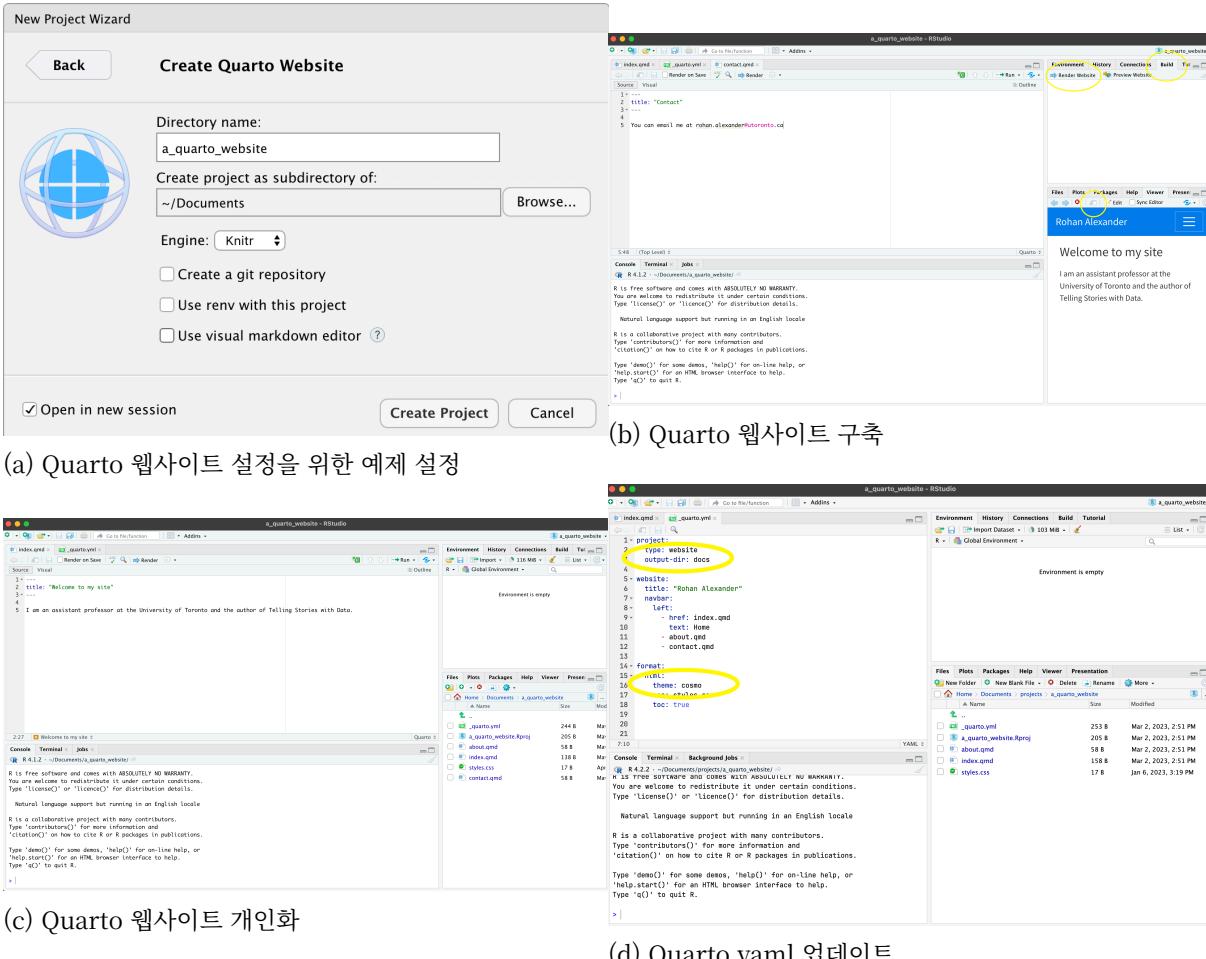


그림 4.1: Quarto를 사용하여 웹사이트 만들기

표시될 수 있지만, 새 창에서도 표시될 수 있습니다. 이 시점에서 우리는 자신의 세부 정보를 반영하도록 변경하고 싶을 수 있습니다. 특히, “index.qmd”的 제목을 변경하고 자신의 세부 정보를 추가하고 싶을 수 있습니다 (그림 ??).

기본 메뉴에 포함된 내용은 “_quarto.yml”에 지정됩니다. “contact.qmd”와 같은 다른 페이지를 추가할 수 있으며, 포함될 내용을 생성하려면 “about.qmd”를 복제한 다음 편집하고 싶을 수 있습니다 (그림 ??). “_quarto.yml”에서 변경할 수 있는 또 다른 측면은 테마입니다. 기본값은 “cosmo”이지만, 여기⁶에 지정된 다른 많은 옵션이 있습니다.

세부 정보가 개인화되고 웹사이트에 불만이 없으면 GitHub로 푸시한 다음 GitHub Pages로 호스팅할 수 있습니다. 이를 활용하려면 먼저 두 가지를 수행해야 합니다. 첫째, “_quarto.yml”을 약간 수정하여 “_site” 대신 “docs” 폴더로 빌드하도록 지정해야 합니다 (그림 ??).

```
#| eval: false
#| echo: true
```

```
project:
  type: website
  output-dir: docs
```

알아야 할 다른 측면은 이 서비스를 사용할 때 기본적으로 GitHub가 사이트를 빌드하려고 시도한다는 것

⁶<https://quarto.org/docs/output-formats/html-themes.html>

입니다. 이는 우리가 원하지 않는 것이므로, 먼저 숨겨진 파일을 추가하여 이를 끄려면 콘솔에서 다음을 실행해야 합니다.

```
#| eval: false  
#| echo: true  
  
file.create(".nojekyll")
```

그런 다음 GitHub가 설정되었다고 가정하고 `usethis`를 사용하여 새로 생성된 프로젝트를 GitHub에 올릴 수 있습니다. `use_git()`을 사용하여 Git 저장소를 초기화한 다음 `use_github()`가 GitHub로 푸시합니다.

```
#| eval: false  
#| echo: true  
  
use_git()  
use_github()
```

프로젝트는 GitHub에 있을 것입니다. GitHub Pages를 사용하여 호스팅할 수 있습니다: “설정 -> 페이지”로 이동한 다음 설정에 따라 소스를 “main” 또는 “master”로 변경하고 마지막으로 “docs”로 변경합니다. 몇 분 동안 다양한 검사를 실행한 후 GitHub는 사이트를 방문할 수 있는 주소를 알려줄 것입니다.

사이트를 업데이트하려면 로컬에서 작업하십시오. 먼저 풀하여 GitHub에서 변경된 내용이 로컬에 있는지 확인한 다음 사이트를 편집하고 다시 렌더링한 다음 일반적인 방식으로 GitHub에 푸시하십시오. 검사가 완료되면 라이브 웹사이트가 업데이트됩니다.

웹사이트에 일반 텍스트 단락, 섹션 제목, 글머리 기호가 포함되어 있는지 확인하십시오. 모든 것이 잘 문서화되고 깔끔하게 서식이 지정되어야 하며 전반적으로 고품질이어야 합니다.

관련된 채점 기준표 구성 요소는 “수업 논문”, “LLM 사용 문서화”, “산문”, “커밋”, “재현 가능한 워크플로우”입니다. 웹사이트 링크를 제출하십시오.

논문

이 시점에서 온라인 부록 “논문”⁷의 Donaldson 논문이 적절할 것입니다.

⁷<https://tellingstorieswithdata.com/23-assessment.html>

제 II 편

소통

5

연구 작성

- i** Chapman and Hall/CRC는 이 책을 2023년 7월에 출판했습니다. 여기^a에서 구매할 수 있습니다. 이 온라인 버전은 인쇄된 내용에 일부 업데이트가 있습니다.

^a<https://www.routledge.com/Telling-Stories-with-Data-With-Applications-in-R/Alexander/p/book/9781032134772>

선행 조건

- By Design: Planning Research on Higher Education 읽기, (Light, Singer, 와/과 Willett 1990)
 - 좋은 연구 질문을 개발하기 위한 전략을 제공하는 2장 “당신의 질문은 무엇인가”에 집중하십시오.
- On Writing Well 읽기 (어떤 판이든 상관없음), (Zinsser 1976)
 - 특히 효과적인 글쓰기 스타일에 대한 “방법”을 제공하는 1부 “원칙”과 2부 “방법”에 집중하십시오.
- Novelist Cormac McCarthy's tips on how to write a great science paper 읽기, (Savage 와/과 Yeh 2019)
 - 이 논문은 글쓰기를 개선할 수 있는 구체적인 팁을 제공합니다.
- Publication, publication 읽기, (G. King 2006)
 - 이 논문은 복제에서 출판 가능한 학술 논문으로 나아가는 전략을 자세히 설명합니다.
- Quantitative Editing 시청, (Bronner 2021)
 - 이 비디오는 FiveThirtyEight의 정량 편집자로서의 경험을 바탕으로 정량 기반 글쓰기 전략을 제공합니다.
- Smoking and carcinoma of the lung 읽기, (Doll 와/과 Hill 1950)
 - 이 논문은 데이터 섹션의 훌륭한 예시를 제공합니다.
- How to write usefully 읽기 (Graham 2020)
 - 독자가 이미 알지 못하는 진실하고 중요한 것을 쓰는 것에 대한 블로그 게시물입니다.
- 다음 잘 쓰여진 정량 논문 중 하나를 읽으십시오.
 - Asset prices in an exchange economy, (R. Lucas 1978)
 - Individuals, institutions, and innovation in the debates of the French Revolution, (Barron 기타 2018)
 - Modeling: optimal marathon performance on the basis of physiological factors, (Joyner 1991)
 - On reproducible econometric research, (Koenker 와/과 Zeileis 2009)
 - Prevented mortality and greenhouse gas emissions from historical and projected nuclear power, (Kharecha 와/과 Hansen 2013)
 - Seeing like a market, (Fourcade 와/과 Healy 2017)
 - Simpson's paradox and the hot hand in basketball, (Wardrop 1995)
 - Some studies in machine learning using the game of checkers, (Samuel 1959)
 - Statistical methods for assessing agreement between two methods of clinical measurement, (Bland 와/과 Altman 1986)
 - Surgical Skill and Complication Rates after Bariatric Surgery, (Birkmeyer 기타 2013)
 - The mundanity of excellence: An ethnographic report on stratification and Olympic swimmers, (Chambliss 1989)
 - The probable error of a mean, (Student 1908)
- 다음 The New Yorker 기사 중 하나를 읽으십시오.
 - Funny Like a Guy, Tad Friend, 2011년 4월 4일
 - Going the Distance, David Remnick, 2014년 1월 19일
 - How the First Gravitational Waves Were Found, Nicola Twilley, 2016년 2월 11일

- Happy Feet, Alexandra Jacobs, 2009년 9월 7일
- Levels of the Game, John McPhee, 1969년 5월 31일
- Reporting from Hiroshima, John Hersey, 1946년 8월 23일
- The Catastrophist, Elizabeth Kolbert, 2009년 6월 22일
- The Quiet German, George Packer, 2014년 11월 24일
- The Pursuit of Beauty, Alec Wilkinson, 2015년 2월 1일
- 다음 다른 출판물의 기사 중 하나를 읽으십시오.
 - Blades of Glory, Holly Anderson, Grantland
 - Born to Run, Walt Harrington, The Washington Post
 - Dropped, Jason Fagone, Grantland
 - Federer as Religious Experience, David Foster Wallace, The New York Times Magazine
 - Generation Why?, Zadie Smith, The New York Review of Books
 - One hundred years of arm bars, David Samuels, Grantland
 - Out in the Great Alone, Brian Phillips, ESPN
 - Pearls Before Breakfast, Gene Weingarten, The Washington Post
 - Resurrecting The Champ, J.R. Moehringer, Los Angeles Times
 - The Cult of "Jurassic Park", Bryan Curtis, Grantland
 - The House that Hova Built, Zadie Smith, The New York Times
 - The Re-Education of Chris Copeland, Flinder Boyd, SB Nation
 - The Sea of Crisis, Brian Phillips, Grantland
 - The Webb Space Telescope Will Rewrite Cosmic History. If It Works., Natalie Wolchover, Quanta Magazine

주요 개념 및 기술

- 글쓰기는 데이터를 분석하는 데 필요한 모든 기술 중 가장 중요한 기술일 수 있습니다. 글쓰기를 잘하는 유일한 방법은 매일 글을 쓰는 것입니다.
- 글을 쓸 때, 이점은 일반적으로 우리 자신에게 돌아오지만, 독자를 위해 글을 써야 합니다. 이는 우리가 전달하고자 하는 하나의 주요 메시지를 가지고, 우리가 어디에 있는지보다는 독자가 어디에 있는지 생각하는 것을 의미합니다.
- 가능한 한 빨리 초고를 작성하고 싶습니다. 아무리 형편없더라도 초고가 존재하는 것과 존재하지 않는 것의 차이는 엄청납니다. 그 시점에서 우리는 다시 쓰기 시작합니다. 그렇게 할 때 불필요한 단어를 제거하여 명확성을 극대화하는 것을 목표로 합니다.
- 우리는 일반적으로 관심 분야에서 시작하여 연구 질문, 데이터 세트 및 분석을 반복적인 방식으로 개발합니다. 이 과정을 통해 우리가 무엇을 하고 있는지 더 잘 이해하게 됩니다.

소프트웨어 및 패키지

- knitr (Xie 2023)
- tidyverse (Wickham 기타 2019)
- tinytable (Arel-Bundock 2024)

```
library(knitr)
library(tidyverse)
library(tinytable)
```

5.1 서론

작가가 되고 싶다면, 무엇보다 두 가지를 해야 합니다. 많이 읽고 많이 써야 합니다. 제가 아는 한 이 두 가지를 피할 방법은 없습니다. 지름길은 없습니다.

S. King (2000, p. 145)

우리는 주로 데이터를 사용하여 이야기를 글로 전달합니다. 글쓰기는 효율적으로 소통할 수 있게 해줍니다. 또한 우리가 무엇을 믿는지 알아내는 방법이며, 우리의 아이디어에 대한 피드백을 받을 수 있게 해줍니다. 효과적인 논문은 간결하게 작성되고 잘 정리되어 있어 이야기가 잘 흐르도록 합니다. 적절한 문장 구조, 철자, 어휘 및 문법은 주의를 분산시키지 않고 이야기의 각 측면을 명확하게 표현할 수 있도록 해주기 때문에 중요합니다.

이 장은 글쓰기에 관한 것입니다. 이장을 마치면, 당신이 원하는 것을 전달하고 독자의 시간을 낭비하지 않는 짧고 상세한 정량 논문을 작성하는 방법에 대해 더 잘 알게 될 것입니다. 우리는 우리 자신을 위해서가 아니라 독자를 위해 글을 씁니다. 특히, 우리는 독자에게 유용하도록 글을 씁니다. 이는 새롭고, 진실하며, 중요한 것을 명확하게 전달하는 것을 의미합니다 (Graham 2020). 그렇다고 해도, 글쓰기의 가장 큰 이점은 종종 작가에게 돌아옵니다. 우리가 독자를 위해 글을 쓸 때도 마찬가지입니다. 이는 글쓰기 과정이 우리가 무엇을 생각하고 어떻게 그것을 믿게 되었는지 알아내는 방법이기 때문입니다.

이 장의 측면들은 목록처럼 느껴질 수 있습니다. 처음에는 이러한 측면들을 빠르게 훑어보고, 필요할 때 다시 돌아오는 것이 유용할 수 있습니다.

5.2 글쓰기

글을 쓰는 방법은 세 번 또는 네 번 반복하는 것이지, 한 번에 끝내는 것이 아닙니다. 저에게 가장 어려운 부분은 먼저, 무엇이든 간에 제 앞에 내놓는 것입니다. 때로는 초조한 광란 속에서 벽에 진흙을 던지듯이 단어를 던지기도 합니다. 무엇이든 간에, 초고로 내뱉고, 쏟아내고, 지껄이십시오.

McPhee (2017, p. 159)

글쓰기 과정은 다시 쓰기 과정입니다. 중요한 과제는 가능한 한 빨리 초고를 작성하는 것입니다. 완전한 초고가 존재하기 전까지는 아무리 나빠 보이더라도 작성된 내용을 삭제하거나 수정하지 않는 것이 유용합니다. 그냥 쓰십시오. (이 조언은 경험이 적은 작가에게 해당됩니다. 경험이 쌓이면 접근 방식이 바뀔 수 있습니다.)

가장 위협적인 단계 중 하나는 빈 페이지이며, 우리는 “서론”, “데이터”, “모델”, “결과”, “논의”와 같은 제목을 즉시 추가하여 이를 처리합니다. 그리고 “제목”, “날짜”, “저자”, “초록”과 같이 필요한 다양한 항목에 대한 필드를 상단 내용에 추가합니다. 이렇게 하면 일반적인 개요가 생성되며, 이는 논문의 mise en place 역할을 할 것입니다. 배경 설명을 하자면, mise en place는 전문 주방에서 재료를 분류하고 준비하여 쉽게 접근할 수 있도록 배열하는 준비 단계입니다. 이는 불필요한 지연 없이 필요한 모든 것을 사용할 수 있도록 보장합니다. 개요를 작성하는 것은 정량 논문을 작성할 때 동일한 역할을 하며, 저녁 식사를 준비하는데 사용할 재료를 카운터에 놓는 것과 유사합니다 (McPhee 2017).

이 일반적인 개요를 설정한 후, 연구 질문에 대해 깊이 생각함으로써 우리가 탐구하는 것에 대한 이해를 발전시켜야 합니다. 이론적으로는 연구 질문을 개발하고, 답하고, 모든 글쓰기를 수행하지만, 실제로는 거의 그렇게 되지 않습니다 (Franklin 2005). 대신, 우리는 일반적으로 질문과 답변의 형태에 대한 아이디어를 가지고 있으며, 글을 쓰면서 이러한 아이디어가 덜 모호해집니다. 이는 글쓰기 과정을 통해 우리의 사고를 다듬기 때문입니다 (S. King 2000, p. 131). 연구 질문에 대한 생각을 정리한 후, 각 섹션에 점을 추가하고, 필요에 따라 정보가 담긴 하위 제목으로 하위 섹션을 추가할 수 있습니다. 그런 다음 해당 점들을 단락으로 확장합니다. 이 작업을 수행하는 동안 우리의 사고는 다른 연구자들의 웹뿐만 아니라 우리의 상황과 환경과 같은 다른 측면에도 영향을 받습니다 (Latour 1996).

초고를 작성하는 동안 당신이 충분히 잘하지 못하거나 불가능하다는 느낌을 무시해야 합니다. 그냥 쓰십시오. 당신은 종이에 단어가 필요합니다. 아무리 나쁘더라도 말입니다. 그리고 초고는 당신이 이것을 달성하는 때입니다. 주의를 분산시키는 것을 제거하고 글쓰기에 집중하십시오. 완벽주의는 적이며, 제쳐두어야 합니다. 때로는 매우 일찍 일어나 글을 쓰거나, 마감일을 정하거나, 글쓰기 그룹을 형성하여 이를 달성할 수 있습니다. 긴급성을 조성하는 것이 유용할 수 있으며, 한 가지 옵션은 진행하면서 적절한 인용을 추가하는 것에 신경 쓰지 않는 것입니다. 이는 속도를 늦출 수 있으며, 대신 “[TODO: 여기에 R 인용 추가]”와 같은 것을 추가하는 것입니다. 그래프와 표도 마찬가지입니다. 즉, 실제 그래프와 표 대신 “[TODO: 여기에 각 국가의 시간 경과에 따른 그래프 추가]”와 같은 텍스트 설명을 포함하십시오. 아무리 나쁘더라도 내용을 추가하는 데 집중하십시오. 이 모든 것이 완료되면 초고가 존재합니다.

이 초고는 형편없이 작성되었고 훌륭함과는 거리가 멀 것입니다. 그러나 나쁜 초고를 작성함으로써 좋은 두 번째 초고, 훌륭한 세 번째 초고, 그리고 결국에는 탁월함에 도달할 수 있습니다 (Lamott 1994, p. 20). 그 초고는 너무 길고, 이해가 되지 않을 것이며, 지지할 수 없는 주장과 해서는 안 되는 주장을 포함할 것입니다. 초고에 대해 부끄럽지 않다면, 충분히 빨리 작성하지 않은 것입니다.

“삭제” 키와 “잘라내기” 및 “붙여넣기”를 광범위하게 사용하여 초고를 두 번째 초고로 바꾸십시오. 초고를 인쇄하고 빨간 펜을 사용하여 단어, 문장 및 전체 단락을 이동하거나 제거하는 것이 특히 도움이 됩니다. 초고에서 두 번째 초고로 넘어가는 과정은 이야기의 흐름과 일관성을 돋기 위해 한 번에 수행하는 것이 가장 좋습니다. 이 첫 번째 다시 쓰기의 한 가지 측면은 우리가 전달하고자 하는 이야기를 향상시키는 것입니다. 또 다른 측면은 이야기가 아닌 모든 것을 제거하는 것입니다 (S. King 2000, p. 57).

초고가 되어가는 내용에 잘 맞지 않더라도 좋아 보이는 작업을 제거하는 것은 고통스러울 수 있습니다. 이 고통을 덜어주는 한 가지 방법은 임시 문서, 예를 들어 “debris.qmd”를 만들어 원치 않는 단락을 즉시 삭제하는 대신 저장하는 것입니다. 또 다른 전략은 단락을 주석 처리하는 것입니다. 그렇게 하면 원본 파일을 계속 볼 수 있고 유용할 수 있는 측면을 알아차릴 수 있습니다.

각 섹션에 작성된 내용을 검토하면서 발전하는 이야기를 뒷받침하는 방식에 특별히 주의를 기울여 의미를 부여하십시오. 이 수정 과정은 글쓰기의 본질입니다 (McPhee 2017, p. 160). 또한 참조를 수정하고 실제 그래프와 표를 추가해야 합니다. 이 다시 쓰기 과정의 일부로 논문의 핵심 메시지가 발전하고 연구 질문에 대한 답변이 더 명확해지는 경향이 있습니다. 이 시점에서 서론과 같은 측면으로 돌아갈 수 있으며, 마지막으로 초록으로 돌아갈 수 있습니다. 오탏 및 기타 문제는 작업의 신뢰성에 영향을 미칩니다. 따라서 두 번째 초고의 일부로 수정해야 합니다.

이 시점에서 초고는 의미를 갖기 시작합니다. 이제 그것을 훌륭하게 만드는 것이 목표입니다. 인쇄하여 다시 종이로 검토하십시오. 이야기에 기여하지 않는 모든 것을 제거하십시오. 이 단계쯤 되면 논문에 너무 가까워지기 시작할 수 있습니다. 이는 다른 사람에게 의견을 요청할 좋은 기회입니다. 이야기의 약점에 대해 피드백을 요청하십시오. 이를 해결한 후에는 논문을 다시 한 번 읽어보는 것이 도움이 될 수 있습니다. 이번에는 소리 내어 읽으십시오. 논문은 결코 “완료”되지 않으며, 특정 시점에서 시간이 다 떨어지거나 지겨워지는 경우가 더 많습니다.

5.3 질문하기

질적 접근 방식과 양적 접근 방식 모두 제자리를 찾고 있습니다. 이 책에서는 양적 접근 방식에 중점을 둡니다. 그럼에도 불구하고 질적 연구는 중요하며, 종종 가장 흥미로운 작업은 두 가지 모두를 포함합니다. 양적 분석을 수행할 때 우리는 데이터 품질, 측정 및 관련성과 같은 문제에 직면합니다. 우리는 종종 인과 관계를 파악하려고 노력하는 데 특히 관심이 있습니다. 그럼에도 불구하고 우리는 세상에 대해 무언가를 배우려고 노력합니다. 우리의 연구 질문은 이 모든 것을 고려해야 합니다.

대략적으로, 그리고 단순화의 위험을 무릅쓰고, 연구를 수행하는 두 가지 방법이 있습니다.

- 1) 데이터 우선; 또는
- 2) 질문 우선.

그러나 이는 이진법이 아니며, 종종 연구는 연구 퍼즐을 중심으로 데이터와 질문 사이를 반복적으로 진행됩니다 (Gustafsson 와/과 Hagström 2017). Light, Singer, 와/과 Willett (1990, p. 39)는 이 접근 방식을 이론 → 데이터 → 이론 → 데이터 등의 나선형으로 설명합니다. 예를 들어, 질문 우선 접근 방식은 이

론 중심 또는 데이터 중심일 수 있으며, 데이터 우선 접근 방식도 마찬가지입니다. 대안적인 틀은 귀납적, 즉 특정에서 일반으로의 접근 방식과 연역적, 즉 일반에서 특정으로의 접근 방식을 비교하는 것입니다.

두 가지 예를 고려해 봅시다.

1. Mok 기타 (2022) 은 100,000명의 Spotify 사용자로부터 80억 개의 고유한 청취 이벤트를 조사하여 사용자가 콘텐츠를 탐색하는 방법을 이해합니다. 그들은 나이와 행동 사이에 명확한 관계가 있음을 발견했습니다. 젊은 사용자는 더 다양한 소비에도 불구하고 나이든 사용자보다 알려지지 않은 콘텐츠를 덜 탐색합니다. 발견 및 탐색에 대한 연구 질문이 이 논문을 이끌고 있다는 것은 분명하지만, 이 데이터 세트에 대한 접근 없이는 불가능했을 것입니다. 궁극적인 일치가 이루어지기 전에 잠재적인 연구 질문과 잠재적인 데이터 세트가 고려되는 반복적인 과정이 있었을 것입니다.
2. ?@sec-fire-hose에서 소개된 신생아 사망률(NMR)을 탐색하고 싶다고 생각해 봅시다. 20년 후 사하라 이남 아프리카에서 NMR이 어떻게 보일지 궁금할 수 있습니다. 이것은 질문 우선 접근 방식이 될 것입니다. 그러나 이 안에는 다음과 같은 것들이 있을 수 있습니다. 다른 양과의 생물학적 관계를 기반으로 무엇을 기대하는지와 같은 이론 중심적인 측면; 또는 예측을 하기 위해 가능한 한 많은 데이터를 수집하는 것과 같은 데이터 중심적인 측면. 대안적인, 순전히 데이터 중심적인 접근 방식은 NMR에 접근한 다음 가능한 것을 알아내는 것입니다.

5.3.1 데이터 우선

데이터 우선일 때, 주요 문제는 사용 가능한 데이터로 합리적으로 답변할 수 있는 질문을 알아내는 것입니다. 이러한 질문을 결정할 때 다음을 고려하는 것이 유용합니다.

- 1) 이론: 인과 관계를 합리적으로 결정할 수 있는 기대가 있습니까? 예를 들어, 마크 크리스텐슨은 주식 시장을 차트화하는 질문이라면 오디세이로 돌아가 볼 위에서 황소 내장을 읽는 것이 더 나을 수 있다고 농담하곤 했습니다. 적어도 그렇게 하면 하루가 끝날 때 먹을 것이 있을 것이기 때문입니다. 질문은 일반적으로 허위 관계를 피하는 데 도움이 되는 그럴듯한 이론적 근거를 가져야 합니다. 주어진 데이터로 이론을 개발하는 한 가지 방법은 “이것은 무엇의 사례인가?”를 고려하는 것입니다 (Rosenau 1999, p. 7). 그 접근 방식을 따르면, 특정 설정을 넘어 일반화하려고 노력합니다. 예를 들어, 특정 내전을 모든 내전의 사례로 생각하는 것입니다. 이것의 이점은 이론 구축에 필요한 일반적인 속성에 주의를 집중시킨다는 것입니다.
- 2) 중요성: 답변할 수 있는 사소한 질문은 많지만, 우리의 시간이나 독자의 시간을 낭비하지 않는 것이 중요합니다. 중요한 질문을 갖는 것은 데이터 정리 및 코드 디버깅에 4주 연속으로 시간을 보내는 것과 같은 상황에서 동기 부여에 도움이 될 수 있습니다. 산업에서는 재능 있는 직원과 자금을 유치하는 데 더 쉽게 만들 수도 있습니다. 그렇다고 해도 균형이 필요합니다. 질문은 답변될 가능성이 높아야 합니다. 세대를 정의하는 질문을 공격하는 것은 여러 부분으로 나누는 것이 가장 좋습니다.
- 3) 가용성: 미래에 추가 데이터가 제공될 합리적인 기대가 있습니까? 이는 관련 질문에 답변하고 하나님의 논문을 연구 의제로 바꿀 수 있게 해줍니다.
- 4) 반복: 이것은 여러 번 실행할 수 있는 것입니까, 아니면 일회성 분석입니까? 전자라면 특정 연구 질문에 답변하기 시작하고 반복할 수 있습니다. 그러나 데이터에 한 번만 접근할 수 있다면 더 광범위한 질문에 대해 생각해야 합니다.

샤오리 멩에게 귀속되는 속담이 있습니다. 모든 통계는 결측 데이터 문제라는 것입니다. 따라서 역설적으로, 데이터 우선 질문을 하는 또 다른 방법은 우리가 가지고 있지 않은 데이터에 대해 생각하는 것입니다. 예를 들어, 이전에 논의된 신생아 및 산모 사망률 예시로 돌아가면 한 가지 문제는 완전한 사망 원인 데이터가 없다는 것입니다. 만약 있다면, 관련 사망자 수를 셀 수 있을 것입니다. ((Castro2023은?) 이 단순한 가설이 실제로는 복잡할 것이라고 상기시킵니다. 왜냐하면 다른 원인과 독립적이지 않은 사망 원인이 때때로 있기 때문입니다.) 결측 데이터 문제가 설정되면 데이터 중심 접근 방식을 취할 수 있습니다. 우리는 우리가 가지고 있는 데이터를 살펴보고, 가상의 데이터 세트를 근사화하는 데 사용할 수 있는 정도를 나타내는 연구 질문을 합니다.

i 거인의 어깨

샤오리 멍은 하버드 대학교의 휘플 V. N. 존스 통계학 교수입니다. 1990년 하버드 대학교에서 통계학 박사 학위를 취득한 후 시카고 대학교에서 조교수로 임명되었고 2000년에 교수로 승진했습니다. 2001년 하버드로 옮겨 2004년부터 2012년까지 통계학과장으로 역임했습니다. 그는 결측 데이터(Meng (1994) 및 Meng (2012)) 및 데이터 품질(Meng (2018))을 포함한 광범위한 주제에 대해 출판했습니다. 그는 2001년 COPSS 회장상을 수상했습니다.

일부 연구자들이 데이터 우선인 한 가지 방법은 특정 지리적 또는 역사적 상황의 데이터에 대한 특정 전문 지식을 개발하는 것입니다. 예를 들어, 그들은 현재의 영국 또는 19세기 후반의 일본에 대해 특히 잘 알고 있을 수 있습니다. 그런 다음 다른 연구자들이 다른 상황에서 묻는 질문을 살펴보고, 그 질문에 자신의 데이터를 가져옵니다. 예를 들어, 특정 질문이 처음에 미국에 대해 질문되고, 그 다음 많은 연구자들이 영국, 캐나다, 호주 및 기타 여러 국가에 대해 동일한 질문에 답변하는 것을 흔히 볼 수 있습니다.

데이터 우선 연구에는 특히 불확실성이 크다는 점을 포함하여 여러 가지 단점이 있습니다. 또한 선택 효과에 대한 우려가 항상 있기 때문에 외부 타당성 확보에 어려움을 겪을 수 있습니다.

데이터 중심 연구의 변형은 모델 중심 연구입니다. 여기서는 연구자가 특정 통계적 접근 방식에 대한 전문가가 된 다음 해당 접근 방식을 적절한 맥락에 적용합니다.

5.3.2 질문 우선

질문 우선을 시도할 때, 데이터 가용성에 대한 역방향 문제가 있습니다. “FINER 프레임워크”는 의학에서 연구 질문 개발을 안내하는 데 사용됩니다. 이는 다음과 같은 질문을 할 것을 권장합니다: 실현 가능하고(Feasible), 흥미롭고(Interesting), 새롭고(Novel), 윤리적이며(Ethical), 관련성 있는(Relevant) 질문(Hulley 기타 2007). (farrugia2010research는?) FINER에 PICOT을 추가하여 인구(Population), 개입(Intervention), 비교 그룹(Comparison group), 관심 결과(Outcome of interest), 시간(Time)과 같은 추가 고려 사항을 권장합니다.

질문을 작성하는 것이 압도적으로 느껴질 수 있습니다. 한 가지 방법은 매우 구체적인 질문을 하는 것입니다. 다른 방법은 기술적, 예측적, 추론적 또는 인과적 분석에 관심이 있는지 결정하는 것입니다. 그러면 이러한 것들은 다른 유형의 질문으로 이어집니다. 예를 들어:

- 기술적 분석: “ x 는 어떻게 생겼습니까?”;
- 예측 분석: “ x 에 무슨 일이 일어날까요?”;
- 추론: “ x 를 어떻게 설명할 수 있습니까?”; 그리고
- 인과: “ x 가 y 에 어떤 영향을 미칩니다?”.

이러한 각각은 역할을 합니다. 신뢰성 혁명 (Angrist 와/과 Pischke 2010) 이후, 특정 접근 방식으로 답변된 인과적 질문이 지배적이었습니다. 이는 일부 이점을 가져왔지만, 비용이 없지는 않았습니다. 기술적 분석은 때로는 더 많은 통찰력을 제공할 수 있으며, 중요합니다 (Sen 1980). 질문의 본질은 그것에 답변하는데 진정으로 관심이 있는지 여부보다 덜 중요합니다.

시간은 종종 제한될 것이며, 흥미로운 방식으로 제한될 수 있으며, 이는 연구 질문의 세부 사항을 안내할 수 있습니다. 유명인의 발표가 주식 시장에 미치는 영향에 관심이 있다면, 발표 전후의 주가를 살펴보면 됩니다. 그러나 암 치료제가 장기적인 결과에 미치는 영향에 관심이 있다면 어떨까요? 효과가 20년이 걸린다면, 우리는 한동안 기다리거나 20년 전에 치료받은 사람들을 살펴봐야 합니다. 그러면 오늘 약을 투여하는 경우와 비교하여 선택 효과와 다른 상황이 발생합니다. 종종 합리적인 유일한 방법은 통계 모델을 구축하는 것이지만, 이는 다른 문제를 야기합니다.

5.4 질문에 답하기

5.4.1 반사실과 편향

질문에 답할 때 반사실을 만드는 것이 종종 중요합니다. 반사실은 “만약”이 거짓인 “만약-그러면” 문장입니다. 루이스 캐럴의 거울 나라의 앤리스에 나오는 험프티 덤프티의 예를 생각해 봅시다.

“정말 쉬운 수수께끼를 내는군요!” 험프티 덤프티가 으르렁거렸다. “물론 그렇게 생각하지 않습니다! 만약 제가 떨어졌다면—그럴 리는 없지만—만약 떨어졌다면—” 여기서 그는 입술을 오므리고 너무나 진지하고 위엄 있게 보여 앤리스는 웃음을 참을 수 없었다. “만약 제가 떨어졌다면,” 그가 계속했다. “왕께서 저에게 약속하셨습니다—그의 입으로 직접—”

Carroll (1871)

험프티는 자신이 결코 떨어지지 않을 것이라고 확신하면서도, 만약 떨어졌을 때 일어날 일에 만족합니다. 질문에 대한 답을 결정하는 것은 종종 이 비교 그룹입니다. 예를 들어, ?@sec-causality-from-observational-data에서는 VO₂ max가 사이클 선수의 경주 우승 확률에 미치는 영향을 고려합니다. 일반 인구에 걸쳐 비교하면 중요한 변수입니다. 그러나 잘 훈련된 운동선수에게만 비교하면 선택 때문에 덜 중요합니다.

연구 질문을 결정할 때 특히 주의해야 할 데이터의 두 가지 측면은 선택 편향과 측정 편향입니다.

선택 편향은 결과가 표본에 누가 포함되는지에 따라 달라질 때 발생합니다. 선택 편향의 해로운 측면 중 하나는 우리가 그것의 존재를 알아야만 그것에 대해 조치를 취할 수 있다는 것입니다. 그러나 많은 기본 진단은 선택 편향을 식별하지 못합니다. ?@sec-hunt-data에서 논의하는 A/B 테스트에서 A/A 테스트는 그룹을 만들고 처치를 적용하기 전에 비교하는 약간의 변형입니다 (따라서 A/A 명명법). 그룹이 처음부터 동일한지 확인하려는 이러한 노력은 선택 편향을 식별하는 데 도움이 될 수 있습니다. 더 일반적으로, 연령 그룹, 성별, 교육과 같은 표본의 속성을 모집단의 특성과 비교하는 것도 도움이 될 수 있습니다. 그러나 선택 편향과 관측 데이터의 근본적인 문제는 우리가 데이터가 있는 사람들은 데이터가 없는 사람들과 적어도 한 가지 면에서 다르다는 것을 알고 있다는 것입니다! 그러나 다른 어떤 면에서 다를 수 있는지는 알 수 없습니다.

선택 편향은 분석의 많은 측면에 스며들 수 있습니다. 처음에는 대표성이 있는 표본도 시간이 지남에 따라 편향될 수 있습니다. 예를 들어, 장 ??에서 논의하는 설문조사 패널은 응답하지 않는 사람들이 생기기 때문에 주기적으로 업데이트해야 합니다.

주의해야 할 또 다른 편향은 측정 편향입니다. 이는 데이터가 수집된 방식에 따라 결과가 영향을 받을 때 발생합니다. 이에 대한 일반적인 예는 응답자에게 소득을 물어보면 온라인 설문조사와 비교하여 직접 대면 설문조사에서 다른 답변을 얻을 수 있다는 것입니다.

5.4.2 추정량

우리는 일반적으로 데이터를 사용하여 질문에 답하는 데 관심이 있으며, 세부 사항에 대해 명확하게 하는 것이 중요합니다. 예를 들어, 흡연이 기대 수명에 미치는 영향에 관심이 있을 수 있습니다. 이 경우, 우리가 결코 알 수 없는 어떤 진정한 효과가 있으며, 그 진정한 효과를 “추정량”이라고 합니다 (Little 와/과 Lewis 2021). 논문의 어느 시점에서, 이상적으로는 서론에서 추정량을 정의하는 것이 중요합니다 (Lundberg, Johnson, 와/과 Stewart 2021). 이는 분석 계획의 특정 측면을 약간 변경하여 우연히 다른 것을 추정하게 되는 것이 쉽기 때문입니다 (Kahan 기타 2022). 일부 의약품 규제 기관에서는 이를 요구하기 시작했습니다 (Kahan 기타 2024). 추정량의 경우 효과가 무엇을 나타내는지에 대한 명확한 설명이 필요합니다 (Kahan 기타 2023). “추정기”는 우리가 가지고 있는 데이터를 사용하여 “추정량”的 “추정치”를 생성하는 과정입니다. Efron 와/과 Morris (1977) 는 추정기 및 관련 문제에 대한 논의를 제공합니다.

Bueno de Mesquita 와/과 Fowler (2021, p. 94)는 추정치와 추정량 간의 관계를 다음과 같이 설명합니다.

$$\text{추정치} = \text{추정량} + \text{편향} + \text{노이즈}$$

편향은 추정기가 추정량과 다른 추정치를 체계적으로 제공하는 문제를 의미하며, 노이즈는 비체계적인 차이를 의미합니다. 예를 들어, 표준 정규 분포를 고려해 봅시다. 우리는 평균을 이해하는 데 관심이 있을 수 있으며, 이는 우리의 추정량이 될 것입니다. 우리는 (실제 데이터에서는 결코 알 수 없는 방식으로) 추정량이 0이라는 것을 알고 있습니다. 그 분포에서 10번을 추출해 봅시다. 추정치를 생성하는 데 사용할 수 있는

표 5.1: 추출 횟수 증가에 따른 무작위 추출 평균의 두 가지 추정기 결과 비교

추출 횟수	추정기 1	추정기 2
10	-	-
	0.58	0.82
100	-0.06	-0.07
1,000	0.06	0.04
10,000	-0.01	-0.01

한 가지 추정기는 추출값을 합산하고 추출 횟수로 나누는 것입니다. 다른 하나는 추출값을 정렬하고 중간 관측값을 찾는 것입니다. 더 구체적으로, 이 상황을 시뮬레이션해 보겠습니다(표 ??).

```
set.seed(853)

tibble(
  num_draws = c(
    rep(10, times = 10),
    rep(100, times = 100),
    rep(1000, times = 1000),
    rep(10000, times = 10000)
  ),
  draw = rnorm(
    n = length(num_draws),
    mean = 0,
    sd = 1
  ) |>
  summarise(
    estimator_one = sum(draw) / unique(num_draws),
    estimator_two = sort(draw)[round(unique(num_draws) / 2, 0)],
    .by = num_draws
  ) |>
  tt() |>
  style_tt(j = 2:3, align = "r") |>
  format_tt(digits = 2, num_mark_big = ",", num_fmt = "decimal") |>
  setNames(c(" 추출 횟수", " 추정치 1", " 추정치 2"))
```

추출 횟수가 증가함에 따라 노이즈의 영향이 제거되고, 우리의 추정치는 추정치의 편향을 보여줍니다. 이 예시에서는 진실이 무엇인지 알지만, 실제 데이터를 고려할 때는 무엇을 해야 할지 알기 더 어려울 수 있습니다. 따라서 추정치를 생성하기 전에 추정량이 무엇인지 명확히 하는 것이 중요합니다.

5.4.3 방향성 비순환 그래프

질문에 답하는 데 사용할 변수에 대해 생각할 때, 우리가 의미하는 바를 구체적으로 명시하는 것이 도움이 될 수 있습니다. 관측 데이터에 간접 스스로를 속이기 쉽습니다. 우리는 열심히 생각하고, 사용 가능한 모든 도구를 사용해야 합니다. 데이터에 대해 열심히 생각하는 데 도움이 될 수 있는 한 가지 프레임워크는 방향성 비순환 그래프(DAG)를 사용하는 것입니다. DAG는 흐름도에 대한 멋진 이름이며, 변수 간의 관계를 나타내기 위해 화살표와 선을 그리는 것을 포함합니다.

이를 구성하기 위해 Graphviz를 사용합니다. Graphviz는 그래프 시각화를 위한 오픈 소스 패키지이며 Quarto에 내장되어 있습니다. 코드는 “R” 대신 “dot” 청크로 래핑해야 하며, 청크 옵션은 “#!” 대신 “//|”로 설정됩니다. 이를 필요로 하지 않는 대안으로는 DiagrammeR (Iannone 2022) 및 ggdag (Barrett 2021b) 사용이 있습니다. 첫 번째 DAG에 대한 전체 청크를 제공하지만, 다른 DAG에 대해서는 코드만 제공합니다.

```

```\{dot
//| label: fig-dot-firstdag-pdf
//| fig-cap: "x가 y에 영향을 미치는 인과 관계"
//| fig-width: 2
digraph D {
 node [shape=plaintext, fontname = "helvetica"];

 {rank=same x y};

 x -> y;
}
```

```

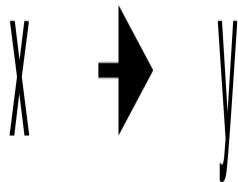


그림 5.1: x가 y에 영향을 미치는 인과 관계

?@fig-dot-firstdag-pdf에서 우리는 x가 y의 원인이라고 생각합니다.

5.4.3.1 교란 변수

상황이 덜 명확한 또 다른 DAG를 만들 수 있습니다. 예시를 더 쉽게 따라갈 수 있도록 소득과 행복 사이의 가상의 관계에 대해 생각하고, 그 관계에 영향을 미칠 수 있는 변수를 고려해 보겠습니다. 이 첫 번째 예시에서는 소득과 행복, 그리고 교육 간의 관계를 고려합니다 (그림 ??).

```

digraph D {

    node [shape=plaintext, fontname = "helvetica"];

    a [label = "소득"];
    b [label = "행복"];
    c [label = "교육"];

    { rank=same a b};

```

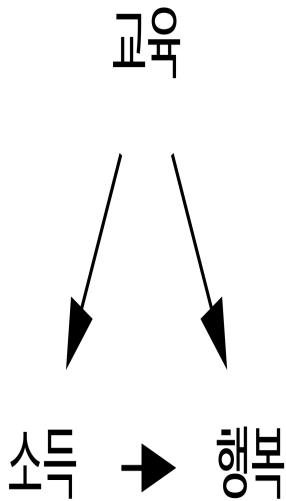
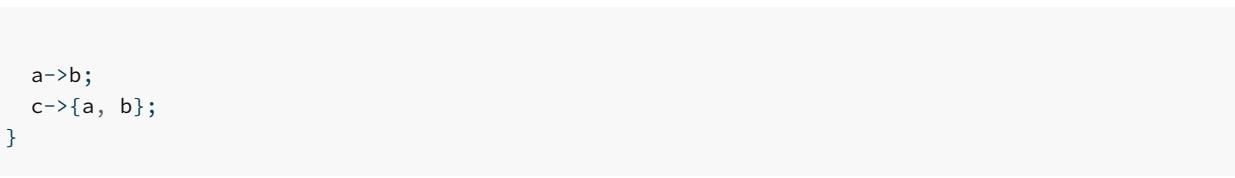


그림 5.2: 교육은 소득과 행복 사이의 관계에 영향을 미치는 교란 변수입니다.

?@fig-dot-educationasconfounder에서 우리는 소득이 행복을 유발한다고 생각합니다. 그러나 우리는 또한 교육이 행복을 유발하고, 교육이 소득도 유발한다고 생각합니다. 이러한 관계는 “백도어 경로”이며, 회귀 분석에서 교육을 조정하지 않으면 관계의 정도를 과대평가하거나, 심지어 소득과 행복 사이에 허위 관계를 만들 수도 있습니다. 즉, 소득의 변화가 행복의 변화를 유발한다고 생각할 수 있지만, 교육이 둘 다를 변화시키는 것일 수도 있습니다. 이 경우 교육이라는 변수는 “교란 변수”라고 불립니다.

Hernán 와/과 Robins (2023, p. 83)은 한 연구자가 한 사람이 하늘을 올려다보는 것이 다른 사람들도 하늘을 올려다보게 하는지 여부에 관심이 있었던 흥미로운 사례를 논의합니다. 두 사람의 반응 사이에는 명확한 관계가 있었습니다. 그러나 하늘에 소음이 있었던 경우도 있었습니다. 두 번째 사람이 첫 번째 사람이 올려다보았기 때문에 올려다보았는지, 아니면 둘 다 소음 때문에 올려다보았는지 불분명했습니다. 실험 데이터를 사용할 때 무작위화는 이러한 우려를 피할 수 있게 해주지만, 관측 데이터에서는 이에 의존할 수 없습니다. 또한 더 큰 데이터가 반드시 이 문제를 해결해 주는 것도 아닙니다. 대신, 상황에 대해 신중하게 생각해야 하며, DAG가 도움이 될 수 있습니다.

교란 변수가 있지만 여전히 인과 효과에 관심이 있다면, 이를 조정해야 합니다. 한 가지 방법은 회귀 분석에 포함하는 것입니다. 그러나 이것의 유효성은 여러 가정을 필요로 합니다. 특히, Gelman 와/과 Hill (2007, p. 169)은 모든 교란 변수를 포함하고 올바른 모델을 가지고 있는 경우에만 우리의 추정치가 표본의 평균 인과 효과에 해당할 것이라고 경고합니다. 두 번째 요구 사항을 제쳐두고 첫 번째 요구 사항에만 집중하면, 교란 변수를 생각하고 관찰하지 않으면 조정하기 어려울 수 있습니다. 그리고 이는 도메인 전문 지식과 이론이 분석에 상당한 비중을 차지할 수 있는 영역입니다.

5.4.3.2 매개 변수

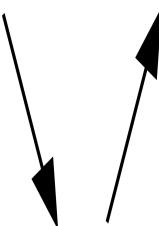
?@fig-dot-luxuryasmediator에서 우리는 소득이 행복을 유발한다고 다시 고려합니다. 그러나 소득이 자녀를 유발하고, 자녀도 행복을 유발한다면, 소득이 행복에 미치는 영향을 이해하기 어려운 상황이 됩니다.

```

digraph D {
    node [shape=plaintext, fontname = "helvetica"];
    a [label = "소득"];
    b [label = "행복"];
    c [label = "자녀"];
    { rank=same a b};
    a->{b, c};
    c->b;
}

```

소득 → 행복



자녀

그림 5.3: 소득과 행복 사이의 매개 변수로서의 자녀

?@fig-dot-luxuryasmediator에서 자녀는 “매개 변수”라고 불리며, 소득이 행복에 미치는 영향에 관심이 있다면 이를 조정하지 않을 것입니다. 만약 조정한다면, 소득에 귀속되는 것 중 일부는 자녀 때문일 것입니다.

5.4.3.3 총돌 변수

마지막으로, ?@fig-dot-residenceascollider에서는 소득이 행복을 유발한다고 생각하는 또 다른 유사한 상황이 있습니다. 그러나 이번에는 소득과 행복 모두 운동을 유발합니다. 예를 들어, 돈이 많으면 운동하기 더 쉬울 수 있지만, 더 행복하면 운동하기 더 쉬울 수도 있습니다.

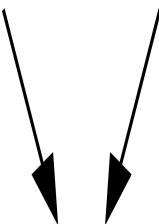
```

digraph D {
    node [shape=plaintext, fontname = "helvetica"];
    a [label = "소득"];
    b [label = "행복"];
}

```

```
c [label = "소득"];
{
  rank=same a b;
  a->{b c};
  b->c;
}
```

소득 → 행복



운동

그림 5.4: 소득과 행복 사이의 관계에 영향을 미치는 충돌 변수로서의 운동

이 경우 운동은 “충돌 변수”라고 불리며, 이를 조건화하면 오해의 소지가 있는 관계를 만들 수 있습니다. 소득은 운동에 영향을 미치지만, 사람의 행복도 이에 영향을 미칩니다. 운동은 예측 변수와 관심 결과 변수 모두가 영향을 미치기 때문에 충돌 변수입니다.

우리는 이것에 대해 명확히 할 것입니다. 우리는 모델을 직접 구성하는 것과 마찬가지로 DAG를 직접 만들어야 합니다. 우리를 위해 그것을 만들어 줄 것은 아무것도 없습니다. 이는 상황에 대해 신중하게 생각해야 함을 의미합니다. DAG에서 무언가를 보고 그것에 대해 조치를 취하는 것은 한 가지 일이지만, 그것이 거기에 있다는 것을 알지 못하는 것은 또 다른 일입니다. McElreath ([2015년] 2020, p. 180)은 이를 유령 DAG라고 설명합니다. DAG는 도움이 되지만, 상황에 대해 깊이 생각하는 데 도움이 되는 도구일 뿐입니다.

모델을 구축할 때 가능한 한 많은 예측 변수를 포함하고 싶은 유혹을 느낄 수 있습니다. DAG는 왜 우리가 더 신중해야 하는지 명확하게 보여줍니다. 예를 들어, 변수가 교란 변수라면 조정하고 싶을 것이고, 변수가 충돌 변수라면 조정하지 않을 것입니다. 우리는 진실을 결코 알 수 없으며, 이론, 관심사, 연구 설계, 데이터의 한계 또는 연구자로서의 우리 자신의 한계와 같은 측면에 의해 정보를 얻습니다. 한계를 아는 것은 모델을 보고하는 것만큼이나 중요합니다. 결함이 있는 데이터와 모델도 그 결함을 인정한다면 여전히 유용합니다. 상황에 대해 생각하는 작업은 결코 끝나지 않으며, 다른 사람에게 의존합니다. 이것이 우리가 모든 작업을 가능한 한 재현 가능하게 만들어야 하는 이유입니다.

5.5 논문의 구성 요소

나는 교수를 시작하기 전에는 아무것도 출판하지 않았지만, 거의 작성하자마자 파괴된 많은 조잡한 노력 속에서, 한때 화려하고 장황한 구성에 가졌을지도 모르는 취향을 극복하고, 평범하고 소박한 것을 선호하게 되었다.

교수 (Brontë 1857)

제목, 초록, 서론, 데이터, 결과, 논의, 그림, 표, 방정식 및 기술 용어와 같은 구성 요소를 논의합니다.¹ 논문 전체에서 가능한 한 간결하고 구체적으로 작성하십시오. 대부분의 독자는 제목을 넘어서지 못할 것입니다. 초록 이상을 읽는 사람은 거의 없을 것입니다. 섹션 및 하위 섹션 제목, 그리고 그래프 및 표 캡션은 주변 텍스트 없이도 자체적으로 작동해야 합니다. 왜냐하면 많은 사람들이 논문을 읽는 방식이 그러한 훑어 보기이기 때문입니다 (Keshav 2007).

5.5.1 제목

제목은 독자를 우리의 이야기에 참여시킬 수 있는 첫 번째 기회입니다. 이상적으로는 우리가 발견한 것을 독자에게 정확히 알려줄 수 있어야 합니다. 효과적인 제목은 중요합니다. 그렇지 않으면 독자들이 논문을 무시할 수 있기 때문입니다. 제목이 “귀엽다”고 할 필요는 없지만, 의미가 있어야 합니다. 즉, 이야기를 명확하게 해야 합니다.

충분히 좋은 제목의 한 가지 예는 “2016년 브렉시트 국민투표에 대하여”입니다. 이 제목은 독자가 논문이 무엇에 관한 것인지 알 수 있기 때문에 유용합니다. 그러나 특히 정보가 많거나 흥미를 끄는 것은 아닙니다. 약간 더 나은 제목은 “2016년 브렉시트 국민투표에서 Vote Leave 결과에 대하여”일 수 있습니다. 이 변형은 정보가 담긴 구체성을 추가합니다. 우리는 가장 좋은 제목은 “2016년 브렉시트 국민투표에서 농촌 지역에서 Vote Leave가 더 나은 성과를 보였습니다: 베이즈 계층 모델의 증거”와 같은 것이라고 주장합니다. 여기에서 독자는 논문의 접근 방식과 주요 요점을 알 수 있습니다.

특히 효과적인 제목의 몇 가지 예를 살펴보겠습니다. Hug 기타 (2019)은 “1990년에서 2017년 사이 신생아 사망률의 국가, 지역 및 전 세계 수준 및 추세, 2030년까지 시나리오 기반 예측: 체계적인 분석”을 사용합니다. 여기에서 논문이 무엇에 관한 것인지, 사용된 방법이 무엇인지 명확합니다. R. Alexander 와/과 Alexander (2021)은 “1901년에서 2018년 사이 호주 연방 의회에서 논의된 주제에 대한 선거 및 총리 변화의 증가된 효과”를 사용합니다. 주요 발견은 논문의 내용에 대한 많은 정보와 함께 제목에서 명확합니다. M. Alexander, Kiang, 와/과 Barbieri (2018)는 “1979-2015년 미국 흑인 및 백인 아편 사망률 추세”를 사용합니다. Frei 와/과 Welsh (2022)는 “미국 세금 허점 폐쇄가 투자자 포트폴리오에 미치는 영향”을 사용합니다. 아마도 역대 최고의 제목 중 하나는 Bickel, Hammel, 와/과 O'Connell (1975)의 “대학원 입학에서의 성별 편향: 버클리 데이터: 편향 측정은 일반적으로 가정하는 것보다 어렵고, 증거는 때때로 예상과 다릅니다”이며, 이는 장 ??에서 다시 다릅니다.

제목은 종종 논문의 마지막 측면 중 하나입니다. 초고를 작성하는 동안에는 일반적으로 작업을 완료하는데 도움이 되는 작업 제목을 사용합니다. 그런 다음 다시 작성하는 과정에서 다듬습니다. 제목은 논문의 최종 이야기를 반영해야 하며, 이는 일반적으로 시작할 때 알 수 있는 것이 아닙니다. 독자가 논문을 읽을 만큼 충분히 흥미를 느끼게 하는 것과 유용하도록 충분한 내용을 전달하는 것 사이에서 균형을 맞춰야 합니다 (Hayot 2014). 두 가지 훌륭한 예시는 토마스 바빙턴 매콜리의 제임스 2세 즉위 이후의 영국사와 윈스턴 처칠의 영어를 사용하는 민족의 역사입니다. 둘 다 내용이 무엇인지 명확하며, 대상 독자에게 흥미를 유발합니다.

한 가지 특정 접근 방식은 “흥미로운 내용: 구체적인 내용” 형식입니다. 예를 들어, “뿌리로 돌아가기: 2016

¹때로는 별도의 문헌 검토 섹션이 필요할 수 있지만, 또 다른 접근 방식은 적절하게 논문 전체에 관련 문헌을 논의하는 것입니다. 예를 들어, 데이터와 관련된 문헌이 있다면 이 섹션에서 논의해야 하며, 모델, 결과 또는 논의와 관련된 문헌은 해당 섹션에서 적절하게 언급해야 합니다.

년 브렉시트 국민투표에서 Vote Leave의 성과 검토”와 같습니다. Kennedy 와/과 Gelman (2021) 는 “인구를 알고 모델을 알라: 모델 기반 회귀 및 사후 계층화를 사용하여 관측된 표본을 넘어 결과를 일반화하기”라는 이 접근 방식의 특히 좋은 예시를 제공하며, Craiu (2019) 도 “고용 캠벨: 투표 데이터 과학자를 찾아서”와 같이 좋은 예시를 제공합니다. 이와 유사한 변형은 “질문? 그리고 접근 방식”입니다. 예를 들어, (cahill2020increase는?) “2030년까지 75%의 수요를 충족시키기 위해 FP2020 국가에서 현대 피임약 사용을 얼마나 늘려야 하는가? 가속화된 전환 방법 및 가족 계획 추정 모델을 사용한 평가”와 같습니다. 이 변형에 익숙해지면, Briggs (2021) 의 “왜 원조는 가장 가난한 사람들을 대상으로 하지 않는가?”와 같이 답변 부분을 생략하면서도 효과적인 상태를 유지하는 것이 적절한 시기를 알 수 있게 됩니다. 또 다른 특정 접근 방식은 “구체적인 내용 다음 광범위한 내용” 또는 그 반대입니다. 예를 들어, “2016년 브렉시트 국민투표에서 농촌성, 엘리트 및 Vote Leave 지지” 또는 “2016년 브렉시트 국민투표에서 Vote Leave 지지, 농촌성 및 엘리트”와 같습니다. 이 접근 방식은 (tolley2021gender가?) “성별, 지방 정당 정치 및 몬트리올의 첫 여성 시장”에서 사용합니다.

때로는 부제목을 포함할 수 있습니다. 이 경우, 이를 활용하는 좋은 방법은 발견한 주요 정량적 결과의 일부 세부 정보를 포함하는 것입니다. 해당 결과에 대한 적절한 수준의 세부 정보와 추상화를 얻는 것은 어렵고, 다시 작성하고 다른 사람의 의견을 구해야 할 것입니다.

5.5.2 초록

10~15페이지 분량의 논문의 경우, 좋은 초록은 3~5문장으로 된 단락입니다. 더 긴 논문의 경우 초록이 약간 더 길어질 수 있습니다. 초록은 논문의 이야기를 명시해야 합니다. 또한 무엇이 수행되었고 왜 중요한지 전달해야 합니다. 이를 위해 초록은 일반적으로 작업의 맥락, 목표, 접근 방식 및 발견에 대해 다룹니다.

더 구체적으로, 초록을 위한 좋은 레시피는 다음과 같습니다. 첫 번째 문장: 논문의 일반적인 영역을 명시하고 독자를 격려합니다. 두 번째 문장: 데이터 세트와 방법을 일반적인 수준에서 명시합니다. 세 번째 문장: 주요 결과를 명시합니다. 그리고 네 번째 문장: 함의에 대해 설명합니다.

우리는 다양한 초록에서 이 패턴을 볼 수 있습니다. 예를 들어, Tolley 와/과 Paquet (2021) 는 첫 문장에서 400년 만에 첫 여성 시장이 선출된 것을 언급하여 독자를 끌어들입니다. 두 번째 문장은 논문에서 무엇이 수행되었는지 명확하게 설명합니다. 세 번째 문장은 설문조사와 같은 방법으로 어떻게 수행되었는지 알려주고, 네 번째 문장은 일부 세부 정보를 추가합니다. 다섯 번째이자 마지막 문장은 주요 요점을 명확하게 합니다.

2017년 몬트리올은 400년 역사상 첫 여성 시장인 발레리 플랑트를 선출했습니다. 이 선거를 사례 연구로 사용하여, 우리는 성별이 결과에 어떻게 영향을 미쳤는지, 그리고 영향을 미치지 않았는지 보여줍니다. 몬트리올 유권자들을 대상으로 한 설문조사는 성별이 투표 선택에 중요한 요소가 아니었음을 시사합니다. 성별이 유권자에게는 크게 중요하지 않았지만, 캠페인과 정당의 조직에는 영향을 미쳤습니다. 우리는 플랑트의 승리가 덜 지도자 중심적인 정당과 정치적 리더십 위치에 대한 여성의 부적합성에 대한 고정관념을 상쇄하는 데 도움이 된 탈 성별화된 캠페인을 보여주는 전략에 부분적으로 설명될 수 있다고 주장합니다.

마찬가지로, Beauregard 와/과 Sheppard (2021) 은 첫 두 문장에서 더 넓은 환경을 명확히 하고, 해당 환경에 대한 이 논문의 구체적인 기여를 명확히 합니다. 세 번째와 네 번째 문장은 데이터 출처와 주요 발견을 명확히 합니다. 다섯 번째와 여섯 번째 문장은 이 초록의 잠재적 독자, 즉 학술 정치 과학자들에게 흥미로울 구체성을 추가합니다. 마지막 문장에서는 저자의 입장이 명확해집니다.

성별 할당량 지지에 대한 이전 연구는 성 평등 및 정부 개입에 대한 태도를 설명으로 집중합니다. 우리는 정치에서 여성의 존재를 늘리는 것을 목표로 하는 정책에 대한 이해에 있어 여성에 대한 태도의 역할이 양면적이라고 주장합니다. 즉, 적대적이고 자비로운 형태의 성차별주의 모두 지지를 이해하는 데 기여하지만, 다른 방식으로 기여합니다. 호주 응답자들의 확

를 기반 표본에 대해 수행된 설문조사에서 얻은 원본 데이터를 사용하여, 우리의 발견은 적대적인 성차별주의자들이 성별 할당량 채택을 통해 정치에서 여성의 존재를 늘리는 것에 더 반대할 가능성이 높다는 것을 보여줍니다. 반면에 자비로운 성차별주의자들은 낮은 수준의 자비로운 성차별주의를 보이는 응답자들보다 이러한 정책을 더 지지할 가능성이 높습니다. 우리는 이것이 자비로운 성차별주의가 여성이 순수하고 보호가 필요하다고 주장하기 때문이라고 주장합니다. 그들은 할당량의 도움 없이는 정치에서 성공하는 데 필요한 것을 가지고 있지 않습니다. 마지막으로, 우리는 여성이 할당량을 더 지지할 가능성이 높지만, 양면적인 성차별주의는 여성과 남성 모두에게 지지와 동일한 관계를 가지고 있음을 보여줍니다. 이러한 발견은 성별 할당량에 대한 대중의 총체적인 지지 수준이 일반적으로 성 평등에 대한 더 큰 수용을 반드시 나타내지는 않는다는 것을 시사합니다.

초록의 또 다른 훌륭한 예시는 Sides, Vavreck, 와/과 Warshaw (2021)입니다. 단 다섯 문장으로, 그들은 무엇을 하는지, 어떻게 하는지, 무엇을 발견하는지, 그리고 왜 중요한지 명확하게 설명합니다.

우리는 2000년부터 2018년까지 미국 선거 결과에 대한 텔레비전 광고의 영향에 대한 포괄적인 평가를 제공합니다. 우리는 대통령, 상원, 하원, 주지사, 법무장관 및 주 재무장관 선거를 포함하고, 광고의 인과 효과를 식별하는데 도움이 되는 차이-차이 및 경계-불연속성 연구 설계를 모두 사용하여 이전 연구를 확장합니다. 우리는 텔레비전 방송 캠페인 광고가 투표 전반에 걸쳐 중요하지만, 대통령 선거보다 하위 투표 선거에서 훨씬 더 큰 영향을 미친다는 것을 발견합니다. 여러 선거 주기의 설문조사 및 유권자 등록 데이터를 사용하여, 우리는 광고 효과의 주요 메커니즘이 당파의 동원이 아니라 설득임을 보여줍니다. 우리의 결과는 캠페인 및 선거 연구뿐만 아니라 유권자 의사 결정 및 정보 처리에도 영향을 미칩니다.

최고의 초록은 내용 대 단어 비율이 너무 높아서 약간 간결하게 느껴질 수도 있습니다. 예를 들어, Touvron 기타 (2023)의 초록에는 낭비되는 단어가 하나도 없으며, 단 네 문장으로 많은 양의 정보를 전달합니다.

우리는 7B에서 65B 매개변수에 이르는 기초 언어 모델 컬렉션인 LLaMA를 소개합니다. 우리는 수조 개의 토큰으로 모델을 훈련했으며, 독점적이고 접근 불가능한 데이터 세트에 의존하지 않고 공개적으로 사용 가능한 데이터 세트만 사용하여 최첨단 모델을 훈련하는 것이 가능함을 보여줍니다. 특히, LLaMA-13B는 대부분의 벤치마크에서 GPT-3 (175B)를 능가하며, LLaMA-65B는 최고의 모델인 Chinchilla-70B 및 PaLM-540B와 경쟁합니다. 우리는 모든 모델을 연구 커뮤니티에 공개합니다.

Kasy 와/과 Teytelboym (2023)은 더 통계적인 초록의 훌륭한 예시를 제공합니다. 그들은 무엇을 하는지, 왜 중요한지 명확하게 식별합니다.

우리는 자원을 참가자에게 매칭하는 것을 반복적으로 선택해야 하고, 개별적으로 선택된 매칭의 수익은 알 수 없지만 학습할 수 있는 실험 설정을 고려합니다. 우리의 설정은 난민 재정착, 사회 주택 할당 및 위탁 양육과 같이 (잠재적으로 복잡한) 용량 제약이 있는 양면 및 단면 매칭을 다룹니다. 우리는 이러한 적응형 조합 할당 문제를 해결하기 위해 톰슨 샘플링 알고리즘의 변형을 제안합니다. 우리는 이 알고리즘에 대한 엄격하고 사전 독립적인 유한 표본 기

대 후회 경계를 제공합니다. 할당 수가 매칭 수에서 기하급수적으로 증가하지만, 우리의 경계는 그렇지 않습니다. 베이즈 계층 모델을 사용한 난민 재정착 데이터를 기반으로 한 시뮬레이션에서, 우리는 이 알고리즘이 고용 확률에 대한 완벽한 지식을 기반으로 한 최적의 매칭에서 얻을 수 있는 고용 증가의 절반을 달성한다는 것을 발견합니다.

마지막으로, Briggs (2021) 는 의심할 여지 없이 사실처럼 보이는 주장으로 시작합니다. 두 번째 문장에서 그는 그것이 거짓이라고 말합니다! 세 번째 문장은 이 주장의 범위를 명시하고, 네 번째 문장은 더 자세한 내용을 제공하기 전에 그가 이 입장에 도달한 방법을 자세히 설명합니다. 마지막 두 문장은 더 넓은 함의와 중요성을 언급합니다.

해외 원조 프로젝트는 일반적으로 지역적 영향을 미치므로, 빈곤을 줄이려면 빈곤층에 가까이 배치되어야 합니다. 저는 지역 인구 수준을 조건으로 할 때, 세계은행(WB) 프로젝트 원조가 국가의 더 부유한 지역을 대상으로 한다는 것을 보여줍니다. 이러한 관계는 시간과 전 세계 지역에 걸쳐 유지됩니다. 저는 WB 태스크 팀 리더(TTL)에 대한 사전 등록된 결합 실험을 사용하여 친부유층 대상에 대한 다섯 가지 기부자 측 설명을 테스트합니다. TTL은 원조를 받는 정부가 원조를 정치적으로 대상으로 하고 구현을 통제하는 데 가장 관심이 있다고 인식합니다. 그들은 또한 원조가 더 가난하거나 더 외딴 지역에서 더 잘 작동하지만, 이러한 지역에서의 구현은 유일하게 어렵다고 믿습니다. 이러한 결과는 분배 정치, 원조에 대한 국제 협상, 국제 기구의 본인-대리인 문제에 대한 논쟁을 시사합니다. 또한 이러한 결과는 프로젝트 구현의 용이성을 덜 중요하게 만들기 위해 WB 인센티브 구조를 조정하면 원조가 국가의 더 가난한 지역으로 흐르도록 장려할 수 있음을 시사합니다.

과학 저널인 Nature는 초록 구성에 대한 가이드를 제공합니다. 그들은 6개 부분으로 구성되고 약 200단어에 달하는 초록 구조를 권장합니다.

- 1) 광범위한 독자가 이해할 수 있는 서론 문장.
- 2) 잠재적 독자에게 관련성 있는 더 자세한 배경 문장.
- 3) 일반적인 문제를 진술하는 문장.
- 4) 주요 결과를 요약하고 설명하는 문장.
- 5) 일반적인 맥락에 대한 문장.
- 6) 마지막으로, 더 넓은 관점에 대한 문장.

초록의 첫 문장은 공허해서는 안 됩니다. 독자가 제목을 지나 계속 읽었다고 가정하면, 이 첫 문장은 우리 가논문을 계속 읽도록 독자를 설득할 수 있는 다음 기회입니다. 그리고 초록의 두 번째 문장 등도 마찬가지입니다. 초록이 너무 좋아서 그것만 읽어도 괜찮을 정도로 작업하고 다시 작업하십시오. 왜냐하면 종종 그렇게 될 것이기 때문입니다.

5.5.3 서론

서론은 자체 포함되어야 하며 독자가 알아야 할 모든 것을 전달해야 합니다. 우리는 미스터리 소설을 쓰는 것이 아닙니다. 대신, 서론에서 가장 중요한 요점을 알려주고 싶습니다. 10~15페이지 분량의 논문의 경우, 서론은 두세 단락의 주요 내용으로 구성될 수 있습니다. Hayot (2014, p. 90)은 서론의 목표는 독자를 참여시키고, 특정 분야와 배경에 위치시키고, 논문의 나머지 부분에서 무엇이 일어나는지 알려주는 것이라고 말합니다. 이는 전적으로 독자 중심이어야 합니다.

서론은 배경을 설정하고 독자에게 배경 지식을 제공해야 합니다. 예를 들어, 우리는 일반적으로 조금 더 넓게 시작합니다. 이는 논문에 대한 일부 맥락을 제공합니다. 그런 다음 논문이 그 맥락에 어떻게 들어맞는지 설명하고, 이야기의 주요 부분인 하나의 핵심 결과에 특히 초점을 맞춰 일부 높은 수준의 결과를 제공합니다. 우리는 초록에서 제공한 것보다 더 자세한 내용을 제공하지만, 전체 범위는 아닙니다. 그리고 한두 문

장으로 다음 단계를 광범위하게 논의합니다. 마지막으로, 논문의 구조를 강조하는 추가적인 짧은 마지막 단락으로 서론을 마무리합니다.

예시 (가상의 세부 정보 포함):

영국 보수당은 항상 농촌 선거구에서 좋은 성과를 거두었습니다. 그리고 2016년 브렉시트 투표도 농촌과 도시 지역 간의 지지율에 상당한 차이가 있었던 점에서 다르지 않았습니다. 그러나 보수적인 문제에 대한 농촌 지지라는 기준에서도 “Vote Leave”에 대한 지지는 이례적으로 강했습니다. “Vote Leave”는 이스트 미들랜드와 잉글랜드 동부에서 가장 강력한 지지를 받았고, “잔류”에 대한 가장 강력한 지지는 그레이터 런던에서였습니다.

이 논문에서 우리는 2016년 브렉시트 국민투표에서 “Vote Leave”的 성과가 농촌성과 그렇게 상관 관계가 있었던 이유를 살펴봅니다. 우리는 투표 지역 수준에서 “Vote Leave”에 대한 지지가 해당 지역의 농장 수, 평균 인터넷 연결성 및 중간 연령으로 설명되는 모델을 구축합니다. 우리는 지역의 중간 연령이 증가함에 따라 해당 지역이 “Vote Leave”를 지지할 가능성이 14% 포인트 감소한다는 것을 발견합니다. 향후 연구에서는 보수당 의원의 영향을 살펴봄으로써 이러한 효과에 대한 더 미묘한 이해를 가능하게 할 수 있습니다.

이 논문의 나머지 부분은 다음과 같이 구성됩니다: 2절은 데이터를 논의하고, 3절은 모델을 논의하며, 4절은 결과를 제시하고, 마지막으로 5절은 우리의 발견과 일부 약점을 논의합니다.

서론은 자체 포함되어야 하며 독자에게 필요한 거의 모든 것을 알려주어야 합니다. 독자는 서론만 읽고도 전체 논문의 모든 주요 측면에 대한 정확한 그림을 가질 수 있어야 합니다. 서론에 그래프나 표를 포함하는 경우는 드뭅니다. 서론은 논문의 구조를 암시하며 마무리되어야 합니다.

5.5.4 데이터

린든 존슨의 전기 작가인 로버트 카로는 전기를 쓸 때 “장소 감각”을 전달하는 것의 중요성을 설명합니다 (Caro 2019, p. 141). 그는 이를 “책의 행동이 일어나는 물리적 환경: 충분히 명확하게, 충분한 세부 사항으로 볼 수 있어서, 행동이 일어나는 동안 마치 자신이 그곳에 있는 것처럼 느끼게 하는 것”이라고 정의합니다. 그는 다음 예를 제공합니다.

레베카가 그 작은 집의 현관문을 나섰을 때, 아무것도 없었습니다—아마도 부리에서 길고 축축한 무언가를 매달고 바위 뒤를 빠르게 지나가는 로드러너나, 흰 꼬리의 번쩍임만 보일 정도로 빠르게 덤불 뒤로 사라지는 토키 외에는 아무것도 없었습니다. 흩어져 있는 나무들의 잎사귀가 흔들리는 것 외에는 움직임이 없었고, 끊임없이 속삭이는 바람 소리 외에는 아무 소리도 없었습니다... 레베카가 거의 절망적으로 집 뒤 언덕을 올랐을 때, 그 정상에서 그녀가 본 것은 더 많은 언덕, 끝없이 펼쳐진 언덕, 단 하나의 집도 보이지 않는 언덕... 아무것도 움직이지 않는 텅 빈 언덕, 그 위에는 텅 빈 하늘; 조용히 머리 위를 맴도는 매는 사건이었습니다. 그러나 무엇보다도 인간적인 것은 아무것도 없었고, 이야기할 사람도 없었습니다.

Caro (2019, p. 146)

존슨의 어머니인 레베카 베인즈 존슨의 상황을 얼마나 철저하게 상상할 수 있습니까? 논문을 작성할 때, 우리는 카로가 힐 카운티에 대해 제공하는 것과 같은 장소 감각을 우리의 데이터에 대해 달성해야 합니다. 우리는 가능한 한 명확하게 함으로써 이를 수행합니다. 우리는 일반적으로 그것에 대한 전체 섹션을 가지고

있으며, 이는 독자에게 우리의 이야기를 뒷받침하는 실제 데이터를 가능한 한 가깝게 보여주기 위해 고안되었습니다.

데이터 섹션을 작성할 때 우리는 우리의 주장에 대한 중요한 질문에 대한 답변을 시작합니다. 즉, 이것을 어떻게 알 수 있는가? (McPhee 2017, p. 78). 데이터 섹션의 훌륭한 예시는 (dolls1950smoking에서?) 제공합니다. 그들은 통제 그룹과 치료 그룹 간의 흡연 효과에 관심이 있습니다. 데이터 세트를 명확하게 설명한 후 관련 교차표를 표시하기 위해 표를 사용하고 그룹을 대조하기 위해 그래프를 사용합니다.

데이터 섹션에서는 우리가 사용하는 데이터 세트의 변수를 철저히 논의해야 합니다. 사용될 수 있었지만 사용되지 않은 다른 데이터 세트가 있다면 언급하고 선택을 정당화해야 합니다. 변수가 구성되거나 결합되었다면 이 과정과 동기를 설명해야 합니다.

우리는 독자가 결과의 기반이 되는 데이터가 어떻게 생겼는지 이해하기를 원합니다. 이는 분석에 사용된 데이터를 그래프로 나타내거나 가능한 한 가깝게 나타내야 함을 의미합니다. 그리고 요약 통계 표도 포함해야 합니다. 데이터 세트가 다른 출처에서 생성되었다면 해당 원본 출처의 예시를 포함하는 것도 도움이 될 수 있습니다. 예를 들어, 데이터 세트가 설문조사 응답에서 생성되었다면 기본 설문조사 질문을 부록에 포함해야 합니다.

데이터 섹션의 그림과 표에 관해서는 어느 정도 판단이 필요합니다. 독자는 세부 사항을 이해할 기회를 가져야 하지만, 일부는 부록에 더 적절하게 배치될 수 있습니다. 그림과 표는 사람들에게 이야기를 설득하는데 중요한 측면입니다. 그래프에서는 데이터를 보여주고 독자가 스스로 결정하도록 할 수 있습니다. 그리고 표를 사용하여 데이터 세트를 요약할 수 있습니다. 최소한 모든 변수는 그래프로 표시되고 표로 요약되어야 합니다. 너무 많다면 일부는 부록으로 강등될 수 있으며, 중요한 관계는 본문에 표시됩니다. 그림과 표는 번호가 매겨지고 텍스트에서 상호 참조되어야 합니다. 예를 들어, “그림 1은 ...을 보여줍니다”, “표 1은 ...을 설명합니다”. 모든 그래프와 표에는 주요 측면을 설명하고 추가 세부 정보를 추가하는 텍스트가 함께 제공되어야 합니다.

장 ??에서 제목과 레이블을 포함한 그래프와 표의 구성 요소를 논의합니다. 그러나 여기서는 텍스트와 그래프 또는 표 사이에 있는 캡션에 대해 논의할 것입니다. 캡션은 정보가 풍부하고 자체 포함되어야 합니다. (borkin2015beyond는?) 시선 추적을 사용하여 시각화가 어떻게 인식되고 회상되는지 이해합니다. 그들은 캡션이 그림의 핵심 메시지를 명확하게 해야 하며, 중복이 있어야 한다고 말합니다. Cleveland ([1985년] 1994, p. 57)가 말했듯이, “그래프, 캡션 및 텍스트 간의 상호 작용은 섬세한 것입니다.” 그러나 독자는 캡션만 읽고도 그래프나 표가 무엇을 보여주는지 이해할 수 있어야 합니다. 두 줄 길이의 캡션이 반드시 부적절한 것은 아닙니다. 그리고 그래프나 표의 모든 측면이 설명되어야 합니다. 예를 들어, Bowley (1901, p. 151)의 그림 ??와 그림 ??를 고려해 보십시오. 둘 다 명확하고 자체 포함되어 있습니다.

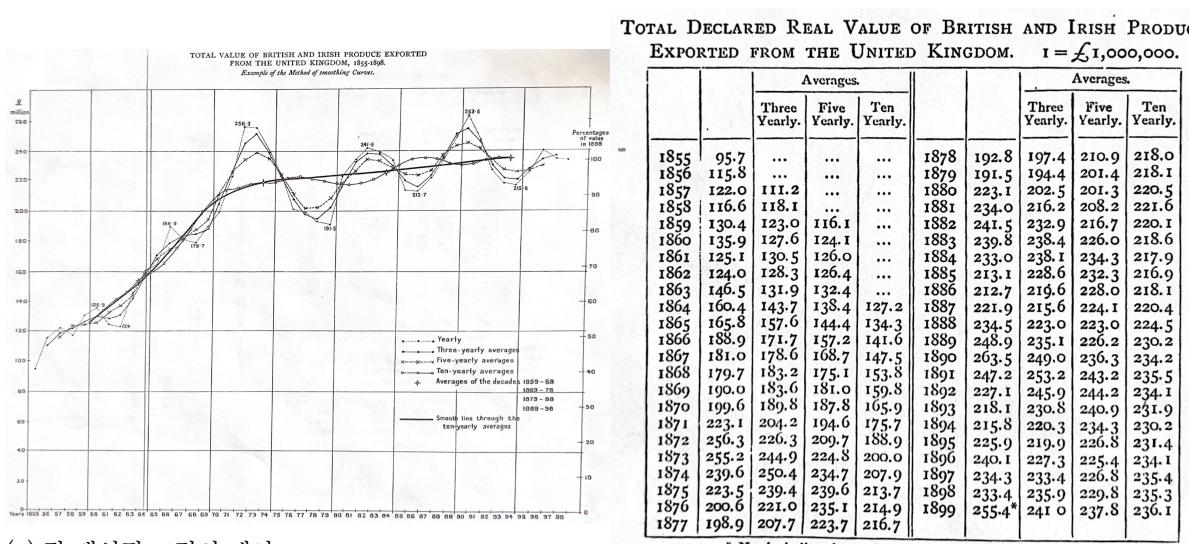


그림 5.5: Bowley (1901) 의 그래프 및 표 예시

표와 그래프 사이의 선택은 전달할 정보의 양에 따라 달라집니다. 일반적으로 요약 통계와 같이 고려해야

할 특정 정보가 있다면 표가 좋은 선택입니다. 독자가 비교하고 추세를 이해하는 데 관심이 있다면 그래프가 좋은 선택입니다 (Gelman, Pasarica, 와/과 Dodhia 2002).

5.5.5 모델

우리는 종종 데이터를 탐색하는 데 사용할 통계 모델을 구축하며, 이에 대한 특정 섹션을 갖는 것이 일반적입니다. 최소한 사용되는 모델을 설명하는 방정식을 지정하고 일반 언어와 상호 참조를 사용하여 구성 요소를 설명해야 합니다.

모델 섹션은 일반적으로 모델이 작성되고 설명되며 정당화되는 것으로 시작됩니다. 예상 독자에 따라 일부 배경 지식이 필요할 수 있습니다. 적절한 수학적 표기법으로 모델을 지정하고 상호 참조한 후, 모델의 구성 요소를 정의하고 설명해야 합니다. 표기법의 각 측면을 정의하려고 노력하십시오. 이는 독자에게 모델이 잘 선택되었음을 확신시키고 논문의 신뢰성을 높이는 데 도움이 됩니다. 모델의 변수는 데이터 섹션에서 논의된 변수와 일치해야 하며, 두 섹션 간에 명확한 연결을 만들어야 합니다.

특징이 모델에 어떻게 들어가는지, 그리고 그 이유에 대한 논의가 있어야 합니다. 몇 가지 예시는 다음과 같습니다.

- 연령 그룹 대신 연령을 사용하는 이유는 무엇입니까?
- 주/도가 수준 효과를 갖는 이유는 무엇입니까?
- 성별이 범주형 변수인 이유는 무엇입니까? 일반적으로 우리는 이것이 상황에 적합한 모델이라는 느낌을 전달하려고 합니다. 우리는 독자가 데이터 섹션에서 논의된 측면이 모델링 결정에 어떻게 나타나는지 이해하기를 원합니다.

모델 섹션은 모델을 뒷받침하는 가정에 대한 논의로 마무리되어야 합니다. 또한 대체 모델 또는 변형에 대한 간략한 논의도 있어야 합니다. 강점과 약점이 명확하고 독자가 이 특정 모델이 선택된 이유를 알기를 원합니다.

이 섹션의 어느 시점에서는 일반적으로 모델을 실행하는 데 사용된 소프트웨어를 지정하고, 모델이 적절하지 않을 수 있는 상황에 대한 생각을 증명하는 것이 적절합니다. 두 번째 요점은 일반적으로 논의 섹션에서 확장될 것입니다. 그리고 모델 검증 및 확인, 모델 수렴 및/또는 진단 문제에 대한 증거가 있어야 합니다. 다시 말하지만, 여기에는 균형이 필요하며, 이 내용 중 일부는 부록에 더 적절하게 배치될 수 있습니다.

기술 용어가 사용될 때는 익숙하지 않은 독자를 위해 일반 언어로 간략하게 설명해야 합니다. 예를 들어, M. Alexander (2019b) 는 지니 계수에 대한 설명을 통합하여 독자를 이해시킵니다.

아기 이름의 집중도를 살펴보려면 각 국가, 성별 및 연도에 대한 지니 계수를 계산해 봅시다. 지니 계수는 빈도 분포 값 간의 분산 또는 불평등을 측정합니다. 0에서 1 사이의 값을 가질 수 있습니다. 소득 분포의 경우, 지니 계수가 1이면 한 사람이 모든 소득을 가지고 있음을 의미합니다. 이 경우, 지니 계수가 1이면 모든 아기가 같은 이름을 가지고 있음을 의미합니다. 반대로, 지니 계수가 0이면 이름이 모든 아기에게 고르게 분포됨을 의미합니다.

통계 모델을 포함하지 않는 논문도 있을 수 있습니다. 이 경우 이 “모델” 섹션은 더 넓은 “방법론” 섹션으로 대체되어야 합니다. 수행된 시뮬레이션을 설명하거나 접근 방식에 대한 더 일반적인 세부 정보를 포함할 수 있습니다.

5.5.6 결과

결과 섹션의 두 가지 훌륭한 예시는 Kharecha 와/과 Hansen (2013) 및 (kiang2021racial에서?) 제공합니다. 결과 섹션에서는 분석 결과를 명확하게 전달하고 합의에 대한 논의에 너무 집중하지 않기를 원합니다. 결과 섹션에는 요약 통계, 표 및 그래프가 필요할 것입니다. 이러한 각 측면은 상호 참조되어야 하며, 각 그림에서 보이는 내용을 자세히 설명하는 텍스트가 함께 제공되어야 합니다. 이 섹션은 결과를 전달해야 합니다. 즉, 결과가 무엇을 의미하는지보다는 결과가 무엇인지에 관심이 있습니다.

이 섹션에는 모델링을 기반으로 한 계수 추정치 그래프 표도 일반적으로 포함될 것입니다. 추정치의 다양한 특징을 논의하고, 모델 간의 차이를 설명해야 합니다. 데이터의 다른 하위 집합을 별도로 고려할 수도 있습니다. 다시 말하지만, 모든 그래프와 표에는 일반 언어로 된 텍스트가 함께 제공되어야 합니다. 대략적인 지침은 텍스트의 양이 표와 그래프가 차지하는 공간의 양과 같거나 그 이상이어야 한다는 것입니다. 예를 들어, 계수 추정치 표를 표시하는 데 전체 페이지가 사용된다면, 해당 표에 대한 전체 페이지 분량의 텍스트가 함께 제공되어야 합니다.

5.5.7 논의

논의 섹션은 논문의 마지막 섹션이 될 수 있으며, 일반적으로 네다섯 개의 하위 섹션으로 구성됩니다.

논의 섹션은 일반적으로 논문에서 수행된 작업에 대한 간략한 요약을 포함하는 하위 섹션으로 시작됩니다. 그 다음에는 이 논문에서 세상에 대해 배우는 핵심 사항에 전념하는 두세 개의 하위 섹션이 이어집니다. 이러한 하위 섹션은 논문에서 이야기되는 이야기의 핵심을 정당화하거나 자세히 설명할 주요 기회입니다. 일반적으로 이러한 하위 섹션에는 새로 도입된 그래프나 표가 없으며, 대신 이전 섹션에서 도입된 것들로부터 배우는 것에 중점을 둡니다. 일부 결과는 다른 사람들이 발견한 것과 관련하여 논의될 수 있으며, 차이점은 여기에서 조정될 수 있습니다.

세상에 대해 배우는 이러한 하위 섹션 다음에는 일반적으로 수행된 작업의 일부 약점을 초점에 맞춘 하위 섹션이 있습니다. 이는 사용된 데이터, 접근 방식 및 모델과 같은 측면에 관련될 수 있습니다. 모델의 경우 우리는 특히 발견에 영향을 미칠 수 있는 측면에 관심이 있습니다. 이는 기계 학습 모델의 경우 특히 어려울 수 있으며, (realm은?) 고려해야 할 측면에 대한 지침을 제공합니다. 그리고 마지막 하위 섹션은 일반적으로 배울 점과 향후 작업이 어떻게 진행될 수 있는지 명시하는 몇 단락입니다.

일반적으로 이 섹션은 전체 논문의 최소 25%를 차지할 것으로 예상됩니다. 즉, 8페이지 논문에서는 최소 2페이지의 논의가 필요합니다.

5.5.8 간결성, 오타 및 문법

간결성은 중요합니다. 이는 부분적으로 우리가 독자를 위해 글을 쓰고, 독자는 다른 우선 순위를 가지고 있기 때문입니다. 그러나 또한 작가로서 가장 중요한 요점이 무엇인지, 어떻게 가장 잘 뒷받침할 수 있는지, 그리고 주장이 가장 약한 부분이 어디인지 고려하도록 강요하기 때문입니다. 장 크레티앙은 전 캐나다 총리입니다. Chrétien (2007, p. 105)에서 그는 "...공무원들에게 문서를 두세 페이지로 요약하고 나머지 자료를 배경 정보로 첨부하도록 요청하곤 했습니다. 저는 곧 이것이 실제로 무엇을 말하는지 모르는 사람들에게만 문제라는 것을 발견했습니다."라고 썼습니다.

이러한 경험은 캐나다에만 국한된 것이 아니며 새로운 것도 아닙니다. (*instituteforgovernment*에서?) 전 영국 내각 의원인 올리버 렛윈은 "일부 부서에서 엄청나게 길고 끔찍한 헛소리가 쏟아져 나온다"고 설명하며, "길이를 4분의 1로 줄여달라"고 요청했다고 말합니다. 그는 부서들이 중요한 것을 잊지 않고 이 요청을 수용할 수 있었다는 것을 발견했습니다. 윈스턴 처칠은 제2차 세계 대전 중 "실제 요점을 간결하게 제시하는 훈련은 더 명확한 사고에 도움이 될 것"이라고 말하며 간결성을 요구했습니다. 맨해튼 프로젝트의 촉매제가 된 실라르드와 아인슈타인의 FDR에게 보낸 편지는 단 두 페이지였습니다!

(zinsser는?) 더 나아가 "좋은 글쓰기의 비결"은 "모든 문장을 가장 깔끔한 구성 요소로 벗겨내는 것"이라고 설명합니다. 모든 문장은 본질로 단순화되어야 합니다. 그리고 기여하지 않는 모든 단어는 제거되어야 합니다.

불필요한 단어, 오타 및 문법 오류는 논문에서 제거되어야 합니다. 이러한 실수는 주장의 신뢰성에 영향을 미칩니다. 독자가 당신이 맞춤법 검사기를 사용하는 것을 신뢰할 수 없다면, 왜 로지스틱 회귀를 사용하는 것을 신뢰해야 합니까? RStudio에는 맞춤법 검사기가 내장되어 있지만, Microsoft Word 및 Google Docs는 유용한 추가 검사 도구입니다. Quarto 문서에서 복사하여 Word에 붙여넣은 다음 빨간색 및 녹색 줄을 찾아 Quarto 문서에서 수정하십시오.

우리는 문법 내용의 n차원에 대해 걱정하지 않습니다. 대신, 우리는 대화 언어 사용에서 발생하는 문법과 문장 구조에 관심이 있습니다 (S. King 2000, p. 118). 편안함을 개발하는 방법은 광범위하게 읽고 다른 사람들에게도 당신의 작업을 읽어달라고 요청하는 것입니다. 또 다른 유용한 전술은 당신의 글을 소리 내어 읽는 것입니다. 이는 소리를 기반으로 이상한 문장을 감지하는 데 유용할 수 있습니다. 정기적으로 나타날 작은 측면 중 하나는 1에서 10까지의 숫자는 단어로 작성해야 하고, 11 이상은 숫자로 작성해야 한다는 것입니다.

5.5.9 규칙

다양한 저자들이 글쓰기 규칙을 정립했습니다. 여기에는 (*politicsandtheenglishlanguage*의?) 규칙이 포함되며, 이는 (*johnsontheeconomist*에?) 의해 재해석되었습니다. 데이터로 이야기를 전달하는 데 초점을 맞춘 글쓰기 규칙의 추가적인 재해석은 다음과 같습니다.

- 독자와 그들의 필요에 집중하십시오. 다른 모든 것은 부차적입니다.
- 구조를 설정하고, 그것에 의존하여 이야기를 전달하십시오.
- 가능한 한 빨리 초고를 작성하십시오.
- 그 초고를 광범위하게 다시 작성하십시오.
- 간결하고 직접적으로 작성하십시오. 가능한 한 많은 단어를 제거하십시오.
- 단어를 정확하게 사용하십시오. 예를 들어, 주가는 개선되거나 악화되는 것이 아니라 오르거나 내립니다.
- 가능한 한 짧은 문장을 사용하십시오.
- 전문 용어를 피하십시오.
- 당신의 작품이 신문 1면에 실릴 것처럼 작성하십시오.
- 참신함이나 “X를 연구한 최초의 사람”이라고 주장하지 마십시오. 항상 먼저 그곳에 도달한 다른 사람이 있습니다.

(*fiske2021words*는?) 과학 논문에 대한 규칙 목록을 가지고 있으며, (*pineau2021improving*의?) 부록은 기계 학습 논문에 대한 체크리스트를 제공합니다. 그러나 아마도 마지막 말은 (*Savage2019*의?) 것이어야 할 것입니다.

[당신이 좋아하는] 논문의 최고의 버전을 쓰려고 노력하십시오. 익명의 독자를 만족시킬 수는 없지만, 자신을 만족시킬 수는 있어야 합니다. 당신의 논문은—바라건대—후세를 위한 것입니다.

Savage 와/과 Yeh (2019, p. 442)

5.6 연습 문제

실습

1. (계획) 다음 시나리오를 고려하십시오: 어린이와 부모가 아파트 창문에서 전차를 지켜봅니다. 매 시간, 8시간 동안, 지나가는 전차의 수를 기록합니다. 데이터 세트가 어떻게 생겼을지 스케치하고, 모든 관측치를 보여주기 위해 만들 수 있는 그래프를 스케치하십시오.
2. (시뮬레이션) 설명된 시나리오를 더 자세히 고려하고 상황을 시뮬레이션하십시오. 그런 다음 시뮬레이션된 데이터를 기반으로 다섯 가지 테스트를 작성하십시오.
3. (획득) 관심 있는 도시의 대중교통 측면에 대한 실제 데이터 소스를 지정하십시오.
4. (탐색) 시뮬레이션된 데이터를 사용하여 그래프와 표를 만드십시오.
5. (공유) 그래프와 표에 대한 텍스트를 작성하십시오. 실제 상황을 반영한 것처럼 작성하십시오. 단락에 포함된 정확한 세부 정보는 사실일 필요는 없지만 합리적이어야 합니다 (즉, 실제로 데이터를 얻거나 그래프를 만들 필요는 없습니다). 코드를 R 파일과 Quarto 문서로 적절하게 분리하십시오. README가 있는 GitHub 저장소 링크를 제출하십시오.

퀴즈

1. 좋은 연구 질문의 세 가지 특징은 무엇입니까 (한두 단락으로 작성)?
2. (*bydesignplanningresearch*는?) 광범위한 주제에서 연구를 자세히 계획하는 방법으로 무엇을 권장합니까 (하나 선택)?
 - a. 전문가와 대화.
 - b. 사용 가능한 데이터 식별.

- c. 구체적인 연구 질문 세트 명확화.
3. (bydesignplanningresearch는?) 연구 질문이 왜 그렇게 중요하다고 믿습니까 (모두 선택)?
 - a. 합리적인 계획 결정을 내리는 유일한 근거입니다.
 - b. 표본을 추출할 대상 인구를 식별합니다.
 - c. 적절한 집계 수준을 결정합니다.
 - d. 결과 변수를 식별합니다.
 - e. 주요 예측 변수를 식별합니다.
 - f. 측정 및 데이터 수집에 대한 과제를 제기합니다.
4. (bydesignplanningresearch에?) 따르면, 연구에서 “이론과 데이터의 나선”의 목적은 무엇입니까 (하나 선택)?
 - a. 이론을 개발하기 전에 데이터를 수집하기 위해.
 - b. 이론과 데이터 사이를 오가며 반복적으로 다듬기 위해.
 - c. 이론적 분석 전에 데이터 수집이 완료되도록 보장하기 위해.
 - d. 데이터 없이 이론적 틀에만 집중하기 위해.
5. 연구 접근 방식의 맥락에서 데이터 우선은 무엇을 의미합니까 (하나 선택)?
 - a. 데이터 가용성을 고려하지 않고 연구 질문을 개발하는 것.
 - b. 미리 정의된 질문에 답하기 위해 특별히 설계된 새로운 데이터를 수집하는 것.
 - c. 경험적 증거보다 이론적 틀을 우선시하는 것.
 - d. 사용 가능한 데이터로 시작하여 답변할 수 있는 질문을 결정하는 것.
6. 데이터 우선 접근 방식의 장점은 무엇입니까 (하나 선택)?
 - a. 이론적 틀의 필요성을 없앱니다.
 - b. 연구자가 사용 가능한 데이터를 기반으로 질문을 공식화할 수 있도록 합니다.
 - c. 인과 관계를 설정할 수 있음을 보장합니다.
 - d. 연구에서 어떤 형태의 편향도 방지합니다.
7. 데이터 우선 접근 방식의 단점은 무엇입니까 (하나 선택)?
 - a. “가로등 아래에서 검색”하는 것에 대한 우려.
 - b. 이론에 기여할 수 있는 능력에 대한 우려.
 - c. 인과 관계를 파악하기 어려울 것이라는 우려.
 - d. 외부 타당성에 대한 우려.
8. 반사실은 무엇입니까 (예시 및 참조를 포함하고 최소 세 단락으로 작성)?
9. 반사실은 무엇입니까 (하나 선택)?
 - a. 주 이론과 모순되는 대안 가설.
 - b. 만약이 거짓인 만약-그러면 문장.
 - c. 논문의 주요 주장에 반대되는 사실.
 - d. 교란 변수를 조정하는 데 사용되는 통계 방법.
10. “FINER” 프레임워크는 무엇을 의미합니까 (하나 선택)?
 - a. 유연하고, 혁신적이며, 중립적이고, 경험적이며, 복제 가능합니다.
 - b. 공식적이고, 해석적이며, 새롭고, 실험적이며, 견고합니다.
 - c. 집중적이고, 통합적이며, 자연적이고, 효율적이며, 신뢰할 수 있습니다.
 - d. 실현 가능하고, 흥미롭고, 새롭고, 윤리적이며, 관련성 있습니다.
11. 추정량은 무엇입니까 (하나 선택)?
 - a. 오류로 측정되는 변수.
 - b. 편향된 추정기.
 - c. 추정치를 계산하기 위해 데이터를 사용하는 과정.
 - d. 우리가 추정하고자 하는 진정한 효과 또는 관심량.
12. 추정량은 무엇입니까 (하나 선택)?
 - a. 관측된 데이터를 기반으로 주어진 양의 추정치를 계산하는 규칙.
 - b. 탐구 대상.
 - c. 특정 데이터 세트와 접근 방식이 주어졌을 때의 결과.
13. 추정기는 무엇입니까 (하나 선택)?
 - a. 관측된 데이터를 기반으로 주어진 양의 추정치를 계산하는 규칙.
 - b. 탐구 대상.
 - c. 특정 데이터 세트와 접근 방식이 주어졌을 때의 결과.
14. 추정기의 역할은 무엇입니까 (하나 선택)?
 - a. 우리가 추정하고자 하는 진정한 효과입니다.
 - b. 데이터에서 추정치를 계산하기 위한 규칙 또는 방법입니다.

- c. 데이터 세트와 방법이 주어졌을 때 계산된 값입니다.
 - d. 통계 모델의 오차 항입니다.
15. 추정치는 무엇입니까 (하나 선택)?
- a. 관측된 데이터를 기반으로 주어진 양의 추정치를 계산하는 규칙.
 - b. 탐구 대상.
 - c. 특정 데이터 세트와 접근 방식이 주어졌을 때의 결과.
16. 선택 편향은 무엇입니까 (하나 선택)?
- a. 참가자가 시간이 지남에 따라 연구에서 이탈할 때.
 - b. 데이터가 측정되는 방식에 따라 결과가 영향을 받을 때.
 - c. 표본이 모집단을 대표하지 않을 때.
 - d. 실험에서 변수가 제대로 통제되지 않을 때.
17. 측정 편향은 무엇입니까 (하나 선택)?
- a. 장비 고장으로 인해 데이터가 부정확하게 기록될 때.
 - b. 데이터 수집 방법이 체계적으로 실제 값을 과대평가하거나 과소평가할 때.
 - c. 측정 과정이 결과에 영향을 미칠 때.
 - d. 결론을 내리기에 표본 크기가 너무 작을 때.
18. 방향성 비순환 그래프(DAG)의 목적은 무엇입니까 (하나 선택)?
- a. 복잡한 모집단에서 무작위 표본을 생성하기 위해.
 - b. 비선형 데이터에 대한 통계 테스트를 수행하기 위해.
 - c. 통계 모델을 자동으로 생성하기 위해.
 - d. 변수 간의 인과 관계를 시각적으로 나타내기 위해.
19. DAG를 구축하는 이점은 무엇입니까 (하나 선택)?
- a. 데이터에서 인과 관계를 자동으로 식별합니다.
 - b. 통계 분석의 필요성을 없앱니다.
 - c. 연구자가 변수 관계에 대해 신중하게 생각하도록 돋습니다.
 - d. 인과 효과에 대한 정확한 추정치를 제공합니다.
20. 교란 변수는 무엇입니까 (하나 선택)?
- a. 예측 변수와 결과 변수 모두에 의해 영향을 받는 변수.
 - b. 예측 변수와 결과 변수 모두에 영향을 미치는 변수.
 - c. 예측 변수에 의해 영향을 받고 결과 변수에 영향을 미치는 변수.
21. 매개 변수는 무엇입니까 (하나 선택)?
- a. 예측 변수와 결과 변수 모두에 의해 영향을 받는 변수.
 - b. 예측 변수와 결과 변수 모두에 영향을 미치는 변수.
 - c. 예측 변수에 의해 영향을 받고 결과 변수에 영향을 미치는 변수.
22. 충돌 변수는 무엇입니까 (하나 선택)?
- a. 예측 변수와 결과 변수 모두에 의해 영향을 받는 변수.
 - b. 예측 변수와 결과 변수 모두에 영향을 미치는 변수.
 - c. 예측 변수에 의해 영향을 받고 결과 변수에 영향을 미치는 변수.
23. (zinsser의?) 2장에 따르면, 좋은 글쓰기의 비결은 무엇입니까 (하나 선택)?
- a. 올바른 문장 구조와 문법.
 - b. 긴 단어, 부사, 수동태 사용.
 - c. 모든 문장을 가장 깔끔한 구성 요소로 벗겨내는 것.
 - d. 철저한 계획.
24. (zinsser의?) 2장에 따르면, 작가는 끊임없이 무엇을 물어야 합니까 (하나 선택)?
- a. 누구를 위해 글을 쓰고 있는가?
 - b. 무엇을 말하려고 하는가?
 - c. 이것을 어떻게 다시 쓸 수 있는가?
 - d. 이것이 왜 중요한가?
25. 논문 작성 과정에서 중요한 과제 중 하나는 무엇입니까 (하나 선택)?
- a. 글쓰기를 시작하기 전에 가능한 한 많은 데이터를 수집하는 것.
 - b. 초고에서 각 문장을 완벽하게 만드는 데 많은 시간을 할애하는 것.
 - c. 가능한 한 빨리 초고를 작성하는 것.
 - d. 글쓰기 전에 상세한 그래프와 표를 만드는 데 집중하는 것.
26. 초고를 빨리 작성하는 것이 왜 중요합니까 (하나 선택)?
- a. 초기 초고에서 실수가 발생하지 않도록 보장합니다.
 - b. 수정하고 개선할 수 있는 완전한 버전을 제공합니다.

- c. 작가가 진행하면서 각 문장을 완벽하게 만들 수 있도록 합니다.
 - d. 글쓰기에 소요되는 전체 시간을 줄입니다.
27. “사랑하는 것을 죽여라”는 무엇을 의미합니까 (하나 선택)?
- a. 논란이 되는 주제에 대해 글을 쓰는 것을 피하기 위해.
 - b. 작업을 개선하기 위해 가혹한 비판을 사용하기 위해.
 - c. 좋아하지만 주요 이야기에 도움이 되지 않는 불필요한 내용을 제거하기 위해.
 - d. 전체 초고를 처음부터 다시 작성하기 위해.
28. 독자에게 초점을 맞출 때도 작가에게 글쓰기의 주요 이점 중 하나는 무엇입니까 (하나 선택)?
- a. 작가가 논문을 다시 작성하는 것을 피할 수 있도록 합니다.
 - b. 작가가 자신이 무엇을 믿고 어떻게 그것을 믿게 되었는지 알아내는 데 도움이 됩니다.
 - c. 동료로부터 필요한 피드백 양을 줄입니다.
 - d. 작가의 작품이 출판되도록 보장합니다.
29. 예를 들어 3장에서 (zinsser의?) 조언을 특징짓는 두 개의 반복되는 단어는 무엇입니까 (하나 선택)?
- a. 다시 쓰기, 다시 쓰기.
 - b. 단순화, 단순화.
 - c. 제거, 제거.
 - d. 적게, 적게.
30. 논문에서 불필요한 단어, 오타 및 문법 오류를 제거하는 주된 이유는 무엇입니까 (하나 선택)?
- a. 단어 수 제한을 충족하기 위해.
 - b. 고급 어휘로 검토자를 감동시키기 위해.
 - c. 논문을 더 길게 만들기 위해.
 - d. 주장의 신뢰성을 높이기 위해.
31. 다음 중 가장 좋은 제목은 무엇입니까 (하나 선택)?
- a. “작은 표본에서 추정치의 표준 오차”
 - b. “표준 오차”
 - c. “문제 세트 2”
32. 제목을 작성하는 한 가지 전략은 무엇입니까 (하나 선택)?
- a. 전문 용어를 사용하여 전문가 독자를 감동시키기 위해.
 - b. 일반적인 주제와 주요 발견에 대한 구체적인 정보를 모두 포함합니다.
 - c. 한두 단어만 사용하여 가능한 한 짧게 만듭니다.
 - d. 독자를 참여시키기 위해 질문 형식으로 제목을 제시합니다.
33. (fourcade2017seeing에?) 대한 새 제목을 작성하십시오.
34. 다음 중 초록을 작성할 때 권장되지 않는 것은 무엇입니까 (하나 선택)?
- a. 핵심 요점을 설명하기 위해 그림이나 표를 추가하는 것.
 - b. 주요 결과 및 함의를 포함하는 것.
 - c. 정확하고 간결한 언어를 사용하는 것.
 - d. 초록을 자체 포함되도록 만드는 것.
35. 초록을 작성하는 일반적인 구조는 무엇입니까 (하나 선택)?
- a. 함의로 시작하여 방법, 그리고 맥락으로 끝납니다.
 - b. 일반적인 영역에 대한 첫 문장, 방법에 대한 두 번째 문장, 주요 결과에 대한 세 번째 문장, 함의에 대한 네 번째 문장.
 - c. 한계로 시작하여 데이터 출처, 그리고 결과로 이어집니다.
 - d. 논문이 답변할 일련의 질문.
36. XKCD Simple Writer²에 따르면, 영어에서 가장 많이 사용되는 1,000개의 단어만 사용하여 (chambliss1989mundanity의?) 초록을 원래 의미를 유지하면서 다시 작성하십시오.
37. (king2006publication에?) 따르면, 소제목의 핵심 과제는 무엇입니까 (하나 선택)?
- a. 논문을 문헌에 통합하기 위해 약어를 사용합니다.
 - b. 독자가 논문의 중요성에 감동하도록 광범위하고 포괄적으로 작성합니다.
 - c. 무작위로 잠들었지만 계속 페이지를 넘기는 독자가 자신이 어디에 있는지 알 수 있도록 합니다.
38. 데이터 섹션에서 무엇을 달성하고 싶습니까 (하나 선택)?
- a. 독자를 감동시키기 위해 데이터의 복잡성을 보여주기 위해.
 - b. 데이터를 철저히 설명하여 장소 감각을 만들기 위해.

²<https://xkcd.com/simplewriter/>

- c. 가능한 한 많은 그래프와 표를 포함하기 위해.
 - d. 데이터의 약점을 숨기기 위해.
39. 연구 논문에서 데이터 섹션의 주요 목표는 무엇입니까 (하나 선택)?
- a. 가능한 한 많은 표와 그래프를 제시하기 위해.
 - b. 독자가 결과의 기반을 이해할 수 있도록 데이터를 철저히 설명하기 위해.
 - c. 분석의 복잡성을 독자에게 설득하기 위해.
 - d. 사용되지 않은 데이터 소스까지 모든 가능한 데이터 소스를 논의하기 위해.
40. (king2006publication에?) 따르면, 표준 오차가 0.05라면 계수에 대한 다음 구체성 중 어떤 것 이 어리석을까요 (모두 선택)?
- a. 2.7182818
 - b. 2.718282
 - c. 2.71828
 - d. 2.7183
 - e. 2.718
 - f. 2.72
 - g. 2.7
 - h. 3
41. 좋은 그림 또는 표 캡션은 무엇을 달성해야 합니까 (하나 선택)?
- a. 가능한 한 간결하게, 이상적으로는 한 줄로.
 - b. 자체 포함되어 주요 요점을 설명합니다.
 - c. 전문 지식을 보여주기 위해 복잡한 전문 용어를 포함합니다.
 - d. 독자가 텍스트를 읽도록 장려하기 위해 최소한의 정보만 제공합니다.
42. 모델 섹션에는 무엇이 있어야 합니까 (하나 선택)?
- a. 문헌에서 사용된 다른 모델 요약.
 - b. 방정식 없이 최종 결과만.
 - c. 수학적 표기법 없는 일반적인 설명.
 - d. 모든 구성 요소의 방정식, 설명 및 정의.
43. 모델 섹션에서 대체 모델 또는 변형을 논의하는 것이 왜 중요합니까 (하나 선택)?
- a. 철저한 고려를 보여주고 선택한 모델을 정당화하기 위해.
 - b. 다른 모델이 열등하다는 것을 보여주기 위해.
 - c. 여러 옵션으로 독자를 혼란시키기 위해.
 - d. 논문의 길이를 늘리기 위해.
44. 결과 섹션의 목적은 무엇입니까 (하나 선택)?
- a. 결과를 해석하고 그 함의를 논의하는 것.
 - b. 다른 연구자들의 발견을 비판하는 것.
 - c. 미래 연구 방향을 제안하는 것.
 - d. 광범위한 해석 없이 분석 결과를 명확하게 제시하는 것.
45. 결과 섹션에서 그래프와 표는 어떻게 통합되어야 합니까 (하나 선택)?
- a. 함께 제공되는 텍스트 없이 독립적으로 존재해야 합니다.
 - b. 혼란을 피하기 위해 최소화되어야 합니다.
 - c. 상호 참조되어 텍스트에서 논의되어야 합니다.
 - d. 흐름을 방해하지 않기 위해 부록에 배치되어야 합니다.
46. 논의 섹션의 목적은 무엇입니까 (하나 선택)?
- a. 결과를 더 자세히 반복하기 위해.
 - b. 상세한 방법론을 제공하기 위해.
 - c. 결과를 해석하고, 함의를 논의하고, 약점을 인정하기 위해.
 - d. 해결책을 제시하지 않고 연구의 모든 한계를 나열하기 위해.
47. (Savage2019는?) 해당 구두점, 단어, 문장, 단락 또는 섹션 없이 원래 메시지를 보존할 수 있는지 스스로에게 물어보라고 권장하는 이유는 무엇입니까 (하나 선택)?
- a. 오류 가능성 줄이기 위해.
 - b. 명확성을 달성하기 위해.
 - c. 논문을 짧게 유지하기 위해.
- d. 다시 작성 과정의 핵심 측면은 무엇입니까 (모두 선택)?
- e. 불필요한 단어를 제거하기 위해 빨간 펜으로 검토하는 것.
 - f. 논문을 인쇄하고 실제 사본을 읽는 것.
 - g. 흐름을 향상시키기 위해 잘라내고 붙여넣는 것.

- h. 소리 내어 읽는 것.
 i. 다른 사람과 교환하는 것.
48. 문법 오류와 오타가 논문의 신뢰성에 영향을 미치는 이유는 무엇입니까 (하나 선택)?
 a. 논문의 캐주얼한 어조를 향상시킵니다.
 b. 논문을 더 짧게 만듭니다.
 c. 세부 사항에 대한 주의 부족을 나타냅니다.
 d. 내용이 좋다면 허용됩니다.

수업 활동

- 연구에 대한 선호하는 접근 방식(데이터 우선/질문 우선/기타)과 그 이유를 논의하십시오.
- 예시를 참조하여 추정량, 추정기 및 추정치가 무엇인지 설명하십시오.
- “선택 편향”을 고려하고, (monicababynames가?) 지니 계수에 대해 설명한 것과 같은 방식으로 한 문장으로 정의를 포함하십시오.
- ChatGPT 또는 동등한 LLM을 사용하여 “선택 효과는 무엇입니까?”라는 질문에 답하는 프롬프트를 만드십시오. 파트너와 함께 컨텍스트, 참조를 추가하고 (필요한 경우) 사실로 만들어 응답을 개선하십시오. 세 가지 측면을 논의하십시오: 1) 프롬프트, 2) 원본 답변, 3) 개선된 답변.
- 잘 쓰여진 정량 논문 중 하나를 선택하십시오.
 - 원본 제목을 작성하십시오. 무엇이 좋고, 무엇이 좋지 않습니까? 대체 제목을 작성하십시오.
 - 초록을 작성하십시오. 무엇이 좋고, 무엇이 좋지 않습니까?
 - ChatGPT 또는 동등한 LLM에게 대체 초록을 만들도록 프롬프트하십시오 (논의할 수 있도록 프롬프트를 복사하십시오).
 - 이 모든 것을 바탕으로 개선된 초록을 작성한 다음 모든 것을 논의하십시오.
- (king2006publication을?) 기반으로 이 수업이 끝날 때까지 의미 있는 논문을 작성하기 위한 계획을 세우십시오. (박사 과정 학생의 경우: 제출할 세 개의 저널/학술 대회를 순서대로 자세히 설명하고, 각 저널/학술 대회에 논문이 적합한 이유를 설명하십시오.)
- 논문 검토: (Gerring2012를?) 읽고 한 페이지 분량의 검토를 작성하십시오.

과제

Caro (2019, p. xii)은 거의 매일 최소 1,000단어를 씁니다. 이 과제의 목적은 당신에게도 그렇게 할 기회를 주는 것입니다. 선행 조건에 명시된 논문 중 하나를 선택하고 다음 작업을 완료하십시오.

- 1일차: 모든 단어를 직접 작성하여 전체 서론을 필사하십시오.
- 2일차: 서론을 5줄 (또는 10%, 더 적은 쪽) 더 짧게 다시 작성하십시오.
- 3일차: 모든 단어를 직접 작성하여 초록을 필사하십시오.
- 4일차: 논문에 대한 새로운 네 문장짜리 초록을 다시 작성하십시오.
- 5일차: 여기³에 정의된 영어에서 가장 많이 사용되는 1,000개의 단어만 사용하여 새 초록의 두 번째 버전을 작성하십시오.
- 6일차: 논문이 작성된 방식 중 마음에 드는 세 가지 점을 자세히 설명하십시오.
- 7일차: 논문이 작성된 방식 중 마음에 들지 않는 한 가지 점을 자세히 설명하십시오.

Quarto를 사용하여 일주일 동안 단일 PDF를 만드십시오. 매일 작업 후 정보가 담긴 커밋 메시지와 함께 작업을 커밋하고 푸시하십시오.

³<https://xkcd.com/simplewriter/>

6

그래프, 표, 지도

i Chapman and Hall/CRC는 이 책을 2023년 7월에 출판했습니다. 여기^a에서 구매할 수 있습니다. 이 온라인 버전은 인쇄된 내용에 일부 업데이트가 있습니다.

^a<https://www.routledge.com/Telling-Stories-with-Data-With-Applications-in-R/Alexander/p/book/9781032134772>

선행 조건

- R for Data Science 읽기, (Wickham, Çetinkaya-Rundel, 와/과 Grolemund [2016년] 2023)
 - ggplot2에 대한 개요를 제공하는 1장 “데이터 시각화”에 집중하십시오.
- Data Visualization: A Practical Introduction 읽기, (Healy 2018)
 - 다른 강조점을 가진 ggplot2에 대한 개요를 제공하는 3장 “플롯 만들기”에 집중하십시오.
- The Glamour of Graphics 시청, (Chase 2020)
 - 이 비디오는 ggplot2로 만든 플롯을 개선하는 방법에 대한 아이디어를 자세히 설명합니다.
- Testing Statistical Charts: What Makes a Good Graph? 읽기, (Vanderplas, Cook, 와/과 Hofmann 2020)
 - 이 기사는 그래프를 만드는 모범 사례를 자세히 설명합니다.
- Data Feminism 읽기, (D'Ignazio 와/과 Klein 2020)
 - 데이터가 맥락 내에서 고려되어야 하는 이유에 대한 예시를 제공하는 3장 “신화적이고, 상상적이며, 불가능한 관점에서 본 합리적이고, 과학적이며, 객관적인 관점”에 집중하십시오.
- Historical development of the graphical representation of statistical data 읽기, (Funkhouser 1937)
 - 이 기사는 그래프가 어떻게 발전했는지 논의하는 2장 “그래픽 방법의 기원”에 집중하십시오.
- Remove the legend to become one 읽기, (Wei 2017)
 - 그래프를 점진적으로 개선하는 과정을 설명합니다. 모든 것이 흥미롭지만, 그래프 측면은 “이것이 선 그래프와 무슨 관련이 있습니까?”로 시작합니다.
- Geocomputation with R, 2장 “R의 지리 데이터” 읽기, (Lovelace, Nowosad, 와/과 Muenchow 2019)
 - 이 장은 R에서 매핑에 대한 개요를 제공합니다.
- Mastering Shiny, 1장 “첫 번째 Shiny 앱” 읽기, (Wickham 2021b)
 - 이 장은 Shiny 앱의 자체 포함된 예시를 제공합니다.

주요 개념 및 기술

- 시각화는 데이터를 이해하고 독자에게 전달하는 한 가지 방법입니다. 데이터 세트의 관측치를 플로팅하는 것이 중요합니다.
- 막대 차트, 산점도, 선 플롯 및 히스토그램을 포함한 다양한 그래프 유형에 익숙해야 합니다. 지오코딩된 데이터가 있는 경우 지도를 일종의 그래프로 간주할 수도 있습니다.
- 표를 사용하여 데이터를 요약해야 합니다. 일반적인 사용 사례에는 데이터 세트의 일부, 요약 통계 및 회귀 결과 표시가 포함됩니다.

소프트웨어 및 패키지

- babynames (Wickham 2021a)
- Base R (R Core Team 2024)
- carData (Fox, Weisberg, 와/과 Price 2022)
- datasauRus (Davies, Locke, 와/과 D'Agostino McGowan 2022)
- ggmap (Kahle 와/과 Wickham 2013)

- `janitor` (Firke 2023)
- `knitr` (Xie 2023)
- `leaflet` (Cheng, Karambelkar, 와/과 Xie 2021)
- `mapdeck` (Cooley 2020)
- `maps` (Becker 기타 2022)
- `mapproj` (McIlroy 기타 2023)
- `modelsummary` (Arel-Bundock 2022)
- `opendatatoronto` (Gelfand 2022b)
- `patchwork` (Pedersen 2022)
- `shiny` (Chang 기타 2021)
- `tidygeocoder` (Cambon 와/과 Belanger 2021)
- `tidyverse` (Wickham 기타 2019)
- `tinytable` (Arel-Bundock 2024)
- `troopdata` (Flynn 2022)
- `usethis` (Wickham, Bryan, 와/과 Barrett 2022)
- `WDI` (Arel-Bundock 2021)

```
library(babynames)
library(carData)
library(datasauRus)
library(ggmap)
library(janitor)
library(knitr)
library(leaflet)
library(mapdeck)
library(maps)
library(mapproj)
library(modelsummary)
library(opendatatoronto)
library(patchwork)
library(tidygeocoder)
library(tidyverse)
library(tinytable)
library(troopdata)
library(shiny)
library(usethis)
library(WDI)
```

6.1 서론

데이터로 이야기를 전달할 때, 우리는 데이터가 독자를 설득하는 데 많은 역할을 하기를 바랍니다. 논문은 매개체이고, 데이터는 메시지입니다. 이를 위해 우리는 독자에게 우리가 이야기를 이해하게 된 기반이 된 데이터를 보여주고 싶습니다. 이를 달성하기 위해 그래프, 표, 지도를 사용합니다.

분석의 기반이 되는 관측치를 보여주려고 노력하십시오. 예를 들어, 데이터 세트가 2,500개의 설문조사 응답으로 구성되어 있다면, 논문의 어느 시점에서는 관심 있는 모든 변수에 대해 2,500개의 관측치 각각을 포함하는 플롯이 있어야 합니다. 이를 위해 우리는 `ggplot2`를 사용하여 그래프를 만듭니다. `ggplot2`는 핵심 `tidyverse`의 일부이므로 별도로 설치하거나 로드할 필요가 없습니다. 이 장에서는 막대 차트, 산점도, 선 플롯 및 히스토그램을 포함한 다양한 옵션을 살펴봅니다.

각 관측치를 보여주는 그래프의 역할과 달리, 표의 역할은 일반적으로 데이터 세트의 일부를 보여주거나 다양한 요약 통계 또는 회귀 결과를 전달하는 것입니다. 우리는 주로 `knitr`를 사용하여 표를 만들 것입니다. 나중에 `modelsummary`를 사용하여 회귀 출력과 관련된 표를 만들 것입니다.

마지막으로, 특정 유형의 데이터를 보여주는 데 사용되는 그래프의 변형인 지도를 다룹니다. `tidygeocoder`를 사용하여 지오코딩된 데이터를 얻은 후 `ggmap`을 사용하여 정적 지도를 만들 것입니다.

6.2 그래프

더 건전하고 풍요로운 문명으로 나아가는 세상은 차트로 나아가는 세상이 될 것입니다.

Karsten (1923, p. 684)

그래프는 설득력 있는 데이터 스토리의 중요한 측면입니다. 그래프는 넓은 패턴과 세부 사항을 모두 볼 수 있게 해줍니다 (Cleveland [1985년] 1994, p. 5). 그래프는 다른 어떤 방법으로도 얻기 어려운 데이터에 대한 친숙함을 가능하게 합니다. 관심 있는 모든 변수는 그래프로 나타내야 합니다.

그래프의 가장 중요한 목표는 실제 데이터와 그 맥락을 가능한 한 많이 전달하는 것입니다. 어떤 면에서 그래프는 정보 인코딩 과정이며, 우리는 청중에게 정보를 전달하기 위해 의도적인 표현을 구성합니다. 청중은 그 표현을 디코딩해야 합니다. 우리 그래프의 성공은 이 과정에서 얼마나 많은 정보가 손실되는지에 달려 있으므로 디코딩은 중요한 측면입니다 (Cleveland [1985년] 1994, p. 221). 이는 우리가 특정 청중에 적합한 효과적인 그래프를 만드는데 집중해야 함을 의미합니다.

그래프가 왜 중요한지 이해하기 위해 `datasaurus`를 설치하고 로드한 후 `datasaurus_dozen` 데이터 세트를 고려해 보십시오.

datasaurus_dozen

```
# A tibble: 1,846 x 3
  dataset     x     y
  <chr>   <dbl> <dbl>
1 dino      55.4  97.2
2 dino      51.5  96.0
3 dino      46.2  94.5
4 dino      42.8  91.4
5 dino      40.8  88.3
6 dino      38.7  84.9
7 dino      35.6  79.9
8 dino      33.1  77.6
9 dino      29.0  74.5
10 dino     26.2  71.4
# i 1,836 more rows
```

데이터 세트는 “x”와 “y” 값을 포함하며, 각각 x축과 y축에 플로팅되어야 합니다. “dataset” 변수에는 “dino”, “star”, “away”, “bullseye”를 포함하여 13가지 다른 값이 있습니다. 우리는 이 네 가지에 초점을 맞추고 각각에 대한 요약 통계를 생성합니다 (표 ??).

```
# Based on: https://juliasilge.com/blog/datasaurus-multiclass/
datasaurus_dozen |>
  filter(dataset %in% c("dino", "star", "away", "bullseye")) |>
  summarise(across(c(x, y), list(mean = mean, sd = sd)),
            .by = dataset) |>
```

표 6.1: 네 가지 datasauRus 데이터 세트에 대한 평균 및 표준 편차

| 데이터 세트 | x 평균 | x 표준 편차 | y 평균 | y 표준 편차 |
|----------|------|---------|------|---------|
| dino | 54.3 | 16.8 | 47.8 | 26.9 |
| away | 54.3 | 16.8 | 47.8 | 26.9 |
| star | 54.3 | 16.8 | 47.8 | 26.9 |
| bullseye | 54.3 | 16.8 | 47.8 | 26.9 |

```
tt() |>
style_tt(j = 2:5, align = "r") |>
format_tt(digits = 1, num_fmt = "decimal") |>
setNames(c("평균", "x 표준", "x 표준 편차", "y 표준", "y 표준 편차"))
```

요약 통계가 유사하다는 점에 유의하십시오 (표 ??). 그럼에도 불구하고 다른 데이터 세트는 실제로는 매우 다른 특성을 가지고 있습니다. 이는 데이터를 플로팅할 때 명확해집니다 (그림 ??).

```
datasaurus_dozen |>
filter(dataset %in% c("dino", "star", "away", "bullseye")) |>
ggplot(aes(x = x, y = y, colour = dataset)) +
geom_point() +
theme_minimal() +
facet_wrap(vars(dataset), nrow = 2, ncol = 2) +
labs(color = "데이터 세트")
```

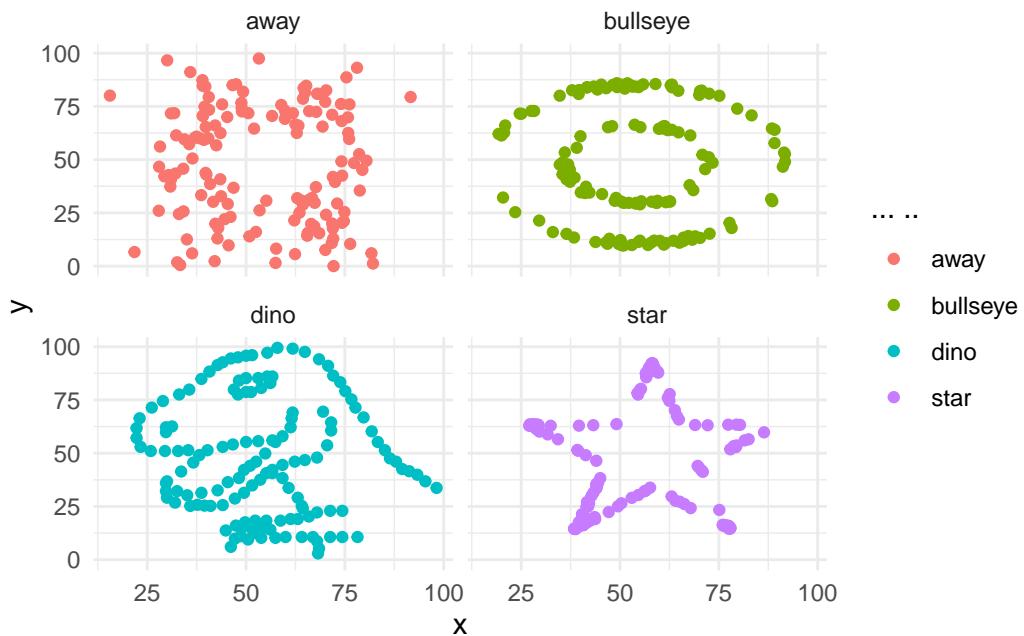


그림 6.1: 네 가지 datasauRus 데이터 세트의 그래프

우리는 20세기 통계학자 프랭크 앤스콤이 만든 “앤스콤의 콰르텟”에서 유사한 교훈을 얻습니다. 즉, 요약 통계에만 의존하지 않고 실제 데이터를 플로팅하는 것이 중요합니다.

표 6.2: 앤스콤의 콰르텟에 대한 평균 및 표준 편차

| 데이터 세트 | x 평균 | x 표준 편차 | y 평균 | y 표준 편차 |
|--------|------|---------|------|---------|
| 1 | 9 | 3.3 | 7.5 | 2 |
| 2 | 9 | 3.3 | 7.5 | 2 |
| 3 | 9 | 3.3 | 7.5 | 2 |
| 4 | 9 | 3.3 | 7.5 | 2 |

```
head(anscombe)
```

```
x1 x2 x3 x4   y1   y2   y3   y4
1 10 10 10 8 8.04 9.14 7.46 6.58
2  8  8  8 8 6.95 8.14 6.77 5.76
3 13 13 13 8 7.58 8.74 12.74 7.71
4  9  9  9 8 8.81 8.77 7.11 8.84
5 11 11 11 8 8.33 9.26 7.81 8.47
6 14 14 14 8 9.96 8.10 8.84 7.04
```

앤스콤의 콰르텟은 4개의 다른 데이터 세트에 대한 11개의 관측치로 구성되며, 각 관측치에 대한 x 및 y 값이 있습니다. “R 필수 사항” 온라인 부록¹에서 논의된 “정돈된” 형식으로 만들기 위해 `pivot_longer()`를 사용하여 이 데이터 세트를 조작해야 합니다.

```
# From: https://www.njtierney.com/post/2020/06/01/tidy-anscombe/
# And the pivot_longer() vignette.
```

```
tidy_anscombe <-
  anscombe |>
  pivot_longer(
    everything(),
    names_to = c(".value", "set"),
    names_pattern = "(.)(.)"
  )
```

먼저 요약 통계를 생성한 다음 (표 ??) 데이터를 플로팅할 수 있습니다 (그림 ??). 이는 요약 통계에만 의존하지 않고 실제 데이터를 그래프로 나타내는 것의 중요성을 다시 한번 보여줍니다.

```
tidy_anscombe |>
  summarise(
    across(c(x, y), list(mean = mean, sd = sd)),
    .by = set
  ) |>
  tt() |>
  style_tt(j = 2:5, align = "r") |>
  format_tt(digits = 1, num_fmt = "decimal") |>
  setNames(c("평균 표준", "x 평균", "x 표준 표준", "y 평균", "y 표준 표준"))
```

```
tidy_anscombe |>
  ggplot(aes(x = x, y = y, colour = set)) +
```

¹https://tellingstorieswithdata.com/20-r_essentials.html

```
geom_point() +
  geom_smooth(method = lm, se = FALSE) +
  theme_minimal() +
  facet_wrap(vars(set), nrow = 2, ncol = 2) +
  labs(colour = "설정 번호") +
  theme(legend.position = "bottom")
```

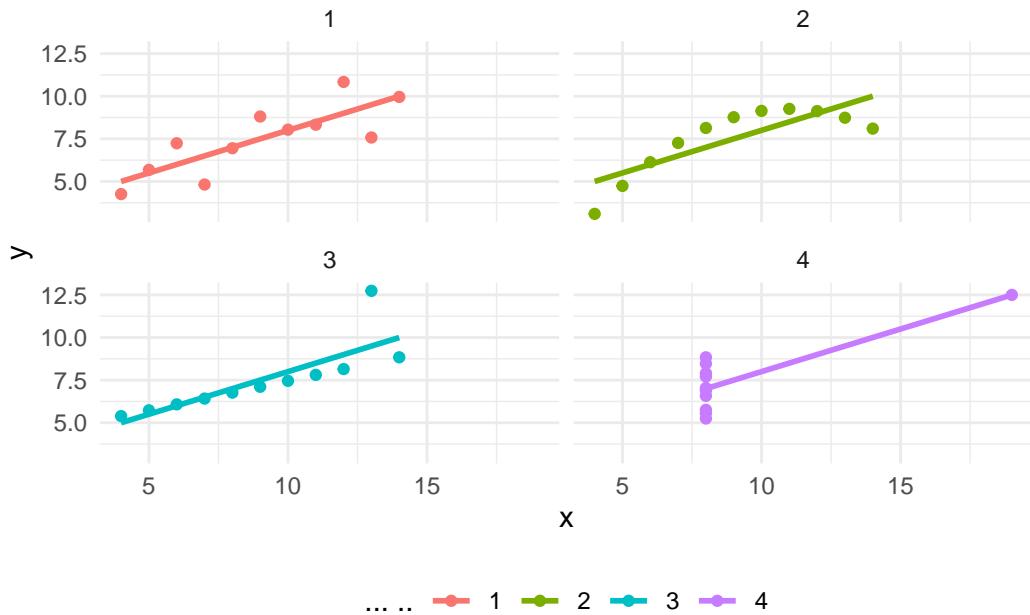


그림 6.2: 앤스콤의 콰르텟 재현

6.2.1 막대 차트

우리는 일반적으로 집중하고 싶은 범주형 변수가 있을 때 막대 차트를 사용합니다. `?@sec-fire-hose`에서 점유된 침대 수를 그래프로 만들 때 이 예시를 보았습니다. 우리가 주로 사용하는 기하학적 객체 (“geom”)은 `geom_bar()`이지만, 특정 상황에 맞는 많은 변형이 있습니다. 막대 차트 사용법을 설명하기 위해 `carData`를 설치하고 로드한 후 (`fox2006effect`가?) 수집하고 BEPS를 통해 제공한 1997-2001년 영국 선거 패널 연구 데이터 세트를 사용합니다.

```
beps <-
  BEPS |>
  as_tibble() |>
  clean_names() |>
  select(age, vote, gender, political_knowledge)
```

데이터 세트는 응답자가 지지하는 정당과 다양한 인구 통계학적, 경제적, 정치적 변수로 구성됩니다. 특히, 응답자의 연령 정보가 있습니다. 먼저 연령에서 연령 그룹을 만들고 `geom_bar()`을 사용하여 각 연령 그룹의 빈도를 보여주는 막대 차트를 만듭니다 (그림 ??).

```
beps <-
  beps |>
  mutate(
    age_group =
```

```

case_when(
  age < 35 ~ "<35",
  age < 50 ~ "35-49",
  age < 65 ~ "50-64",
  age < 80 ~ "65-79",
  age < 100 ~ "80-99"
),
age_group =
  factor(age_group, levels = c("<35", "35-49", "50-64", "65-79", "80-99"))
)

```

```

beps |>
  ggplot(mapping = aes(x = age_group)) +
  geom_bar() +
  theme_minimal() +
  labs(x = "연령 그룹", y = "인구 수")

beps |>
  count(age_group) |>
  ggplot(mapping = aes(x = age_group, y = n)) +
  geom_col() +
  theme_minimal() +
  labs(x = "연령 그룹", y = "인구 수")

```

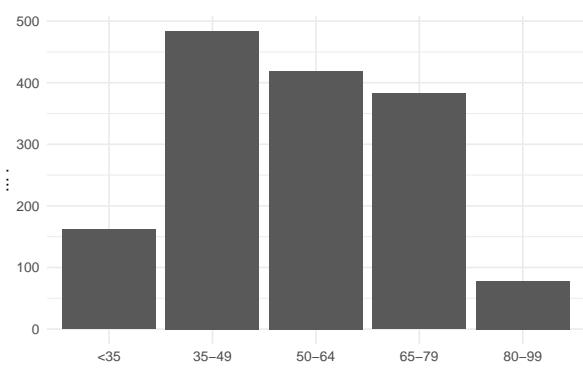
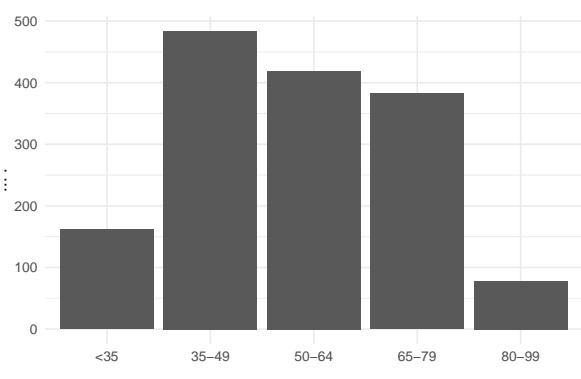
(a) `geom_bar()` 사용(b) `count()` 및 `geom_col()` 사용

그림 6.3: 1997-2001년 영국 선거 패널 연구의 연령 그룹 분포

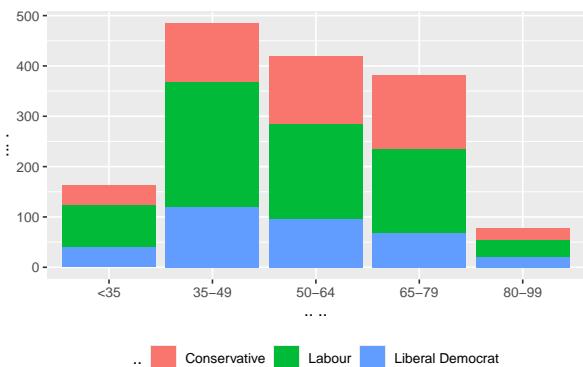
`ggplot2`에서 사용하는 기본 축 레이블은 관련 변수의 이름이므로, 더 자세한 내용을 추가하는 것이 종종 유용합니다. 우리는 `labs()`를 사용하여 변수와 이름을 지정하여 이를 수행합니다. `?@fig-bepfitst-1`의 경우 x축과 y축에 대한 레이블을 지정했습니다.

기본적으로 `geom_bar()`는 각 연령 그룹이 데이터 세트에 나타나는 횟수를 계산합니다. `geom_bar()`의 기본 통계 변환("stat")이 "count"이기 때문에 이렇게 합니다. 이는 우리가 직접 통계를 만들 필요가 없도록 해줍니다. 그러나 이미 카운트를 구성했다면 (예: `beps |> count(age_group)`), y축에 대한 변수를 지정하고 `geom_col()`을 사용할 수 있습니다 (그림 ??).

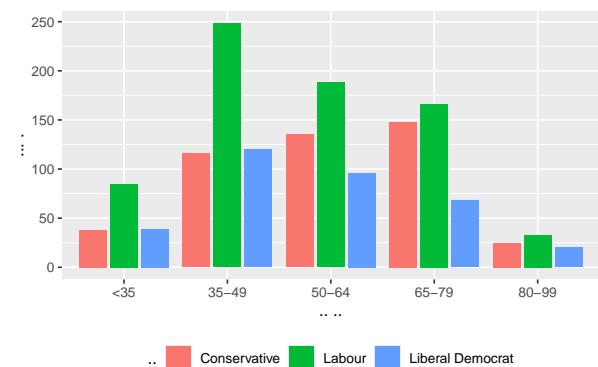
다른 통찰력을 얻기 위해 데이터를 다양한 그룹으로 나누어 고려할 수도 있습니다. 예를 들어, 색상을 사용하여 연령 그룹별로 응답자가 지지하는 정당을 살펴볼 수 있습니다 (그림 ??).

```
beps |>
  ggplot(mapping = aes(x = age_group, fill = vote)) +
  geom_bar() +
  labs(x = "연령 그룹", y = "투표 수", fill = "당派") +
  theme(legend.position = "bottom")

beps |>
  ggplot(mapping = aes(x = age_group, fill = vote)) +
  geom_bar(position = "dodge2") +
  labs(x = "연령 그룹", y = "투표 수", fill = "당派") +
  theme(legend.position = "bottom")
```



(a) geom_bar() 사용



(b) geom_bar()와 dodge2 사용

그림 6.4: 1997-2001년 영국 선거 패널 연구의 연령 그룹 및 투표 선호도 분포

기본적으로 이러한 다른 그룹은 쌓여 있지만, position = "dodge2"를 사용하여 나란히 배치할 수 있습니다 (그림 ??). ("dodge" 대신 "dodge2"를 사용하면 막대 사이에 약간의 공간이 추가됩니다.)

6.2.1.1 테마

이 시점에서 그래프의 전반적인 모양을 다루고 싶을 수 있습니다. ggplot2에는 다양한 테마가 내장되어 있습니다. 여기에는 theme_bw(), theme_classic(), theme_dark(), theme_minimal()가 포함됩니다. 전체 목록은 ggplot2 치트 시트²에서 확인할 수 있습니다. 이러한 테마는 레이어로 추가하여 사용할 수 있습니다 (그림 ??). ggthemes (Arnold 2021) 및 hrbrthemes (Rudis 2020)를 포함한 다른 패키지에서 더 많은 테마를 설치할 수도 있습니다. 심지어 우리만의 테마를 만들 수도 있습니다!

```
theme_bw <-
beps |>
  ggplot(mapping = aes(x = age_group)) +
  geom_bar(position = "dodge") +
  theme_bw()

theme_classic <-
beps |>
  ggplot(mapping = aes(x = age_group)) +
  geom_bar(position = "dodge") +
  theme_classic()

theme_dark <-
```

²<https://github.com/rstudio/cheatsheets/blob/main/data-visualization.pdf>

```
beps |>
ggplot(mapping = aes(x = age_group)) +
geom_bar(position = "dodge") +
theme_dark()

theme_minimal <-
beps |>
ggplot(mapping = aes(x = age_group)) +
geom_bar(position = "dodge") +
theme_minimal()

(theme_bw + theme_classic) / (theme_dark + theme_minimal)
```

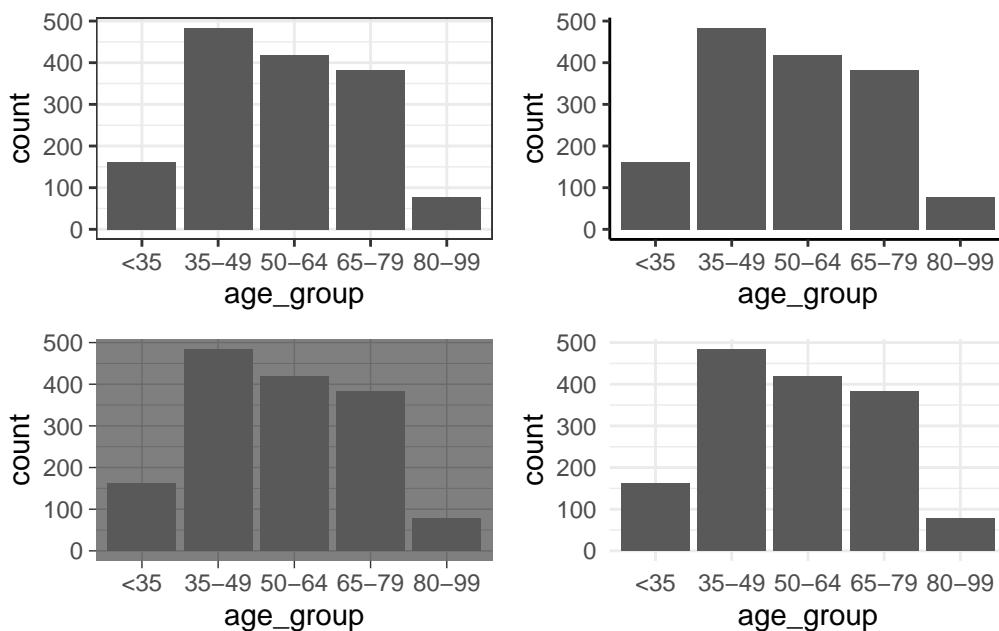


그림 6.5: 1997-2001년 영국 선거 패널 연구의 연령 그룹 및 투표 선호도 분포, 다양한 테마 및 patchwork 사용법 설명

?@fig-bepthemes에서 우리는 patchwork를 사용하여 여러 그래프를 함께 가져옵니다. 이를 위해 패키지를 설치하고 로드한 후 그래프를 변수에 할당합니다. 그런 다음 “+”를 사용하여 서로 옆에 있어야 할 것을 나타내고, “/”를 사용하여 위에 있어야 할 것을 나타내며, 괄호를 사용하여 우선 순위를 나타냅니다.

6.2.1.2 패싯

우리는 하나 이상의 변수를 기반으로 변화를 보여주기 위해 패싯을 사용합니다 (L. Wilkinson 2005, p. 219). 패싯은 다른 변수의 변화를 강조하기 위해 이미 색상을 사용했을 때 특히 유용합니다. 예를 들어, 연령과 성별에 따른 투표를 설명하는 데 관심이 있을 수 있습니다 (그림 ??). 겹침을 피하기 위해 guides(x = guide_axis(angle = 90))로 x축을 회전합니다. 또한 theme(legend.position = "bottom")으로 범례의 위치를 변경합니다.

```
beps |>
ggplot(mapping = aes(x = age_group, fill = gender)) +
geom_bar() +
theme_minimal() +
```

```

  labs(
    x = "영국 선거 패널",
    y = "투표 선호도",
    fill = "성별"
  ) +
  facet_wrap(vars(vote)) +
  guides(x = guide_axis(angle = 90)) +
  theme(legend.position = "bottom")

```

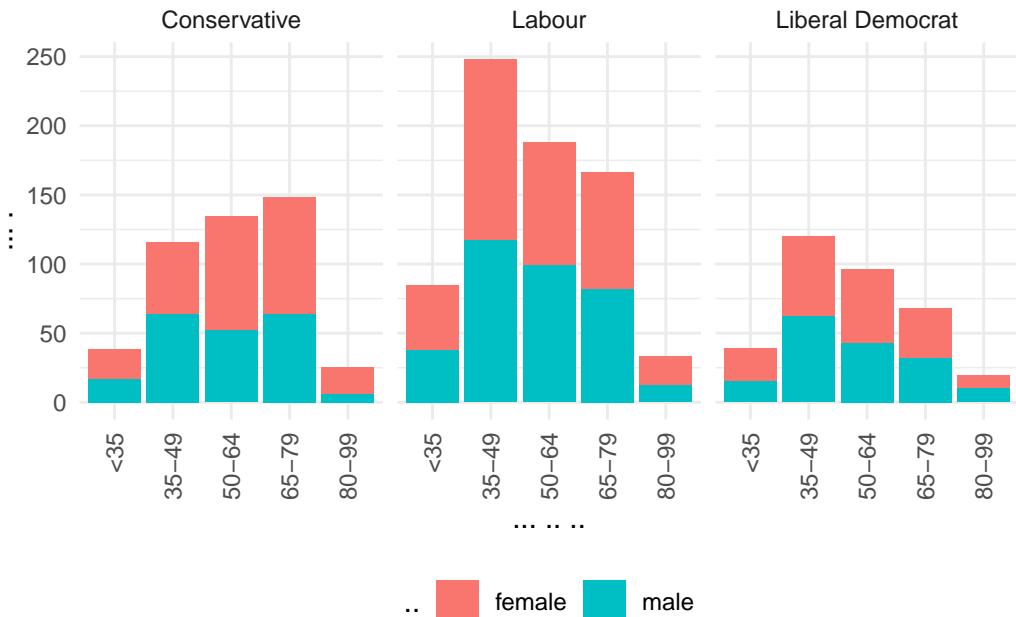


그림 6.6: 1997-2001년 영국 선거 패널 연구의 성별 연령 그룹 및 투표 선호도 분포

`facet_wrap()`을 `dir = "v"`로 변경하여 수평 대신 수직으로 래핑할 수 있습니다. 또는 `nrow = 2`와 같이 몇 줄을 지정하거나 `ncol = 2`와 같이 열 수를 지정할 수 있습니다.

기본적으로 두 패싯 모두 동일한 x축과 y축을 가집니다. `scales = "free"`를 사용하여 두 패싯 모두 다른 스케일을 갖도록하거나, `scales = "free_x"`를 사용하여 x축만, 또는 `scales = "free_y"`를 사용하여 y축만 갖도록 할 수 있습니다 (그림 ??).

```

beps |>
  ggplot(mapping = aes(x = age_group, fill = gender)) +
  geom_bar() +
  theme_minimal() +
  labs(
    x = "영국 선거 패널",
    y = "투표 선호도",
    fill = "성별"
  ) +
  facet_wrap(vars(vote), scales = "free") +
  guides(x = guide_axis(angle = 90)) +
  theme(legend.position = "bottom")

```



그림 6.7: 1997-2001년 영국 선거 패널 연구의 성별 연령 그룹 및 투표 선호도 분포

마지막으로, `labeller()`를 사용하여 패싯의 레이블을 변경할 수 있습니다 (그림 ??).

```

new_labels <-
  c("0" = "00 00", "1" = "00 00",
    "2" = "00 00", "3" = "00 00")

beps |>
  ggplot(mapping = aes(x = age_group, fill = vote)) +
  geom_bar() +
  theme_minimal() +
  labs(
    x = "000 00 00",
    y = "000 0",
    fill = "00"
  ) +
  facet_wrap(
    vars(political_knowledge),
    scales = "free",
    labeller = labeller(political_knowledge = new_labels)
  ) +
  guides(x = guide_axis(angle = 90)) +
  theme(legend.position = "bottom")
  
```

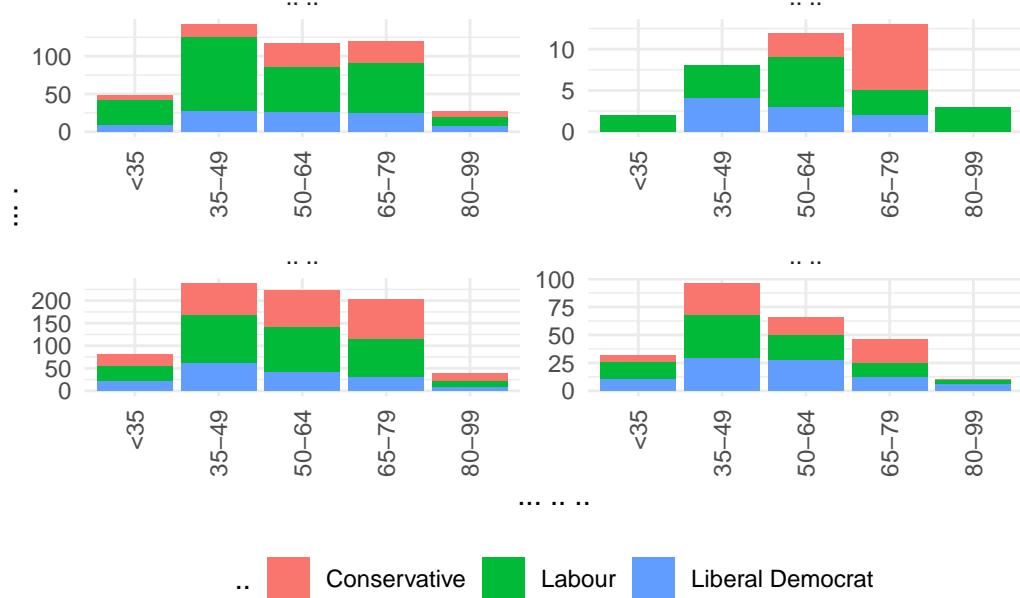


그림 6.8: 1997-2001년 영국 선거 패널 연구의 정치 지식별 연령 그룹 및 투표 선호도 분포

이제 여러 그래프를 결합하는 세 가지 방법이 있습니다. 하위 그림, 패싯, 그리고 patchwork입니다. 이들은 다른 상황에서 유용합니다.

- 하위 그림 - ?@sec-reproducible-workflows에서 다루었습니다 - 다른 변수를 고려할 때;
- 패싯 - 범주형 변수를 고려할 때; 그리고
- patchwork - 완전히 다른 그래프를 함께 가져오는 데 관심이 있을 때.

6.2.1.3 색상

이제 그래프에 사용된 색상으로 넘어갑니다. 색상을 변경하는 다양한 방법이 있습니다. RColorBrewer (Neuwirth 2022)에서 사용할 수 있는 많은 팔레트는 scale_fill_brewer()를 사용하여 지정할 수 있습니다. viridis (Garnier 기타 2021)의 경우 scale_fill_viridis_d()를 사용하여 팔레트를 지정할 수 있습니다. 또한 viridis는 색맹 팔레트에 특히 중점을 둡니다 (그림 ??). RColorBrewer와 viridis는 tidyverse의 일부인 ggplot2가 우리를 위해 처리하기 때문에 명시적으로 설치하거나 로드할 필요가 없습니다.

i 개인의 어깨

“brewer” 팔레트의 이름은 신디 브루어 (G. Miller 2014)를 의미합니다. 1991년 미시간 주립 대학교에서 지리학 박사 학위를 취득한 후 샌디에이고 주립 대학교에서 조교수로 근무했으며, 1994년 펜실베이니아 주립 대학교로 옮겨 2007년 정교수로 승진했습니다. 그녀의 가장 잘 알려진 책 중 하나는 Designing Better Maps: A Guide for GIS Users (C. Brewer 2015)입니다. 2019년 그녀는 1968년 설립된 O. M. 밀러 지도학 메달을 수상한 아홉 번째 인물이 되었습니다.

```
# Panel (a)
beps |>
  ggplot(mapping = aes(x = age_group, fill = vote)) +
  geom_bar() +
  theme_minimal() +
  labs(x = "연령 그룹", y = " ", fill = " ") +
  theme(legend.position = "bottom") +
  scale_fill_brewer(palette = "Blues")
```

```
# Panel (b)
beps |>
  ggplot(mapping = aes(x = age_group, fill = vote)) +
  geom_bar() +
  theme_minimal() +
  labs(x = "나이 그룹", y = "표", fill = "투표") +
  theme(legend.position = "bottom") +
  scale_fill_brewer(palette = "Set1")

# Panel (c)
beps |>
  ggplot(mapping = aes(x = age_group, fill = vote)) +
  geom_bar() +
  theme_minimal() +
  labs(x = "나이 그룹", y = "표", fill = "투표") +
  theme(legend.position = "bottom") +
  scale_fill_viridis_d()

# Panel (d)
beps |>
  ggplot(mapping = aes(x = age_group, fill = vote)) +
  geom_bar() +
  theme_minimal() +
  labs(x = "나이 그룹", y = "표", fill = "투표") +
  theme(legend.position = "bottom") +
  scale_fill_viridis_d(option = "magma")
```

미리 만들어진 팔레트를 사용하는 것 외에도 우리만의 팔레트를 만들 수 있습니다. 그렇다고 해도 색상은 신중하게 고려해야 할 사항입니다. 색상은 전달되는 정보의 양을 늘리는 데 사용되어야 합니다 (Cleveland [1985년] 1994). 색상은 불필요하게 그래프에 추가되어서는 안 됩니다. 즉, 어떤 역할을 해야 합니다. 일반적으로 그 역할은 다른 그룹을 구별하는 것이며, 이는 색상을 서로 다르게 만드는 것을 의미합니다. 색상과 변수 사이에 어떤 관계가 있다면 색상도 적절할 수 있습니다. 예를 들어, 망고와 라즈베리의 가격 그래프를 만들 때, 색상이 각각 노란색과 빨간색이라면 독자가 정보를 해독하는 데 도움이 될 수 있습니다 (Franconeri 기타 2021, p. 121).

6.2.2 산점도

우리는 종종 두 개의 숫자 또는 연속 변수 간의 관계에 관심이 있습니다. 이를 보여주기 위해 산점도를 사용할 수 있습니다. 산점도가 항상 최선의 선택은 아닐 수 있지만, 나쁜 선택인 경우는 거의 없습니다 (Weissgerber 기타 2015). 일부는 산점도를 가장 다재다능하고 유용한 그래프 옵션으로 간주합니다 (Friendly 와/과 Wainer 2021, p. 121). 산점도를 설명하기 위해 `WDI`를 설치하고 로드한 다음 이를 사용하여 세계은행에서 일부 경제 지표를 다운로드합니다. 특히, `WDIsearch()`를 사용하여 다운로드를 용이하게 하기 위해 `WDI()`에 전달해야 하는 고유 키를 찾습니다.

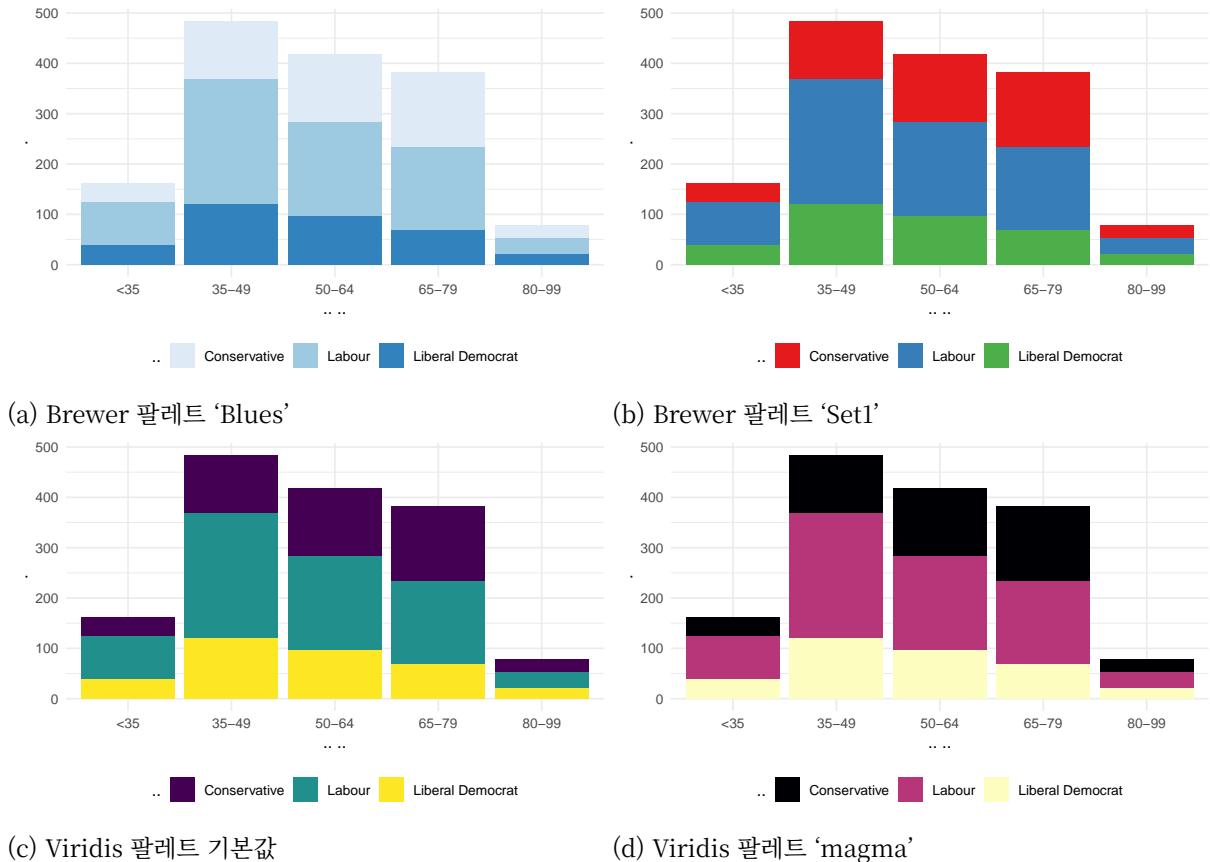


그림 6.9: 1997-2001년 영국 선거 패널 연구의 연령 그룹 및 투표 선호도 분포, 다양한 색상 설명

i 아, 우리가 그 데이터에 대한 좋은 데이터를 가지고 있다고 생각하는군요!

OECD (2014, p. 15)에 따르면 국내총생산(GDP)은 “주어진 기간 동안 주어진 국가의 모든 기업, 비영리 기관, 정부 기관 및 가구가 생산하는 모든 생산량(또는 생산)을 이중 계산 없이 단일 수치로 결합하며, 생산되는 재화 및 서비스의 유형에 관계없이 생산이 해당 국가의 경제 영토 내에서 이루어지는 경우”입니다. 현대적 개념은 20세기 경제학자 사이먼 쿠즈네츠에 의해 개발되었으며 널리 사용되고 보고됩니다. 국가의 경제 활동만큼 복잡한 것을 설명하는 데 명확하고 구체적인 단일 숫자를 갖는 것은 어느 정도 편안함을 줍니다. 이러한 요약 통계를 갖는 것은 유용하고 유익합니다. 그러나 모든 요약 통계와 마찬가지로 그 강점은 또한 약점입니다. 단일 숫자는 구성 요소에 대한 정보를 필연적으로 잃게 되며, 분해된 차이가 중요할 수 있습니다 (Moyer 와/과 Dunn 2020). 이는 장기적인 개선보다 단기적인 경제 발전을 강조합니다. 그리고 “주정치의 정량적 명확성은 불완전한 데이터에 대한 의존성과 그 결과로 발생하는 총계 및 구성 요소 모두에 적용될 수 있는 광범위한 오류 마진을 쉽게 잊게 만듭니다” (Kuznets, Epstein, 와/과 Jenks 1941, p. xxvi). 경제 성과의 요약 측정은 국가 경제의 한 측면만을 보여줍니다. 많은 강점이 있지만 GDP가 약한 점은 영역도 있습니다.

```
WDIsearch("gdp growth")
WDIsearch("inflation")
WDIsearch("population, total")
WDIsearch("Unemployment, total")
```

```
world_bank_data <-
```

```
WDI(
  indicator =
  c("FP.CPI.TOTL.ZG", "NY.GDP.MKTP.KD.ZG", "SP.POP.TOTL", "SL.UEM.TOTL.NE.ZS"),
  country = c("AU", "ET", "IN", "US")
)
```

변수 이름을 더 의미 있게 변경하고 필요한 변수만 유지할 수 있습니다.

```
world_bank_data <-
world_bank_data |>
rename(
  inflation = FP.CPI.TOTL.ZG,
  gdp_growth = NY.GDP.MKTP.KD.ZG,
  population = SP.POP.TOTL,
  unem_rate = SL.UEM.TOTL.NE.ZS
) |>
select(country, year, inflation, gdp_growth, population, unem_rate)

head(world_bank_data)
```

```
# A tibble: 6 x 6
  country     year   inflation   gdp_growth   population   unem_rate
  <chr>     <dbl>      <dbl>       <dbl>        <dbl>       <dbl>
1 Australia  1960      3.73        NA        10276477       NA
2 Australia  1961      2.29        2.48       10483000       NA
3 Australia  1962     -0.319      1.29       10742000       NA
4 Australia  1963      0.641       6.22      10950000       NA
5 Australia  1964      2.87       6.98       11167000       NA
6 Australia  1965      3.41       5.98       11388000       NA
```

시작하려면 `geom_point()`를 사용하여 국가별 GDP 성장률과 인플레이션을 보여주는 산점도를 만들 수 있습니다 (그림 ??).

```
# Panel (a)
world_bank_data |>
ggplot(mapping = aes(x = gdp_growth, y = inflation, color = country)) +
geom_point()

# Panel (b)
world_bank_data |>
ggplot(mapping = aes(x = gdp_growth, y = inflation, color = country)) +
geom_point() +
theme_minimal() +
labs(x = "GDP 성장률", y = "인플레이션", color = "나라")
```

막대 차트와 마찬가지로 테마를 변경하고 레이블을 업데이트할 수 있습니다 (그림 ??).

산점도의 경우 막대 차트에서 사용했던 “fill” 대신 “color”를 사용합니다. 점을 사용하기 때문입니다. 이는 팔레트를 변경하는 방식에도 약간 영향을 미칩니다 (그림 ??). 그렇다고 해도 `shape = 21`과 같은 특정 유형의 점에서는 `fill`과 `color` 미학을 모두 가질 수 있습니다.

```
# Panel (a)
world_bank_data |>
ggplot(aes(x = gdp_growth, y = inflation, color = country)) +
```

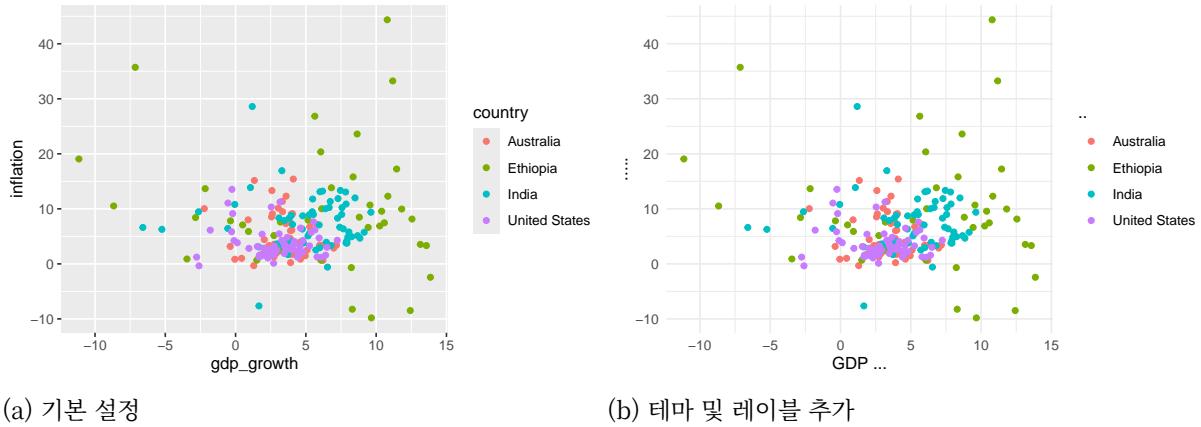


그림 6.10: 호주, 에티오피아, 인도, 미국의 인플레이션과 GDP 성장률 간의 관계

```

geom_point() +
theme_minimal() +
labs(x = "GDP 성장률", y = "인플레이션", color = "나라") +
theme(legend.position = "bottom") +
scale_color_brewer(palette = "Blues")

# Panel (b)
world_bank_data |>
  ggplot(aes(x = gdp_growth, y = inflation, color = country)) +
  geom_point() +
  theme_minimal() +
  labs(x = "GDP 성장률", y = "인플레이션", color = "나라") +
  theme(legend.position = "bottom") +
  scale_color_brewer(palette = "Set1")

# Panel (c)
world_bank_data |>
  ggplot(aes(x = gdp_growth, y = inflation, color = country)) +
  geom_point() +
  theme_minimal() +
  labs(x = "GDP 성장률", y = "인플레이션", color = "나라") +
  theme(legend.position = "bottom") +
  scale_colour_viridis_d()

# Panel (d)
world_bank_data |>
  ggplot(aes(x = gdp_growth, y = inflation, color = country)) +
  geom_point() +
  theme_minimal() +
  labs(x = "GDP 성장률", y = "인플레이션", color = "나라") +
  theme(legend.position = "bottom") +
  scale_colour_viridis_d(option = "magma")

```

산점도의 점들이 때때로 겹칩니다. 이 상황은 다양한 방법으로 해결할 수 있습니다 (그림 ??).

- 1) “alpha”를 사용하여 점에 투명도를 추가합니다 (그림 ??). “alpha” 값은 완전히 투명한 0에서 완전히 불투명한 1까지 다양할 수 있습니다.
- 2) `geom_jitter()`을 사용하여 약간의 노이즈를 추가하여 점을 약간 이동시킵니다 (그림 ??). 기본적

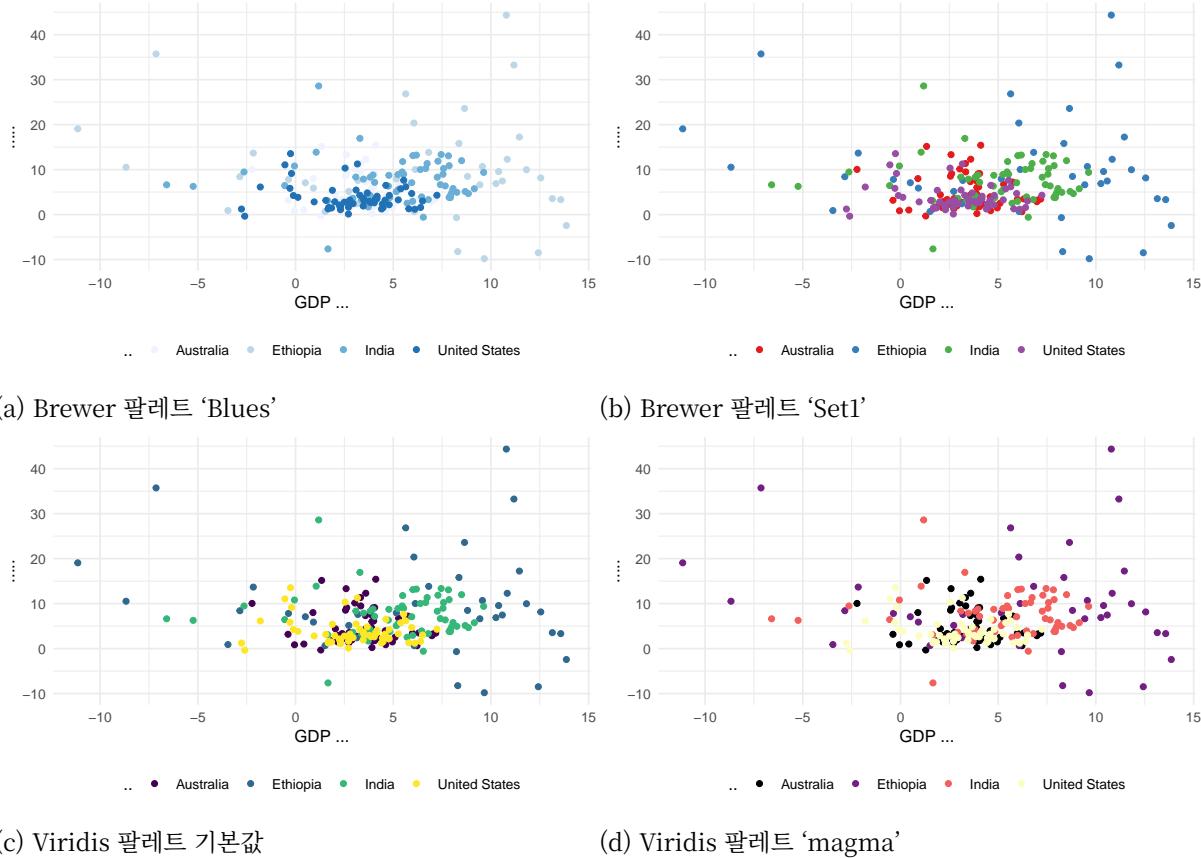


그림 6.11: 호주, 에티오피아, 인도, 미국의 인플레이션과 GDP 성장을 간의 관계

으로 이동은 양방향으로 균일하지만, “width” 또는 “height”를 사용하여 이동 방향을 지정할 수 있습니다. 이 두 옵션 간의 결정은 정확성이 얼마나 중요한지, 그리고 점의 수에 따라 달라집니다. 점의 상대적 밀도를 강조하고 개별 점의 정확한 값을 반드시 강조하지 않을 때 `geom_jitter()`를 사용하는 것이 종종 유용합니다. `geom_jitter()`를 사용할 때는 `?@sec-fire-hose`에서 소개된 재현성을 위해 시드를 설정하는 것이 좋습니다.

```
set.seed(853)

# Panel (a)
world_bank_data |>
  ggplot(aes(x = gdp_growth, y = inflation, color = country)) +
  geom_point(alpha = 0.5) +
  theme_minimal() +
  labs(x = "GDP 성장", y = "인플레이션", color = "색상")

# Panel (b)
world_bank_data |>
  ggplot(aes(x = gdp_growth, y = inflation, color = country)) +
  geom_jitter(width = 1, height = 1) +
  theme_minimal() +
  labs(x = "GDP 성장", y = "인플레이션", color = "색상")
```

우리는 종종 산점도를 사용하여 두 연속 변수 간의 관계를 설명합니다. `geom_smooth()`를 사용하여 “요약” 선을 추가하는 것이 유용할 수 있습니다 (그림 ??). “method”를 사용하여 관계를 지정하고, “color”로 색상을

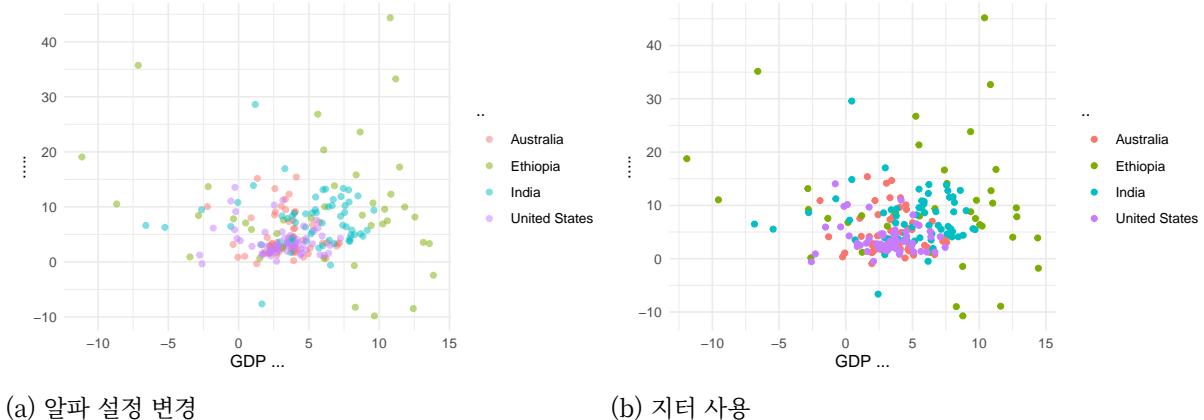


그림 6.12: 호주, 에티오피아, 인도, 미국의 인플레이션과 GDP 성장률 간의 관계

변경하고, “se”로 표준 오차를 추가하거나 제거할 수 있습니다. 일반적으로 사용되는 “method”는 `lm`이며, 이는 `lm()` 함수를 사용하는 것과 유사하게 단순 선형 회귀선을 계산하고 플로팅합니다. `geom_smooth()`를 사용하면 그래프에 레이어가 추가되므로 `ggplot()`에서 미학을 상속합니다. 예를 들어, 그림 ?? 및 그림 ??에서 각 국가에 대해 하나의 선이 있는 이유입니다. 특정 색상을 지정하여 이를 덮어쓸 수 있습니다 (그림 ??). 스플라인과 같은 다른 유형의 적합선이 선호되는 상황도 있습니다.

```
# Panel (a)
world_bank_data |>
  ggplot(aes(x = gdp_growth, y = inflation, color = country)) +
  geom_jitter() +
  geom_smooth() +
  theme_minimal() +
  labs(x = "GDP 성장률", y = "인플레이션", color = "나라")

# Panel (b)
world_bank_data |>
  ggplot(aes(x = gdp_growth, y = inflation, color = country)) +
  geom_jitter() +
  geom_smooth(method = lm, se = FALSE) +
  theme_minimal() +
  labs(x = "GDP 성장률", y = "인플레이션", color = "나라")

# Panel (c)
world_bank_data |>
  ggplot(aes(x = gdp_growth, y = inflation, color = country)) +
  geom_jitter() +
  geom_smooth(method = lm, color = "black", se = FALSE) +
  theme_minimal() +
  labs(x = "GDP 성장률", y = "인플레이션", color = "나라")
```

6.2.3 선 플롯

경제 시계열과 같이 함께 연결되어야 하는 변수가 있을 때 선 플롯을 사용할 수 있습니다. 세계은행 데이터 세트를 계속 사용하고 `geom_line()`을 사용하여 미국의 GDP 성장에 초점을 맞춥니다 (그림 ??). 데이터 출처는 `labs()` 내의 “caption”을 사용하여 그래프에 추가할 수 있습니다.

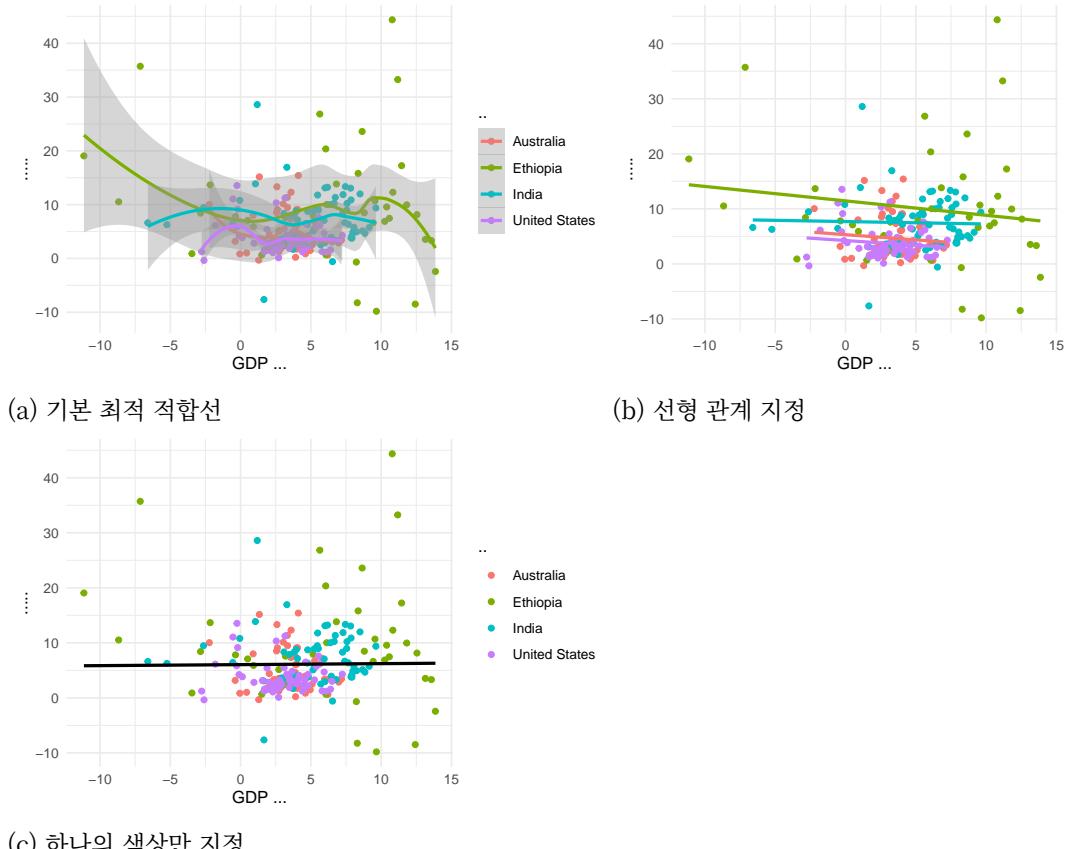


그림 6.13: 호주, 에티오피아, 인도, 미국의 인플레이션과 GDP 성장을 간의 관계

```
# Panel (a)
world_bank_data |>
  filter(country == "United States") |>
  ggplot(mapping = aes(x = year, y = gdp_growth)) +
  geom_line() +
  theme_minimal() +
  labs(x = "연도", y = "GDP 성장", caption = "호주 등: 1960-2010.")

# Panel (b)
world_bank_data |>
  filter(country == "United States") |>
  ggplot(mapping = aes(x = year, y = gdp_growth)) +
  geom_step() +
  theme_minimal() +
  labs(x = "연도", y = "GDP 성장", caption = "호주 등: 1960-2010.")
```

geom_line()의 약간 변형된 geom_step()를 사용하여 연도별 변화에 집중할 수 있습니다 (그림 ??).

필립스 곡선은 시간 경과에 따른 실업률과 인플레이션 간의 관계를 나타내는 플롯의 이름입니다. 때때로 데이터에서 역관계가 발견되기도 합니다. 예를 들어, 1861년에서 1957년 사이의 영국에서 그러했습니다 (Phillips 1958). 우리는 데이터에서 이 관계를 조사하는 다양한 방법을 가지고 있습니다.

- 1) 그라프에 두 번째 선을 추가합니다. 예를 들어, 인플레이션을 추가할 수 있습니다 (그림 ??). 이

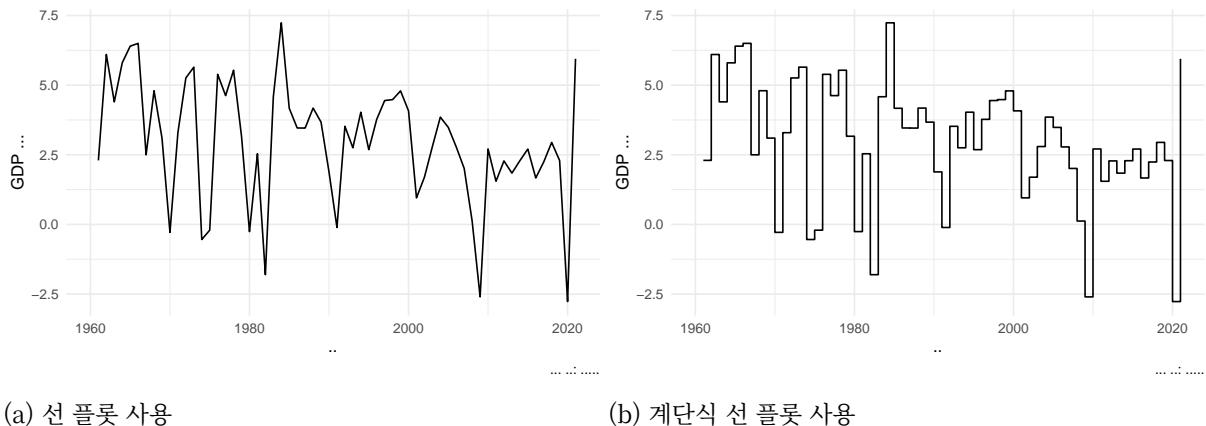


그림 6.14: 미국 GDP 성장률 (1961-2020)

를 위해서는 데이터가 정돈된 형식인지 확인하기 위해 “R 필수 사항” 온라인 부록³에서 논의된 `pivot_longer()`를 사용해야 합니다.

- 2) `geom_path()`를 사용하여 데이터 세트에 나타나는 순서대로 값을 연결합니다. `?@fig-notphillips-2`에서는 1960년에서 2020년 사이의 미국에 대한 필립스 곡선을 보여줍니다. `?@fig-notphillips-2`는 실업률과 인플레이션 사이에 명확한 관계를 보여주지 않는 것 같습니다.

```
world_bank_data |>
  filter(country == "United States") |>
  select(-population, -gdp_growth) |>
  pivot_longer(
    cols = c("inflation", "unem_rate"),
    names_to = "series",
    values_to = "value"
  ) |>
  ggplot(mapping = aes(x = year, y = value, color = series)) +
  geom_line() +
  theme_minimal() +
  labs(
    x = "연도", y = "값", color = "시리즈",
    caption = "미국 경제: 1960-2020."
  ) +
  scale_color_brewer(palette = "Set1", labels = c("인플레이션", "실업률")) +
  theme(legend.position = "bottom")

world_bank_data |>
  filter(country == "United States") |>
  ggplot(mapping = aes(x = unem_rate, y = inflation)) +
  geom_path() +
  theme_minimal() +
  labs(
    x = "실업률", y = "인플레이션",
    caption = "미국 경제: 1960-2020."
  )
```

³https://tellingstorieswithdata.com/20-r_essentials.html

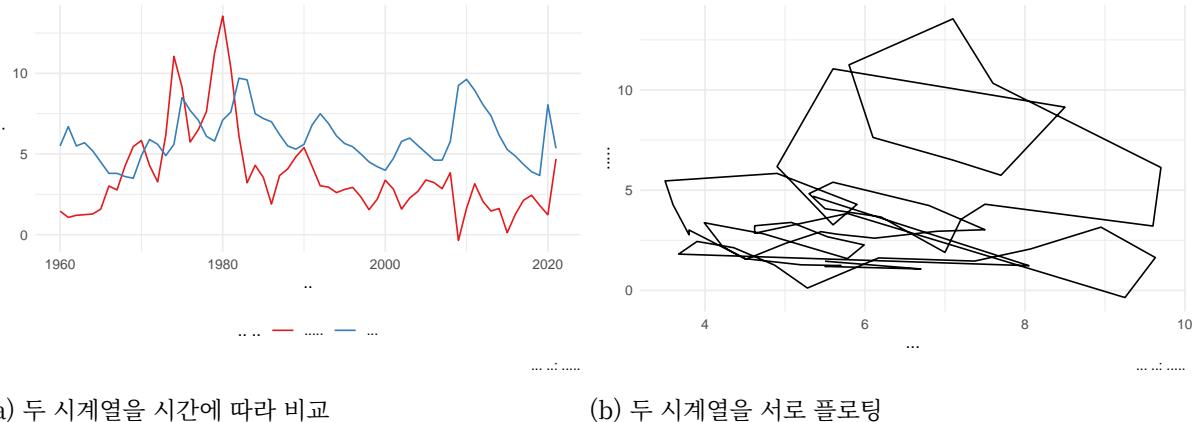


그림 6.15: 미국의 실업률과 인플레이션 (1960-2020)

6.2.4 히스토그램

히스토그램은 연속 변수의 분포 모양을 보여주는 데 유용합니다. 데이터 값의 전체 범위는 “빈”이라고 불리는 간격으로 나뉘며, 히스토그램은 각 빈에 몇 개의 관측치가 속하는지 계산합니다. ?@fig-histogramone에서는 에티오피아의 GDP 분포를 살펴봅니다.

```
world_bank_data |>
  filter(country == "Ethiopia") |>
  ggplot(aes(x = gdp_growth)) +
  geom_histogram() +
  theme_minimal() +
  labs(
    x = "GDP 増加",
    y = "頻度 頻度",
    caption = "エチオピア: GDP."
  )
```

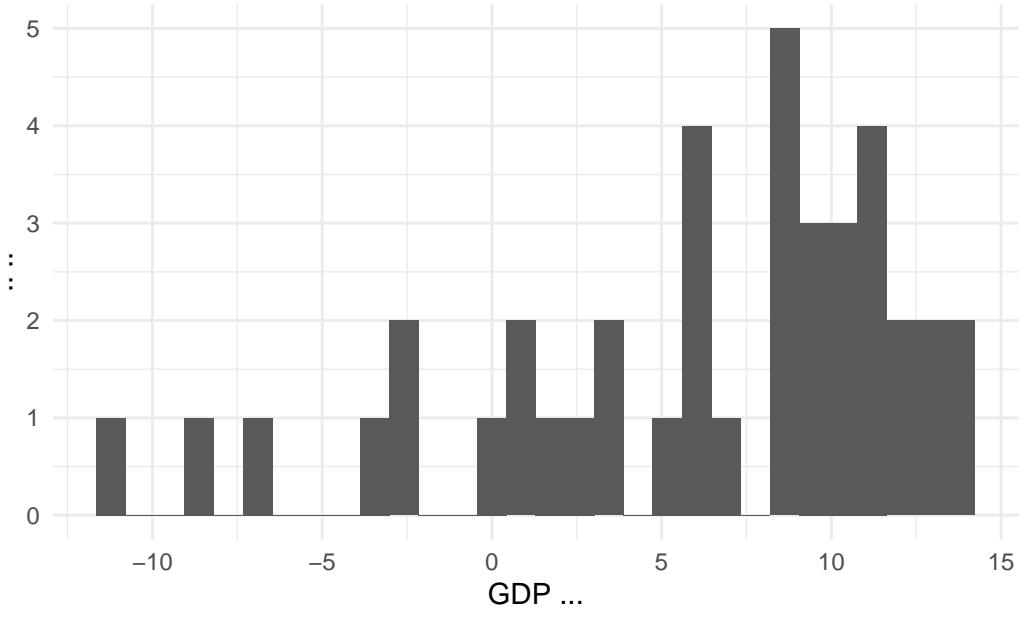


그림 6.16: 에티오피아의 GDP 성장률 분포 (1960-2020)

히스토그램의 모양을 결정하는 핵심 구성 요소는 빈의 수입니다. 이는 두 가지 방법으로 지정할 수 있습니다 (그림 ??).

- 1) 포함할 “빈”的 수를 지정하거나;
- 2) “빈 너비”를 지정합니다.

```
# Panel (a)
world_bank_data |>
  filter(country == "Ethiopia") |>
  ggplot(aes(x = gdp_growth)) +
  geom_histogram(bins = 5) +
  theme_minimal() +
  labs(
    x = "GDP 成長",
    y = "頻度 頻率"
  )

# Panel (b)
world_bank_data |>
  filter(country == "Ethiopia") |>
  ggplot(aes(x = gdp_growth)) +
  geom_histogram(bins = 20) +
  theme_minimal() +
  labs(
    x = "GDP 成長",
    y = "頻度 頻率"
  )

# Panel (c)
world_bank_data |>
```

```

filter(country == "Ethiopia") |>
ggplot(aes(x = gdp_growth)) +
geom_histogram(binwidth = 2) +
theme_minimal() +
labs(
  x = "GDP 성장률",
  y = "빈도"
)

# Panel (d)
world_bank_data |>
filter(country == "Ethiopia") |>
ggplot(aes(x = gdp_growth)) +
geom_histogram(binwidth = 5) +
theme_minimal() +
labs(
  x = "GDP 성장률",
  y = "빈도"
)

```

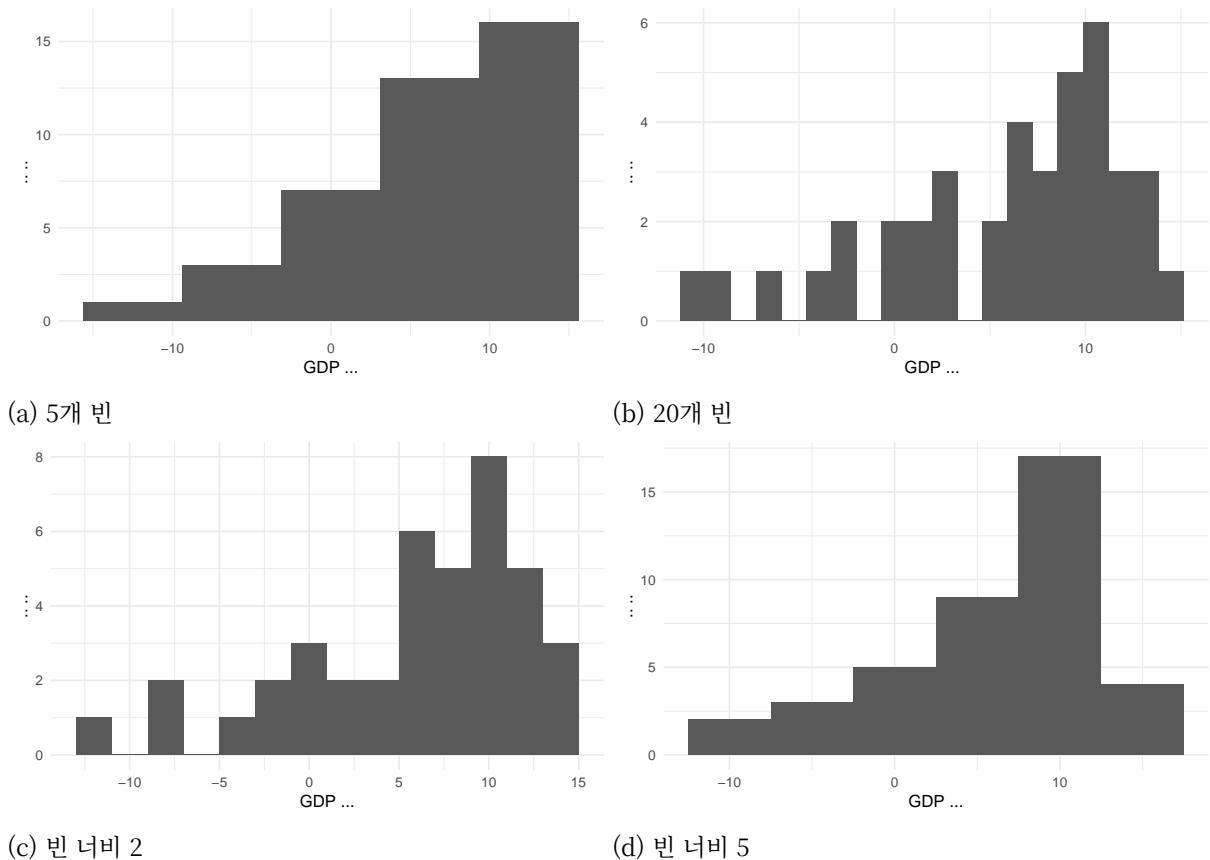


그림 6.17: 에티오피아의 GDP 성장을 분포 (1960-2020)

히스토그램은 데이터를 국소적으로 평균화하는 것으로 생각할 수 있으며, 빈의 수는 이러한 현상이 얼마나 발생하는지에 영향을 미칩니다. 빈이 두 개뿐인 경우 상당한 평활화가 발생하지만, 정확성을 많이 잃게 됩니다. 빈이 너무 적으면 편향이 더 커지고, 빈이 너무 많으면 분산이 더 커집니다 (Wasserman 2005, p. 303). 빈의 수 또는 너비에 대한 우리의 결정은 편향과 분산의 균형을 맞추는 것과 관련이 있습니다. 이

는 주제와 목표를 포함한 다양한 고려 사항에 따라 달라집니다 (Cleveland [1985년] 1994, p. 135). 이것 이 (Denby2009가?) 히스토그램을 탐색 도구로서 특히 가치 있다고 간주하는 이유 중 하나입니다.

마지막으로, “fill”을 사용하여 다른 유형의 관측치를 구별할 수 있지만, 상당히 지저분해질 수 있습니다. 일반적으로 다음이 더 좋습니다.

1. `geom_freqpoly()`로 분포의 윤곽을 그립니다 (그림 ??).
2. `geom_dotplot()`로 점 스택을 만듭니다 (그림 ??).
3. 투명도를 추가합니다. 특히 차이가 더 뚜렷한 경우 (그림 ??).

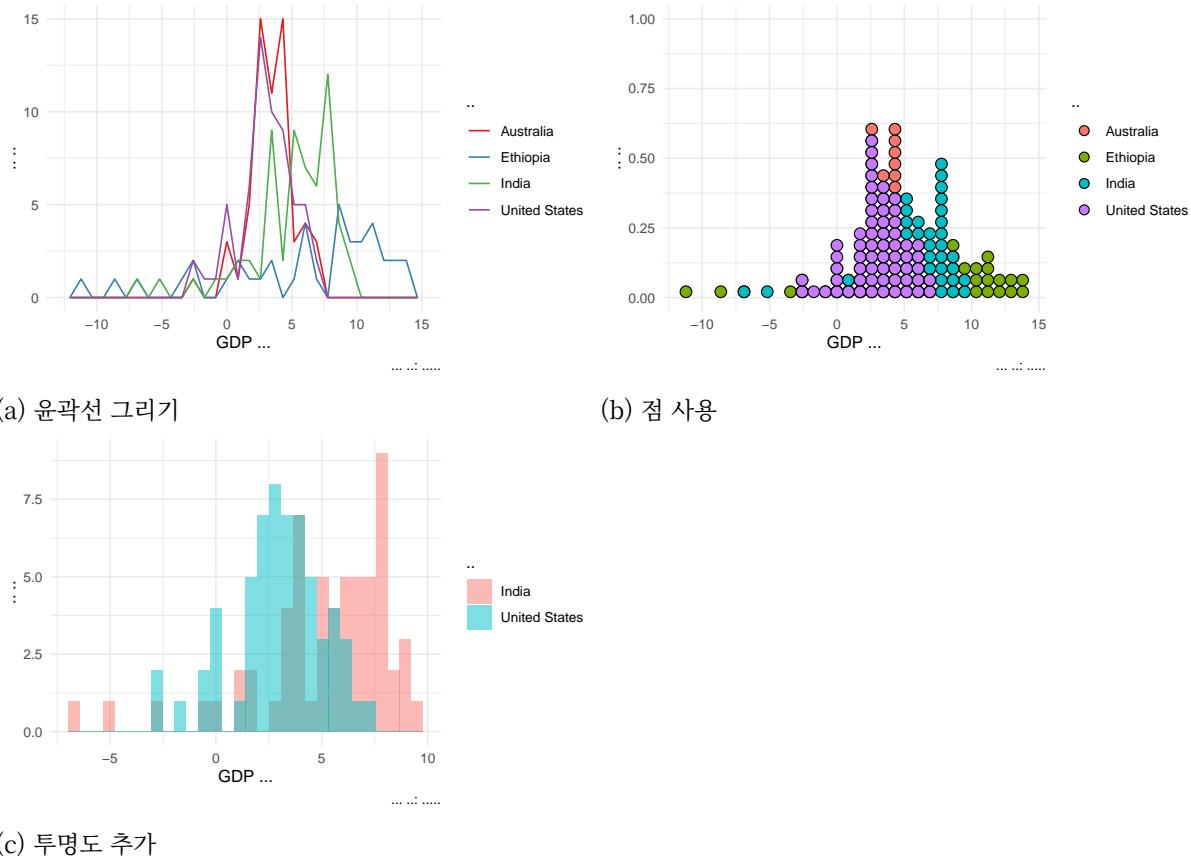
```
# Panel (a)
world_bank_data |>
  ggplot(aes(x = gdp_growth, color = country)) +
  geom_freqpoly() +
  theme_minimal() +
  labs(
    x = "GDP ䷂",
    y = "䷂ ䷂",
    color = "䷂",
    caption = "䷂ ䷂: ䷂")
) +
  scale_color_brewer(palette = "Set1")

# Panel (b)
world_bank_data |>
  ggplot(aes(x = gdp_growth, group = country, fill = country)) +
  geom_dotplot(method = "histodot") +
  theme_minimal() +
  labs(
    x = "GDP ䷂",
    y = "䷂ ䷂",
    fill = "䷂",
    caption = "䷂ ䷂: ䷂")
) +
  scale_color_brewer(palette = "Set1")

# Panel (c)
world_bank_data |>
  filter(country %in% c("India", "United States")) |>
  ggplot(mapping = aes(x = gdp_growth, fill = country)) +
  geom_histogram(alpha = 0.5, position = "identity") +
  theme_minimal() +
  labs(
    x = "GDP ䷂",
    y = "䷂ ䷂",
    fill = "䷂",
    caption = "䷂ ䷂: ䷂")
) +
  scale_color_brewer(palette = "Set1")
```

히스토그램의 흥미로운 대안은 경험적 누적 분포 함수(ECDF)입니다. 이와 히스토그램 사이의 선택은 청중에 따라 달라지는 경향이 있습니다. 덜 정교한 청중에게는 적절하지 않을 수 있지만, 청중이 정량적으로 편안하다면 히스토그램보다 평활화가 덜하기 때문에 훌륭한 선택이 될 수 있습니다. `stat_ecdf()`를 사용하여 ECDF를 만들 수 있습니다. 예를 들어, `?@fig-ecdfismyfavo` `hidonthavefavs`는 `?@fig-histogramone`과 동일한 ECDF를 보여줍니다.

```
world_bank_data |>
  ggplot(mapping = aes(x = gdp_growth, color = country)) +
```



(c) 투명도 추가

그림 6.18: 다양한 국가의 GDP 성장을 분포 (1960-2020)

```
stat_ecdf(geom = "point") +
theme_minimal() +
labs(
  x = "GDP 성장", y = "분포", color = "국가",
  caption = "국가별 GDP 성장 분포")
) +
theme(legend.position = "bottom")
```

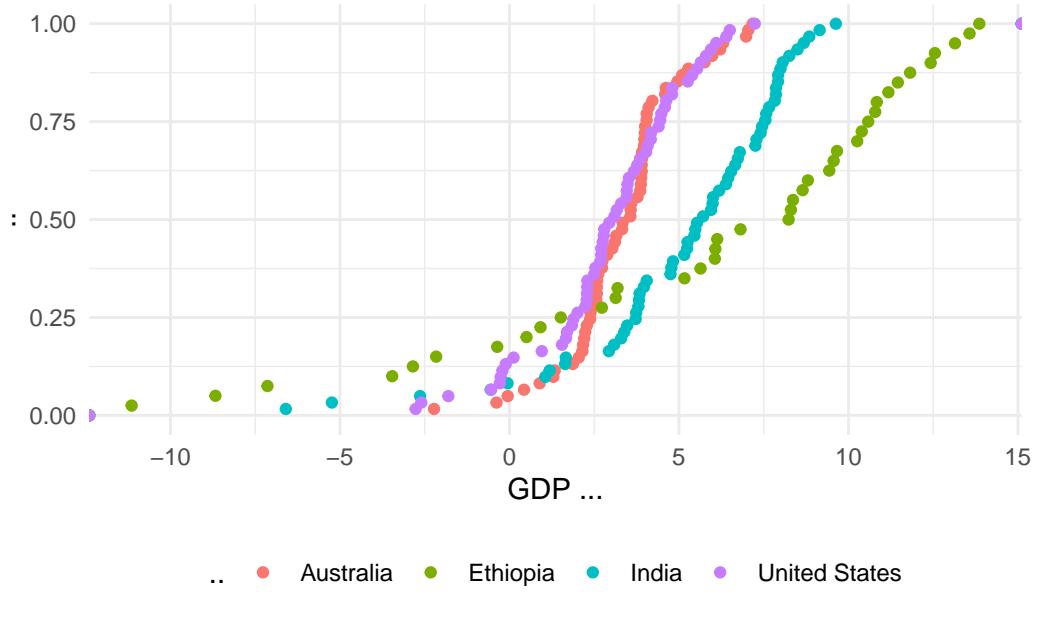


그림 6.19: 네 개 국가의 GDP 성장률 분포 (1960-2020)

6.2.5 상자 그림

상자 그림은 일반적으로 다섯 가지 측면을 보여줍니다. 1) 중앙값, 2) 25번째 백분위수, 3) 75번째 백분위수. 네 번째와 다섯 번째 요소는 세부 사항에 따라 다릅니다. 한 가지 옵션은 최소값과 최대값입니다. 다른 옵션은 75번째 백분위수와 25번째 백분위수 사이의 차이인 사분위수 범위(IQR)를 결정하는 것입니다. 네 번째와 다섯 번째 요소는 25번째 및 75번째 백분위수에서 $1.5 \times \text{IQR}$ 이내의 극단적인 관측치입니다. 후자의 접근 방식은 ggplot2의 geom_boxplot에서 기본적으로 사용됩니다. Spear (1952, p. 166)는 범위와 중앙값 및 범위와 같은 다양한 요약 통계에 초점을 맞춘 차트 개념을 도입했으며, Tukey (1977)는 어떤 요약 통계에 초점을 맞추고 이를 대중화했습니다 (Wickham 와/과 Stryjewski 2011).

그래프를 사용하는 한 가지 이유는 데이터가 얼마나 복잡한지 이해하고 받아들이는 데 도움이 되기 때문입니다. 데이터를 숨기거나 평활화하려고 노력하는 대신 말입니다 (Armstrong 2022). 상자 그림의 적절한 사용 사례는 (Bethlehem 2022에서 와?) 같이 많은 변수의 요약 통계를 한 번에 비교하는 것입니다. 그러나 상자 그림만으로는 데이터 분포를 보여주기보다는 숨기기 때문에 최선의 선택인 경우는 거의 없습니다. 동일한 상자 그림이 매우 다른 분포에 적용될 수 있습니다. 이를 이해하기 위해 두 가지 유형의 베타 분포에서 시뮬레이션된 데이터를 고려해 보십시오. 첫 번째는 두 개의 베타 분포에서 추출한 것입니다. 하나는 오른쪽으로 치우쳐 있고 다른 하나는 왼쪽으로 치우쳐 있습니다. 두 번째는 왜곡이 없는 베타 분포에서 추출한 것입니다. Beta(1, 1)은 Uniform(0, 1)과 동일합니다.

```
set.seed(853)

number_of_draws <- 10000

both_left_and_right_skew <-
  c(
    rbeta(number_of_draws / 2, 5, 2),
    rbeta(number_of_draws / 2, 2, 5)
  )

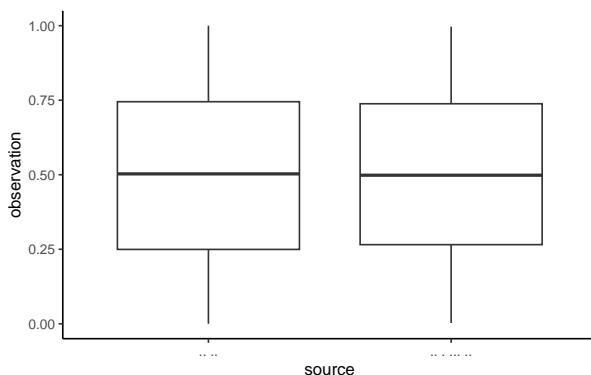
no_skew <-
  rbeta(number_of_draws, 1, 1)
```

```
beta_distributions <-
  tibble(
    observation = c(both_left_and_right_skew, no_skew),
    source = c(
      rep("한국 및 미국", number_of_draws),
      rep("한국", number_of_draws)
    )
  )
```

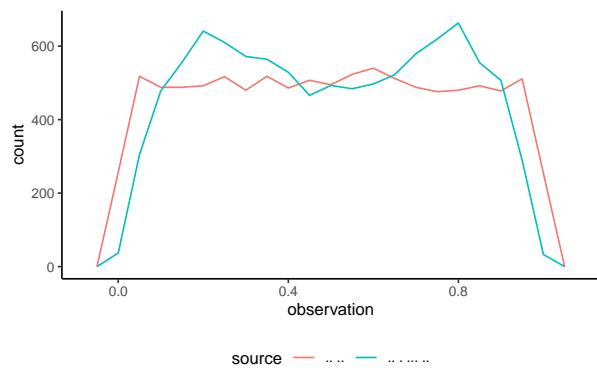
먼저 두 시리즈의 상자 그림을 비교할 수 있습니다 (그림 ??). 그러나 실제 데이터를 플로팅하면 얼마나 다른지 알 수 있습니다 (그림 ??).

```
beta_distributions |>
  ggplot(aes(x = source, y = observation)) +
  geom_boxplot() +
  theme_classic()

beta_distributions |>
  ggplot(aes(x = observation, color = source)) +
  geom_freqpoly(binwidth = 0.05) +
  theme_classic() +
  theme(legend.position = "bottom")
```



(a) 상자 그림으로 설명



(b) 실제 데이터

그림 6.20: 다른 매개변수를 가진 베타 분포에서 추출한 데이터

상자 그림을 사용해야 한다면, 한 가지 방법은 상자 그림 위에 실제 데이터를 레이어로 포함하는 것입니다. 예를 들어, 그림 ?? 에서는 네 개 국가의 인플레이션 분포를 보여줍니다. 이것이 잘 작동하는 이유는 실제 관측치와 요약 통계를 모두 보여주기 때문입니다.

```
world_bank_data |>
  ggplot(mapping = aes(x = country, y = inflation)) +
  geom_boxplot() +
  geom_jitter(alpha = 0.3, width = 0.15, height = 0) +
  theme_minimal() +
  labs(
    x = "나라",
    y = "인플레이션",
    caption = "인플레이션: 나라별."
  )
```

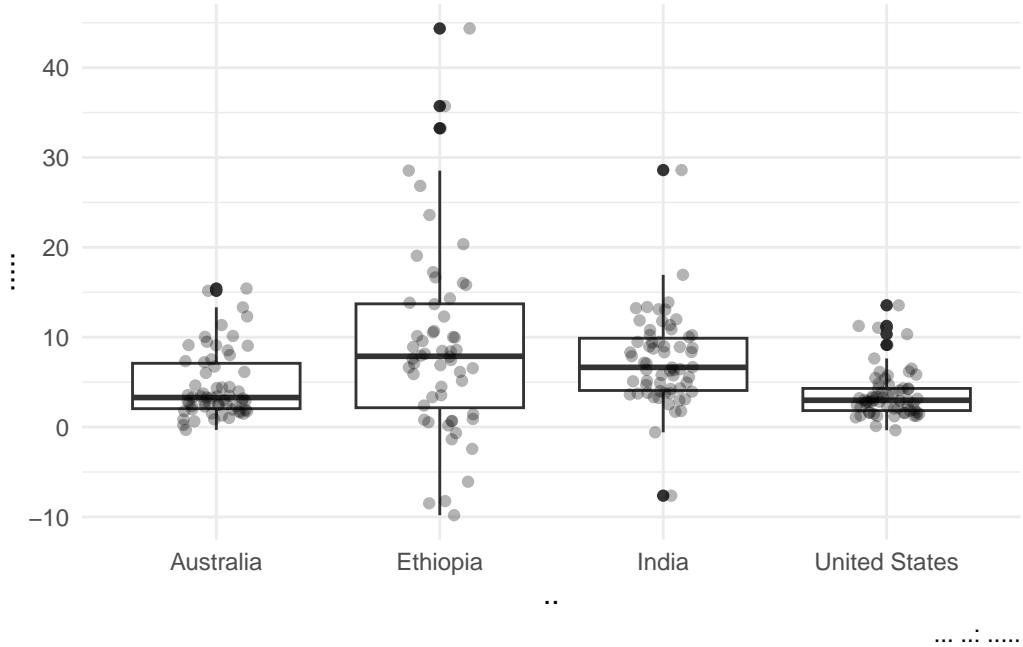


그림 6.21: 네 개 국가의 인플레이션 데이터 분포 (1960-2020)

6.2.6 대화형 그래프

shiny (Chang 기타 2021)는 R을 사용하여 대화형 웹 애플리케이션을 만드는 방법입니다. 재미있지만 약간 까다로울 수 있습니다. 여기서는 shiny를 활용하는 한 가지 방법을 단계별로 설명합니다. 즉, 그래프에 빠르게 상호 작용 기능을 추가하는 것입니다. 이는 작은 일처럼 들리지만, 왜 그렇게 강력한지에 대한 훌륭한 예시는 (theeconomistforecasts에서?) 제공합니다. 그들은 2022년 프랑스 대통령 선거 예측이 시간이 지남에 따라 어떻게 변했는지 보여줍니다.

babynames (Wickham 2021a)의 “babynames” 데이터 세트를 기반으로 대화형 그래프를 만들 것입니다. 먼저 정적 버전을 만들 것입니다 (그림 ??).

```
top_five_names_by_year <-
  babynames |>
  arrange(desc(n)) |>
  slice_head(n = 5, by = c(year, sex))

top_five_names_by_year |>
  ggplot(aes(x = n, fill = sex)) +
  geom_histogram(position = "dodge") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1") +
  labs(
    x = "인기 이름 수",
    y = "연도",
    fill = "성별"
  )
```

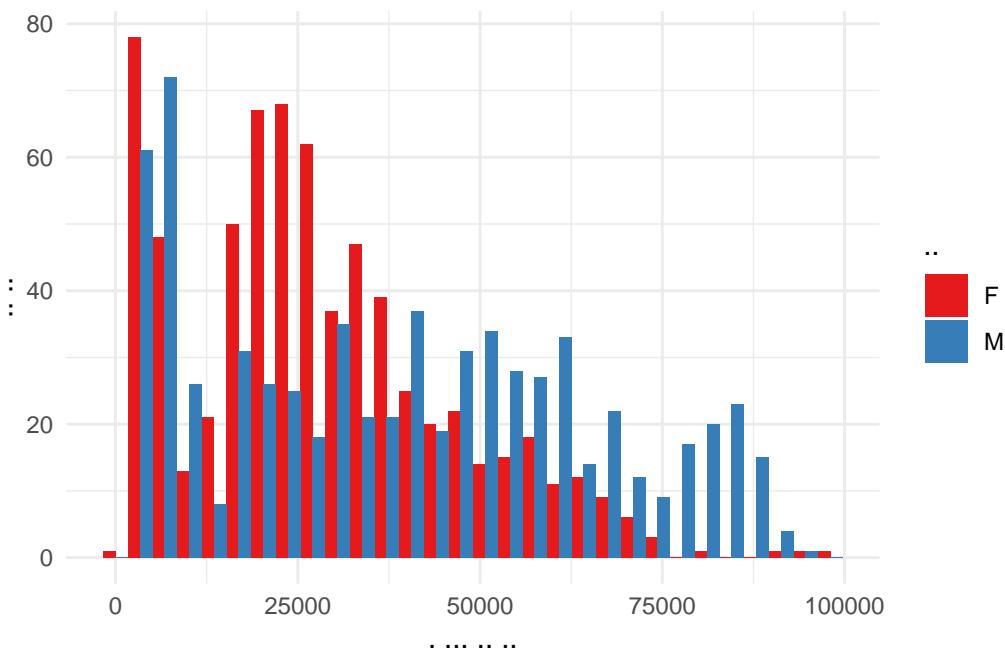


그림 6.22: 인기 있는 아기 이름

우리가 관심 있을 수 있는 한 가지는 “bins” 매개변수의 효과가 우리가 보는 것에 어떻게 영향을 미치는지입니다. 우리는 상호 작용을 사용하여 다른 값을 탐색하고 싶을 수 있습니다.

시작하려면 새 shiny 앱을 만드십시오 (“파일” -> “새 파일” -> “Shiny 웹 앱”). “not_my_first_shiny”와 같은 이름을 지정하고 다른 모든 옵션은 기본값으로 두십시오. 새 파일 “app.R”이 열리고 “앱 실행”을 클릭하여 어떻게 보이는지 확인합니다.

이제 해당 파일 “app.R”的 내용을 아래 내용으로 바꾼 다음 다시 “앱 실행”을 클릭하십시오.

```
library(shiny)

# Define UI for application that draws a histogram
ui <- fluidPage(
  # Application title
  titlePanel("인기 있는 아기 이름"),

  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
      sliderInput(
        inputId = "number_of_bins",
        label = "Bins:",
        min = 1,
        max = 50,
        value = 30
      )
    ),
    # Show a plot of the generated distribution
    mainPanel(plotOutput("distPlot"))
  )
)
```

```
# Define server logic required to draw a histogram
server <- function(input, output) {
  output$distPlot <- renderPlot({
    # Draw the histogram with the specified number of bins
    top_five_names_by_year |>
      ggplot(aes(x = n, fill = sex)) +
      geom_histogram(position = "dodge", bins = input$number_of_bins) +
      theme_minimal() +
      scale_fill_brewer(palette = "Set1") +
      labs(
        x = "婴儿的名字",
        y = "出现次数",
        fill = "性别"
      )
  })
}

# Run the application
shinyApp(ui = ui, server = server)
```

빈의 수를 변경할 수 있는 대화형 그래프를 만들었습니다. ?@fig-shinyone과 같아야 합니다.

Count of names for five most popular names each year.

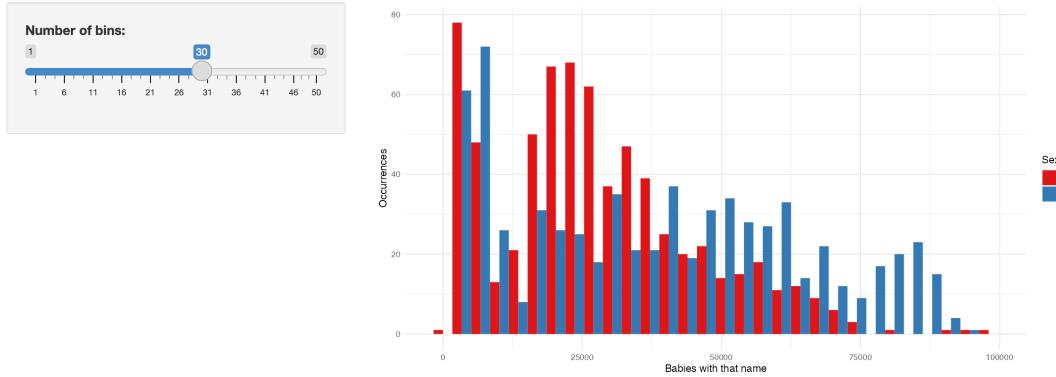


그림 6.23: 사용자가 빈의 수를 제어하는 Shiny 앱 예시

6.3 표

표는 설득력 있는 이야기를 전달하는 데 중요한 부분입니다. 표는 그래프보다 적은 정보를 전달할 수 있지만, 높은 충실도로 전달합니다. 특히 몇 가지 특정 값을 강조하는 데 유용합니다 (Andersen 와/과 Armstrong 2021). 이 책에서는 주로 세 가지 방식으로 표를 사용합니다.

1. 데이터 세트의 일부를 보여주기 위해.
2. 요약 통계를 전달하기 위해.
3. 회귀 결과를 표시하기 위해.