

오딧세이 바이블 The Book of OHDSI

한국 오딧세이

2019-12-02

Contents

서문	xi
이 책의 목표	xi
이 책의 구성	xi
기여자	xii
소프트웨어 버전	xiii
The Book of OHDSI가 쓰여진 과정	xiii
이 책이 번역된 과정	xiv
한국판 번역에 부쳐	xv

I OHDSI 커뮤니티 1

1 OHDSI 커뮤니티	3
1.1 데이터에서 근거로의 여정	3
1.2 OMOP (Observational Medical Outcomes Partnership)	5
1.3 개방형 과학 공동체로서의 OHDSI	6
1.4 OHDSI의 역사	8
1.5 OHDSI와의 협업	10
1.6 한국 OHDSI의 역사	10
1.7 요약	12

2 OHDSI 시작하기 13

2.1 여정에 동참하기	13
2.2 적합한 위치	20
2.3 요약	22

3 오픈 사이언스	23
3.1 오픈 사이언스	23
3.2 실천의 오픈 사이언스: 스터디톤	25
3.3 공개 표준 Open Standards	25
3.4 오픈 소스 Open Source	26
3.5 공개 데이터 Open Data	26
3.6 열린 담론 Open Disclosure	27
3.7 OHDSI와 FAIR의 가이드 원칙	27
II 공통 데이터 모델	31
4 공통 데이터 모델	33
4.1 설계 원리	34
4.2 데이터 모델 규칙	36
4.3 표준화된 CDM 테이블	42
4.4 부가 정보	56
4.5 요약	56
4.6 예제	56
5 OMOP 표준 용어	59
5.1 왜 용어 Vocabulary인가, 그리고 왜 표준화인가?	59
5.2 개념 Concept	62
5.3 관계 Relationships	71
5.4 계층 Hierarchy	74
5.5 내부 참조 테이블	76
5.6 특수 상황	76
5.7 요약	78
5.8 예제	79
6 추출 변환 적재 Extract Transform Load	81
6.1 서론	81
6.2 1단계: ETL 설계	82
6.3 2단계: 용어 매핑 생성	92
6.4 3단계: ETL 수행	99
6.5 4단계: 질 관리	100

6.6 ETL 협약 convention과 테마스 THEMIS	101
6.7 CDM과 ETL의 유지	102
6.8 ETL에 대한 마지막 생각	103
6.9 요약	103
6.10 예제	104
III 데이터 분석	107
7 데이터 분석 이용 사례	109
7.1 임상적 특성 분석	109
7.2 인구 수준 추정	111
7.3 환자 수준 예측	112
7.4 고혈압 이용 사례 예	113
7.5 관찰 연구의 한계	114
7.6 요약	115
7.7 예제	115
8 OHDSI 분석 툴	117
8.1 분석 구현	117
8.2 분석 전략	118
8.3 ATLAS	120
8.4 Methods Library	122
8.5 설치 전략	130
8.6 요약	131
9 SQL과 R	133
9.1 SqlRender	134
9.2 DatabaseConnector	143
9.3 CDM 쿼리 실행	147
9.4 쿼리 작성 시 OMOP 용어 사용하기	151
9.5 QueryLibrary	152
9.6 간단한 연구 구성하기	154
9.7 SQL과 R을 사용하여 연구 구현	154
9.8 요약	161
9.9 예제	162

10 코호트 만들기	163
10.1 코호트란 무엇인가?	164
10.2 규칙 기반 코호트 정의	165
10.3 개념 모음	167
10.4 확률적 코호트 정의	168
10.5 코호트 정의 유효성	168
10.6 고혈압 환자 코호트 작성하기	169
10.7 ATLAS를 이용해 코호트 작성하기	169
10.8 SQL을 사용하여 코호트 구현하기	181
10.9 요약	190
10.10 예제	190
11 임상적 특성 분석	193
11.1 데이터베이스 수준의 특성 분석 Database Level Characterization	194
11.2 코호트 특성 분석 Cohort Characterization	194
11.3 치료 경로 Treatment Pathways	195
11.4 발생 Incidence	196
11.5 고혈압 환자의 임상적 특성 분석	197
11.6 ATLAS를 활용한 데이터베이스의 특성 분석	197
11.7 ATLAS를 이용한 코호트 특성 분석	198
11.8 R을 이용한 코호트 특성 분석	207
11.9 ATLAS에서 코호트 경로 Cohort Pathways in ATLAS	212
11.10 ATLAS를 이용한 발생률 분석	217
11.11 요약	221
11.12 예제	222
12 연구 수준 추정	223
12.1 코호트 방법론 설계	224
12.2 자가 통제 코호트 연구 설계 Self-Controlled Cohort Design	228
12.3 환자-대조군 연구 설계 Case-Control Design	229
12.4 환자-교차 연구 설계 Case-Crossover Design	230
12.5 자기 대조 환자군 연구 설계 Self-Controlled Case Series Design	231
12.6 고혈압 연구 설계하기	232
12.7 ATLAS를 사용한 연구 구현하기	234
12.8 R을 사용한 연구 구현하기	248

12.9 연구 결과물	258
12.10 요약	264
12.11 예제	264
13 환자 수준 예측	267
13.1 예측 문제	268
13.2 데이터 추출	270
13.3 모델 적합	272
13.4 예측 모델 평가	278
13.5 환자-수준 예측 연구 설계	282
13.6 ATLAS에서의 연구 구현하기	286
13.7 R에서의 연구 실행	299
13.8 결과 보급	306
13.9 추가적 환자-수준 예측 변수	314
13.10 요약	317
13.11 예제	317
IV 근거의 질	319
14 근거의 질	321
14.1 신뢰성 있는 근거의 속성	321
14.2 근거의 질에 대한 이해	323
14.3 근거 품질의 전달	324
14.4 요약	324
15 데이터의 질	327
15.1 데이터 품질 문제에 대한 원인	328
15.2 보편적인 데이터 품질	329
15.3 연구 별 검사	334
15.4 ACHILLES 실습	336
15.5 Data Quality Dashboard 실습 (Data Quality Dashboard in Practice)	339
15.6 연구별 검사 실습	340
15.7 요약	343
15.8 예제	343
16 임상적 타당성	345

16.1 보건의료 데이터의 특성	345
16.2 코호트 유효성 검사	346
16.3 원천 기록 검증	349
16.4 PheEvaluator	352
16.5 근거의 일반화	363
16.6 요약	363
17 소프트웨어의 타당성	365
17.1 분석 코드의 타당성	366
17.2 연구 방법론 라이브러리 소프트웨어의 개발 과정	368
17.3 연구방법론 라이브러리 기능 검사	370
17.4 요약	371
18 방법론적 타당성	373
18.1 설계별 진단	374
18.2 모든 추정을 위한 진단법	374
18.3 실무에서의 연구 방법론 검증	382
18.4 OHDSI 방법론 벤치마크	390
18.5 요약	391
V OHDSI 연구	395
19 연구단계	397
19.1 일반 모범 사례 지침	398
19.2 세부 연구 단계	401
19.3 요약	407
20 OHDSI 네트워크 리서치	409
20.1 연구 네트워크로서의 OHDSI	409
20.2 OHDSI 네트워크 연구	410
20.3 OHDSI 네트워크 연구 수행하기	414
20.4 미래의 모습: 네트워크 연구의 자동화	417
20.5 OHDSI 네트워크 연구의 정석	418
20.6 요약	420
Appendix	420

A Glossary	421
B Cohort definitions	427
B.1 ACE Inhibitor 사용 코호트	427
B.2 ACE inhibitor 단독 요법의 새로운 사용자 코호트 New User	428
B.3 심근경색 Acute myocardial infarction(AMI) 코호트	432
B.4 혈관부종 Angioedema	433
B.5 Thiazide 계 이뇨제 Thiazide-Like Diuretics 단독 요법 새로운 사용자 New User 코호트	434
B.6 고혈압 일차 요법 사용자 코호트	438
B.7 3년 이상 follow-up 된 고혈압 일차 요법 사용자 코호트	442
B.8 ACE Inhibitor 사용자 코호트	442
B.9 Angiotensin Receptor Blocker (ARB) 사용자 코호트	443
B.10 Thiazide 이뇨제 사용자 코호트	444
B.11 Dihydropyridine Calcium Channel Blocker (dCCB) 사용자 코호트	444
B.12 Non-Dihydropyridine Calcium Channel Blocker (ndCCB) 사용자 코호트	445
B.13 Beta-Blocker Use	445
B.14 Loop 이뇨제 사용자 코호트	446
B.15 칼륨보존 이뇨제 사용자 코호트	446
B.16 Alpha-1 길항제 사용자 코호트	447
C Negative controls	449
C.1 ACE inhibitor 와 thiazide 계 이뇨제	449
D Protocol template	453
E Suggested Answers	457
E.1 공통 데이터 모델	457
E.2 OMOP 표준 용어	462
E.3 추출 변환 적재	462
E.4 데이터 분석 이용 사례	464
E.5 SQL과 R	464
E.6 코호트 만들기	467
E.7 임상적 특성 분석	472
E.8 인구 수준 추정	479
E.9 환자 수준 예측	485

E.10 데이터의 질	487
Bibliography	489
Index	501

서문

이 책은 오딧세이 OHDSI, Observational Health Data Science and Informatics 커뮤니티가 작성한 The Book of OHDSI이다. 이 책은 OHDSI 관련 모든 지식의 중앙저장소 역할을 담당하고자 쓰여졌으며 오픈소스 개발 도구들을 통해 커뮤니티에 의해 관리되는 생명력있는 문서로 계속 진화하고 있다. 영문판: ohdsi.github.io/TheBookOfOhdsi/

이 책의 목표

이 책은 OHDSI 관련 모든 지식의 중앙저장소 역할을 담당하고자 쓰여졌으며 OHDSI 커뮤니티, CDM 데이터 기준과 OHDSI 도구에 중점을 두었다. OHDSI 초보자와 숙련자 모두를 위해 현실적으로 필요한 이론과 사용방법에 대한 교육을 제공하는 실용적인 목표를 가지고 있다. 이 책을 읽은 뒤 당신은 OHDSI란 무엇인가, 또한 그 여성에 어떻게 동참할 것인가에 관하여 이해하게 될 것이다. 또한 CDM과 표준화된 용어들이 무엇인지, 이러한 것들이 관찰 보건 데이터베이스의 표준화에 어떻게 사용되는지 알게 될 것이다. 이 데이터에 대해 임상적 특성 분석 clinical characterization, 인구 수준 추정 population-level estimation, 환자 수준 예측 patient-level prediction, 이 3가지 주요 이용 사례들을 배우게 될 것이다. 이 책을 통해 이 3가지 활동을 지원하는 OHDSI의 오픈 소스 도구와 사용법에 대해 익히게 될 것이다. 데이터 품질, 임상적 타당성, 소프트웨어 타당성, 방법론적 타당성 등에 관한 장들에서 CDM에서 생성된 근거들의 품질을 어떻게 확립했는지를 설명할 것이다. 마지막으로, 분산 연구망에서 이러한 연구들을 실행하기 위해 OHDSI를 어떻게 사용하는지를 배우게 될 것이다.

이 책의 구성

이 책은 5개의 주요 섹션으로 정리되어 있다:

- I) 오딧세이 커뮤니티

- II) 단일한 데이터 표현
- III) 데이터 분석법
- IV) 근거의 품질
- V) 오딧세이 연구

각 섹션은 다수의 장 Chapter으로 구성되어 있으며 각 장은 아래의 순서대로 해당 장에 맞게 구성되어 있다: 서론, 이론, 실행, 요약, 예제.

기여자

원문의 각 장은 해당 장을 이끈 주요 작성자들을 표기하고 있다. 그러나 주요 작성자 외에도 이 책을 완성하는데 기여를 한 많은 사람들이 있으며 아래의 기여자들에게 감사를 표한다:

Hamed Abedtash	Mustafa Ascha	Mark Beno
Clair Blacketer	David Blatt	Brian Christian
Gino Cloft	Frank DeFalco	Sara Dempster
Jon Duke	Sergio Eslava	Clark Evans
Thomas Falconer	George Hripcak	Vojtech Huser
Mark Khayter	Greg Klebanov	Kristin Kostka
Bob Lanese	Wanda Lattimore	Chun Li
David Madigan	Sindhoosha Malay	Harry Menegay
Akihiko Nishimura	Ellen Palmer	Nirav Patil
Jose Posada	Nicole Pratt	Dani Prieto-Alhambra
Christian Reich	Jenna Reps	Peter Rijnbeek
Patrick Ryan	Craig Sachson	Izzy Saridakis
Paola Saroufim	Martijn Schuemie	Sarah Seager
Anthony Sena	Sunah Song	Matt Spotnitz
Marc Suchard	Joel Swerdel	Devin Tian
Don Torok	Kees van Bochove	Mui Van Zandt
Erica Voss	Kristin Waite	Mike Warfe
Jamie Weaver	James Wiggins	Andrew Williams
Seng Chan You		

Table 1: Versions of packages in the Methods Library used in this book.

Package	Version
CaseControl	1.6.0
CaseCrossover	1.1.0
CohortMethod	3.1.0
Cyclops	2.0.2
DatabaseConnector	2.4.1
EmpiricalCalibration	2.0.0
EvidenceSynthesis	0.0.4
FeatureExtraction	2.2.4
MethodEvaluation	1.1.0
ParallelLogger	1.1.0
PatientLevelPrediction	3.0.6
SelfControlledCaseSeries	1.4.0
SelfControlledCohort	1.5.0
SqlRender	1.6.2

소프트웨어 버전

이 책의 많은 부분은 OHDSI의 오픈소스 소프트웨어를 다루고 있으며 이 소프트웨어는 시간이 지나면서 계속 진화해 나갈 것이다. 개발자들은 사용자에게 일관되고 안정적인 경험을 제공하고자 최선을 다할 것이나, 시간이 지나면서 소프트웨어의 개선으로 인해 불가피하게 이 책의 내용이 더 이상 맞지 않는 경우가 발생할 것이다. 이를 보완하기 위해 커뮤니티는 온라인 버전을 통해 변화를 계속 업데이트할 예정이며 새로운 에디션의 실물 책을 출간할 예정이다. 이 책이 쓰여진 버전의 소프트웨어 버전은 아래와 같다:

- ACHILLES: version 1.6.6
- ATLAS: version 2.7.3
- EUNOMIA: version 1.0.0
- Methods Library packages: 테이블 1참조

The Book of OHDSI가 쓰여진 과정

이 책의 원문인 The Book of OHDSI는 bookdown 패키지를 사용한 RMarkdown으로 쓰여졌다. 온라인 버전은 지속적 통합 시스템인 “travis”를 통해서 <https://github.com/OHDSI/TheBookOfOhdsi>의 저장소를 사용해 자동작성 되었다. 이러한 온라인 버전은 정기적으로 스냅샷 형식으로 저장되며 이렇게 저장된 파일을 “에디션”이라 표기한다. 이 에디션의 실물 책자는 아마존에서 구입 가능하다.

이 책이 번역된 과정

2019년 OHDSI 심포지엄에서 The Book of OHDSI가 배포된 이후, 한국 OHDSI 연구자들이 공동으로 번역작업을 진행하였다. 원문과 마찬가지로 bookdown 패키지를 동일하게 사용하여 작성하였다. 원문 또는 번역의 오류가 발견된다면 한국 오딧세이 포럼 <http://www.ohdsikorea.org/>에 활발한 의견 개진을 바란다.

한국 및 국제 OHDSI 네트워크의 발전을 위하여 대가를 바라지 않고, 번역 작업에 힘써주신 다음의 공동 번역자들에게 큰 감사의 말씀을 드린다.

이름	소속
강미라	성균관대학교
김도엽	아주대학교
김민아	삼성서울병원
김이석	한양대학교
김청수	아주대학교
박래옹	아주대학교
박유진	아주대학교
박지명	아주대학교
박철형	아주대학교
신서정	아주대학교
양영모	아주대학교
오송희	아주대학교
유승찬	아주대학교
유재용	성균관대학교
윤선영	삼성서울병원
이선경	아주대학교
이성원	아주대학교
이일동	성균관대학교
임지연	동국대학교
장동경	성균관대학교
장진성	삼성서울병원
전명훈	아주대학교
전호균	아주대학교
조재형	아주대학교

이름	소속
차원철	성균관대학교

한국판 번역에 부쳐

Martijn Schumie, David Madigan

한국 오딧세이 Korean Chapter of OHDSI는 유전체 및 방사선 영상 자료 등을 위한 공통 데이터 모델 Common Data Model의 확장, 새로운 OHDSI 소프트웨어 개발 및 OHDSI 기반 주요 임상 연구 등 OHDSI의 다양한 분야에 혁혁한 공헌을 해왔다. CDM의 광범위한 채택으로 인해 한국은 전국 규모의 탄탄한 분산연구망을 구축하였다. 한국의 OHDSI 연구자들은 한국 오딧세이 심포지엄을 조직하여 전세계의 연구자들을 한국으로 초대할 뿐 아니라, 미국, 유럽, 아시아 각국의 OHDSI 심포지엄에도 열성적으로 참여하고 있다. 우리는 OHDSI의 공동체 정신을 한국에서 생생하게 느낄 수 있다.

이 책의 원문인 The Book of OHDSI는 한국의 연구자들을 포함한 전세계 OHDSI 커뮤니티에 의해 OHDSI 커뮤니티를 위하여 작성되었다. 우리는 한국 오딧세이가 이 책을 단기간 내에 한국어로 번역한 것에 대해 경탄을 금치 않으며, 이 책의 번역이 한국 오딧세이에 중요한 이정표가 되리라 믿어 의심치 않는다.

Part I

OHDSI 커뮤니티

Chapter 1

OHDSI 커뮤니티

Chapter leads: Patrick Ryan & George Hripcsak

함께 모이면 시작되고, 함께 지내면 진보하고, 함께 일하면 성공한다. -헨리 포드

1.1 데이터에서 근거로의 여정

대학병원과 의원, 규제 기관 및 의료 제품 제조업체, 보험 회사 및 정책 기관, 그리고 환자와 의료 제공자 간의 모든 상호관계를 포함하는 전 세계 보건 의료의 어느 곳에서나 다음과 같은 공통적인 과제가 있다. 우리는 과거를 통해 배운 것을 어떻게 미래를 위하여 적용하여 더 나은 결정을 내릴 수 있을 것인가?

10년이 넘도록, 많은 사람들이 스스로 학습하는 보건의료 체계 learning healthcare system의 비전에 대해서 논의해 왔다. “그것은 각 환자와 의료 제공자가 함께 의료 행위를 결정할 때 필요한 최상의 근거를 생성하고 적용하기 위함이다. 또한, 환자 치료의 부산물로서 새로운 의학적 발견이 가능하도록 유도하며, 보건의료의 혁신, 질, 안전 및 가치를 보장하기 위함이다.” (Olsen et al., 2007) 이 원대한 계획의 주요한 요소는 일상적인 임상 치료 과정에서 수집된 환자 수준 patient-level의 데이터를 분석하여 실세계 근거 real-world evidence를 생성할 수 있으며, 의료 시스템에 전파되어 실제 임상에 정보를 제공 할 수 있으리라는 야심 찬 전망에 있다. 미국 의학 연구소 Institute of Medicine의 근거 중심 의학 원탁회 Roundtable on Evidence-Based Medicine은 2007년 보고서에서 “2020년까지 90%의 임상 결정이 정확하고, 시기적절하고, 최신의 임상 정보에 의해 뒷받침될 것이며, 그것은 가능한 최선의 근거를 반영할 것이다.”라고 예측했다. (Olsen et al., 2007) 비록 여러 가지 면에서 엄청난 발전이 있었지만, 우리는 여전히 이 위대한 열망에는 한참 미치지 못하고 있다.

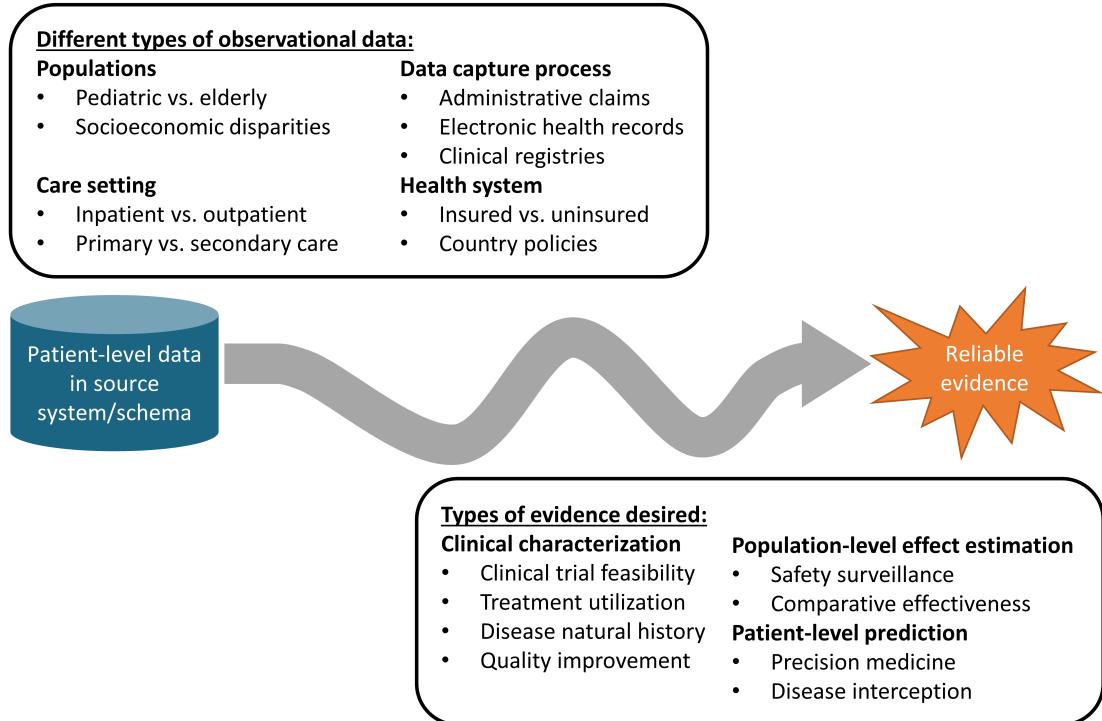


Figure 1.1: 데이터에서 근거로의 여정

무엇 때문인가? 부분적으로는 환자 수준의 데이터에서 신뢰할만한 근거를 생성하는 여정이 몹시 고되기 때문일 것이다. 데이터로부터 근거를 생성하는 과정에는 정해진 하나의 길이 없으며, 어떠한 지도도 그 길을 안내해주지 않는다. 사실, “데이터 data”가 무엇인지, 그리고 “근거 evidence”가 무엇인지에 대한 통일된 관념도 존재하지 않는다.

원천 시스템에는 다양한 환자 수준의 데이터를 수집하는 여러 유형의 관찰 데이터베이스 observational database가 있다. 이 데이터베이스는 서로 다른 의료 시스템 내부의 인구, 치료 설정 및 데이터 수집 프로세스의 이질성만큼 다양하다. 의사 결정에 도움이 될 수 있는 다양한 유형의 근거가 있으며, 분석 방법론에 따라 임상적 특성 분석 clinical characterization, 인구 수준 추정 population-level estimation 및 환자 수준 예측 patient-level prediction으로 분류할 수 있다. 출발지(원천 데이터)와 원하는 목적지(근거) 자체에 대한 이해뿐 아니라, 광범위한 임상, 과학 및 기술 역량들이 모두 필요하기에 이 여정은 더욱 어려워진다. 보험 청구나 진료 과정이 데이터로 수집되면서 보건 정책이나 보험 환급과 관련된 행동 동기들로 인해 데이터 수집 및 정제 과정에서 발생할 수 있는 비뚤림을 비롯하여 환자와 의료 제공자 간의 진료 현장에서 원천 데이터가 수집되는 전반적인 과정에 대한 철저한 의료정보학적 이해도 필요하다. 임상적 의문으로부터 해답을 도출하는 데 적합한 관찰 연구를 설계하기 위해선 역학 원칙과 통계적 방법도 숙지하고 있어야 한다. 수백만 명 환자의 수년간의 종적 추적에 걸친 수십억 건의 임상 관찰을 가진 데이터 세트에 대해 계산적으로 효율적인 데이터

처리 알고리즘을 구현하고 실행할 수 있는 기술적 능력 역시 필요하다. 관찰형 연구를 통해 습득한 내용을 다른 근거와 통합하고, 이 새로운 지식이 건강 정책 및 임상 관행에 어떤 영향을 미칠지 고려하기 위해서는 임상 지식 또한 필요하다. 그렇기 때문에 한 개인이 데이터를 이용하여 근거를 성공적으로 만들어 내는 데 필요한 기술과 자원을 모두 보유하는 것은 매우 드문 일이다. 따라서 이용 가능한 최선의 데이터를 가장 적절한 방법으로 분석하여 모든 이해당사자가 그들의 의사결정 과정에 믿고 사용할 수 있는 근거를 생산하기 위해서는, 종종 많은 개인이나 기관과의 협력이 필요하다.

1.2 OMOP (*Observational Medical Outcomes Partnership*)

협력 관찰형 연구 모델의 주목할만한 예시로 OMOP(*Observational Medical Outcomes Partnership*)이 있다. OMOP은 미국 식품의약국 FDA이 주관하고, 미국 국립 보건원 National Institutes of Health 관리하에 학술 연구자, 보건 데이터 파트너 및 협력 제약사간의 컨소시엄으로 구성되었으며, 관찰형 보건의료 데이터를 이용하여 능동적 의료 제품 안전성 감시의 발전을 꾀하고자 만들어진 민관 협력체였다. (Stang et al., 2010) OMOP은 다수의 이해관계자 간의 거버넌스 구조를 확립했고, 다수의 청구자료 및 전자 의무 기록 데이터베이스에 적용하여 참인 약물 안전성 연관성과 거짓 양성 소견을 식별할 수 있는 대안적인 역학 설계 및 통계 방법의 성능을 경험적으로 검증하는 일련의 방법론적 실험을 설계하였다.

분산된 관찰형 데이터베이스를 통해 연구를 진행하면서 기술적인 난제를 인식하고, 연구진들은 데이터의 구조, 내용 및 용어를 표준화하여 하나의 통계 분석 코드가 모든 데이터 파트너에서 공통으로 사용될 수 있도록, OMOP 공통 데이터 모델 Common Data Model(CDM)을 설계하였다. (Overhage et al., 2012) OMOP 실험은 공통 데이터 모델과 표준화된 어휘를 확립하는 것이 가능하다는 것을 증명하였으며, 이는 서로 다른 의료체계에서 다른 용어체계를 통해서 생성된 다른 데이터 유형을 수용하여 기관 간 협업과 계산적으로 효율적인 분석을 용이하게 할 수 있는 방식으로 구현되었다.

OMOP은 처음부터 오픈 사이언스 정책을 채택하여 연구 설계, 데이터 표준, 분석 코드, 경험적 결과 등 모든 작업의 결과를 공공에 배포함으로써 투명성을 증진하고, OMOP이 수행하고 있는 연구에 대한 신뢰를 쌓을 뿐 아니라, 또한 다른 이들의 연구 목적을 위하여 발전할 수 있도록 하였다. OMOP의 원래 초점은 약물 안전성이었지만, 의학적 개입이나 보건 시스템 정책에 대한 비교 효과연구를 포함하여 다양한 분석 사용사례를 지원하기 위해 지속적으로 발전하고 있다.

OMOP은 대규모의 경험적 실험 large-scale empirical experiments을 성공적으로 수행하였고, (Ryan et al., 2012, 2013b) 방법론적인 혁신을 이루고, (Schuemie et al., 2014) 관찰형 데이터를 이용한 안정성에 관련된 의사결정에 유용한 지식 생성을 위한 적절한 방법론을 제시하였다. (Madigan et al., 2013b,a) OMOP 프로젝트는 종료되었지만, 오픈 사이언스 원칙과 함께 OMOP의 유산은 OHDSI

가 이어받았다.

OMOP 프로젝트가 FDA의 능동 감시에 도움을 줄 수 있는 관찰형 연구를 완료하고 종료된 이후, 사람들은 OMOP의 종료가 새로운 여정의 시작이 되어야 한다고 생각했다. OMOP의 방법론적 연구가 관찰 데이터에서 생성되는 근거의 품질을 명시적으로 개선할 수 있는 모범 사례 best practice 를 제시하였지만, 많은 연구자들은 이러한 모범 사례를 채택하기를 주저했다. 이를 방해하는 몇 가지 장애물들이 있었는데, 1) 방법론적인 혁신을 내세우기 전 관찰형 자료의 품질에 대한 근본적인 우려 2) 방법론적 문제와 해결책에 대한 불충분한 개념적 이해 3) 개별 데이터 파트너의 로컬 환경 내에서 솔루션을 독립적으로 구현할 수 없다는 점 4) 이러한 접근방식이 다른 연구자들이 관심이 있는 임상적 문제에 적용 가능한지에 대한 불확실성 등이었다. 이러한 모든 장애물에 대해 변화를 만들기 위해서는 한 개인의 힘이 아니라, 여러 사람이 협력하여야만 한다는 것을 깨달을 수 있었다. 다음과 같은 협력이 필요했다:

- 기초 데이터 품질에 대한 신뢰도를 높이며 구조, 콘텐츠 및 의미론적 일관성을 촉진하여 표준화된 분석이 가능하도록 개방형 커뮤니티 open community의 데이터 구조, 어휘 및 추출 변환 적재 Extract-Transform-Load(ETL) 표준규약 구축을 위한 협업
- 약물 안전성 연구 외에도 임상적 특성 분석, 인구 수준 추정 및 환자 수준 예측을 위한 보다 광범위한 모범 사례를 확립하기 위한 협업. 방법론적 연구를 통해 입증된 과학적 모범 사례를 코드로 구현하고 연구자들이 쉽게 채택할 수 있는 오픈 소스 분석 소프트웨어 개발에 대한 협업
- 주요한 보건 문제를 해결할 공통의 질문에 대한 임상 적용을 위한 협업으로써, 커뮤니티를 아울러서 데이터에서 근거로의 여정을 총괄적으로 인도해줄 수 있는 협업체계

이러한 통찰을 통해 OHDSI가 태어났다.

1.3 개방형 과학 공동체로서의 OHDSI

OHDSI(Observational Health data Sciences and Informatics)는 보다 더 나은 의료 결정과 더 나은 보건 관리를 촉진할 수 있는 과학적 근거를 공동으로 생성하도록 함으로써 보건 수준을 향상하는 것을 목표로 하는 개방형 과학 공동체다. (Hripcsak et al., 2015) OHDSI는 관찰 보건 데이터 observational health data의 적절한 사용에 대한 과학적 모범 사례를 확립하기 위한 방법론적 연구를 수행하고, 이러한 연구방법론을 일관되고 투명하며 재현 가능한 솔루션으로 코드화하는 오픈 소스 분석 소프트웨어를 개발하여, 보건의료 정책 및 환자 치료에 도움이 될 수 있는 임상적 근거를 마련하는 데에 적용할 수 있도록 노력한다.

1.3.1 OHDSI의 사명 Mission

더 나은 의학적 결정과 의료 발전을 촉진할 수 있는 근거를 상호협력하에 생성할 수 있도록 공동체에 힘을 실어줌으로써 보건을 개선한다.

To improve health by empowering a community to collaboratively generate the evidence that promotes better health decisions and better care.

1.3.2 OHDSI의 이상 Vision

관찰형 연구를 통해 건강과 질병에 대한 포괄적인 이해가 가능한 세상

A world in which observational research produces a comprehensive understanding of health and disease.

1.3.3 OHDSI의 목표 Objectives

- 혁신 Innovation: 관찰 연구는 파괴적 사유를 통해서 가장 큰 혜택을 얻을 수 있는 분야이다. 우리는 우리의 업무에 새로운 방법론적인 접근을 적극적으로 찾고 격려한다.

Observational research is a field which will benefit greatly from disruptive thinking.
We actively seek and encourage fresh methodological approaches in our work.

- 재현 Reproducibility: 보건향상을 위해서는 정확하고 재현 가능하며 잘 보정된 근거가 필요하다.

Accurate, reproducible, and well-calibrated evidence is necessary for health improvement.

- 공동체 Community: 우리는 OHDSI에 적극적으로 참여하는 모든 사람 (환자, 의료직 전문가, 연구자, 또는 단순히 우리의 주장을 믿는 사람)을 환영한다.

Everyone is welcome to actively participate in OHDSI, whether you are a patient, a health professional, a researcher, or someone who simply believes in our cause.

- 협력 Collaboration: 우리는 우리 공동체 참여자들의 현실적 요구를 최우선으로 다루기 위해서 함께 일한다.

We work collectively to prioritize and address the real world needs of our community's participants.

- **개방 Openness:** 우리는 방법론, 도구, 우리가 생성하는 근거 등 우리 공동체의 모든 진행 사항을 개방하고 공개적으로 접근 가능할 수 있도록 최대한 노력한다.

We strive to make all our community's proceeds open and publicly accessible, including the methods, tools and the evidence that we generate.

- **선행 Beneficence:** 우리는 우리 공동체에 속한 개인과 기관의 권리를 보호하기 위해서 항상 노력한다.

We seek to protect the rights of individuals and organizations within our community at all times.

1.4 OHDSI의 역사

OHDSI는 2014년 설립된 이래 성장을 지속하여 컴퓨터 과학, 역학, 통계, 의생명 정보학, 보건 정책 및 임상 의학 등 다양한 분야를 대표하는 학계, 의료 제품 산업, 규제 기관, 정부, 보험자, 기술 제공자, 의료 시스템, 임상의사 및 환자 집단 등 2,500명 이상의 다양한 이해관계자가 온라인 포럼에서 활동하고 있다. OHDSI 협력체로써 자발적으로 보고한 기관 및 데이터베이스의 리스트는 OHDSI 웹사이트에서 확인할 수 있다.¹ OHDSI 협력자 지도 (Figure 1.2)는 폭넓은 국제 공동체로서의 다양성을 상기시킨다.

OHDSI는 OMOP-CDM이라는 개방형 공동체 데이터 표준 기반으로 2019년 8월 기준으로 20여 개국, 100개 이상의 의료 데이터베이스들로 구성된 분산 연구망 distributed research network(DRN)를 구축했다. 분산 연구망에서는 환자 수준의 데이터를 개인이나 조직 간에 공유할 필요가 없다. 분산 연구망에서는 데이터를 기관 폐쇄망 안에 두고 연구자는 프로토콜 형태의 분석 코드/프로그램을 공유하기 때문이다. 데이터 파트너들은 연구자의 요청에 따라 기관 안에서 연구 프로토콜을 실행해 자동으로 생성되는 요약 집합정보 (평균, 합, 표준편차, 교차비, 위험도 등)만 연구자에게 회신하는 방식으로, 연구자는 폐쇄망 안에 있는 환자의 개별 정보를 보거나 취득하지 않는다. OHDSI 분산 망에서 각 데이터 파트너는 환자 수준 데이터의 사용에 대한 완전한 자율성을 유지하고, 각 기관이 기관의 데이터 거버넌스 정책을 지속해서 준수할 수 있다.

OHDSI 개발자 커뮤니티는 3가지의 사용 사례를 지원하기 위해 OMOP CDM 위에 다음 3가지의 강력한 오픈 소스 분석 소프트웨어 라이브러리를 구축하였는데 이는 다음과 같다. 1) 임상적 특성 분석: 질병의 자연 경과, 치료 행태 및 질 향상을 위한 임상 특성 분석 2) 인구 수준 추정: 의약품 안전성 감시 및 비교 효과 연구에서의 인과성 분석 3) 환자 수준 예측: 기계학습 알고리즘을 활용한 정밀 의학 또는 의료 개입. OHDSI 개발자들은 OMOP CDM의 채택, 데이터 품질 평가, OHDSI 네트

¹<https://www.ohdsi.org/who-we-are/collaborators/>

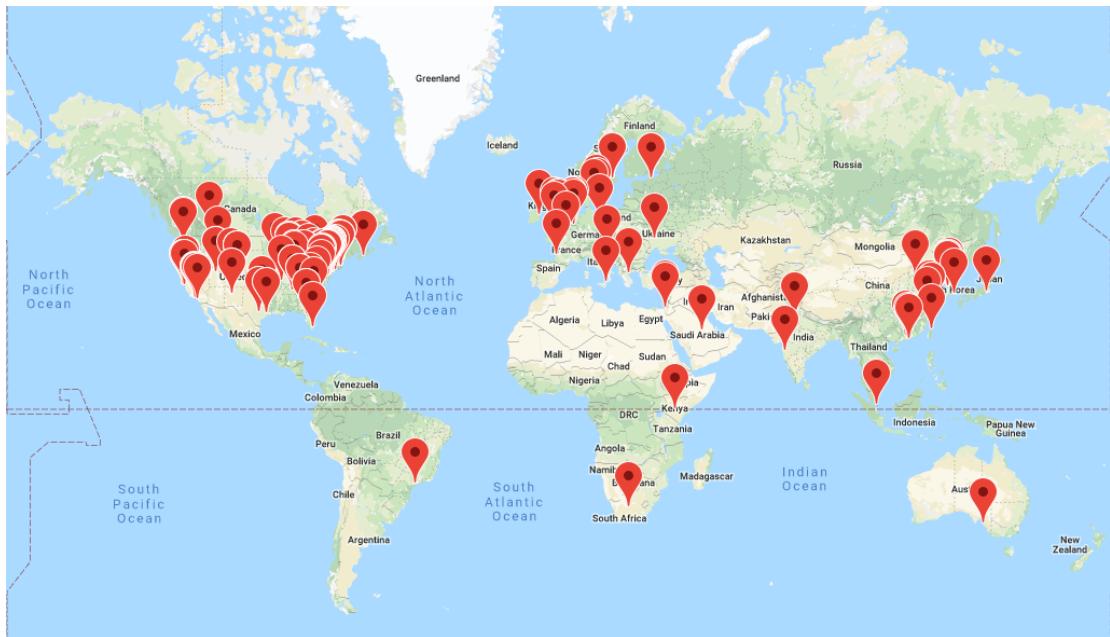


Figure 1.2: 2019년 8월 기준 OHDSI 협력자 지도

워크 연구의 촉진을 지원하는 애플리케이션을 개발하고 있다. 이러한 소프트웨어에는 R과 Python에 내장된 백 엔드 통계 패키지 및 HTML과 Javascript로 개발된 프론트엔드 웹 어플리케이션이 포함된다. 모든 OHDSI 소프트웨어들은 오픈 소스 정책을 채택하여 Github을 통해 공개된다.²

오픈 소스 소프트웨어들과 함께, OHDSI의 개방형 과학 공동체적 접근은 관찰형 연구의 발전을 가능하게 했다. 첫 번째 OHDSI 네트워크 연구는 당뇨, 우울증, 고혈압의 3가지 만성 질병에 대한 치료 패턴을 분석하는 것이었다. PNAS(Proceedings of the National Academy of Science)에 출판된 연구는, 그때까지 수행된 최대 규모의 관찰 연구로써 11개의 데이터베이스에서 2억 5천만 명의 환자 데이터를 이용하여 이전에 보고된 적 없는 치료 패턴의 지역적 차이 및 환자별 치료 선택에 대한 이질성에 대해 발표하였다. (Hripcsak et al., 2016) OHDSI는 교란변수를 통제하는 새로운 통계적 방법론을 제시하였고, (Tian et al., 2018) 인과성 검증 능력에 대해 검증하였고, (Schuemie et al., 2018a) 이러한 방법론을 뇌전증 약제의 개별 안전성 연구 (Duke et al., 2017) 및 당뇨병의 이차 약제의 비교 효과 연구 (Vashisht et al., 2018), 우울증 치료의 대규모 비교 효과 연구 (Schuemie et al., 2018b), 고혈압 환자의 이제 병합 요법의 비교 효과 연구 (You et al., 2019), 대규모 고혈압 약제 비교 연구 (Suchard et al., 2019)에 활용하였다. OHDSI 공동체는 또한 관찰형 보건의료 데이터의 기계학습 알고리즘을 활용한 프레임 워크를 구축 (Reps et al., 2018) 하여 다양한 치료 분야에 활용하였다. (Johnston et al., 2019; Cepeda et al., 2018; Reps et al., 2019)

²<https://github.com/OHDSI>

1.5 OHDSI와의 협업

OHDSI는 근거를 생성하기 위해 협업을 강화하는 것을 목표로 하는 공동체인데, OHDSI 참가자가 된다는 것은 무엇을 의미하는가? 만약 당신이 OHDSI의 사명을 믿고 데이터에서 근거에 이르는 여정의 어디든지 기여를 하는 데 관심이 있다면, OHDSI는 당신을 위한 공동체가 될 수 있다. OHDSI 참가자는 보건 의료 데이터에 접근이 가능하고, 이를 활용해 의학적 근거를 생성하고 싶은 개인일 수 있다. OHDSI 참가자는 과학적 모범 사례를 수립하고 대안적 접근법을 평가하는 데 관심이 있는 방법론 연구자일 수 있다. OHDSI 참가자는 OHDSI의 타 연구자들이 사용할 수 있는 도구를 만들기 위해 프로그래밍 기술을 적용하는 데 관심이 있는 소프트웨어 개발자일 수 있다. OHDSI 참가자는 중요한 의학 보건학적 질문을 가지고 있고 논문 발표 등을 통해 그러한 질문들에 대한 근거를 더 확장 더 큰 의료 커뮤니티에 제공하고자 하는 임상 연구자일 수 있다. OHDSI 참가자는 공공 보건을 위해 이러한 공통적인 사명과 가치를 믿고 해당 지역의 공동체가 OHDSI 관련 교육과 심포지엄 개최를 포함하여, 그 임무를 지속할 수 있도록 자원을 제공하는 개인 또는 단체일 수도 있다. 당신의 배경이나 소속과 관계없이, OHDSI는 개개인이 공통의 목적을 위해 함께 일할 수 있는 공동체가 되기를 추구하고 있으며, 각 개인이 공동으로 의료를 발전시킬 수 있는 기여를 하고 있다. 이 여정에 함께하고 싶다면, 2장 (“OHDSI 시작하기”)을 통해 어떻게 시작하는지 배울 수 있다.

1.6 한국 OHDSI의 역사

OMOP의 창립자 중 한 명인 Martijn Schuemie는 OMOP의 연구성과 중 하나로서 관찰형 자료에서 confounding by indication을 찾을 수 있는 LEOPARD 알고리즘을 고안하였고, 2010년 남아프리카 케이프타운에서 열린 세계의료정보학회(IMIA)에서 그 내용을 발표하였다. 당시 세계의료정보학회에 참석하였던 박래웅은 우연히 Martijn Schuemie의 LEOPARD 알고리즘과 OMOP을 접하게 되었고 강한 흥미와 유대감을 느꼈다. 이후 그는 2012년 국내 4개 대학병원의 다기관 임상 의료 정보 통합시스템 개발을 진행하였고 이를 위해 자체적으로 고안한 CDM을 적용하였다. 2013년 세계약물역학학회(ICPE)가 그 해 9월 캐나다 몬트리올에서 열렸고 이 학회에서 다시 만난 Martijn Schuemie, Patrick Ryan과 박래웅은 각자 진행하던 프로젝트에 대해서 논의하였고 향후 긴밀한 협조를 결의하였다. 이후 그는 빠른 시간 내에 아주대병원 전자의무기록을 CDM으로 변환 완료하고 2014년 OHDSI의 결성을 알리는 컬럼비아대학에서 열린 첫 번째 face-to-face 모임에 참여하면서 변환 완료된 아주대병원의 CDM과 Achilles 웹페이지를 전격 공개하였다. 미국 이외의 국가에서 변환된 첫 번째 CDM이며 전 세계에서 첫 번째로 공개된 Achilles 페이지였다. Achilles

그는 2014년 6월 이후 본격적으로 한국 사회에 OHDSI를 알리기 시작하였고, 이후 국민건강보험공단을 시작으로 가천길병원 등이 OHDSI에 참여하기 시작하였다. 이후 계속 국내외에서 OMOP-CDM, OHDSI 전파를 위해 노력한 결과, 2016년부터는 최초로 국제 OHDSI committee에서 개별 국가를



Figure 1.3: 2014년 최초 개최된 OHDSI Face to Face 미팅에서 아주대학교병원의 CDM과 Achilles 웹페이지를 소개하였다. 사진의 좌측하단에 Christopher Knoll이 아주대 CDM Achilles 화면을 살펴보고 있으며 많은 참석자가 각별한 관심을 보였다.

위한 포럼 Korean chapter 을 개설하고, 한국의 OHDSI 참여가 본격화되었다. 첫 한국 국제 OHDSI 심포지엄은 2017년 3월 아주대학교에서 튜토리얼, 리더십 미팅을 포함하여 3일간 개최되었다.

한국 OHDSI 네트워크에 참여를 희망하는 병원 관계자들과 함께 2017년 3월 7일 첫 번째 리더십 미팅을 가진 후 현재까지 2달마다 전국의 의과대학/병원을 순회하며 총 15회 이상의 한국 OHDSI 리더십 미팅을 개최하며 OHDSI 전파 및 상호 협력을 꾀하고 있다.



Figure 1.4: 2017년 한국에서의 OHDSI 국제 심포지엄

1.7 요약



- OHDSI의 사명은 참여 공동체의 상호협력 하에 의료 발전을 촉진하는 근거를 생성하는 능력을 부여하는 것이다.
- OHDSI의 이상은 혁신, 재현, 공동체, 개방, 협력, 선행의 목표를 바탕으로 의료 빅데이터의 분석을 통해 세계에 건강과 질병에 대한 포괄적인 이해를 제공하는 것이다.
- OHDSI 참가자들은 개방형 공동체로서의 데이터 표준, 방법론 연구, 오픈소스 분석 소프트웨어 개발 및 임상적 적용을 통해 데이터에서부터 근거로의 여정을 발전시키고자 노력한다.

Chapter 2

OHDSI 시작하기

Chapter leads: Hamed Abedtash & Kristin Kostka

“천리길도 한 걸음부터” - 노자

OHDSI 커뮤니티는 학계, 산업계 및 정부 기관 전반에 걸쳐 다양한 이해관계자들을 대표하고 있다. 본 커뮤니티의 작업으로 의료 시스템뿐 아니라 환자, 의료제공자, 연구자들을 포함한 다양한 개인들과 기관들이 혜택을 받게 된다. 이러한 이점은 의료 데이터를 더 유용하도록 개선할 뿐만 아니라 의료데이터 분석의 질을 향상함으로써 얻어지게 된다. 관찰 연구는 파괴적인 생각 disruptive thinking으로부터 크게 혜택을 받을 수 있는 분야이다. 이 분야에서는 적극적인 새로운 방법론적 도입이 필요하다.

2.1 여정에 동참하기

환자, 의료 전문가, 연구자 혹은 단순히 OHDSI의 목적에 동감하는 사람이면 누구든지 OHDSI 커뮤니티에 적극적으로 참여할 수 있다. OHDSI는 포용적 멤버십 모델을 추구하며 OHDSI의 공동 연구자가 되기 위한 멤버십 비용은 없다. 참여를 원하는 사람은 단지 손을 들기만 하면 매년 OHDSI 멤버십 카운트에 포함될 것이다. 참여는 전적으로 자의에 의한 것이며 매주 커뮤니티의 네트워크 스터디나 OHDSI 작업 그룹에 참여하는 것만으로도 충분하다. 꼭 데이터를 보유하고 있어야만 OHDSI 커뮤니티의 활동 멤버가 되는 것은 아니다. 본 커뮤니티는 데이터 보유자, 연구자, 헬스케어 제공자, 환자와 소비자 모두에게 도움을 주고자 한다. 공동연구자의 프로필은 OHDSI 웹사이트에서 관리되고 정기적으로 업데이트되고 있다. 멤버십은 OHDSI 커뮤니티 원격회의, 워크그룹, 지역별 모임을 통해 육성되고 있다.

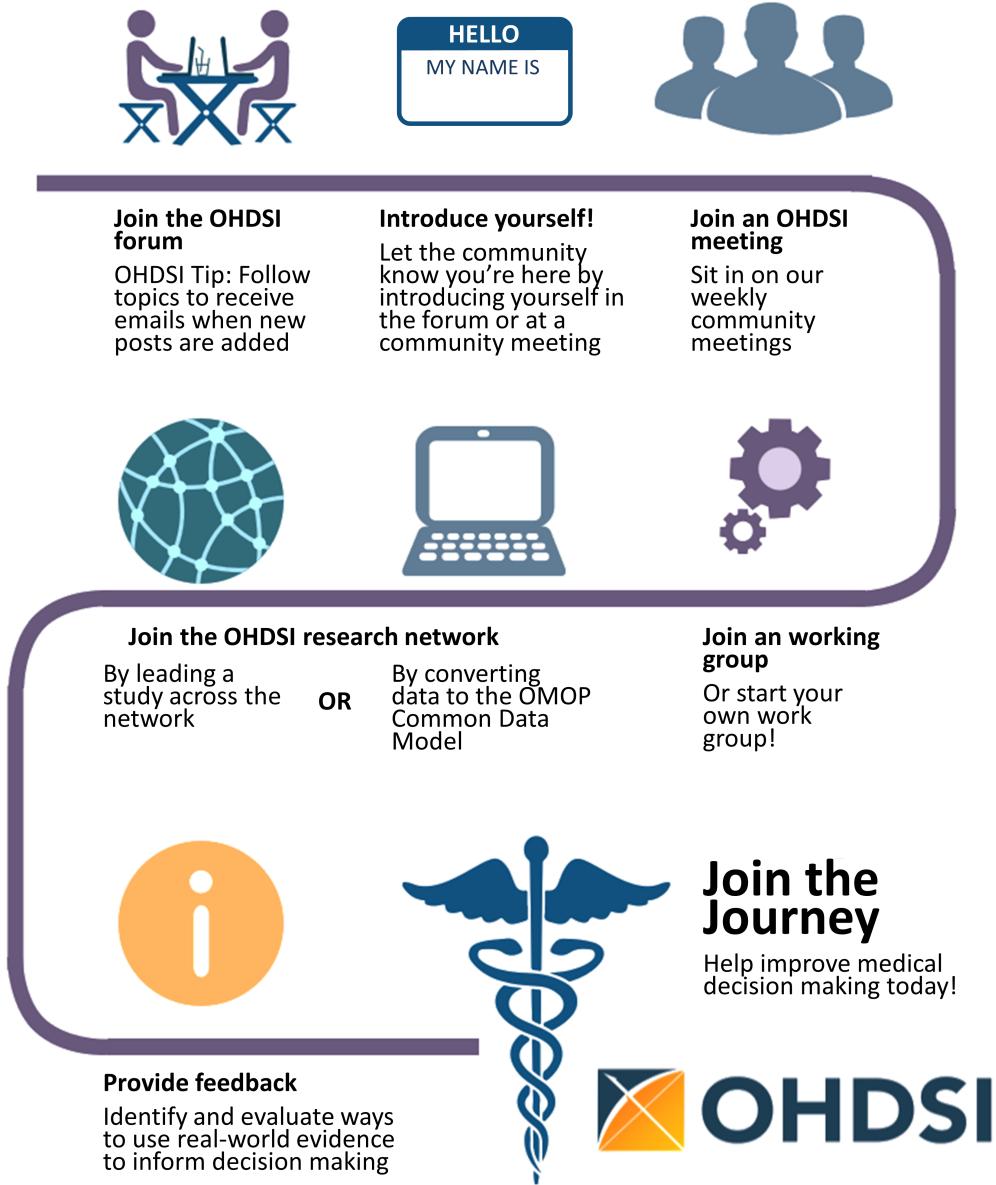


Figure 2.1: 여정에 동참하십시오 - OHDSI의 협력자가 되는 법.

2.1.1 OHDSI 포럼

OHDSI 포럼¹은 OHDSI 커뮤니티 공동연구자들이 메시지를 올리는 형식을 통해 대화하는 온라인 토론 사이트이다. 포럼은 트리와 같은 구조로 구성되었다. 가장 상위에는 “카테고리”가 있으며 관련성 있는 토론 카테고리로 나눠진다. 각 카테고리 아래로는 하위 포럼과 추가적인 하위 포럼들로 구성된다. 각 주제 (스레드라고도 불림)의 가장 낮은 하위 포럼에서 포럼 멤버들 간의 토론 혹은 포스트가 작성된다.

OHDSI 포럼에서는 다음을 포함한 콘텐츠 카테고리를 찾을 수 있다:

- **일반 General:** OHDSI 커뮤니티와 참여 방법에 대한 전반적인 토론
- **구현 Implementers:** 로컬 환경에서 공동 데이터 모델과 OHDSI 분석 프레임워크를 구현하는 방법에 대한 토론
- **개발자 Developers:** OHDSI 어플리케이션의 오픈 소스 개발과 OMOP CDM과의 균형을 위한 도구에 관한 논의
- **연구자 Researchers:** OHDI 연구 네트워크 기반의 근거 생성, 공동 연구, 통계적 방법과 기타 CDM 기반 연구에 대한 토론
- **CDM 개발자 CDM Builders:** 진행 중인 CDM을 위한 조건, 용어 vocabulary 그리고 엔지니어링 요소들에 관한 토론
- **용어 사용자 Vocabulary Users:** 용어 vocabulary 콘텐츠에 관한 토론
- **지역 지부 Regional Chapters(예를 들면, 한국, 중국, 유럽):** 지역별 언어로 진행되며 지역별 OMOP 구현과 OHDSI 커뮤니티 활동에 관한 토론

개별적인 주제로 포스팅을 올리려면 먼저 계정 등록을 해야 한다. 포럼 계정을 만들고 나면 General Topic 아래 “Welcome to OHDSI! – Please introduce yourself”라는 토픽에 다음과 같이 본인 소개를 하는 것을 추천한다. 1) 본인 소개 및 본인의 업무 소개 2) 커뮤니티 안에서 어떤 방식으로 도움을 줄 수 있는지 (예를 들면, 소프트웨어 개발, 연구, 논문 작성 등)를 본인 소개에 설명한다. 이제 당신은 OHDSI 여정에 동참하였다! 이후엔 토론에 참여하는 것을 권장한다. OHDSI 커뮤니티 포럼을 통해 자신의 질문을 포스팅하고 새로운 아이디어를 내고, 협업에 참여하길 바란다.



토픽을 “watch” 할 수도 있다. 이 뜻은 관심 있는 토픽에 새로운 포스트가 올라올 경우, 이메일로 안내를 받고 이메일 회신을 통해 다시 답장을 보낼 수도 있다는 것이다. 앞으로 다가올 미팅에 대한 아젠다도 확인할 수 있으며 공동작업 기회와 주간 OHDSIダイジェスト를 이메일로 받을 수 있다.

¹<http://forum.ohdsi.org>

2.1.2 OHDSI 이벤트

OHDSI는 정기적으로 직접 참여가 가능한 이벤트를 개최하여 공동연구자들이 서로 학습하고 향후 협력 관계를 강화할 기회를 제공한다. 이러한 이벤트는 OHDSI 웹사이트를 통해 전달되며 참석에 관심이 있는 사람들에게 무료로 제공된다.

OHDSI 심포지엄은 미국, 유럽, 아시아 등에서 매년 개최되는 과학 컨퍼런스로, 이를 통해 공동 연구자들은 총회, 포스터 발표 및 소프트웨어 시연 등을 통해 각각의 최신 연구를 발표할 수 있다. OHDSI 심포지엄은 OHDSI 커뮤니티에서 진행되고 있는 최신의 동향을 배울 수 있는 최적의 장소이다. 일반적으로 OHDSI 심포지엄에서는 새로운 커뮤니티 참여자들이 데이터 표준이나 분석 방법 등에 대해 배울 수 있도록 OHDSI 튜토리얼을 같이 진행한다.

OHDSI 공동연구자들의 대면 이벤트 face-to-face event는 좀 더 규모가 작은 포럼인데, 일반적으로 공동으로 관심이 있는 특정 문제들을 중심으로 구성된다. 지난 이벤트 중에는 표현형 해커톤 phenotype hack-a-thon, 데이터 질 해커톤 data quality hack-a-thon, 오픈소스 다큐멘테이션론 documentation-a-thon 등이 있었다. OHDSI는 다양한 스터디톤 study-a-thon 이벤트를 개최해 왔으며, 이를 통해 공동연구자들이 며칠간 함께 팀이 되어 특정 연구주제를 위한 적절한 분석을 수행하고 OHDSI 네트워크에 관한 학습, 많은 사람에게 알릴 수 있는 근거를 생성할 기회를 제공하였다. 이런 행사를 통해 공동의 문제를 해결하려는 열망뿐 아니라, 배움과 지속적인 발전을 도모하는 우호적 환경을 OHDSI 내에서 제공하고자 하는 관심도 대두되었다.

OHDSI 커뮤니티의 힘을 보다 자세히 배우기 바란다. OHDSI 웹사이트의 OHDSI Past Events section에서 지난 심포지엄, 대면 이벤트, OHDSI 튜토리얼 등을 접할 수 있다.

2.1.3 OHDSI 커뮤니티 원격회의

OHDSI 커뮤니티 주간 원격회의 OHDSI call는 매주 OHDSI 커뮤니티 안에서 발생하는 활동들에 대해 배울 기회이다. 한국 시간으로 매주 수요일 새벽 2시 (미국 동부 시간 기준 화요일 오후 12시부터 1시)에 원격회의로 진행되고 있으며 OHDSI 소프트웨어의 최근 개발 사항뿐 아니라 개별 공동 연구자들 및 그룹 활동과 커뮤니티의 전체적인 성과를 알 기회이다. 이 미팅은 모두 녹화되고 있으며 과거 발표자료들은 OHDSI 웹사이트 리소스에서 확인할 수 있다.

우리는 모든 OHDSI 공동 연구자들이 주간 원격회의에 참석하고 커뮤니티 토론을 위한 주제를 제안하기를 바란다. OHDSI 커뮤니티 원격회의는 연구 결과를 공유하고 현재 활발히 진행 중인 작업에 대한 의견을 제시하고 피드백을 얻으며, 개발 중인 오픈소스 소프트웨어를 시연하고, 데이터 모델링과 분석에 대한 모범사례를 커뮤니티와 함께 논의하고, 연구비 grant/논문 출판/컨퍼런스 워크샵 등을 위한 미래의 공동 작업 기회에 대해 많은 아이디어를 논의하는 장이 될 수 있다. 만약 원격회의

발표와 관련한 아이디어가 있다면 OHDSI 포럼에 글을 올릴 수 있다.

OHDSI 신입이라면 원격회의를 통해 OHDSI 네트워크 내에서 일어나는 일들에 대하여 알아가는 것이 좋을 것이다. OHDSI 원격회의에 참여하기 원한다면 OHDSI Wiki를 참고하기 바란다. 커뮤니티 원격회의의 주제는 매우 다르다. OHDSI 포럼의 OHDSI 주간 다이제스트를 통해 매주 발표주제에 관한 정보를 받을 수 있다. 원격회의마다 처음으로 참여하는 사람들의 배경과 OHDSI 가입 동기에 관한 소개를 받는 시간을 가진다.

2.1.4 OHDSI 워크그룹

OHDSI에는 워크그룹 Workgroup 팀들이 이끌어가는 다양한 프로젝트가 있다. 각각의 워크그룹은 커뮤니티에 기여하기 위한 프로젝트의 목적, 목표, 세부사항 등을 결정하는 리더십을 가지고 있다. 프로젝트 목적과 목표에 기여하고 싶은 참가자라면 누구나 워크그룹에 참여할 수 있다. 워크그룹은 장기적인 목표를 위해 오랫동안 유지되기도 하고, 커뮤니티의 특정 필요를 충족시키기 위한 단기 프로젝트를 위해 짧게 유지되기도 한다. 워크그룹의 정기 미팅은 프로젝트 리더들에 의해 결정되며 그룹마다 각각 다르다. 활동 중인 워크그룹의 리스트는 OHDSI Wiki에서 관리되고 있다.

테이블 2.1은 활동 중인 OHDSI 워크그룹의 레퍼런스를 제공한다. 해당 프로젝트에 적극적으로 참여하여 배우길 바란다.

Table 2.1: 주목할 만한 OHDSI 워크그룹

이름	목적	대상 인원
Atlas & WebAPI	Atlas & WebAPI는 OHDSI 오픈소스 소프트웨어 중 하나로 OMOP CDM 기반의 표준화된 분석 기능을 제공하고 있다. 이를 발전시키는 것이 목적	오픈소스 Atlas/WebAPI 플랫폼의 개선과 기여하고 싶은 Java와 JavaScript 소프트웨어 개발자들
CDM & Vocabulary	임상 환자 빅데이터의 대규모 분석을 위한 체계적이고 표준화된 OMOP CDM의 지속적인 개발. 타 워크그룹에 의해 개발된 표준화된 분석을 지원하고, 국제 코딩 시스템의 커버리지를 확장하기 위해 표준화된 Vocabulary의 질적 개선이 목적	OMOP CDM의 데이터 구조와 표준 용어를 개선하고 싶은 사람

이름	목적	대상 인원
Genomics	다양한 시퀀싱 작업의 결과로 나오는 유전자 변이 정보를 OMOP CDM과 연동 가능한 형태로 저장하기 위한 Genomic CDM 확장 모델 개발	유전자 데이터 연구에 관심이 있거나 경험이 있는 사람
Population-Level	정확하고 믿을 수 있으며 재현 가능한 관찰	제한 없음
Estimation	연구의 방법론을 개발하며 이러한 방법의 사용을 촉진	
Natural Language Processing	OHDSI 관찰 데이터베이스에서 문서 데이터의 사용을 촉진하고 이를 위한 소프트웨어와 방법론을 개발	제한 없음
Patient-Level	정확하고 잘 보정된 환자 중심의 표준화된 머신러닝 예측 모델 프로세스를 구축하여	제한 없음
Prediction	다양한 관심 영역에 사용할 수 있게 하며, 또한 어떤 소집단 환자의 데이터에도 적용할 수 있도록 함	
Gold Standard	OHDSI 참여자들이 함께 검증한 표현형 phenotype 정의와 다른 커뮤니티에서 개발한 표현형 정의를 발견, 평가, 활용하도록 함	표현형(Phenotype)의 큐레이션과 입증에 관심이 있는 사람
Phenotype Library	OMOP CDM과 FHIR 통합에 대한 로드맵을 수립하고 OHDSI와 FHIR 상호 간에 서로의 도구와 API를 활용하여 데이터와 연구의 발전을 꾀함	상호 운용성 interoperability에 관심 있는 사람
FHIR Workgroup	OMOP CDM을 확장하여 OHDSI 도구를 활용하여 환경지리적 요인과 환자의 임상적 표현형간의 관련성을 조사	보건 관련 지리학적 연구에 관심 있는 사람
Clinical Trials	OHDSI 플랫폼과 에코시스템을 활용하여 임상 시험 개념 및 수행을 지원하고자 함	임상 시험에 관심 있는 사람

이름	목적	대상 인원
THEMIS	OMOP CDM ETL 프로토콜들의 질과 재현성을 촉진하기 위하여 기본적인 OMOP CDM 규칙에 더하여 표준 규약을 발전시킴	제한 없음
Metadata &	수작업 또는 컴퓨터로 작성한 메타데이터 저장의 표준 프로세스와 CDM의 주석	제한 없음
Annotations	annotation 형태를 정의하여 연구자들이 관찰 데이터 세트에서 유용한 데이터 아티팩트를 소비하고 만들어 낼 수 있도록 함	
Patient Generated Health Data (PGHD)	스마트폰, 앱, 웨어러블 기기를 통해 생성된 PGHD 데이터의 ETL 규칙, 임상 데이터와의 통합, PGHD의 분석 Data(PGHD)프로세스의 개발	제한 없음
Women of OHDSI Steering Committee	OHDSI 커뮤니티 내부의 여성들이 함께 모여 과학계, 테크놀로지, 엔지니어링, 수학 (STEM) 분야에서 여성으로 겪는 도전을 나누기 위한 포럼 제공. 여성들의 입장에서 관점, 우려 사항, 아이디어를 나누며 OHDSI 커뮤니티가 STEM 분야의 여성들을 지원할 수 있을지에 대한 의견 교환. 궁극적으로 여성들이 존경받는 분야에서 여성의 리더가 될 수 있도록 장려 모든 OHDSI 활동과 이벤트가 발전해나가는 커뮤니티의 필요사항과 부합하고 있는지 확인함으로 OHDSI의 사명과 비전, 가치를 유지함. 또한 미래 방향에 대한 지침을 제공함으로 콜럼비아 대학에 기반을 둔 OHDSI coordination center의 자문그룹 역할을 수행	이 목표에 동감하는 사람 커뮤니티 내의 리더들

2.1.5 OHDSI 지역 지부

OHDSI 지역 지부 Regional Chapter는 각각의 지리적 위치의 특정 문제를 해결하기 위해 로컬 네트워킹 이벤트 및 회의를 개최하고자 하는 지리적 영역에 위치한 OHDSI 공동 작업자 그룹을 대표한다. 현재 OHDSI 지역 지부는 한국², 유럽³, 중국⁴ 등이 있다. 한국 지부 포럼에서는 한국말을 이용하여 질문과 생각을 올릴 수 있다. 만약 본인의 지역에 OHDSI 지역 지부를 설립하고 싶다면 OHDSI website에 설명된 OHDSI 지역 지부 프로세스를 따라 진행할 수 있다.

2.1.6 OHDSI 연구 네트워크

다수의 OHDSI 공동연구자들은 자신의 데이터를 OMOP CDM으로 변환하는 것에 관심이 있다. OHDSI 연구 네트워크는 OMOP CDM의 호환성을 준수하기 위해 추출 변환 적재 Extract Transform Load(ETL) 프로세스를 거친 관찰형 데이터베이스의 다양하고 범세계적 커뮤니티를 대표한다. 만약 OHDSI 커뮤니티에서 당신의 역할에 데이터 변환이 포함되어 있다면 OMOP CDM 및 용어 vocabulary에 대한 튜토리얼, 변환을 지원하는 무료 소프트웨어, 특정 도메인 또는 데이터 타입의 유형을 타깃으로 하는 워크그룹이 있다. OHDSI 공동연구자들은 OHDSI 포럼을 활용하여 CDM 변환 중에 발생하는 문제를 논의하고 해결하는 것을 권장한다.

2.2 적합한 위치

이제 지금쯤이면 과연 나는 OHDSI 커뮤니티의 어디에 어울릴까? 라는 고민을 할 것이다.

나는 연구를 시작하려는 임상 연구자입니다. 만약 당신이 OHDSI 연구 네트워크를 사용하여 특정 임상 질문에 답을 하는 연구를 하고 싶다거나, 논문을 제출하려는 임상 연구자라면, 맞게 찾아온 것이다. 우선 OHDSI 포럼의 OHDSI Researchers Topic에 당신의 아이디어를 게시할 수 있다. 이것은 당신과 비슷한 관심사를 가진 연구자와 연결하는 데 도움이 된다. OHDSI는 논문출판을 적극 권장하며 당신의 연구 주제를 데이터 분석 및 논문으로 신속하게 전환할 수 있는 많은 자원을 보유하고 있다. 이에 관한 자세한 내용은 11장, 12장, 13장에서 확인할 수 있다.

OHDSI 커뮤니티가 생산하는 정보를 읽고 소비하고 싶습니다. 당신이 환자, 임상의사 혹은 의료 분야 세부 전문가이든, OHDSI는 임상에 대해 더 잘 이해할 수 있도록 고품질의 근거를 제공하고자 한다. 어쩌면 당신은 오랫동안 코딩을 해 온 프로그래밍 전문가일 수도 있고, 프로그래밍을 한 번도 해본 적이 없을 수도 있다. 어찌되었든 당신은 이 커뮤니티의 일환이 될 수 있다. 우리는 당신을 근거 소비자 *evidence consumer* – OHDSI 연구를 행동으로 옮기는 개인- 라고 부른다. 당신은 OHDSI

²<http://forums.ohdsi.org/c/For-collaborators-wishing-to-communicate-in-Korean>

³<https://www.ohdsi-europe.org/>

⁴<https://ohdsichina.org/>

가 어떤 근거를 만들었거나, 만들고 있는지를 파악하고 싶거나, 혹은 당신과 관련된 질문들 하고 싶을 수도 있다. 이런 당신을 토론에 초대한다. OHDSI 포럼에 질문을 올리기 바란다. 커뮤니티 원격회의에 참석하여 최신 연구를 들어보고, OHDSI 심포지엄 및 대면 미팅에 참석하여 커뮤니티에 직접 참여하도록 하라. OHDSI는 당신의 질문을 중요하게 여긴다. 당신이 어떤 근거를 찾고 있는지 우리가 알 수 있도록 목소리를 높여주기를 바란다!

나는 보건의료 분야에서 의사결정을 할 수 있는 위치에 있습니다. 나는 데이터 소유자거나 그 소유자를 대표할 수 있습니다. 나는 내 기관에서 OMOP CDM 및 OHDSI 분석 도구의 유용성을 평가하고 있습니다. 조직의 관리자/리더로서 OHDSI에 관해 들어봤을 수 있으며 OMOP CDM이 어떻게 당신의 경우에 이용될 수 있는지 궁금할 수 있다. 그렇다면, OHDSI Past Events의 자료를 통해 OHDSI의 성과를 보는 것으로 시작할 수 있다. 커뮤니티 원격회의에 참여하여 단순히 듣기만 해도 된다. 7장(데이터 분석 이용 사례)은 OMOP CDM 및 OHDSI 분석 도구를 활용하여 시행할 수 있는 연구의 종류를 이해하는 데 도움이 될 것이다. 당신을 위해 OHDSI 커뮤니티가 당신의 여정에 함께 할 수 있다. 관심 있는 특정 영역이 있다면 이에 대한 예제를 물어보는 것에 두려워하지 마라. 전 세계 200개 이상의 조직이 OHDSI 내에서 협력하고 있으며 이 커뮤니티의 가치를 보여주는 데 도움이 되는 성공 사례가 많다.

나는 내 기관의 데이터를 ETL 및 변환하여 OMOP CDM으로 변환하고자 하는 데이터베이스 관리자입니다. 당신의 데이터를 “OMOP-CDM”으로 변환하고자 하는 것은 고귀하고 가치 있는 일이다. 만약 ETL 프로세스를 막 시작하는 경우에는 OHDSI 커뮤니티 ETL 튜토리얼 슬라이드를 참조하거나 다가오는 OHDSI 심포지엄에 등록하여 튜토리얼을 듣기 바란다. THEMIS 워크그룹 원격회의에 참여하거나 OHDSI 포럼에 질문을 올릴 수도 있다. OMOP CDM의 성공적인 구현을 돋는 것에 관심이 많은 OHDSI 커뮤니티에서 풍부한 지식을 찾을 수 있을 것이다. 부끄러워하지 말고 모르는 것이 있으면 언제든 질문하라!

나는 OHDSI 툴 스택에 기여를 하고 싶은 생물통계학자 혹은 방법론 개발자입니다. 무엇보다도 OHDSI method 라이브러리에 당신의 전문 지식을 이용하여 기여하고자 하는 마음과, 방법론을 더욱 잘 개발하기 위한 당신의 열정에 감사를 표한다. 우선 인구 수준 추정이나 환자 수준 예측 워크그룹 원격회의에 참여하여 커뮤니티의 현 우선순위에 대하여 자세히 들어 보기를 추천한다. OHDSI 도구를 사용하면서 각 GitHub Repo에 문제를 제기할 수도 있다. (예를 들면, SQL 렌더 패키지의 문제일 경우 OHDSI/SqlRender에 대한 GitHub Repo에 문제를 제기하면 된다) 당신의 기여를 환영한다!

나는 OHDSI 툴 스택을 보완하는 도구 만드는 것에 관심이 있는 소프트웨어 개발자입니다. 커뮤니티에 온 것을 환영한다! OHDSI 임무의 일환으로 우리의 툴은 오픈소스로서 Apache licenses에 따라 관리된다. 우리는 OHDSI 툴 스택을 보완하는 솔루션 개발을 환영한다. 언제든 워크그룹에 참여하여

아이디어를 제안해 주길 바란다. 다만, OHDSI는 오픈 사이언스 기반의 개방형 협업에 많은 투자를 하는 점을 유의하자. 독점적인 알고리즘과 소프트웨어 솔루션도 환영하지만, 우리 소프트웨어 개발 작업에서 그런 작업은 주요 관심사가 아니다.

나는 OHDSI 커뮤니티에 조언하고 싶은 컨설턴트입니다. 환영한다! 당신의 전문 지식은 매우 귀중하다. 필요에 따라 OHDSI 포럼에 적절히 본인의 서비스를 홍보해도 된다. OHDSI 튜토리얼에 참여하길 바라며 당신의 전문성으로 매년 열리는 심포지엄 및 OHDSI 대면 미팅 행사 개최 등을 도와줄 수 있으면 좋겠다.

나는 OHDSI에 대하여 더 배우고 싶은 학생입니다. 올바르게 찾아왔다! OHDSI 커뮤니티 원격 회의에 참여하여 본인을 소개하는 것을 고려하자. OHDSI 튜토리얼을 참고하고 OHDSI 심포지엄의 대면 미팅에 참여하여 OHDSI 커뮤니티가 제공하는 방법과 도구에 관하여 자세히 알아보라. 만약 특정 연구에 관심이 있다면 OHDSI 포럼의 연구자 토픽에 글을 올려보기 바란다. 다양한 조직에서 OHDSI가 후원하는 연구 기회 (예를 들면 박사후과정, 연구 펠로우십)를 제공한다. OHDSI 포럼은 이러한 기회 등에 대한 최신 정보를 제공할 것이다.

2.3 요약



- OHDSI 커뮤니티를 시작하기란 매우 쉽다! **OHDSI Forum**에 글을 올리고 원격 회의에 참여하라.
- OHDSI 포럼에 본인의 연구나 CDM, ETL에 관한 질문을 올리기 바란다.

Chapter 3

오픈 사이언스

Chapter lead: Kees van Bochove

OHDSI 창립 당시로부터 지금까지 OHDSI 커뮤니티의 목표는 오픈 소스 소프트웨어의 사용이나, 모든 컨퍼런스의 절차 및 자료의 공공적 가용성 그리고 생산된 의학적 근거의 투명한 공개와 같이 오픈 사이언스의 가치를 구축함으로써 국제적 협력체계를 구축하는 것이었다. 그러나, 오픈 소스 소프트웨어란 정확히 무엇을 말하는가? 통상적으로 의료데이터는 매우 민감한 개인 정보로 인식되며 선한 의도에 대해서도 개방되기 어려운데, OHDSI는 어떻게 의료 데이터 사이언스 분야에 개방형 데이터 전략이나 오픈 사이언스 전략을 견지할 수 있었을까? 분석의 재현성 reproducibility 이 왜 그렇게 중요할까? 그리고 OHDSI 커뮤니티는 어떻게 이 목표를 달성하고자 하는가? 이는 우리가 이 장에서 다룰 중요한 문제 중 몇 가지이다.

3.1 오픈 사이언스

‘오픈 사이언스 Open Science’ 용어 자체는 90년대부터 사용되어 왔지만, 이는 OHDSI가 생겨난 2010년대부터 실제 견인력을 얻기 시작했다. 위키피디아 (Wikipedia, 2019a) 에는 “만들어진 과학 연구의 결과 및 부산물(논문, 데이터, 실제 샘플 및 소프트웨어 포함)에 대해, 아마추어 또는 전문가 모두가 접근할 수 있도록 하는 운동”이라 정의하고 있으며 더 나아가 일반적으로 공동 네트워크를 통해 개발된다고 써있다. OHDSI 커뮤니티는 자체적으로 ‘오픈 사이언스’ 집단 혹은 네트워크라고 정의하지 않았으나 이 용어는 OHDSI의 개념과 원칙을 사용하는 데 자주 사용된다. 예를 들어 2015년 Jon Duke는 OHDSI를 “의료 근거 생성에 관한 오픈 사이언스 접근법”¹ 이라 말하였으며 2019

¹https://www.ohdsi.org/wp-content/uploads/2014/07/ARM-OHDSI_Duke.pdf

년에는 EHDEN 컨소시엄의 입문용 웹 세미나에서는 OHDSI 네트워크 접근 방식을 “21세기 실세계 오픈 사이언스”² 라고 극찬하였다. 이번 장에서 보게 되겠으나 OHDSI 커뮤니티의 곳곳에서 오픈 사이언스의 정신을 발견할 수 있다. 어떤 이들은 OHDSI 커뮤니티는 의료 근거 생성의 투명성과 신뢰성을 개선하기 위한 공동의 욕구에 의해 시작된 풀뿌리 오픈 사이언스 집단이라고 주장하기도 한다.

오픈 사이언스 또는 “사이언스 2.0” (Wikipedia, 2019b) 접근법은 현재 의학 연구계의 알려져 있는 여러 가지 문제를 해결하기 위해 고안되었다. 정보 기술은 데이터 생성 및 분석 방법의 폭발적인 성장으로 이어졌으며 개별 연구원의 경우 전문 분야에 발표된 모든 문헌을 따라잡기가 매우 어렵다. 특히 평소 진료일을 하면서도 최신 의학에 뒤쳐지지 않아야 하는 임상 의사의 경우는 훨씬 더 그렇다. 게다가, 많은 수의 실험들이 열악한 통계 설계, 출판 비뚤림 publication bias, p-hacking 및 유사한 통계적 문제로 영향을 받고 있으며 재현하기 어렵다는 우려가 커지고 있다. 동료 심사 peer review를 통한 논문 출판과 같은 전통적인 방식은 종종 이러한 문제를 인식하거나 해결하는데 실패하곤 한다. 2018년 Nature의 특집호에서 “재현할 수 없는 연구의 어려움 Challenges in irreproducible research”³은 몇 가지 예를 보여주었다. 한 저자 그룹은 자신들이 속한 분야의 논문들을 대상으로 체계적 문헌 고찰을 적용하려 하였으나, 여러 가지 이유로 식별된 논문 오류를 수정하기 어렵다는 것을 발견하였다. 특히 결함이 있는 디자인으로 시작한 실험은 특히 수정하기 어렵다. Ronald Fisher에 의하면 “실험을 마친 후 통계학자와 상담하는 것은 마치 그에게 사체 부검을 해달라는 것과 같다. 아마 무엇 때문에 그 실험이 사망했는지는 말해줄 수 있겠지요.” (Wikiquote, 2019) 연구자들은 잘못된 통계적 유의성을 야기하는 부적절한 무작위 설계, 메타분석 상의 잘못된 계산, 부적절한 과거력 비교와 같은 통계적 문제들을 흔히 만난다. (Allison et al., 2016) 같은 호의 다른 논문에서는 물리학의 경험을 예로 들어서, 기본 데이터 뿐 아니라 데이터 처리 및 분석 코드에 대한 상세한 문서를 다른 연구자들이 사용할 수 있도록 제공하여 연구 재현성을 높여야 한다고 주장한다. (Chen et al., 2018)

OHDSI 커뮤니티는 이러한 어려움을 스스로 해결하기 위해 노력하고, 대규모로 의학 근거를 만드는 것에 역점을 두고 있다. Schuemie et al. (2018b)에서 언급하였듯이 현 패러다임은 “특정 연구 설계를 이용해서 한 번에 신뢰성이 떨어지는 하나의 결과치만을 추정하여 하나의 논문에 출판” 하고 있으나 OHDSI 커뮤니티는 일관되고 표준화된 방법에 기반한 대규모 분석을 통한 관찰 연구를 지지하며, 평가, 교정 및 편견 없는 결과발표를 통해 더욱 안정적이고 완전한 근거 기반을 만들 수 있다. 이는 OMOP CDM로 변환된 의료 데이터 네트워크와 모든 사람이 사용할 수 있고 증명할 수 있는 오픈 소스 분석 코드, 그리고, howoftten.org에서 발표한 것과 같이 질환 발생과 관련된 대규모 과거력 데이터들을 조합하여 이를 수 있다. 다음에는 구체적인 예시 함께, 공개 표준 open standards, 오픈 소스 open source, 공개 데이터 open data, 열린 담론 open discourse의 4가지

²<https://www.ehden.eu/webinars/>

³<https://www.nature.com/collections/prbfkwmwvz>

원칙을 이용하여 OHDSI의 오픈 사이언스 접근 방식을 더욱 자세히 설명할 것이다. 이번 장은 오픈 사이언스의 관점에서 OHDSI 의 전망과 FAIR 원칙에 대해 간략히 참고하며 마무리할 예정이다.

3.2 실천의 오픈 사이언스: 스터디톤

커뮤니티 내부의 최근 동향은 ‘스터디톤 Study-a-thons’의 출현이다. 스터디톤이란 OMOP CDM과 OHDSI 툴을 사용하여, 중요하고 임상적으로 관련이 있는 연구 질문에 대답하기 위해 여러 학문 분야에 걸친 과학자들이 모여서 짧고 집중된 대면회의 face-to-face meeting를 하는 모임이다. 이에 관한 좋은 예는 EHDEN 웨비나에서 설명한 2018 Oxford 스터디톤인데, 행사 과정을 단계별로 나눠 제공하고 공개적으로 사용할 수 있는 결과를 강조하고 있다. 스터디톤이 이어지는 기간 동안, 참가자는 의학적으로 관련이 있는 연구 주제를 제안하고 하나 이상의 연구 주제를 스터디톤 기간 동안 진행할 수 있도록 선정한다. 스터디톤 동안 참가자들은 OMOP CDM 형태의 환자 레벨 데이터에 접근할 수 있고, 쿼리를 이용해 데이터를 추출할 수 있다. 실제 스터디톤 시간의 대부분은 통계적 접근법 (2장 참조), 데이터 소스의 적합성, 상호작용으로 만들어진 결과와 이러한 결과에 의해 필연적으로 제기되는 후속 질문에 대해 논의하는 데 사용된다. Oxford 스터디톤의 경우 다양한 무리를 관절 대체 수술 후 발생하는 부작용에 대한 연구를 중심으로 질문이 이루어졌으며 OHDSI 포럼 및 도구를 이용하여 study-a-thon이 진행하는 동안 대화식으로 결과를 발표하였다. (8장 참조) ATLAS와 같은 OHDSI 도구는 코호트 정의의 신속한 생성, 교환, 토론 및 평가를 용이하게 하여, 문제 정의와 방법론 선택에 대한 합의에 도달하는 초기 프로세스를 아주 빠르게 가속화 해준다. 관련 데이터 소스와 OHDSI 오픈소스 환자 수준 예측 patient level prediction 패키지 (13장 참조)의 사용성 덕분에, 하루 만에 수술 후 90일 내 사망률에 대한 예측 모델을 만들고, 다음 날 여러 대규모 데이터 소스에서 이 모델에 대한 외부 검증이 가능했다. 또한 study-a-thon은 전통적인 학술 논문 (무리를 관절 전체 성형술 부작용에 대한 patient-level 예측 모델 개발 및 검증, Ross Williams, Daniel Prieto-Alhambra et al., 논문 작성 중) 을 만들어 냈는데, Peer review를 통해서 진행되었다면 몇 달 걸렸을 작업이다. 다수의 의료 데이터베이스를 이용하여 수억 명의 환자 기록을 이용한 연구가 스터디톤을 통하여 낙서 같은 초안으로 시작하여 연구 설계, 근거 생산 및 논문 작성까지 1주일 안에 가능했다는 사실은 OHDSI 가 근거를 만드는데 필요한 처리 기간을 몇 달에서 단 몇 일로 감소시켜 의학 연구의 근본적인 혁신을 이루어냈다는 것을 증명한다. - (역자 주: Burn et al. (2019) 가 Oxford 스터디톤 당시 연구하였던 무리를 관절 치환 수술 관련 연구는 2019년 11월 Lancet Rheumatology에 출판되었다)

3.3 공개 표준 Open Standards

OHDSI 커뮤니티에서 유지 관리하는 매우 중요한 커뮤니티 리소스는 OMOP 공통 데이터 모델 (4장 참조)와 관련 표준용어 (5장 참조)이다. 모델 자체는 관찰 의료 데이터를 수집하기 위해 범위가 정해졌으며 원래는 약물, 시술, 의료기기 등에 노출되는 것과 진단 및 검사와 같은 결과 간의 연관성을 분석하기 위한 것이었으나 이제는 다양한 분석 사용 사례로 확장되었다. (7장 참조) 그러나 다양한 코딩 시스템, 의료 패러다임 및 다양한 유형의 의료 소스를 가진 전 세계의 의료 데이터를 통일시키려면 소스 코드와 가장 가까운 표준화된 용어 간에 엄청난 양의 '매핑'이 필요하다. OMOP 표준용어는 7장에서 추가로 설명한다. OMOP 표준용어는 전 세계적으로 사용되는 수백 개의 의료 용어 시스템과 매핑을 포함하고 있으며 OHDSI Athena 를 통해 열람할 수 있다. 이러한 용어와 매핑을 자유롭게 사용할 수 있는 리소스를 커뮤니티에 제공함으로써, OMOP과 OHDSI 커뮤니티는 의료 데이터 분석에 상당한 기여를 하고 있으며, 이러한 목적을 위한 가장 포괄적인 모델로 전 세계적으로 약 12억 명의 의료 기록을 대표하고 있다.⁴ (Garza et al., 2016) - (역자 주: 최근 조사자료에 의하면 약 21억 명, 미국을 제외 시 약 3억 8천만 명 자료. 기관 간 자료가 연계되지 않으므로 한 환자의 자료가 여러 번 중복됨으로 인해서 포함된 실제 고유 환자 수보다 더 많게 평가됨)

3.4 오픈 소스 Open Source

OHDSI 커뮤니티가 제공하는 또 다른 핵심 리소스는 오픈 소스 프로그램이다. 여기에는 데이터를 OMOP에 매핑하기 위한 보조 소프트웨어 (6장 참조), 일반적으로 많이 사용되는, 강력한 통계 패키지를 포함하는 OHDSI 메소드 라이브러리, 관찰 연구를 위한 오픈 소스 코드, OHDSI 에코시스템을 뒷받침하는 ATLAS, Athena 및 기타 인프라 관련 소프트웨어 (8장 참조)로 나눌 수 있다. 오픈 사이언스 관점에서, 가장 중요한 리소스 중 하나는 OHDSI 연구 네트워크와 같은 실제 연구 실행을 위한 코드이다(20장 참조). 이 프로그램들은 GitHub을 통해 점검, 검토 및 기여가 가능한 오픈 소스 OHDSI 스택을 최대한 활용한다. 예를 들어, 네트워크 연구를 위한 코딩은 분석법 이용 사례에 대한 통계적 방법을 일관되게 재사용 할 수 있는 라이브러리를 기반으로 하는 경우가 많다. OHDSI의 오픈 소스 소프트웨어 사용과 협력이 의학적 근거 생성의 품질과 신뢰성을 어떻게 뒷받침하는지에 대한 자세한 내용은 17장을 참고하기 바란다.

3.5 공개 데이터 Open Data

개인 정보 보호에 민감한 의료 데이터의 특성 때문에 포괄적인 환자 수준의 데이터는 일반적으로 완전히 개방할 수 없다. 그러나 앞서 언급된 <http://howoften.org> 및 <http://data.ohdsi.org> 에 게시된 공개 결과 세트들과 같이 중요한 집계 데이터나 분석 결과를 게시하기 위해 OMOP CDM

⁴<https://www.ema.europa.eu/en/events/common-data-model-europe-why-which-how>

데이터를 활용할 수 있다. 또한, OHDSI 커뮤니티는 테스트와 개발을 위해 SynPUF와 같은 시뮬레이션 데이터 세트를 제공하며, OMOP CDM으로 변환된 데이터 소스들의 네트워크 안에서 연구를 수행하는 데 OHDSI 연구 네트워크 (20장 참조) 가 이용될 수 있다. 소스 데이터와 OMOP CDM 간의 매핑을 투명하게 하기 위해서는, CDM 변환시 OHDSI가 제공하는 ETL 또는 용어 매핑 툴을 사용하고 용어 매핑을 오픈 소스로 공개하는 것이 바람직하다.

3.6 열린 담론 *Open Disclosure*

공개 표준, 오픈 소스, 공개 데이터는 훌륭한 자산이지만, 그 자체로는 진료 행위에 큰 영향은 없을 것이다. OHDSI의 오픈 사이언스 활동과 영향의 핵심은 의학적 근거 생성을 구현하여 과학으로부터 진료 현장으로 이행하도록 해주는 것이다. OHDSI 커뮤니티는 미국, 유럽, 아시아에서 여러 연례 OHDSI 심포지엄을 개최하고 있으며, 특히 한국과 중국의 혁신적인 커뮤니티들을 보유하고 있다. 이들 심포지엄에서는 통계적 방법론, 데이터 및 소프트웨어 사용법, 표준 용어, 그리고 OHDSI 오픈 소스 커뮤니티의 다른 모든 측면에서의 발전에 대해 논의한다. OHDSI 포럼⁵과 위키⁶는 전 세계 수천 명의 연구자가 관찰 연구를 수행하도록 돋는다. 커뮤니티 원격회의⁷와 GitHub⁸의 코드, 이슈, pull requests는 코드, CDM과 같은 오픈 커뮤니티의 자산을 지속해서 발전시키고 전 세계적으로 수억 명의 환자 기록을 이용하여 개방적이고 투명한 방법으로 범세계적 관찰 연구가 OHDSI 네트워크 연구로써 수행되고 있다. 개방성과 열린 담론은 커뮤니티 전반에 걸쳐 권장되며 바로 이 책은 OHDSI 위키, 커뮤니티 원격회의, GitHub repository에 의해 촉진되는 오픈 프로세스를 통해 쓰인다.⁹ 그러나 OHDSI 공동연구자들이 없다면 프로세스와 도구는 빈 껍데기가 될 것이라는 점을 강조할 필요가 있다. 실제로 OHDSI 커뮤니티의 진정한 가치는 1장에서 논의한 바와 같이 협력과 오픈 사이언스를 통해 건강을 증진한다는 비전을 공유하는 회원들과 함께한다고 말할 수 있다.

3.7 OHDSI와 FAIR의 가이드 원칙

3.7.1 FAIR

이 장의 마지막 단락은 Wilkinson et al. (2016) 이 발표한 FAIR 데이터 가이드 원칙 (Findability, Accessibility, Interoperability, Reusability)에 입각하여 OHDSI 커뮤니티와 도구의 현재 상태를 살펴본다.

⁵<https://forums.ohdsi.org>

⁶<https://www.ohdsi.org/web/wiki>

⁷<https://www.ohdsi.org/web/wiki/doku.php?id=projects:overview>

⁸<https://github.com/ohdsi>

⁹<https://github.com/OHDSI/TheBookOfOhdsi>

3.7.2 검색성 Findability

OMOP CDM으로 변환되어 분석에 사용되는 모든 의료 데이터베이스는 과학적 관점에서 향후 참조와 재현성을 위해 관리되어야 한다. 개별적인 OMOP 데이터베이스를 위한 영구 식별자를 사용하는 것이 아직 널리 확산되지는 않았는데, 부분적으로는 보통 데이터베이스가 방화벽 안에 숨겨져 있거나 내부 네트워크에 있어 원격 접속이 가능하지는 않기 때문이다. 그러나 인용 등을 위한 데이터베이스에 대한 요약 설명문을 공유하는 것은 가능하다. EMIF 카탈로그¹⁰의 예를 보면, 이 카탈로그는 데이터 수집 목적, 소스 데이터, 용어, 액세스 제어 메커니즘, 라이센스, 등의 등의 측면에서 데이터베이스에 대한 포괄적인 기록을 제공한다. (Oliveira et al., 2019) 이 접근 방식은 IMI EHDEN 프로젝트에서 더욱 심층 개발되었다.

3.7.3 접근성 Accessibility

오픈 프로토콜을 통해 OMOP CDM으로 변환된 데이터는 일반적으로 SQL 인터페이스를 통해 접근할 수 있는데, 이 인터페이스는 OMOP CDM과 결합하여 표준화되고 문서화된 데이터 접근 방식을 제공한다. 그러나, 위에서 논의한 바와 같이, CDM 데이터는 보안상의 이유로 원격을 통해 외부에서 접근할 수 없는 경우가 많다. IMI EHDEN과 같은 프로젝트의 주요 연구 주제와 운영 목표는 연구원들이 접근할 수 있는, 안전한 전 세계 의료 데이터 네트워크를 만드는 것이다. 환자 수준의 데이터는 공개되지 못하더라도, LEGEND와 <http://howoften.org> 과 같은 OHDSI 이니셔티브를 통해 보이듯 다수의 OMOP 데이터베이스의 분석 결과를 공개적으로 게시 가능하다.

3.7.4 상호운용성 Interoperability

상호운용성 Interoperability은 OMOP CDM과 OHDSI 도구들의 가장 강력한 장점이다. 근거 생성을 위해 활용할 수 있는 전 세계적 의료 데이터 네트워크를 구축하기 위해서는 데이터베이스 간의 상호운용성을 달성하는 것이 핵심이며, 이는 OMOP CDM 모델과 표준용어집 Standardized Vocabularies을 통해 달성 가능하다. 단순히 자료 구조 및 용어의 표준화를 넘어서, 코호트 정의와 통계적 접근법을 공유함으로써 OHDSI 커뮤니티는 의료 데이터 분석 방법론의 상호운용이 가능한 플랫폼을 제공한다. 병원과 같은 의료 시스템은 종종 CDM 데이터에 대한 기록의 소스이기 때문에, OHDSI 접근방식의 상호운용성은 HL7 FHIR, HL7 CIMI 및 OpenEHR과 같은 운영적인 의료 상호운용성 표준과의 상호운용성을 가짐으로써 더욱 향상될 수 있다. CDISC나 생물 의학 온톨로지 같은 임상적 상호운용성 표준과의 정렬도 마찬가지다. 특히 종양학과 같은 분야에서 이것은 중요한 주제로서, OHDSI 커뮤니티의 Oncology 워크그룹과 Clinical Trials 워크그룹은 이러한 문제가 적극적으로 논의되는 포럼의 좋은 예를 보여준다. 다른 데이터의 참조 및 특히 온톨로지 용어 측면에서, ATLAS와 OHDSI Athena는 사용되고 있는 서로 다른 의료 용어 시스템의 맥락에서 OMOP

¹⁰<https://emif-catalogue.eu>

표준용어집을 탐색할 수 있어서 중요한 도구이다.

3.7.5 재사용 가능성 Reusability

재사용 가능성 Reusability에 관한 FAIR 원칙은 데이터 라이선스, 출처 (데이터가 어떻게 존재했는지 명확화) 및 관련 커뮤니티 표준과의 연결과 같은 중요한 문제에 초점을 맞추고 있다. 데이터 라이센스는 복잡한 주제로서, 특히 서로 다른 사법권 간에는 더욱 복잡하며, 광범위하게 다루기에는 이 책의 범위를 벗어난다. 그러나 당신의 데이터 (예를 들면, 분석 결과)를 다른 사용자가 자유롭게 사용할 수 있도록 하려는 경우 데이터 라이선스를 정의함으로써 통해 이러한 권한을 명시적으로 제공하는 것이 좋다. 그러나 이는 아직 인터넷에서 찾을 수 있는 대부분의 데이터에 대한 일반적인 관행이 아니며 불행히도 OHDSI 커뮤니티 역시 예외가 아니다. CDM 데이터베이스의 데이터 출처와 관련하여, CDM 버전, 표준용어집 배포, 사용자 정의 코드 목록 등과 같이 자동화된 방식으로 메타 데이터를 사용할 수 있도록 하기 위해 아직은 개선할 수 있는 부분들이 있다. OHDSI ETL 툴은 현재 이 정보를 자동으로 생성하지 않지만, Data Quality 워크 그룹과 Metadata 워크그룹 등은 이에 대해 활발하게 작업 중이다. 또 다른 중요한 측면은 기본 데이터베이스 자체의 출처 검증이다. 병원이나 일반 의용 정보시스템이 교체되었는지 또는 변경되었는지, 그리고 알려진 데이터 누락이나 다른 데이터 문제가 과거에 언제 발생했는지를 아는 것이 중요하다. OMOP CDM에서 이러한 메타데이터를 체계적으로 연결하는 방법을 탐색하는 것이 Metadata 그룹의 역할이다.



- OHDSI 커뮤니티는 의료 근거 생성의 상호 운용성과 재현성을 적극적으로 추구하는 오픈 사이언스 커뮤니티로 볼 수 있다.
- OHDSI는 기존의 단일 추정 의학 연구 패러다임에서, 실세계 의료 자료을 이용하여 기초 발생률, 중재 및 치료 효과의 통계적 추정치 등을 대규모 체계적 근거 생성 시스템을 기반으로 생성하는 패러다임으로의 전환을 지지하고 있다.

Part II

공통 데이터 모델

Chapter 4

공통 데이터 모델

Chapter leads: Clair Blacketer

관찰 데이터는 환자가 진료를 받는 동안 어떤 일이 일어나는지를 보여준다. 전 세계적으로 점점 더 많은 수의 환자에 대한 데이터가 빅 데이터라고 불리는 형태로 수집 및 저장되고 있다. 이러한 데이터의 일차적 목적은 다음과 같은 세 가지로 설명할 수 있다. (i) 직접적으로 (많은 경우에 설문 조사 및 레지스트리 정보를 활용하여) 연구를 용이하게 하기 위해, (ii) 의료 행위 수행을 지원하기 위해 (이를 보통 전자 의무 기록 Electronic Health Records(EHR)이라고 함), 또는 (iii) 의료비 지불 관리를 위함(청구 데이터). 세 가지 목적 모두 임상 연구에 보편적으로 사용되나, 두, 세 번째 항목은 이차적인 목적으로 사용된다. 위 세 가지 모두 일반적으로 고유한 내용의 형식 및 인코딩으로 이루어져 있다.

관찰형 의료 데이터 Observational healthcare data에 공통 데이터 모델이 필요한 이유는 무엇일까?

일차적인 목적에 의해 모든 임상적인 사건을 동일하게 포착하는 관찰형 데이터베이스 Observational database는 없다. 따라서, 여러 다른 데이터 출처에서 연구 결과를 도출하고 데이터를 포착하는 과정에서 발생하는 잠재적 비뚤림 bias의 영향을 이해하기 위해 결과들을 비교 및 대조해야 한다. 또한 통계적 검증력을 갖춘 결론을 도출하려면 많은 수의 환자가 필요하다. 이는 여러 데이터 출처를 동시에 평가하고 분석해야 할 필요성을 설명한다. 그러기 위해서는 데이터를 공통 데이터 표준 common data standard으로 변환할 필요가 있다. 게다가 환자 데이터는 민감 정보이기에 높은 수준의 보안이 필수적이다. 기존에 그래왔듯이 분석을 목적으로 하는 데이터 추출은 엄격한 데이터 사용 계약 및 복잡한 접근 제어 방식이 필요하다. 공통 데이터 표준은 연구자의 분석환경으로 데이터를 추출하지 않고, 연구자가 만든 분석 프로세스를 데이터가 있는 곳으로 전달하는 방식으로 작동할 수 있어,

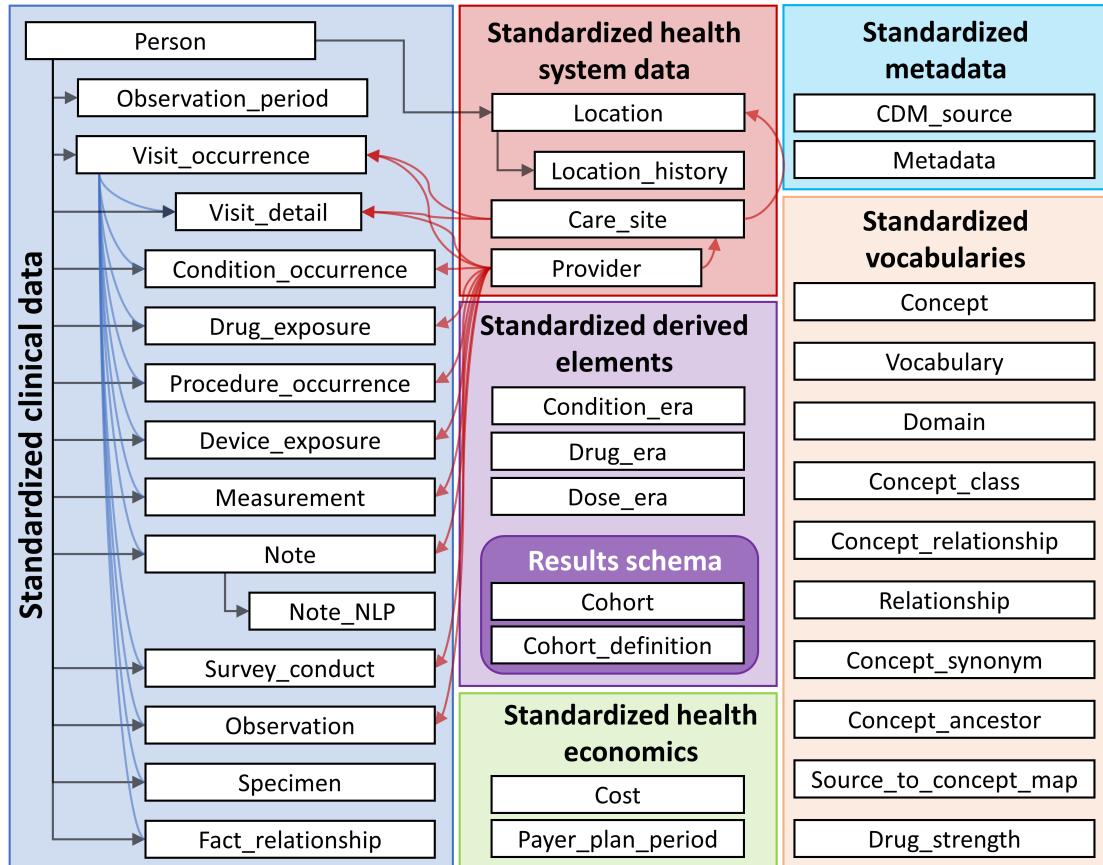


Figure 4.1: CDM 6.0 버전의 모든 테이블에 대한 개요. 테이블 간의 모든 관계가 표시된 것은 아님.

데이터 추출 단계를 생략하고 기본 환경의 데이터에 대해 표준화된 분석을 실행할 수 있도록 하여 개인 정보 유출 위험 및 추가 보안의 필요성을 줄여 줄 수 있다.

이러한 표준은 공통 데이터 모델 Common Data Model(CDM)에 의해 제공된다. CDM은 표준화된 데이터를 기반으로 (5장 참조) 연구 방법들을 효과적으로 비교할 수 있고, 재현 가능한 결과를 얻을 수 있는 체계적인 환경을 제공한다. 이 장에서는 데이터 모델 아키텍처를 비롯하여 CDM 디자인, 규칙 및 테이블 선택에 대한 논의를 제공하고자 한다.

CDM 내의 모든 테이블에 대한 개요는 그림 4.1에서 살펴볼 수 있다.

4.1 설계 원리

OMOP CDM은 다음과 같은 목적의 전형적인 관찰 연구에 최적화되어 있다.

- 특정한 의료 행위의 개입(약물 노출, 시술 procedure, 의료 정책 변경 등)이 있거나 의료 관련

결과(질환, 시술, 기타 약물 노출에 대한)를 포함하는 환자 집단 확인

- 인구 통계학적 정보, 질병의 자연사, 의료 서비스 전달, 활용 및 비용, 병적 상태, 치료 및 치료 과정 등과 같은 다양한 매개 변수에 대한 환자 집단의 특성 확인
- 개별 환자에서 결과의 발생 예측 - 13장 참고
- 앞서 설명한 의료 행위의 개입이 인구 수준에 미치는 영향 추정 - 12장 참고

이러한 목표를 달성하기 위해서 OMOP CDM의 개발은 다음과 같은 설계 요소를 따른다:

- **목적에 대한 적합성:** CDM은 의료 서비스 제공이나 보험청구 업무를 해결하기 위한 목적보다는 데이터 분석 연구에 최적화된 방식으로 구성된 데이터를 제공하는 것을 목표로 한다.
- **데이터 보호:** 이름, 일 단위의 생년월일 등 환자의 신원 및 안전을 위협할 수 있는 모든 데이터는 제한되어 있다. 영어에 대한 연구를 위한 정확한 생년 월일과 같은 보다 자세한 정보가 명시적으로 필요한 경우에는 예외가 가능하다.
- **도메인 설계:** 도메인은 사람 중심 관계형 데이터 모델 person-centric relational data model로 모델링 되며 기록마다 개인의 신원과 날짜 정보가 최소한으로 수집된다. 여기서 관계형 데이터 모델은 데이터가 기본 키 primary key와 외래 키 foreign key로 연결된 테이블로 표현되는 모델이다.
- **도메인의 이론적 근거:** 개체-관계 모델 entity-relationship model에서 도메인은 분석 이용 사례가 있는지 (예를 들면, 질환 conditions) 그리고 달리 적용 가능한 방안이 없는 특정한 속성 attributes이 있는지에 따라서 별도로 정의된다. 다른 모든 데이터는 개체-속성-값 구조 entity-attribute-value structure를 가진 Observation 테이블에 관찰 데이터로 저장할 수 있다.
- **표준화된 용어:** 기록을 표준화하기 위해, CDM은 필수적이고 적절한 표준 건강 관리 개념 concept 을 포함하는 표준 용어에 의존한다. (역자 주: OMOP 표준 용어 및 개념은 OMOP-CDM의 가장 큰 특장점이다. 표준 용어집은 일종의 사전이며, concept 은 OMOP CDM 의 하나하나의 단어/어휘로 생각할 수 있다. ‘Concept’을 우리 말로 번역하면 ‘개념’ 이지만, 문맥에 따라 이해하기 어려운 경우 concept 영문을 그대로 사용한다.)
- **기존 용어 재사용:** 이러한 concept은 국립 의학 도서관 NLM, 재향 군인 담당 부서 Department of Veteran’s Affairs, 질병 예방센터 CDC 등과 같은 국가 및 산업 표준화 또는 용어 정의 주도 기관이나 협회에서 만든 용어를 재사용하기도 한다.
- **원천 코드 유지 관리:** 모든 의료 코드가 표준화된 용어에 매핑되어 있더라도 정보가 소실되지 않도록 원천 코드도 같이 저장한다.
- **기술 중립성:** CDM은 특정 기술만을 채택하지 않는다. Oracle, SQL Server 등과 같은 관계형 데이터베이스 또는 SAS 분석 데이터 모음으로도 구현할 수 있다.
- **확장성:** CDM은 데이터 처리 및 컴퓨터를 이용한 분석에 최적화되어 있기 때문에 수십억 건에 달하는 임상 관찰을 비롯하여 수억 명이 포함된 데이터베이스 등 다양한 크기의 원천

데이터를 수용할 수 있다.

- **이전 버전과의 호환성:** 이전 CDM으로부터의 모든 변경 사항은 Github 저장소 (<https://github.com/OHDSI/CommonDataModel>)에 명확하게 서술되어 있다. 버전이 업데이트 되어도, 이전 버전의 CDM은 최신 버전을 이용해 쉽게 만들 수 있으며, 이전 버전에 존재하던 정보는 손실되지 않는다.

4.2 데이터 모델 규칙

CDM에 채택된 많은 묵시적 혹은 명시적인 규칙이 있다. 따라서, CDM에 관련된 메소드 개발자는 이러한 규칙을 잘 이해하고 있어야 한다.

4.2.1 모델의 일반적인 규칙

mn CDM은 “사람 중심”的 모델로서, 모든 임상적인 사건에 대한 테이블이 PERSON 테이블을 중심으로 연결되어 있다. 시작 날짜 및 기타 날짜 정보와 더불어 이는 모든 의료 관련 사건에 대해 사람별로 종적 관찰이 가능하도록 한다. 이 규칙에 예외적으로, 표준화된 의료체계 데이터 테이블들 standardized health system data tables은 다양한 도메인의 사건에 직접 연결되어 있다.

4.2.2 스키마의 일반적인 규칙

데이터베이스 관리자는 스키마 또는 데이터베이스 별로 읽기 용 테이블과 읽기/쓰기 테이블을 분리할 수 있다. 임상 사건 및 용어 테이블은 “CDM” 스키마에 저장하며, 이를 사용자 end user에게 읽기 전용으로 설정하는 것을 추천한다. ATLAS 및 사용자가 조작할 필요가 있는 테이블은 “Result” 스키마에 저장하여 “CDM” 데이터가 저장되는 데이터베이스와 분석시마다 중간 결과 등이 저장되는 “Result” 스키마 또는 데이터베이스를 분리하는 것이 좋다. “Result” 스키마에 들어가는 두 테이블은 COHORT와 COHORT_DEFINTION이다. 이 테이블은 10장에 자세히 설명된 것처럼 사용자가 정의할 수 있는 관심 그룹 또는 코호트에 대한 정보 및 설명을 저장하기 위한 테이블들이다. 이는 분석 중에 테이블이 작성될 수 있음을, 즉 사용자별로, 분석별로 개별 테이블을 생성하여 정보를 저장할 수 있음을 의미한다. 읽기-쓰기가 가능한 스키마를 하나로만 설정함으로써, 다수의 사용자들의 이용을 운영 및 제어하는 것은 그에 맞춘 CDM의 구현에 달려 있다.

4.2.3 데이터 테이블의 일반적인 규칙

CDM은 데이터 베이스 플랫폼에 독립적이다. 데이터 유형은 일반적으로 ANSI SQL 데이터 유형 (VARCHAR, INTEGER, FLOAT, DATE, DATETIME, CLOB)을 사용하여 정의된다. VARCHAR에서만 (문자열 길이와 같은) 추가적인 정밀도가 필요하다. 이는 필요한 최소 문자열 길이를 반영하

지만, 구체적인 CDM 인스턴스 내에서 확장할 수 있다. CDM은 날짜 및 날짜 시간 형식을 규정하지 않는다. CDM에 대한 표준 쿼리는 로컬 인스턴스 및 날짜/시간 구성에 따라 달라질 수 있다.

참고: 데이터 모델 자체는 플랫폼에 독립적이지만, 데이터 모델과 함께 작동하도록 구축된 여러 도구는 특정 사양이 요구된다. 이에 대한 자세한 내용은 8장 참조.

4.2.4 도메인의 일반적인 규칙

서로 다른 성격의 기록은 도메인 domain 별로 정리되어 있다. 이러한 기록은 도메인별로 테이블과 필드에 저장되고, 표준화된 용어에 정의된 대로 도메인별 표준 개념 standard concept으로 표현된다 (5.2.3절 참조). 각 표준 개념에 고유한 도메인이 할당되는데, 이는 어떤 테이블에 기록되어야 하는지를 정의한다. 정확한 도메인 할당은 커뮤니티 내에서 항상 논의의 대상이 되지만, 엄격한 도메인-테이블-필드 간 대응 규칙은 어떠한 코드나 concept에 대해서도 항상 정확성을 보장한다. 예를 들어, 증상 및 진단 concept은 Condition 도메인에 속하며 CONDITION_OCCURRENCE 테이블의 CONDITION_CONCEPT_ID로 기록된다. 소위 말하는 시술 시 사용되는 약품은 일반적으로 원천 데이터에서는 Procedure 테이블에 Procedure 코드로 기록되는 경우가 많지만, CDM에서는 이러한 정보는 매핑된 표준 개념이 약물 도메인에 할당되어있기 때문에 DRUG_EXPOSURE 테이블에 저장한다. 다시 말해 모든 concept은 도메인을 가지며, 이 도메인이 해당 기록이 저장될 수 있는 위치를 제한한다. 원천 데이터 베이스 상 KCD-7 코드로 진단 테이블에 입력되어 있는 정보라도, 매핑 후의 standard concept이 Procedure 도메인을 가진다면, 해당 기록은 CDM의 CONDITION_OCCURRENCE 가 아니라 PROCEDURE_OCCURRENCE 테이블에 기록되어야 한다. 표 4.1과 같이 총 30개의 도메인이 있다.

Table 4.1: 각 도메인에 속하는 표준 개념 standard concept의 수.

개념의 개수	도메인 ID	개념의 개수	도메인 ID
1731378	Drug	183	Route
477597	Device	180	Currency
257000	Procedure	158	Payer
163807	Condition	123	Visit
145898	Observation	51	Cost
89645	Measurement	50	Race
33759	Spec Anatomic Site	13	Plan Stop Reason
17302	Meas Value	11	Plan
1799	Specimen	6	Episode
1215	Provider Specialty	6	Sponsor

개념의 개수	도메인 ID	개념의 개수	도메인 ID
1046	Unit	5	Meas Value Operator
944	Metadata	3	Spec Disease Status
538	Revenue Code	2	Gender
336	Type Concept	2	Ethnicity
194	Relationship	1	Observation Type

4.2.5 개념을 통한 내용 표현

CDM 테이블에서는 각 정보의 내용이 완전히 정규화되어 개념 concept으로 저장된다. 개념은 CONCEPT 테이블의 외래 키 역할을 하는 각각의 CONCEPT_ID 값이 할당되어 사건 테이블에 저장된다. CDM의 모든 인스턴스는 (개념에 대한 참고 자료로써 공통 데이터 모델과 함께 상호운용의 핵심 메커니즘이자 OHDSI 연구 네트워크의 기반인) 동일한 CONCEPT 테이블을 사용한다. 표준 개념이 없거나 식별되지 않는 경우에는 CONCEPT_ID가 존재하지 않는 개념이거나 알 수 없음 또는 매팅이 불가능함을 의미하는 0으로 설정된다 (즉, CONCEPT_ID = 0).

CONCEPT 테이블의 정보는 각각의 개념에 대한 상세 정보 (이름, 도메인, 클래스 등)를 포함하고 있다. Concepts, Concept Relationships, Concept Ancestors 및 다른 개념과 관련 있는 정보는 표준화된 용어에 포함되어 있다 (5장 참조).

4.2.6 필드 명명 규칙

모든 테이블의 변수명은 하나의 규칙을 따른다:

Table 4.2: 필드명 규칙.

명명	설명
[Event]_ID	테이블 별 각 행의 고유 식별자로, 테이블 간 관계를 설정하는 외래 키 역할을 한다. 예를 들어 PERSON_ID는 각 개인을 고유하게 식별한다. VISIT_OCCURRENCE_ID는 방문을 고유하게 식별한다.

명명	설명
[Event]_CONCEPT_ID	CONCEPT 참고 테이블의 standard concept에 대한 외래 키. 이는 모든 분석에 기반이 되는 사건의 주요 표현이다. 이 필드에는 OMOP 용어집에 정의된 concept ID 만을 사용할 수 있다. 예를 들어 CONDITION_CONCEPT_ID = 31967에는 SNOMED concept인 “오심 Nausea”에 대한 참조 값을 포함하고 있다.
[Event]_SOURCE_CONCEPT_ID	CONCEPT 참고 테이블의 행에 대한 외래 키. 이 concept 은 원천값 source value(아래)과 동등하며, 이때 [EVENT_CONCEPT_ID]와 동일한 표준 개념이거나 또 다른 비-표준 non-standard 개념일 수 있다. 예를 들어, Condition_SOURCE_CONCEPT_ID = 42496858는 KCD7 용어집의 “지속적 심방세동(I48.1)” concept을 나타내며 이는 non-standard concept이다. 유사한 CONDITION_CONCEPT_ID는 표준 SNOMED-CT concept으로 4232697이다. 이 필드에도 OMOP 용어집에 정의된 concept ID 만을 사용할 수 있다는 점에 유의하라. 표준 개념만이 사건의 의미를 모호하지 않게 표현하므로 표준 분석에 응용 시 상호 운용성이 없는 원천 개념 (SOURCE_CONCEPT)을 사용하는 것은 바람직하지 않다. 즉, 42496858(KCD7 I48.1에 대한 OMOP concept ID) 대신 4232697(SNOMED code 440059007 (Persistent atrial fibrillation)의 OMOP concept ID)를 사용하는 것을 추천한다
[Event]_TYPE_CONCEPT_ID	원천 정보의 출처를 나타내는 CONCEPT 참고 테이블에 대한 외래 키. 이는 사건의 유형이나 개념의 유형을 나타내는 것이 아니라 이 기록을 생성한 메커니즘에 대한 정보를 수집하는 것을 의미한다. 예를 들면, DRUG_TYPE_CONCEPT_ID는 이 기록이 약국에서의 처방 (“Pharmacy dispensing”)으로 부터 발생하였는지 혹은 전자 처방 신청서 (“Prescription written”)로부터 발생하였는지를 구분한다. 역시 OMOP 용어집에서 정의된 concept ID 만을 사용할 수 있다.

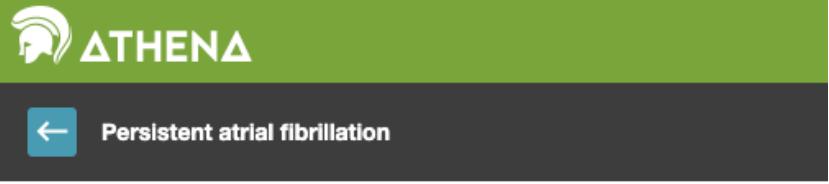
명명	설명
[Event]_SOURCE_VALUE	이 사건이 원천 데이터에 표현된 방식 그대로 쓰인 코드 혹은 자유 서술 문자열이다. 이 원천 값은 데이터 원천 간에 통일되어 있지 않으므로 표준 분석 방식에 사용하는 것은 좋지 않다. 예를 들면, CONDITION_SOURCE_VALUE는 KCD7 코드 I48.1에 점을 제외하고 “I481”라는 기록을 포함할 수 있다. 이 필드를 이용하여 원천값을 유지할 수 있다.

4.2.7 개념과 원천값과의 차이

많은 테이블이 원천값 source value, 원천 개념 source concept, 표준 개념 standard concept으로 다양한 위치에 동일한 정보를 포함하고 있다.

- **원천 값 Source Values** 은 원천 데이터에서의 사건 기록의 본래 표현이다. 이는 ICD9CM, NDC 또는 Read와 같은 널리 사용되는 공공 도메인의 코딩 시스템이나 CPT4, GPI 또는 MedDRA와 같이 독점적인 코딩 시스템, 혹은 남성은 M 여성은 F와 같이 원천 데이터에서만 사용되는 제한된 어휘의 코드일 수 있다. 또한, 표준화 및 제어되지 않은 짧은 자유 텍스트 문구일 수도 있다. 원천 값은 데이터 테이블의 [Event]_SOURCE_VALUE 필드에 저장된다.
- **개념 Concepts** 은 임상적 요소의 의미를 일반화하는 CDM 특이적인 개체이다. 대부분 개념은 이미 의료계에 존재하는 공개 되었거나 독점적인 코딩 체계를 기반으로 하고 있지만, 일부는 새롭게 생성되었다 (CONCEPT_CODE는 “OMOP”으로부터 시작됨). 개념은 모든 도메인에 걸쳐 고유한 ID(concept ID)를 가지고 있다.
- **원천 개념 Source Concepts** 은 원자료에서 사용된 코드를 나타내는 개념 concept이다. 원천 개념은 OMOP 기반의 개념이 아니라 기존에 존재하는 공개 되었거나 독점적인 코딩 체계만을 위해 사용한다. 원천 개념은 데이터 테이블의 [Event]_SOURCE_CONCEPT_ID 필드에 저장된다.
- **표준 개념 Standard Concepts** 은 모든 데이터베이스에서 고유하게 임상적인 개체의 의미를 정의하는 데에 사용되고 원천에서 사용한 코딩 체계와는 독립적인 개념이다. 표준 개념은 일반적으로 이미 공개되어 있거나 독점적인 용어 원천에서 가져온다. 표준 개념과 동일한 의미를 가진 비표준 개념 Non-standard concept은 표준 용어의 표준 개념에 매핑되어 있다(KCD7의 OMOP Concept은 비표준 개념이며, 이는 표준 개념인 SNOMED Concept에 매핑된다). 표준 개념은 데이터 테이블의 [Event]_CONCEPT_ID 필드에서 참조된다.

Source value는 편의 및 품질 보증 Quality Assurance(QA) 목적으로만 제공된다. 여기에는 특정



The screenshot shows the ATHENA interface with the search term "Persistent atrial fibrillation". Below the search bar is a table titled "DETAILS" containing the following information:

Domain ID	Condition	
Concept Class ID	KCD7 code	
Vocabulary ID	KCD7	(?)
Concept ID	I48.1	
Concept code	I48.1	
Invalid reason	Valid	
Standard concept	Non-standard	
Synonyms	지속성 심방세동	
Valid start	07/01/2017	
Valid end	12/31/2099	

Figure 4.2: Persistent atrial fibrillation의 KCD7 코드

데이터 원천의 맥락에서만 의미 있는 정보가 포함될 수 있다. Source value나 source concept을 사용하는 것은 선택사항이지만, 원천 데이터가 코딩 시스템을 사용하는 경우 **강력하게 권장된다**. 하지만 표준 개념의 경우 **필수 사항**이다. 이 표준 개념을 필수적으로 사용하면 모든 CDM 인스턴스가 동일한 언어를 사용할 수 있다. 예를 들면 “지속성 심방세동” (Persistent atrial fibrillation, 그림 4.2 참조)의 condition은 지속성 심방세동에 대한 KCD7 코드가 I48.1임을 나타낸다.

문맥이 없으면, I48.1은 ICD10 또는 ICD10CM 코드 시스템의 Persistent atrial fibrillation 으로 해석될 수도 있다. 하지만 source code와 달리 concept ID는 OMOP 용어 체계 내에서 유일하다. 즉, persistent atrial fibrillation에 대한 KCD7, ICD10, ICD10CM 코드는 모두 I48.1로 동일하지만 concept ID는 42488512 (KCD7), 45548021 (ICD10), 35207785 (ICD10CM) 으로 concept ID

TERM CONNECTIONS (2)			
RELATIONSHIP	RELATES TO	CONCEPT ID	VOCABULARY
Is a	Atrial fibrillation and flutter	42488510	KCD7
Non-standard to Standard map (OMOP)	Persistent atrial fibrillation	4232697	SNOMED

Figure 4.3: Persistent atrial fibrillation의 SNOMED Concept ID

만으로도 모두 구별이 가능하다. KCD7의 지속적 심방세동의 source concept은 그림 4.3과 같이 “OMOP (Non-standard to Standard Map)”관계를 통해 SNOMED 어휘에서 standard concept 4232697로 매핑된다. 또다른 예로 코드 011은 ICD9 에서는 Pulmonary tuberculosis 이지만, UB04 에서는 “Hospital Inpatient (Including Medicare Part A)”로 해석되거나, DRG 용어의 “Nervous System Neoplasms without Complications, Comorbidities”로 완전히 다르게 해석될 수 있다. 이것이 원천과 표준 모두의 concept ID가 중요한 이유이다. Standard SNOMED concept을 참조하는 모든 연구가 지원되는 모든 원천 코드를 포함할 수 있도록 Read, KCD7, ICD10, CIEL 및 MeSH 코드에도 동일한 매핑 관계가 존재한다.

Standard concept과 Source concept의 관계를 보여주는 예가 표 4.7에 나와 있다.

4.3 표준화된 CDM 테이블

CDM에는 16개의 임상 사건 테이블, 10개의 용어 테이블, 2개의 메타데이터 테이블, 4개의 보건 시스템 데이터 테이블, 2개의 보건 경제학 데이터 테이블, 3개의 파생 테이블 및 2개의 결과 스키마 테이블이 포함되어 있다. 이 테이블은 CDM Wiki에 전체 명시되어 있다.¹

이러한 테이블이 실제로 어떻게 사용되는지를 설명하기 위해, 한 사람의 데이터를 이 장의 나머지 부분에서 걸쳐 공통으로 사용할 것이다.

4.3.1 실행 예제: 자궁내막증

자궁내막증은 보통 여성의 자궁 내의 자궁내막 세포가 신체 다른 곳에서 생겨나는 고통스러운 질환이다. 심하면 불임, 장, 방광 문제를 일으킬 수 있다. 이번 절에서는 한 환자의 이 질병에 대한 경험과 이 질병이 공통 데이터 모델로 어떻게 표현되는지를 상세하게 설명하고자 한다.

¹<https://github.com/OHDSI/CommonDataModel/wiki>



나는 이 고통스러운 여정의 모든 과정마다 내가 얼마만큼의 고통을 받고 있는지를 모두에게 납득시켜야 했다.

Lauren은 수년 동안 자궁 내막증 증상을 겪어 왔다. 그러나 진단을 받기 전에 난소에서 낭종이 파열되었다. Lauren에 대한 자세한 내용은 <https://www.endometriosis-uk.org/laurens-story>에서 확인할 수 있다.

4.3.2 PERSON 테이블

Lauren에 대해서 우리가 알고 있는 것은?

- 그녀는 36세 여성이다
- 그녀의 생년월일은 1982년 3월 12일이다
- 그녀는 백인이다
- 그녀는 영국인이다

이를 염두에 두면 PERSON 테이블을 다음과 같이 나타낼 수 있다:

Table 4.3: PERSON 테이블.

필드 Column 이름	값	설명
PERSON_ID	1	PERSON_ID는 원천에서 직접적으로 생성되거나 변환 과정의 일부분으로 생성된 정수여야 한다.
GENDER_CONCEPT_ID	8532	여성을 의미하는 concept ID는 8532이다.
YEAR_OF_BIRTH	1982	
MONTH_OF_BIRTH	3	
DAY_OF_BIRTH	12	

필드 Column 이름	값	설명
BIRTH_DATETIME	1982-03-12 00:00:00	시간을 정확히 알 수 없는 경우 자정으로 한다.
DEATH_DATETIME		
RACE_CONCEPT_ID	8527	백인을 의미하는 concept ID는 8527이다. 영국인이라는 race는 4093769이다. 둘 다 해당할 경우 후자를 활용한다. 민족성은 ETHNICITY_CONCEPT_ID가 아닌 race의 일부로써 여기에 저장된다는 점에 유의하라.
ETHNICITY_CONCEPT_ID	38003564	이는 히스패닉을 다른 사람과 구분하기 위해 사용되는 전형적인 미국식 표기법이다. 이 경우 영국인이라는 민족성은 RACE_CONCEPT_ID에 저장된다. 미국 이외의 지역에서는 사용되지 않는다. 38003564 는 “히스패닉이 아님”을 나타낸다.
LOCATION_ID		주소는 알려지지 않았다.
PROVIDER_ID		일차 진료 제공자는 알려지지 않았다.
CARE_SITE		일차 진료 장소는 알려지지 않았다.
PERSON_SOURCE_ID	1	대부분 PERSON_ID 와 동일 하지만
VALUE		일반적으로 이는 원천 데이터에서의 그녀의 식별자가 될 것이다.
GENDER_SOURCE_ID	F	원천 데이터에 나타난 성별에 대한 값이 여기에 저장되어 있다.
VALUE		
GENDER_SOURCE_ID	0	원천 데이터의 성별에 대한 값이 OHDSI에서 지원하는 코딩 체계를 사용한 경우 해당 concept이 여기에 해당한다. 예를 들어, 그녀의 성별이 원천에서 “sex-F”이고 해당 PCORNet 용어에 대한 OMOP concept ID 44814665를 이 필드에 입력한다.
RACE_SOURCE_ID	white	인종 값이 원천에 있는 대로 여기에 저장된다.
VALUE		
RACE_SOURCE_ID	0	GENDER_CONCEPT_ID와 같은 원리 적용
CONCEPT_ID		

필드 Column 이름	값	설명
ETHNICITY_SOURCE_ VALUE	english	민족성 값이 원천에 나와 있는 대로 여기에 저장된다.
ETHNICITY_SOURCE_ CONCEPT_ID	0	GENDER_SOURCE_CONCEPT_ID와 같은 원리 적용

4.3.3 OBSERVATION_PERIOD 테이블

OBSERVATION_PERIOD 테이블은 민감성과 특수성을 고려하여 환자의 최소한의 인구통계, 질환, 시술 및 약물이 원천 시스템에 기록될 수 있도록 관찰된 기간을 합리적인 예상을 통해 정의하도록 설계되었다. 보험 데이터의 경우 일반적으로 환자의 등록 시기이다 (역자 주: 한국 보험공단의 경우 국민건강보험에 가입되어 있는 기간을 뜻한다). 대부분의 의료 시스템이 어떤 의료 기관이나 제공업체를 방문할지 결정해두지 않기 때문에 전자 의무 기록(EHR)에서는 더욱 까다롭다. 전자의 무기록의 경우, 차선책으로서 시스템의 첫 번째 기록 OBSERVATION_PERIOD의 시작일이 되고, 마지막 기록은 종료일로 간주한다. 즉, 청구기록의 경우에는 환자의 등록 기간에 맞춰 개인마다 여러 개의 (만약 개인이 보험 가입과 탈퇴를 반복한 경우) observation period를 가질 수 있지만, 전자의무기록의 경우에는 일반적으로는 개인당 하나의 observation period를 가지게 된다.

Lauren의 Observation Period는 어떻게 정의될까?

표 4.4에 나타난 Lauren의 정보가 전자 의무 기록 시스템에 기록되었다고 가정하자. 그녀의 관찰 기간에서 얻어진 방문기록은 다음과 같다:

Table 4.4: Lauren의 의료 기관 방문.

Encounter ID	Start date	Stop date	Type
70	2010-01-06	2010-01-06	outpatient
80	2011-01-06	2011-01-06	outpatient
90	2012-01-06	2012-01-06	outpatient
100	2013-01-07	2013-01-07	outpatient
101	2013-01-14	2013-01-14	ambulatory
102	2013-01-17	2013-01-24	inpatient

방문기록을 기반으로 했을 때 그녀의 OBSERVATION_PERIOD 테이블은 다음과 같을 것이다:

Table 4.5: OBSERVATION_PERIOD 테이블.

필드 이름	값	설명
OBSERVATION_PERIOD_ID	1	이는 일반적으로 테이블의 각 기록에 대한 고유 식별자로서 자동으로 생성되는 값이다.
PERSON_ID	1	PERSON 테이블에서 Laura의 기록에 대한 외래 키이며 PERSON을 OBSERVATION_PERIOD 테이블에 연결한다.
OBSERVATION_PERIOD_START_DATE	2010-01-06	이는 기록상 그녀의 가장 처음 방문했을 때 시작 날짜이다.
OBSERVATION_PERIOD_END_DATE	2013-01-24	이는 기록상 그녀의 가장 마지막 방문했을 때 마지막 날짜이다.
PERIOD_TYPE_ID	44814725	Concept의 class가 “Obs Period Type”인 concept 중 가장 좋은 선택은 44814724이며, 이는 “의료 관련 방문 기간(Period covering healthcare encounters)”을 나타낸다.
CONCEPT_ID		

4.3.4 VISIT_OCCURRENCE

VISIT_OCCURRENCE에서는 환자의 의료 시스템에 방문한 정보에 대해 저장되어 있다. OHDSI 언어 내에서 이를 Visit이라고 하며 주요한 사건으로 간주한다. 의료 서비스가 제공될 수 있는 다양한 환경을 나타내는 광범위한 계층 구조를 가진 열두 가지 주요 방문 카테고리가 있다. 가장 일반적인 Visit 기록은 입원 inpatient, 외래 outpatient, 응급실 emergency department 및 비-의료 기관방문 non-medical institution Visits이다.

Lauren의 방문을 Visit으로 어떻게 표현할 수 있을까?

예를 들어 VISIT_OCCURRENCE 테이블의 표 4.4의 입원 환자 방문을 나타내어 보자.

Table 4.6: VISIT_OCCURRENCE 테이블.

필드 이름	값	설명
VISIT_OCCURRENCE_ID	514	이는 일반적으로 테이블의 각 기록에 대한 고유 식별자로서 자동으로 생성되는 값이다.

필드 이름	값	설명
PERSON_ID	1	PERSON 테이블에서 Laura의 기록에 대한 외래 키이며 PERSON을 VISIT_OCCURRENCE에 연결한다.
VISIT_CONCEPT_ID	9201	입원 환자 방문을 나타내는 외래 키는 9201이다.
VISIT_START_DATE	2013-01-17	Visit의 시작 날짜.
VISIT_START_DATE	2013-01-17	Visit의 날짜와 시간. 시간을 알 수 없기 때문에 자정으로 나타낸다.
VISIT_END_DATE	2013-01-24	Visit의 종료 날짜. 일일 방문이라면 이 값이 시작 날짜와 동일해야 한다.
VISIT_END_DATETIME	2013-01-24 00:00:00	Visit의 종료 날짜와 시간. 시간을 알 수 없기 때문에 자정으로 나타낸다.
VISIT_TYPE_CONCEPT_ID	32035	이는 방문 기록의 보험 청구, 병원 청구, 전자 의무 기록(EHR)과 같은 제공처에 대한 정보를 제공한다. 해당 예에서는 방문 기록이 전자 의무 기록과 유사하므로 concept ID 32035 (“Visit derived from EHR encounter record”)이 사용되었다.
PROVIDER_ID*	NULL	방문 기록에 해당 제공자와 관련된 ID가 있으면 이 필드에 기록한다. 이는 PROVIDER 테이블의 PROVIDER_ID 필드의 내용이어야 한다.
CARE_SITE_ID	NULL	방문 기록에 치료 제공 장소와 관련된 ID가 있으면 이 필드에 기록한다. 이는 CARE_SITE 테이블의 CARE_SITE_ID 필드의 내용이어야 한다.
VISIT_SOURCE_VALUE	inpatient	출처에 나와 있는 방문 값 그대로 여기에 입력한다. Lauren의 데이터에는 존재하지 않는다.

필드 이름	값	설명
VISIT_SOURCE_	0	출처의 방문 값이 OHDSI에서 통용되는 용어를 사용하여 코딩된 경우 원천 코드를 나타내는 CONCEPT_ID 값을 여기에 넣는다. Lauren의 데이터에는 존재하지 않는다.
CONCEPT_ID		
ADMITTED_FROM_	0	환자가 어디에서부터 입원해 왔는지 알 수 있는 경우 이를 나타내는 concept을 포함하고 있다.
CONCEPT_ID		이 concept은 “Visit”的 도메인을 가지고 있어야 한다. 예를 들어 만약 환자가 집에서 병원으로 입원한 경우 8536 (“Home”)값일 것이다.
ADMITTED_FROM_	NULL	환자가 어디에서부터 입원해 왔는지를 나타내는 원천 값이다. 위의 예를 활용하면 여기에는 “Home”이 들어가야 한다.
SOURCE_VALUE		
DISCHARGE_TO_	0	환자가 어디로 퇴원 되었는지 알 수 있는 경우 이를 나타내는 concept을 나타낸다. 이 concept은 “Visit” 도메인을 가지고 있어야 한다. 예를 들면, 만약 환자가 보조 생활 시설로 보내졌을 경우 concept ID는 8615 (“Assisted Living Facility”)일 것이다.
CONCEPT_ID		
DISCHARGE_TO_	0	환자가 퇴원한 곳을 나타내는 원천 값이다.
SOURCE_VALUE		위의 예를 활용하면 “보조 생활 시설”이 된다.
PRECEDING_VISIT_	NULL	현재 Visit의 바로 이전의 방문을 나타낸다.
OCCURRENCE_ID		ADMITTED_FROM_CONCEPT_ID와 달리 Visit Concept이 아닌 실제 Visit Occurrence 기록에 연결된다. 또한, Visit Occurrence에 따른 기록은 없으며 Visit Occurrence는 이 필드를 통해서만 연결되어 있다.

- 환자는 입원하는 경우와 마찬가지로 한번 방문하는 동안 여러 의료 제공자와 상호 작용할 수 있다. 이러한 상호작용은 VISIT_DETAIL 테이블에 기록될 수 있다. 이 장에서는 자세히 다루지 않지만, CDM wiki에서 VISIT_DETAIL 테이블에 대한 자세한 내용을 확인할 수 있다.

4.3.5 CONDITION_OCCURRENCE

CONDITION_OCCURRENCE 테이블의 기록은 제공자가 관찰하거나 환자가 보고한 상태의 진단, 징후 또는 증상이다.

Lauren의 condition은 무엇일까?

그녀의 예로 돌아가자면 그녀는 다음과 같이 말한다:

3년 정도 전쯤 그동안 매우 통증이 심했던 월경이 점점 더 고통스러워지고 있다는 것을 알아챘다. 나는 내 대장 바로 옆에서 날카롭게 쑤시는 통증을 느끼기 시작했고 꼬리뼈와 아랫골반 부위가 따갑고 부풀어 오르는 것을 느꼈다. 내 월경이 너무 고통스러워져서 일을 한 달에 하루 이틀 쉬었다. 진통제가 가끔 고통을 줄여 주긴 했지만, 보통은 그렇지 않았다.

월경통이라고 하는 고통스러운 월경 경련의 SNOMED 코드는 266599000이다. 표 4.7은 CONDITION_OCCURRENCE 테이블에 어떻게 표시되는지를 보여준다:

Table 4.7: CONDITION_OCCURRENCE 테이블.

필드 이름	값	설명
CONDITION_OCCURRENCE_ID	964	이는 일반적으로 테이블의 각 기록에 대한 고유 식별자로서 자동으로 생성되는 값이다.
PERSON_ID	1	이는 PERSON 테이블에서 Laura의 기록에 대한 외래 키이며 PERSON을 CONDITION_OCCURRENCE에 연결한다.
CONDITION_CONCEPT_ID	194696	SNOMED 코드 266599000을 나타내는 외래 키 (concept ID): 194696.
CONDITION_START_DATE	2010-01-06	Condition의 인스턴스가 기록된 날짜이다.
CONDITION_START_DATETIME	2010-01-06 00:00:00	Condition의 인스턴스가 기록된 날짜 및 시간이다. 시간을 알 수 없으므로 자정으로 입력한다.
CONDITION_END_DATE	NULL	이는 인스턴스가 종료된 것으로 여겨지는 날짜지만 거의 기록되지 않는다.

필드 이름	값	설명
CONDITION_END_DATE	NULL	Condition의 인스턴스가 종료된 것으로 여겨지는 날짜 및 시간이 알려져 있으면 입력한다.
CONDITION_TYPE_ID	32020	이 열은 기록의 출처, 즉 보험 청구, 병원 청구 기록, 전자 의무 기록(EHR) 등에서 얻어졌다는 정보를 제공하기 위한 것이다. 해당 예에서는 방문 기록이 EHR과 유사하기 때문에 concept ID 32020 (“EHR encounter diagnosis”)을 사용한다. 이 필드에 있는 concept은 “Condition Type” 용어에 있는 것이 여야 한다.
CONDITION_STATUS_ID	0	상황에 대해 알려진 것이 있는 경우 입력한다. 예를 들어, concept ID에 4203942가 사용되었을 경우 Condition이 인정된 진단명일 수 있다.
STOP_REASON	NULL	Condition이 더 이상 존재하지 않는 이유가 알려져 있으면 원천 데이터에 있는 대로 입력한다.
PROVIDER_ID	NULL	만약 condition 기록에 진단의 제공자가 수록되어 있으면 해당 제공자의 ID를 이 필드에 입력한다. 이는 방문 시 제공자를 나타내는 PROVIDER 테이블의 PROVIDER_ID의 내용이어야 한다.
VISIT_OCCURRENCE_ID	509	Condition이 진단되었을 당시 Visit 값 (VISIT_OCCURRENCE 테이블의 VISIT_OCCURRENCE_ID에 대한 외래 키)

필드 이름	값	설명
CONDITION_SOURCE_ VALUE	266599000	Conditions를 나타내는 원래의 원천 값. Lauren의 월경 곤란의 사례에서는 해당 Condition에 대한 SNOMED 코드가 여기에 저장되고 코드를 나타내는 concept은 CONDITION_SOURCE_CONCEPT_ID로 이동했으며 이로부터 매핑된 standard concept은 CONDITION_CONCEPT_ID 필드에 저장된다.
CONDITION_SOURCE_ CONCEPT_ID	194696	원천의 질환 값이 OHDSI에서 활용하는 용어로 코드화되어 있는 경우 그 값을 나타내는 concept ID를 여기에 입력한다. 월경 곤란의 예에서는 그 원천 값이 SNOMED 코드이므로 코드를 나타내는 concept은 194696이다. 이 경우에는 CONDITION_CONCEPT_ID 영역과 같은 값이다.
CONDITION_STATUS_ SOURCE_VALUE	0	원천 데이터에서 질환의 상태에 대한 값이 있다면 여기에 입력한다.

4.3.6 DRUG_EXPOSURE

DRUG_EXPOSURE 테이블은 환자에게 약물을 투여하고자 한 의도나 실제 투여에 대한 기록을 수집한다. 의약품에는 처방전이 필요한 전문의약품과 처방전 없이 살 수 있는 일반 의약품, 백신 및 고분자 생물학적 제제를 포함한다. 약물 노출은 처방, 처방전, 약품 불출, 시술 시 사용된 약품, 기타 환자가 보고한 정보와 같은 임상적인 사건에서 유추된다.

Lauren의 약물 노출은 어떻게 나타낼 수 있을까?

월경통을 완화하기 위해 Lauren은 2010년 1월 6일 방문하여 아세트아미노펜 375mg (일명 Paracetamol, 예를 들어, 미국에서 NDC 코드 69842087651로 판매되었다) 경구 제제 60알을 30일 치 쳐방으로 받았다. 이는 DRUG_EXPOSURE 테이블에서 다음과 같이 나타난다:

Table 4.8: DRUG_EXPOSURE 테이블.

필드 이름	값	설명
DRUG_EXPOSURE_ID	1001	이는 일반적으로 테이블의 각 기록에 대한 고유 식별자로서 자동으로 생성되는 값이다.
PERSON_ID	1	PERSON 테이블에서 Laura의 기록에 대한 외래 키이며 PERSON을 DRUG_EXPOSURE에 연결한다.
DRUG_CONCEPT_ID	1127433	의약품에 대한 concept. 아세트아미노펜에 대한 NDC 코드는 concept 1127433으로 표시되는 RxNorm 코드 313782에 매핑된다.
DRUG_EXPOSURE_START_DATE	2010-01-06	약물에 노출되기 시작한 날짜
DRUG_EXPOSURE_START_DATETIME	2010-01-06 00:00:00	약물에 노출되기 시작한 날짜 및 시각. 알 수 없는 경우 자정을 입력
DRUG_EXPOSURE_END_DATE	2010-02-05	약물 노출이 종료되는 날짜. 서로 다른 출처에서 알려진 날짜나 추정된 날짜일 수 있으며 환자가 약물에 노출 지속한 날짜의 마지막 날을 의미한다. 해당 사례에서는 Lauren이 30일 동안 받았음을 알기에 이 날짜가 추론될 수 있다.
DRUG_EXPOSURE_END_DATETIME	2010-02-05 00:00:00	약물 노출 종료 날짜 및 시간. DRUG_EXPOSURE_END_DATE와 비슷한 규칙이 적용된다. 알 수 없는 경우 자정을 입력
VERBATIM_END_DATE	NULL	원천에서 실제 약물 노출 종료 날짜를 명시한 경우. 아닐 경우 end date는 환자가 전체 날짜에 약물에 노출되었다고 가정하여 유추되어 결정된다.
DRUG_TYPE_	38000177	해당 열은 보험 청구, 처방 기록 등에서 비롯된 기록의 출처에 대한 정보를 제공하기 위해 작성이 되었다. 이 예에서는 concept 38000177 (“Prescription written”)이 사용되었다.
CONCEPT_ID		

필드 이름	값	설명
STOP_REASON	NULL	약물 투여가 중단된 이유. 요법 완료, 변경 제거 등이 이유에 포함된다. 이 정보가 수집되는 경우는 거의 없다.
REFILLS	NULL	대다수의 나라에서 처방 시스템의 일부인 초기 처방 이후 자동 재조제 횟수. 초기 처방은 세지 않고 NULL로 시작한다. Lauren의 아세트아미노펜의 경우 재조제되지 않았으므로 NULL이다.
QUANTITY	60	최초 처방전 또는 조제 기록에 기록된 약물의 양
DAY_SUPPLY	30	처방 된 약의 투여 일수
SIG	NULL	최초 처방전 또는 조제 기록에 기록되어 미국 처방 시스템의 용기에 인쇄된 약 처방전의 지침 (“signetur”). 약물 지침은 CDM에서 아직 표준화되지 않았으며 표기된 그대로 입력된다.
ROUTE_CONCEPT_ID	4132161	이 concept은 환자의 약물 투여 경로를 나타낸다. Lauren은 아세트아미노펜을 경구 복용하였으므로, concept ID 4132161을 사용하였다.
LOT_NUMBER	NULL	제조업체로부터의 특정 수량 또는 의약품에 할당된 식별자. 이 정보는 거의 수집되지 않는다.
PROVIDER_ID	NULL	약품 기록에 처방자에 대한 정보가 있으면 해당 공급자의 ID가 해당 영역에 들어간다. 이때 PROVIDER 테이블의 PROVIDER_ID를 사용한다.
VISIT_OCCURRENCE_ID	509	약물 처방 시 VISIT_OCCURRENCE 테이블에 대한 외래 키
VISIT_DETAIL_ID	NULL	약물 처방 시 VISIT_DETAIL 테이블에 대한 외래 키
DRUG_SOURCE_VALUE	69842087651	원천 데이터에 나와 있는 의약품의 원천 코드. Lauren의 예에서 NDC 코드가 저장된다.

필드 이름	값	설명
DRUG_SOURCE_	750264	이는 약물의 원천 값을 나타내는 concept이다.
CONCEPT_ID		750264 concept은 “Acetaminophen 325 MG Oral Tablet”의 NDC 코드를 나타낸다.
ROUTE_SOURCE_	NULL	원천 기록에 나와 있는 그대로의 등록 경로를
VALUE		나타낸다.

4.3.7 PROCEDURE_OCCURRENCE

PROCEDURE_OCCURRENCE 테이블에는 의료 서비스 제공자가 진단 또는 치료 목적으로 환자에게 주문하거나 시행한 활동 또는 과정에 대한 기록이 포함되어 있다. Procedure는 다양한 수준의 표준화를 통해 다양한 형태로 여러 데이터 출처에 존재한다. 예를 들면 다음과 같다:

- 수행된 시술을 포함한 의료 서비스에 대한 청구의 일부로 시술 코드를 포함하는 의료 청구
- 처방/시행 정보로부터 시술에 대한 정보를 수집하는 전자 의무 기록

Lauren이 받은 시술은 무엇일까?

Lauren의 설명에서 2013년 1월 14일에 4x5cm 낭종을 왼쪽 난소 초음파를 통해 확인했다는 것을 알 수 있다. PROCEDURE_OCCURRENCE 테이블에 나타내는 방법은 다음과 같다:

Table 4.9: PROCEDURE_OCCURRENCE 테이블.

필드 이름	값	설명
PROCEDURE_	1277	이는 일반적으로 테이블의 각 기록에 대한 고유 식별자로서 자동으로 생성되는 값이다.
OCCURRENCE_ID		
PERSON_ID	1	PERSON 테이블에서 Laura의 기록에 대한 외래 키이며 PERSON을 PROCEDURE_OCCURRENCE에 연결한다.
PROCEDURE_	4127451	골반 초음파에 대한 SNOMDE 시술 코드는 304435002이고, concept ID 4127451로 나타낼 수 있다.
CONCEPT_ID		
PROCEDURE_DATE	2013-01-14	시술이 시행된 날짜
PROCEDURE_DATE	2013-01-14	시술이 시행된 날짜 및 시각. 시간을 알 수 없는 경우 자정으로 입력한다.
DATETIME	00:00:00	

필드 이름	값	설명
PROCEDURE_TYPE_	38000275	해당 열은 보험 청구, 전자 의무 기록(EHR)
CONCEPT_ID		내의 처방,시행 기록과 같은 시술 기록의 출처에 대한 정보를 제공하기 위한 것이다. 해당 예제에서 concept ID 38000275 (“EHR order list entry”)이 전자 의무 기록으로부터의 시술 기록으로 사용된다.
MODIFIER_CONCEPT_ID	0	이는 시술에 대한 한정어를 나타내는 concept ID 위한 부분이다. 예를 들어, 기록에 CPT4의 시술이 양측에서 수행되었다고 한다면 concept ID 42739579 (“Bilateral procedure”)가 사용되는 것이다.
QUANTITY	0	처방 또는 등록된 시술의 수. 수량이 누락되거나 숫자 0 또는 1일 경우 모두 같은 뜻이다.
PROVIDER_ID	NULL	시술 기록에 제공자가 기록되어 있으면, 제공자의 ID를 이 영역에 입력한다. 이는 PROVIDER 테이블에 있는 PROVIDER_ID의 외래 키 여야만 한다.
VISIT_OCCURRENCE_ID	740	시술가 실행된 방문 정보 (VISIT_OCCURRENCE 테이블의 VISIT_OCCURRENCE_ID로 표시됨)를 알 수 있는 경우 입력한다.
VISIT_DETAIL_ID	NULL	시술이 실행된 방문에 대한 세부 사항 (VISIT_DETAIL 테이블의 VISIT_DETAIL_ID로 표시됨)이 있는 경우 입력한다.
PROCEDURE_SOURCE_VALUE	304435002	원천 데이터에 있는 그대로의 시술에 대한 코드 및 정보
PROCEDURE_SOURCE_CONCEPT_ID	4127451	시술의 원천 값을 나타내는 concept
MODIFIER_SOURCE_VALUE	NULL	원천 데이터에 나타난 그대로의 원천 코드에 대한 한정어

4.4 부가 정보

이 장에서는 데이터 표현 방법의 예로 CDM에서 사용할 수 있는 일부 테이블만을 다룬다. 자세한 정보는 위키 사이트²에서 참고할 수 있다.

4.5 요약



- CDM은 광범위한 관찰 연구 활동을 지원하도록 설계되었다.
- CDM은 사람 중심 모델이다.
- CDM은 데이터 구조를 표준화할 뿐만 아니라 표준화된 용어를 통해 내용 표현을 표준화한다.
- 충분한 추적 가능성을 위하여 원천 코드가 CDM에 유지된다.

4.6 예제

전제 조건

첫 번째 연습에서는 앞에서 설명한 CDM 테이블을 확인해야 하며, ATHENA³ 또는 ATLAS⁴를 통해 용어에 있는 concept을 찾아야 할 것이다.

Exercise 4.1. John은 1974년 8월 4일에 태어난 흑인 남자이다. 이 정보를 인코딩하는 PERSON 테이블 항목을 정의하라.

Exercise 4.2. John은 2015년 1월 1일에 현재 이용하는 보험에 등록했다. 그의 보험 데이터베이스의 데이터는 2019년 7월 1일에 추출되었다. 이 정보를 인코딩하는 OBSERVATION_PERIOD 테이블 항목을 정의하라.

Exercise 4.3. John은 2019년 5월 1일에 Ibuprofen 200 MG Oral 정제 (NDC 코드: 76168009520)를 30일간 투여하도록 처방되었다. 이 정보를 인코딩하는 DRUG_EXPOSURE 테이블 항목을 정의하라.

²<https://github.com/OHDSI/CommonDataModel/wiki>

³<http://athena.ohdsi.org/>

⁴<http://atlas-demo.ohdsi.org>

전제 조건

해당 마지막 세 연습 문제에서는 8.4.5절에 설명된 것과 같이 R, R-Studio 그리고 Java가 설치되었 있다고 가정한다. SqlRender, DatabaseConnector 및 Eunomia 패키지가 요구되고 아래 내용을 통해 설치할 수 있다:

```
install.packages(c("SqlRender", "DatabaseConnector", "devtools"))
devtools::install_github("ohdsi/Eunomia", ref = "v1.0.0")
```

Eunomia 패키지는 로컬 R 세션 내에서 실행될 CDM의 가상 데이터 모음을 제공한다. 연결 세부 사항은 아래 내용을 통하여 얻을 수 있다:

```
connectionDetails <- Eunomia::getEunomiaConnectionDetails()
```

CDM 데이터베이스 스키마는 “main”이다.

Exercise 4.4. SQL과 R을 사용하여 “Gastrointestinal hemorrhage” (concept ID 192671) 질환의 모든 기록을 검색하라.

Exercise 4.5. SQL과 R을 사용하여 원천 코드로 “Gastrointestinal hemorrhage” 질환의 모든 기록을 검색하라. 이 데이터베이스는 ICD-10을 사용하며 관련 ICD-10 코드는 “K92.2”이다.

Exercise 4.6. SQL과 R을 사용하여 PERSON_ID가 61인 사람의 관찰 기간을 검색하라.

답변은 부록 E.1에서 확인 할 수 있다.

Chapter 5

OMOP 표준 용어

Chapter leads: Christian Reich & Anna Ostropolets

흔히 “용어 Vocabulary”라고 불리는 OMOP 표준 용어 Standardized Vocabularies는 OHDSI 연구 네트워크의 기초적인 부분이자, 공통 데이터 모델(CDM)의 핵심 부분이다. OMOP 표준 용어는 데이터의 내용을 정의함으로써 분석 방법, 정의, 결과를 표준화하여 진정한 원격 (방화벽 안에서) 네트워크 연구와 분석을 가능하게 한다. 일반적으로, 코딩 체계를 사용한 구조화된 데이터이든 혹은 자유 진술 문으로 구성된 데이터이든, 관찰 의료 데이터의 내용을 찾아 해석하는 것은 임상 사건을 설명하는 수많은 방법을 고민하는 연구 실무자에게까지 전달되기 마련이다. OHDSI는 표준화된 형식뿐 아니라 엄격한 표준 콘텐츠와의 조화가 필요하다.

이 장에서는 먼저 기초적인 부분을 이해하고 활용하기 위해, 표준 용어의 주요 원칙, 구성 요소 및 관련 규칙, 규약 및 일반적인 상황에 관해 설명하고자 한다. 또한 이를 지속해서 개선하기 위해 커뮤니티의 지원이 필요한 부분을 언급할 것이다.

5.1 왜 용어 Vocabulary인가, 그리고 왜 표준화인가?

의학 용어의 역사는 흑사병 plague 및 기타 질병을 관리하기 위해 사용했던, 중세 런던의 사망 증명서 Bills of Mortality까지 거슬러 올라간다. (그림 5.1 참조)

그 후, 의학 용어 분류는 규모와 복잡성이 크게 확대되면서 시술, 서비스, 약물, 의료기기 등, 의료의 다른 측면으로 널리 전파되었다. 의학 용어 분류의 주요 원칙은 동일하게 유지된다. 즉, 일부 의료 커뮤니티가 환자 데이터를 획득, 분류 및 분석하기 위한 목적으로 동의한 통제 어휘, 전문용어,

1660.

A General BILL for this present Year,

Ending the 11th Day of December 1660.

According to the Report made to the King's most excellent Majesty,
By the Company of Parish Clerks of LONDON, &c.

DISEASES and CASUALTIES.

A	Bortive and Stillborn	421	Flox and Small Pox	—	—	1523	Palsy	—	—	—	17
	Aged	909	Found dead in the Streets,	{	2		Plague	—	—	—	36
	Ague and Fever	—	Fields, &c.				Plurify	—	—	—	12
	Apoplexy and Suddenly	91	French Pox	—	—	51	Quinny and fore Throat	—	—	—	21
	Blafted and Planet	3	Gout	—	—	4	Rickets	—	—	—	44 ¹
	Bleeding and bloody Issue	7	Grief	—	—	13	Rising of the Lights	—	—	—	210
	Bloody Flux, Scowring, and Flux	346	Griping in the Guts	—	—	253	Rupture	—	—	—	12
	Burnt and Scalded	6	Hanged and made away them-selves	{	11		Scurvy	—	—	—	82
	Cancer, Gangrene and Fistula	63	Head-ach and Headmouldshot	—	35		Shot	—	—	—	7
	Canker, fore Mouth and Thrush	73	Jaundies	—	—	102	Shingles	—	—	—	1
	Childbed	226	Imposthume	—	—	105	Sores, Ulcers, broken and bruised Limbs	{	61		
	Chrifomes and Infants	858	Killed by several Accidents	—	55		Spleen	—	—	—	7
	Cold, Cough and Hiccough	33	King's Evil	—	28		Spotted Fever and Purples	—	—	—	368
	Colick and Wind	116	Lethargy	—	6		Starved	—	—	—	7
	Consumption and Tiffick	2982	Livergrown	—	8		Strangury	—	—	—	22
	Convulsion	742	Lunatick and Frenzy	—	14		Stopping of the Stomach	—	—	—	186
	Cut of the Stone and Stone	46	Megrims	—	5		Surfeit	—	—	—	202
	Dropfy and Tympany	646	Measles	—	6		Swine Pox	—	—	—	2
	Drowned	57	Mother	—	1		Teeth and Worms	—	—	—	839
	Executed	7	Murthered	—	7		Vomiting	—	—	—	8
	Falling Sickness	—	Overlaid and Starved at Nurse	46			Wen	—	—	—	1

Figure 5.1: 1660 London Bill of Mortality, 당시 알려진 62가지 질병의 분류 체계를 사용하여 사망한 거주자의 사망 원인을 보여준다.

계층 hierarchy 및 언어 개념 ontology이다. 이러한 용어집의 상당수는 공공기관과 정부 기관에서 장기적으로 의무 관리하고 있다. 예를 들면, 세계보건기구(WHO)는 최근 국제 질병분류(ICD)에 11차 개정판(ICD11)을 추가하였다. 지역 정부는 KCD7(한국), ICD10CM(미국), ICD10GM(독일) 등과 같은 국가별 버전을 만들고 있다. 정부는 또한 의약품의 마케팅과 판매를 통제하고 인증된 의약품의 목록을 운영하고 있다. 용어집은 상업용 제품 또는 내부용으로 민간 부문에서도 사용된다. 예를 들면, 전자 의무 기록 Electronic Health Records(EHR) 시스템과 의료보험청구용이 있다.

그 결과, 각 국가, 지역, 의료시스템과 의료기관은 그 용어가 사용되는 지역에서만 쓰이는 자체 질병 분류체계를 갖고 있을 가능성이 크다. 이렇게 무수히 많은 용어집은 사용 중인 시스템의 상호운용성을 방해한다. 표준화는 환자 데이터 교환을 가능하게 하고, 전 세계적 수준의 의료 데이터 분석의 길을 열어주고, 성능 특성 분석 및 품질 평가를 포함한 체계적이고 표준화된 연구를 가능하게 하는 핵심 요소이다. 이러한 문제를 해결하기 위해, 위에서 언급된 WHO와 the Standard Nomenclature of Medicine(SNOMED) 또는 Logical Observation Identifiers Names and Codes(LOINC) 같은 다국적 기관이 생겨나고 광범위한 표준을 만들기 시작했다. 미국의 보건 IT 표준 위원회 Health IT Standards Committee(HITAC)는 다양한 단체 간의 건강 정보 교환을 위한 공통 플랫폼에서 사용하기 위해 ONC(National Coordinator for Health IT)의 표준으로 SNOMED, LOINC 및 약물 용어인 RxNorm을 사용할 것을 권장하고 있다.

OHDSI는 관찰 연구를 위한 국제 표준인 OMOP CDM을 개발했다. CDM의 일부로, OMOP 표준 용어는 다음 두 가지 목적으로 사용 할 수 있다:

- 커뮤니티에서 사용되는 모든 용어의 공통 저장 자료
- 연구에 사용하기 위한 표준화와 매핑

표준화된 용어는 커뮤니티에 무료로 제공되며, **OMOP CDM 실제 사용 시마다 필수 참조 테이블로 사용되어야 한다.**

5.1.1 표준화된 용어 구축

표준 용어의 모든 용어는 같은 공통 형식으로 통합된다. 이를 통해 연구자가 기존 용어의 여러 가지 형식과 수명 주기 규칙을 이해하거나 처리할 필요가 없다. 모든 용어는 Pallas 시스템¹을 사용하여 정기적으로 업데이트되고, 통합된다. 용어는 OMOP CDM 워크그룹의 일부인 OHDSI Vocabulary 팀이 만들어 운영하고 있다. 오류가 발견되면 OHDSI Forums² 또는 CDM Github 페이지³에 게시하여 오류를 보고하고 리소스를 개선할 수 있도록 도와주길 바란다.

¹<https://github.com/OHDSI/Vocabulary-v5.0>

²<https://forums.ohdsi.org>

³<https://github.com/OHDSI/CommonDataModel/issues>

5.1.2 표준 용어 이용하기

표준 용어 정보를 얻기 위해, Pallas를 직접 실행할 필요는 없다. 대신, ATHENA⁴에서 최신 버전을 다운로드하여 로컬 데이터베이스에 적재하면 된다. ATHENA는 강력한 용어 검색 기능도 가지고 있다.

모든 표준 용어 테이블에 포함된 zip 파일을 다운로드하려면, OMOP CDM에 필요한 모든 용어집을 선택해야 한다. 표준 개념 standard concept을 가진 용어집은 (5.2.6절 참조) 미리 선택되어 있다. 원천 데이터에 사용되는 용어를 추가한다. 저작권이 있는 용어집은 선택할 수 없다. 해당 용어집을 리스트에 포함하려면 “License required” 버튼을 클릭해야 한다. 용어팀이 당신에게 연락할 것이고, 당신이 라이센스를 제출하도록 요청하거나, 해당 라이센스를 얻는데 적절한 사람과 연결해줄 것이다.

5.1.3 용어의 원천: 도입 vs 구축

OHDSI는 일반적으로 용어집을 새로 구축하기보다는 기존에 사용되는 용어집을 채택하는 것을 선호한다. 왜냐하면 (i) 많은 용어집이 이미 공동체의 관찰 데이터에 사용되어 왔으며 (ii) 용어집의 구성 및 유지 관리가 복잡하여 오랜 기간 동안 많은 이해관계자의 의견을 수렴해야 하기 때문이다. 이러한 이유로, 전담 조직은 생성, 사용 중단, 병합 및 분할의 수명 주기에 따라 용어집을 제공하고 있다. (5.2.10절 참조) 현재 OHDSI는 Type Concept (예를 들어, condition type concept)와 같은 내부 관리 용어만 생성하고 있다. 유일한 예외는 RxNorm Extension인데, RxNorm Extension은 미국 내부에서 사용되지 않는 약물들에 대한 용어집이다. (5.6.9절 참조)

5.2 개념 Concept

OMOP CDM의 모든 임상 기록은 개념 concept으로 표현되며, 이는 각 기록의 의미를 나타낸다. CDM에서 concept은 데이터 기록의 기본적인 구성 요소로써, 모든 테이블을 거의 예외 없이 완전 정규화한다. CONCEPT 테이블은 CDM 내의 사전과 같으며, CDM 내부의 모든 concept 은 이 사전 안에 정의되어 있다. 달리 말해 CONCEPT 테이블에 없다면, 해당 용어는 CDM 내의 concept ID를 갖지 못하고, 데이터를 정규화하여 기록하기 힘들어진다. (그림 5.2 참조)

그러므로 이 시스템은 포괄적이지 않으면 안 된다. 즉, 환자의 의료 경험 (예를 들어, 진단명, 시술, 약물 노출 등) 및 의료시스템의 일부 관리 정보 (예를 들어, 병원 방문, 관리 부위 등)과 관련된 모든 이벤트를 포괄할 만큼 충분히 많은 concept이 있어야 한다.

⁴<http://athena.ohdsi.org>

CONCEPT_ID	313217	Primary key
CONCEPT_NAME	Atrial fibrillation	English description
DOMAIN_ID	Condition	Domain
VOCABULARY_ID	SNOMED	Vocabulary
CONCEPT_CLASS_ID	Clinical Finding	Class in vocabulary
STANDARD_CONCEPT	S	Standard, Source of Classification
CONCEPT_CODE	49436004	Code in vocabulary
VALID_START_DATE	01-Jan-1970	
VALID_END_DATE	31-Dec-2099	Valid during time interval
INVALID_REASON		

Figure 5.2: OMOP CDM에서 vocabulary의 표준 표현. 위의 예는 심방세동의 SNOMED 코드에 대한 CONCEPT 테이블 레코드이다.

5.2.1 개념 ID

각각의 concept은 concept ID를 기본 키 primary key로 가진다. 무의미한 정수로 이루어진 이 ID는, 해당 원본 코드를 대신하여 CDM 테이블에 데이터를 기록하는 데 사용된다.

5.2.2 개념 이름 Concept name

각 concept에는 하나의 이름이 있다. 그 이름은 항상 영어로 되어있고, 원천 source으로부터 가져온다. 원천 용어에 둘 이상의 이름을 가지고 있으면, 가장 표현력이 높은 이름이 선택되고, 나머지 이름은 동일한 CONCEPT_KEY로 CONCEPT_SYNONYM 테이블에 저장된다. 영어 이외의 이름은 CONCEPT_SYNONYM에 기록되며, 적합한 language concept ID가 LANGUAGE_CONCEPT_ID 필드에 기록된다. 이름은 255자까지의 길이를 가지는데, 너무 긴 이름은 잘라내고 최대 1,000자까지 저장 가능한 다른 이름의 동의어로 기록된다.

5.2.3 도메인

각 concept에는 DOMAIN_ID 필드에 도메인이 할당되는데, 정수로 되어 있는 CONCEPT_ID와 달리 도메인의 경우 대소문자를 구분하면서 길이가 짧은 고유한 문자 ID이다. 이러한 각 도메인의 예로는 “Condition”, “Drug”, “Procedure”, “Visit”, “Device”, “Specimen” 등이 있다. 모호하거나 조합된 개념 pre-coordinated (combination) concept의 경우 combination 도메인에 속할 수 있으나, standard concept에는 (5.2.6절 참조) 항상 단일 도메인이 할당된다. 도메인은 또한 어떤 임상 사건 또는 임상 속성 등이 어떤 CDM 테이블과 필드에 기록되어야 하는지 알려준다. 도메인 할당은 Pallas

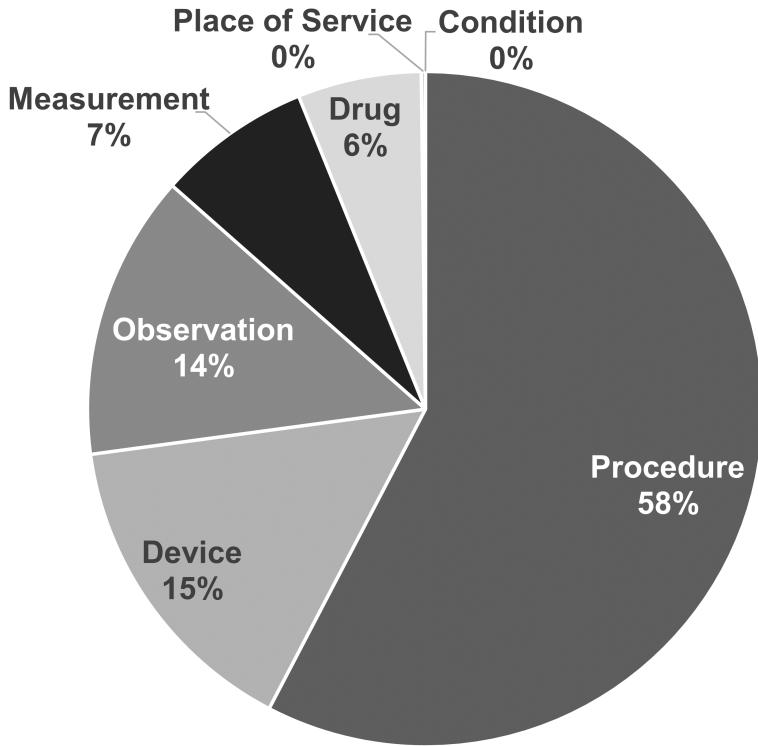


Figure 5.3: 시술 용어인 CPT4와 HCPCS의 도메인 할당 비율. 직관적으로, 이러한 용어집은 한 도메인의 코드와 개념을 포함하고 있어야 하지만, 실제로는 여러 도메인이 혼재되어 있다.

내에 경험적 지식을 이용한 용어 수집 중에 수행되는 OMOP 고유의 특징이다. Source vocabularies 는 서로 다른 도메인이 함께 혼재된 경우가 많으나, 그 정도는 각기 다르다 (그림 5.3 참조).

경험적 지식을 이용한 도메인 할당 방법은 도메인의 정의를 따라 진행한다. 이러한 정의는 CDM 의 테이블 및 필드 정의에서 파생된다 (4장 참조). 경험적 지식은 완벽하지 않으며, 불분명하다 (5.6 절의 “Special Situations” 참조). 만일, 도메인이 잘못 지정된 concept을 발견한다면, Forums 또는 CDM issue 게시판을 통하여, 문제점을 보고하고 개선하도록 도와주기 바란다.

5.2.4 용어집 Vocabularies

각 용어집에는 대소문자를 구분하는 고유한 문자 ID가 있으며, 일반적으로 용어집의 약어 이름을 쓰고, 대시는 생략한다. 예를 들자면, ICD-9-CM의 용어 ID(vocabulary ID)는 “ICD9CM”, KCD-7 의 용어 ID 는 “KCD7”이다. 현재 OHDSI가 지원하는 용어집은 111개로, 그중 78개가 외부 source 에서 채택되었고, 나머지는 OMOP 내부 용어집이다. 이러한 용어는 일반적으로 분기별 일정에 따라 갱신된다. 용어집의 버전은 VOCABULARY reference file에 따라 정의되어 있다.

5.2.5 개념 계층 Concept Classes

일부 용어집은 대소문자를 구분하는 고유한 문자 ID를 통해 내부의 코드나 concept을 분류한다. 예를 들어, SNOMED에는 “semantic tag”라고 불리는 33가지 concept class을 가지고 있다. “clinical finding”, “social context”, “body structure” 등과 같은 concept class은 concept의 수직적 구분을 의미한다. MedDRA 또는 RxNorm과 같은 다른 concept은 계층 내에 수평적인 수준을 분류하는 concept class 개념 계층을 가지고 있다. 다시 말해 concept class는 각 용어 vocabulary 내의 세부 분류로 이해할 수 있다. HCPCS와 같이 concept class가 없는 용어는 vocabulary ID ('HCPCS')가 그대로 concept class ID로 사용된다.

Table 5.1: 개념 계층에서 수평 및 수직 하위분류 원칙에 따른 용어집

	Concept class subdivision principle	Vocabulary
Horizontal		all drug vocabularies, ATC, CDT, Episode, HCPCS, HemOnc, ICDs, MedDRA, OSM, Census
Vertical		CIEL, HES Specialty, ICDO3, MeSH, NAACCR, NDFRT, OPCS4, PCORNET, Plan, PPI, Provider, SNOMED, SPL, UCUM
Mixed		CPT4, ISBT, LOINC
None		APC, all Type Concepts, Ethnicity, OXMIS, Race, Revenue Code, Sponsor, Supplier, UB04s, Visit

수평적 개념 계층을 사용하면 특정 계층 수준을 지정할 수 있게 해준다. 예를 들어, 약물 용어의 RxNorm에서 “성분 Ingredient” 개념 계층은 약물의 최상위층 계급 레벨을 정의한다 (역자 주: RxNorm 에서는 Ingredient 보다 상위의 class 가 없다). 수직적 모델에서 concept class 요소는 최상위에서 맨 아래까지 모든 class 중의 하나일 수 있다.

5.2.6 표준 개념

각 임상 사건의 의미를 나타내는 하나의 개념을 표준 개념 Standard concept이라고 부른다. 예를 들면, MESH 코드 D001281, CIEL 코드 148203, SNOMED 코드 49436004, KCD7 코드 I48, ICD9CM 코드 427.31 및 Read 코드 G573000은 모두 condition 도메인에서 “심방세동 atrial fibrillation”을 정의하지만, condition 데이터에서는 SNOMED concept만이 표준이고 질환을 의미할 수 있다. 나머지는 non-standard concept 또는 원천 개념 source concept으로, standard concept에 매핑되어야 한다. Standard concept은 STANDARD_CONCEPT 필드에 “S”라고 표시한다. 그리고 이러한 standard concept만이 “_CONCEPT_ID”로 끝나는 CDM 필드에 데이터를 기록하는 데 사용된다. CONCEPT 테이블의 INVALID_REASON은 NULL이어야 한다. (역자주: CONCEPT 테이블에서 standard concept의 INVALID_REASON 필드를 NULL이 아니라 공란("")으로 놔둘 경우, 표준 CDM 분석 파이프라인에서 오류가 발생하며, 이것은 가장 흔하게 발생하는 문제 중 하나이다.)

5.2.7 비표준 개념

비표준 개념 Non-Standard concept은 임상 사건을 나타내는 데 사용되지 않으나, 여전히 표준 용어집의 일부를 구성하고 원천 데이터에서 주로 발견된다. 이런 이유로, 이것을 “원천 개념 source concepts”이라고 부른다. source concept을 standard concept으로 변환하는 과정을 “매핑 mapping”이라고 부른다 (5.3.1절 참조). Non-standard concept 또는 source concept들은 각 테이블의 _SOURCE_CONCEPT_ID 필드에 저장될 수 있다. CONCEPT 테이블에서 Non-standard concept은 STANDARD_CONCEPT 필드에 값이 없다(NULL).

5.2.8 분류 개념

분류 개념 Classification concept은 표준이 아니므로 데이터를 나타내는 데 사용할 수 없다. 하지만 standard concept과 어우러져 계층 구조를 나타냄으로써, 계층 쿼리 hierarchical queries를 수행하는데 사용할 수 있다. 예를 들어, MedDRA 코드 10037908의 모든 하위 항목에 대한 쿼리를 사용하면 (MedDRA license를 받지 않는 사용자에게는 보이지 않음, 5.1.2절 액세스 제한 참조) 심 방세동 atrial Fibrillation에 대한 표준 SNOMED concept이 검색된다. (CONCEPT_ANCESTOR 테이블을 사용한 계층 쿼리는 5.4절 참조) - 그림 5.4 참조.

Concept을 standard, non-standard, classification으로 분류하는 작업은 각 도메인별로 용어 레벨로 결정된다. 이는 concept의 질, 내장된 계층구조 및 그 용어가 선언된 목적에 따라 행해진다. 또한, 한 용어의 모든 concept이 standard concept으로 사용되는 것은 아니다. 표준 용어의 지정 방식은 도메인마다 분리되어 있고, 각 concept은 유효 valid해야 하며 (5.2.10절 참조), 같은 의미에 대해 서로 다른 용어집에서 하나 이상의 concept이 경쟁하는 경우 우선순위에 따라 결정할 수 있다.

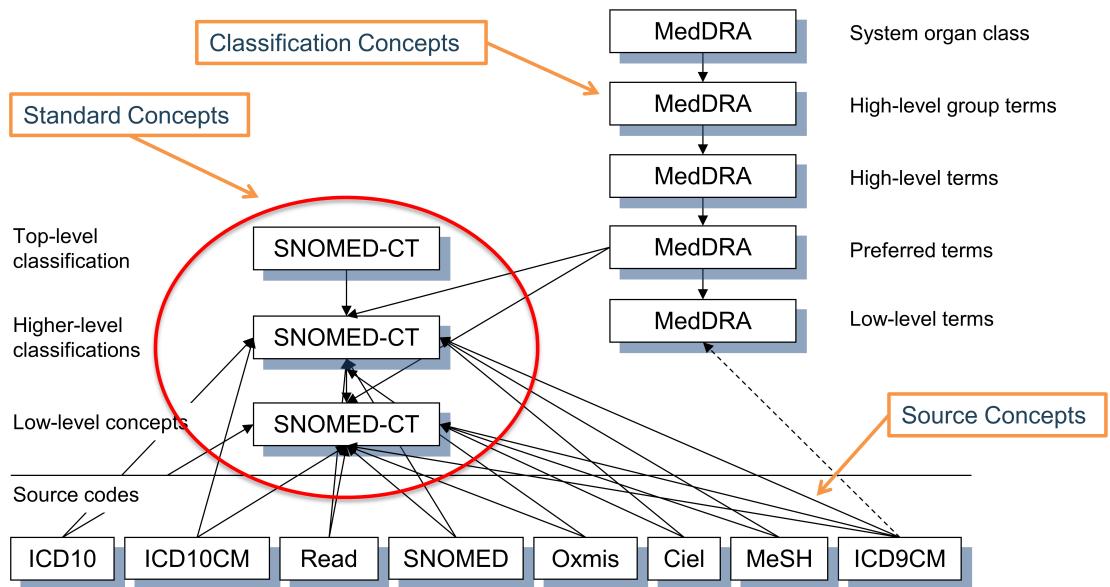


Figure 5.4: 표준, 비표준 원천 및 분류 개념과 condition 도메인에서의 계층적 관계. SNOMED는 대부분 standard condition concept (일부 ICDO3에서 파생된 종양학 관련 concept)에 사용되고, MedDRA concept은 계층 분류 concept에 사용되며, 그 외 다른 모든 용어집은 non-standard concept이나 혹은 계층 구조에 포함되지 않는 source concept을 가지고 있다.

즉, 하나의 표준 용어 standard vocabulary는 존재하지 않는다. SNOMED vocabulary는 광범위한 도메인에서 표준으로 채택하고 있지만, 그렇다고 SNOMED의 concept 모두가 standard concept인 것은 아니다. 도메인별 표준 용어의 예는 표 5.2를 참고하기 바란다.

Table 5.2: 표준, 비표준 및 분류 개념 할당에 활용되는 용어 목록

도메인	for standard	for classification	
	concepts	for source concepts	concepts
Condition	SNOMED, ICDO3	SNOMED Veterinary	MedDRA
Procedure	SNOMED, CPT4, HCPCS, ICD10PCS, ICD9Proc, OPCS4	SNOMED Veterinary, HemOnc, NAACCR	현재까지 없음

도메인	for standard concepts	for source concepts	for classification concepts
Measurement	SNOMED, LOINC	SNOMED Veterinary, NAACCR, CPT4, HCPCS, OPCS4, PPI	현재까지 없음
Drug	RxNorm, RxNorm Extension, CVX	HCPCS, CPT4, HemOnc, NAAACCR	ATC
Device	SNOMED	Others, currently not normalized	현재까지 없음
Observation	SNOMED	Others	현재까지 없음
Visit	CMS Place of Service, ABMT, NUCC	SNOMED, HCPCS, CPT4, UB04	현재까지 없음

5.2.9 개념 코드

개념 코드 Concept code 는 원천 용어 내에서 사용하는 식별자이다. 예를 들어, ICD9CM 또는 NDC 코드 자체는 해당 필드 (concept code)에 저장되고, OMOP 테이블은 CONCEPT 테이블의 concept ID를 외래 키로 사용을 한다. 그 이유는, concept code의 네임 스페이스가 용어집에 걸쳐 겹치기 때문이고, 동일한 코드가 완전히 다른 의미로 다른 용어에 존재할 수 있기 때문이다. (역자 주: 개념 코드는 한 용어집 내에서 유일한 코드로 사용되지만, 같은 코드가 다른 용어집에서 나타날 수 있으나 다른 의미로 사용될 수 있다) (표 5.3 참조)

Table 5.3: 같은 개념 코드 1001이 다른 용어집에서 다른 도메인, 다른 개념 클래스로 사용된다.

Concept ID	Code	Concept		Vocabulary ID	Concept Class
		Concept Name	Domain ID		
35803438	1001	Granulocyte colony-stimulating factors	Drug	HemOnc	Component Class
35942070	1001	AJCC TNM Clin T	Measurement	NAACCR	NAACCR Variable
1036059	1001	Antipyrine	Drug	RxNorm	Ingredient
38003544	1001	Residential Treatment - Psychiatric	Revenue Code	Revenue	Revenue Code
43228317	1001	Aceprometazine maleate	Drug	BDPM	Ingredient
45417187	1001	Brompheniramine Maleate, 10mg/mL injectable solution	Drug	Multum	Multum
45912144	1001	Serum	Specimen	CIEL	Specimen

5.2.10 수명 주기

하나의 용어집이 영원불변으로 고정되어 있는 경우는 드물다. 오히려, 코드와 concept이 더해지며 꾸준히 수정된다. OMOP CDM은 장기에 걸친 환자 데이터를 지원하기 위한 모델이므로, 과거에 사용되었지만, 지금은 더 이상 사용되지 않는 concept을 지원해야 할 뿐 아니라, 신설된 concept을 더하여 상황에 맞게 지원해야 한다. CONCEPT 테이블에는 사용 가능한 수명 주기 상태를 설명하는 세 개의 필드가 있다: VALID_START_DATE, VALID_END_DATE, INVALID_REASON 그 값은 각 개념의 수명 주기 상태에 따라 달라진다.

- 유효한 concept 및 새로운 concept

- 설명: 사용 중인 concept.
 - VALID_START_DATE: concept이 사용되기 시작한 날, 용어집에 concept을 통합한 날을 알지 못하거나, 알려지지 않은 경우 1970-1-1.
 - VALID_END_DATE: “지금은 활성화되어 있으나, 다음에는 무효가 될 수가 있음.”을 나타내는 규칙으로 2099-12-31을 설정.
 - INVALID_REASON: NULL
- **후속 코드 없이 더 이상 사용되지 않는 concept**
 - 설명: concept이 비활성화 상태여서 표준으로 사용할 수 없다. (5.2.6절 참조)
 - VALID_START_DATE: concept이 사용되기 시작한 날, 용어집에 concept을 통합한 날을 알지 못하거나, 알려지지 않은 경우 1970-1-1.
 - VALID_END_DATE: 사용 중단을 나타내는 과거의 날짜, 또는 해당 날짜를 알 수 없는 경우, concept이 누락되거나 비활성화된 용어집 갱신일
 - INVALID_REASON: “D”
 - **후속 코드가 있는 업그레이드 된 concept**
 - 설명: 비활성이지만, 후속 코드가 정의된 concept. 이는 일반적으로 중복 제거를 통해 제거된 concept.
 - VALID_START_DATE: concept이 사용되기 시작한 날, 용어집에 concept을 통합한 날을 알지 못하거나, 알려지지 않은 경우 1970-1-1.
 - VALID_END_DATE: 업그레이드를 나타내는 과거의 날짜, 또는 해당 날짜를 알 수 없는 경우, 업그레이드가 추가된 용어집 갱신일
 - INVALID_REASON: “U”
 - **다른 새로운 concept에 대한 재사용 코드**
 - 설명: 용어집은 새로운 개념을 위해 없어진 concept의 concept 코드를 재사용했다.
 - VALID_START_DATE: concept이 사용되기 시작한 날, 용어집에 concept을 통합한 날을 알지 못하거나, 알려지지 않은 경우 1970-1-1.
 - VALID_END_DATE: 사용 중단을 나타내는 과거의 날짜, 또는 해당 날짜를 알 수 없는 경우, concept이 누락되거나 비활성화로 설정된 용어집 갱신일
 - INVALID_REASON: “R”

일반적으로 concept 코드는 재사용하지 않는다. 하지만 이 규칙에서 벗어나는 몇몇 용어집으로 HCPCS, NDC, DRG가 있다. 이 용어집에서는 동일한 concept 코드가 같은 용어집 내에서 하나 이상의 concept에 사용된다. 하지만 CONCEPT_ID 값은 고유값을 유지한다. 재사용된 concept 코드는 INVALID_REASON 필드에 “R”로 표시되며 VALID_START_DATE부터 VALID_END_DATE 까지의 기간을 이용하여 concept 코드는 같지만 서로 다른 concept을 구별하는 데 사용해야 한다.

5.3 관계 Relationships

두 개념이 동일한 도메인 또는 vocabulary에 속하는지 여부와 관계없이 두 개념은 특정 관계를 맺을 수 있다. 관계의 특성은 CONCEPT_RELATIONSHIP 테이블의 RELATIONSHIP_ID 필드에서 대소문자를 구분하는 고유한 짧은 문자 ID로 표시한다. 각 관계에 대해서 상호가 바뀐 대칭 관계가 존재하여, CONCEPT_ID_1, CONCEPT_ID_2 필드의 내용이 교환되고, RELATIONSHIP_ID는 반대로 바뀌게 된다. 예를 들어, “Maps to” 관계는 “Mapped from”과 반대의 관계를 갖는다.

CONCEPT_RELATIONSHIP 테이블 레코드에는 수명 주기 필드인 RELATIONSHIP_START_DATE, RELATIONSHIP_END_DATE, INVALID_REASON이 있다. 그러나, INVALID_REASON = NULL인 유효한 기록만 ATHENA에서 이용할 수 있다. 비활성화된 관계는 Pallas 시스템 내에서 내부처리 용도로만 사용되도록 보관된다. RELATIONSHIP 테이블은 전체 relationship IDs 목록 및 그 반대의 목록과 함께 참조로 사용된다.

5.3.1 매핑 관계 Mapping Relationships

이러한 관계는 두 개의 relationship ID 쌍을 이용하여 Non-standard concept를 standard concept으로 변환하게 해준다. (표 5.4 참조).

Table 5.4: 매핑 관계의 유형.

Relationship ID 쌍	목적
“Maps to” and “Mapped from”	표준 개념에 매핑. standard concept은 자기 자신에게 매핑되고, non-standard concept은 standard concept으로 매핑된다. 대부분의 non-standard concept과 모든 standard concept은 한 standard concept과 이러한 관계를 맺는다. non-standard concept은 *_SOURCE_CONCEPT_ID에 저장되고 standard concept은 *_CONCEPT_ID 필드에 저장된다. 분류 개념은 매핑되지 않는다.
“Maps to value” and “Value mapped from”	MEASUREMENT와 OBSERVATION 테이블의 VALUE_AS_CONCEPT_ID 필드에 배치할 값을 나타내는 개념에 매핑.

이러한 매핑 관계를 사용하는 목적은 같은 concept 간의 교차를 통해 OMOP CDM에서 임상 사건이 표현되는 방식을 조화롭게 해주는 것이다. 이는 표준화된 용어가 이루어 낸 주요 성과이다.

“동등 개념 Equivalent concept”은 동일한 의미가 있으며, 중요한 것은 계층적으로 하위 개념 descen-

dant concept이 동일한 의미론적 공간을 가진다는 것이다. 다시 말해, 두 개의 concept의 의미와 그 범위가 정확히 일치한다는 것이다. Equivalent concept을 사용할 수 없고, 그 concept이 표준이 아닌 경우, 그 concept은 약간 더 넓은 의미를 가진 개념으로 매핑 (소위, “uphill-mappings”)된다. 예를 들어, ICD10CM W61.51 “Bitten by goose”에 해당하는 concept은 SNOMED의 standard condition concept에는 없다. 대신 SNOMED concept ID 217716004 “Peck by bird”에 매핑되지만, 그 새가 거위라는 의미는 잊게 된다. Up-hill 매핑은 정보의 구체성이 유실돼도 표준 연구 사례를 진행하는 데 문제없다고 간주할 때만 사용해야 한다.

일부 매핑은 source concept을 둘 이상의 standard concept에 연결한다. 예를 들어, ICD9CM의 070.43 “Hepatitis E with hepatic coma”는 SNOMED의 235867002 “Acute hepatitis E”뿐 아니라 SNOMED의 72836002 “Hepatic Coma”에도 매핑되어 있다. 그 이유는 기존의 source concept이 간염 hepatitis와 혼수 coma라는 두 가지 조건의 선 조합 pre-coordinated이기 때문이다. SNOMED에는 해당 조합이 없음으로, ICD9CM 레코드에 기록된 두 개의 레코드 (각각 매핑된 standard concept이 있는 레코드)가 생성된다.

“Maps to value” 관계는 Entity-Attribute-Value(EAV) 모델에 따라 OMOP CDM 테이블의 값을 나누기 위한 목적이 있다. 일반적으로 다음과 같은 경우이다:

- 테스트와 결과값으로 구성된 측정값
- 개인 또는 가족의 질병력
- 물질에 대한 알레르기
- 예방접종 필요

이런 상황에서 source concept은 속성(test or history)과 값(test result or disease)의 조합이다. “Maps to” 관계는 소스를 속성 concept에 매핑하고, “Maps to value” 관계는 값 concept을 매핑한다. (예로 그림 5.5 참조)

용어 매핑은 네트워크 연구를 수행하는 구성원의 노력을 도와주는 지원과, 무료로 제공되는 OMOP 표준 용어집의 또 다른 핵심 기능이다. 현재 KCD-7 source concept과 SNOMED standard concept 간의 매핑 관계 역시 ATHENA에 공개되어 있다. 매핑 관계는 외부 소스에서 파생되거나 용어팀에 의해 수작업으로 유지 관리된다. 사람이 하는 일이기에 매핑 관계는 완벽하지 않을 수 있다. 잘못 되거나 이의가 있는 매핑 관계를 발견한 경우 Forums 또는 CDM issue 게시판을 통해 알려줘서 프로세스를 개선하도록 돋는 것이 중요하다.

매핑 규칙에 대한 자세한 설명은 OHDSI Wiki에서 확인할 수 있다.⁵

⁵<https://www.ohdsi.org/web/wiki/doku.php?id=documentation:vocabulary:mapping>

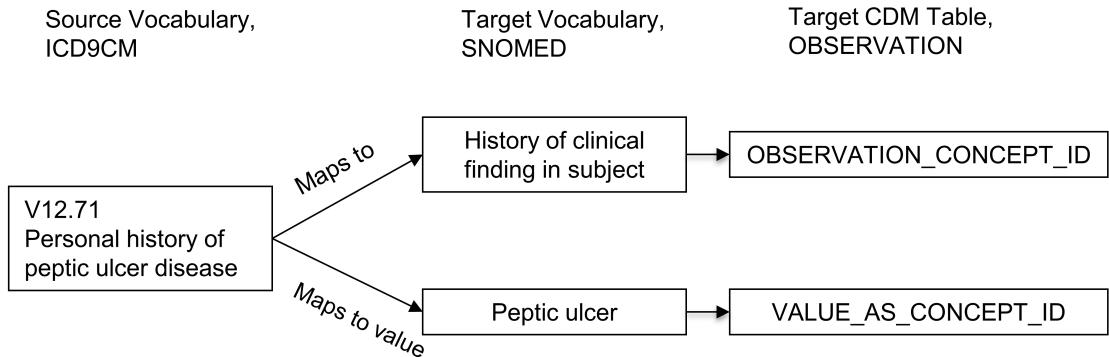


Figure 5.5: 원천 개념 source concept과 표준 개념 standard concept 사이의 일-대-다 매핑. 선 조합 pre-coordinated 개념은 두 가지 개념으로 나뉘는데, 하나는 속성(여기서는 임상 소견의 과거력)과 다른 하나는 값(소화성 궤양 peptic ulcer)이다. "Maps to" 관계는 measurement 또는 observation 도메인에 매핑되는 반면, "Maps to value" 개념에는 도메인 제한이 없다.

5.3.2 계층적 관계

계층을 나타내는 관계는 “Is a” – “Subsumes” 관계를 통해 정의된다. 계층적 관계는 자식 개념 child concept이 하나 이상의 추가 속성이나 더욱 정밀하게 정의된 속성에 더하여, 부모 개념 parent concept의 모든 속성을 갖도록 정의된다. 예를 들어, SNOMED의 49436004 “심방세동 atrial fibrillation”은 “Is a” 관계를 통해 SNOMED의 17366009 “심방 부정맥 atrial arrhythmia”에 연결된다. 두 concept 모두 부정맥 arrhythmia 형태를 제외하고 다른 속성은 동일하다, 즉, 한 concept에는 세동 fibrillation으로 정의되고, 다른 concept에서는 세동으로 정의되지 않는다. Concept에는 둘 이상의 부모와 둘 이상의 자식 개념이 있을 수 있다. 이 예에서 SNOMED의 49436004 “심방세동 atrial fibrillation”은 SNOMED의 40593004 “세동 fibrillation”과도 “Is-a” 관계를 맺는다.

5.3.3 서로 다른 용어집에서 온 개념 간의 관계

이 관계는 일반적으로 “Vocabulary A – Vocabulary B equivalent”的 유형으로, 기존 용어집 소스에서 제공되거나, OHDSI 용어팀에 의해 구축된다. 그것은 대략적인 매핑 역할을 할 수 있지만, 종종 잘 정리된 매핑보다는 관계의 정확도가 떨어진다. High-quality equivalence 관계 (예를 들어 Source – RxNorm equivalent)는 항상 “Maps to” 관계에 의해 복제된다.

5.3.4 동일 용어집 내 개념 간의 관계

한 용어집 내부의 관계는 일반적으로 용어집 제공자가 보급한다. 전체적인 설명은 OHDSI Wiki의 개별 용어집 아래의 용어집 설명서에서 찾을 수 있다. 이 중 다수는 임상 사건 간의 관계를 정의하여 정보 검색에 이용될 수 있다. 예를 들어, 요도 장애 disorders of the urethra는 “finding site of” 관계를 따라가면 찾을 수 있다. (표 5.5 참조)

Table 5.5: “요도 urethra”의 “Finding site of” 관계는 해부학적 구조에 있는 모든 조건을 나타낸다.

CONCEPT_ID_1	CONCEPT_ID_2
4000504 “Urethra part”	36713433 “Partial duplication of urethra”
4000504 “Urethra part”	433583 “Epispadias”
4000504 “Urethra part”	443533 “Epispadias, male”
4000504 “Urethra part”	4005956 “Epispadias, female”

이러한 관계의 품질과 포괄성은 기존 용어집의 질에 따라 다르다. 일반적으로, SNOMED와 같은 standard concept을 도출하는데 사용되는 용어집은 자료 분류와 구조화를 더 잘해주기 때문에, 내부 관계의 품질을 높여주는 데 도움이 된다.

5.4 계층 Hierarchy

도메인 내에서 표준 및 분류 체계는 계층 구조로 구성되며, CONCEPT_ANCESTOR 테이블에 저장된다. 이를 통해 개념과 모든 계층적 하위 항목을 큐리로 검색할 수 있게 된다. 이 하위 항목은 상위 항목과 같은 속성을 가지고 있지만, 추가로 정의된 것도 있다.

계층적 관계를 통해 연결된 모든 가능한 개념을 내포한 CONCEPT_RELATIONSHIP 테이블로부터 자동으로 CONCEPT_ANCESTOR 테이블이 생성된다. 이는 “Is a” – “Subsumes” (그림 5.6 참조), 또는 다른 관계를 서로 다른 용어집 간에 계층화하여 연결한다. 한 관계가 계층 구조 생성자에 참여할 것인가, 말 것인가의 선택은 RELATIONSHIP 참조 테이블의 DEFINIES_ANCESTRY라는 표시를 달아서 정의해준다.

상위와 하위 항목 사이의 단계 step 개수를 의미하는 ancestor degree는 MIN_LEVELS_OF_SEPARATION과 MAX_LEVELS_OF_SEPARATION으로 표현되고, 최단 또는 최장의 가능한 연결 정도를 정의해준다. 모든 계층적 관계가 분리 수준 계산에 동일하게 기여하는 것은 아니다. Degree를 구하기 위한 단계는 각 relationship ID에 대한 RELATIONSHIP 참조 테이블의 IS_HIERARCHICAL 표시 flag로 결정된다.

현재, 고품질의 포괄적인 계층구조는 약물 drug과 상태 condition 두 개의 도메인에만 존재한다. Procedure, measurement, observation 도메인은 부분적으로만 적용되며 현재 구축 중이다. 조상 개념 ancestry concept은 출처, 상품명 또는 다른 속성과 무관하게 특정 성분이나 약물 등급의 구성원을 가지는 모든 약물을 탐색할 수 있어서 특히 약물 도메인에서 유용하다.

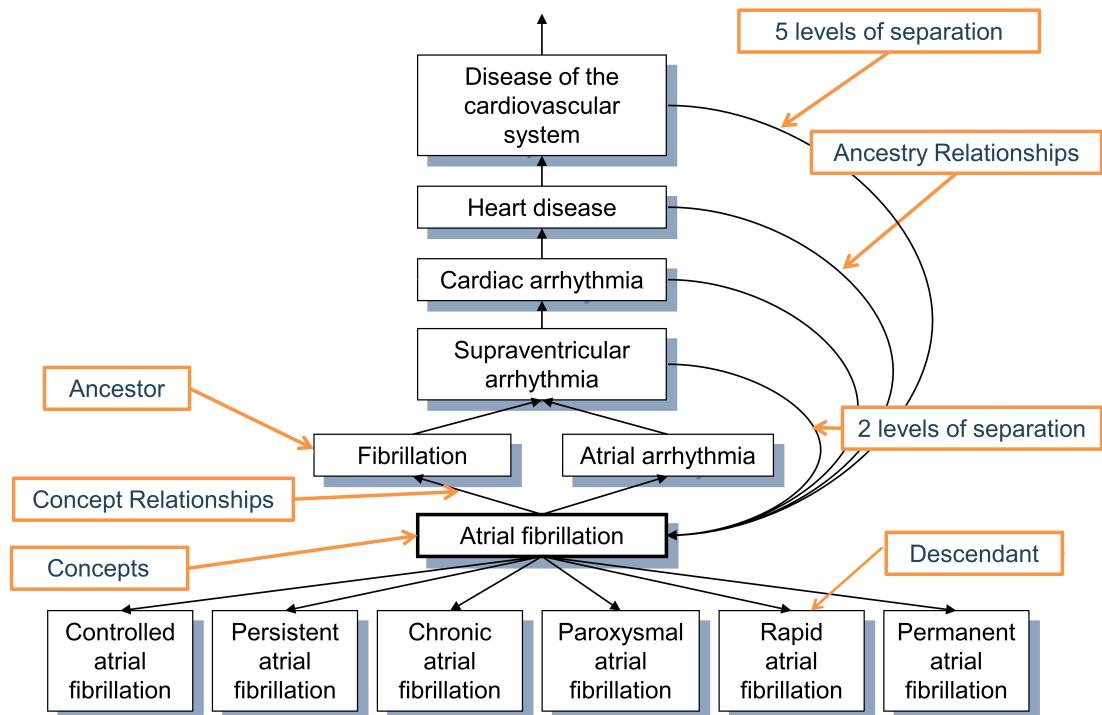


Figure 5.6: “심방세동 atrial fibrillation” condition의 계층. First degree ancestry는 “Is a”와 “Subsumes” 관계를 통해 정의되며, 모든 higher degree 관계는 추론되고, CONCEPT_ANCESTOR 표에 저장된다. 각 개념은 분리 수준이 0인 자기 자신의 하위 개념이다.

5.5 내부 참조 테이블

DOMAIN_ID, VOCABULARY_ID, CONCEPT_CLASS_ID (셋 모두 CONCEPT 레코드 안에 있는) 및 CONCEPT_RELATIONSHIP_ID(CONCEPT_RELATIONSHIP 안에 있는)는 모두 그들 자체 용어집에 의해 제어된다. 이는 4개의 참조 테이블인 DOMAIN, VOCABULARY, CONCEPT_CLASS 및 RELATIONSHIP에 정의되어 있으며, *_ID 필드를 기본 키 primary keys로 하여 더욱 자세한 *_NAME 필드 및 *_CONCEPT_ID 필드를 포함하는 CONCEPT 테이블에 대한 참조가 포함된 4개의 참조 테이블에 정의되어 있다. 이 중복 레코드의 목적은 정보 모델이 자동 탐색 엔진을 허용하도록 지원하기 위한 것이다.

VOCABULARY 테이블에는 기존 vocabulary의 source와 버전을 참조하는 VOCABULARY_REFERENCE 및 VOCABULARY_VERSION 필드가 포함되어 있다. RELATIONSHIP 테이블에는 추가 필드인 DEFINES_ANCESTRY, IS_HIERARCHICAL 및 REVERSE_RELATIONSHIP 필드가 있다. 후자는 한 쌍의 관계에 대해 반대의 counter relationship ID를 정의한다.

5.6 특수 상황

5.6.1 성별

OMOP CDM 및 표준 용어집의 성별은 출생 시의 생물학적 성별 (남성 및 여성)을 나타내지만, 종종 이외의 성별을 어떻게 정의하는지 의문이 제기된다. 이러한 사례는 OBSERVATION 테이블을 이용해야 하는데, 여기에 개인이 자체적으로 정의한 성별이 저장된다 (원천 데이터에 그런 정보가 포함된 경우).

5.6.2 인종과 민족

인종 race과 민족 ethnicity은 미국 정부가 지정한 정의를 따른다. 민족은 인종과 상관없이, Hispanic 또는 non-Hispanic 인구에서 분화된 세부 항목으로 구분한다. 인종은 공통적인 상위 5개 인종으로 나뉘며, 계층적 후손으로서 민족성을 가지고 있다. 혼혈 인종은 포함되지 않는다.

5.6.3 진단 코딩 체계 및 OMOP Conditions

ICD-9 또는 ICD-10, KCD-7과 같은 일반적으로 사용되는 코딩 체계는 적절한 진단 작업에 기반하여, 다소 잘 정의된 진단을 명시한다. KCD7 의 모든 코드에 condition 도메인이 할당되지는 않으며, Condition이 즉 KCD-7 또는 ICD-10은 아니다(즉, 의미론적 공간이 동일하지 않고 KCD-7

과 Condition 이 부분적으로 겹친다고 봐야한다). 예를 들어, conditions에는 진단이 도출되기 전에 기록된 징후와 증상도 포함되어있으며, ICD 코드에는 다른 도메인에 속하는 개념 (예를 들어, procedures)이 포함되어 있다.

5.6.4 시술 Procedure 코드 시스템

마찬가지로, HCPCS 및 CTP4와 같은 코딩 체계는 의료 시술의 목록으로 간주한다. 실제로, 이 체계는 의료비 청구 타당성을 선택하는 메뉴에 가깝다. 청구 코드의 많은 부분이 procedure 도메인 하에 포함되어 있지만, 또한 많은 concept 이 도메인 밖에 벗어나 있다.

5.6.5 기기 Device

기기 개념 Device concept은 원천 표준 개념에 사용될 수 있는 표준화된 코드 체계를 가지고 있지 않다. 많은 원천 데이터에서 기기는 코드화되어 있지 않고, 외부 코딩 체계에도 포함되지 않는다. 이러한 이유로 device에는 현재 사용 가능한 계층 시스템이 없다.

5.6.6 방문 및 서비스

방문 개념 visit concept은 의료 서비스의 성격을 정의한다. 많은 보건 시스템에서 병원과 같은 일부 기관과 같이, 물리적 구조를 나타내는 장소를 의미하며 '서비스 공간 Place of Service'이라고 명명한다. 다른 곳에서는 서비스 service라고 한다. Visit concept은 또한 국가마다 다르며, 정의하기 쉽지 않다. 진료 장소 care sites는 일반적으로 주로 방문하는 병원으로 정의할 수 있으나(예를 들어, XYZ 병원), 이를 이용하여 visit concept을 정의해서는 안 된다(예를 들어, XYZ 병원에서 조차 환자는 진료 목적이 아닌 방문을 할 수도 있기 때문이다).

5.6.7 제공자 및 전문의

모든 보건의료 제공자는 provider 도메인에 정의되어 있다. 이 제공자는 의사 및 간호사와 같은 의료 전문가일 수도 있지만, 검안사나, 신발 제조업자와 같은 비의료 서비스 제공자 non-medical provider 일 수도 있다. 전문의는 제공자인 “의사 Physician”의 하위 개념이다. 진료 장소는 전문성 specialty 을 보유할 수 없고, 주요 직원의 전문성에 의해 정의되는 경우가 많다 (“외과 Surgical department”).

5.6.8 특별한 요구사항이 있는 치료 영역

표준 용어는 포괄적인 방식으로 의료의 모든 측면을 다루고 있다. 하지만, 일부 치료영역에서는 특별한 요구를 가지고 있으며, 특별한 용어집이 필요하다. 예로, 종양학, 방사선학, 유전체학 같은 것이다. Special OHDSI 워크 그룹은 이러한 확장 기능을 개발한다. 결과적으로, OMOP 표준 용어

집은 통합 시스템을 구성하여 서로 다른 기원과 목적으로 생긴 concept이 모두 동일한 도메인 특화 계층 domain-specific hierarchy 내에 존재하게 된다.

5.6.9 약물 도메인 내의 표준 개념

Drug 도메인의 많은 개념은 미국 국립 의학 도서관 National Library of Medicine이 제공하는 용어 집인 RxNorm에서 제공하고 있다. 하지만, 미국 외 지역의 의약품은 미국에서 시판되는 성분, 형태, 강도의 조합 인지에 따라 RxNorm에 존재하지 않을 수도 있다. 미국 시장에 없는 약물은 OHDSI 용어팀에 의해 RxNorm 확장판, RxNorm Extension이라는 용어집으로 추가되며, 이것은 OHDSI에 의해 유일하게 생성된 거대한 용어집이다.

5.6.10 NULL의 특색

많은 용어집에는 정보가 없는 코드를 포함하고 있다. 예를 들어, 5개의 성별 개념인 8507 “Male”, 8532 “Female”, 8570 “Ambiguous”, 8551 “Unknown”, 8521 “기타” 중 앞의 2개인 8507 “Male”, 8532 “Female”만 표준이며, 나머지 3개는 매핑이 되어있지 않은 원천 개념이다. 표준 용어집에서는 왜 정보가 없는지에 대한 구분은 없다. 환자가 정보를 철회하거나, 결측값, 정의되지 않거나 표준화되지 않은 값 때문일 수도 있으며 COCEPT_RELATIONSHIP에 매핑 레코드가 없는 경우 일 수도 있다. 이러한 개념은 매핑되지 않으며, 이들의 standard concept으로의 기본 매핑은 concept ID=0 이다.

5.7 요약



- 모든 이벤트 및 관리되는 사건은 OMOP 표준 용어집에 개념 concept, 개념 관계 concept relationship 및 개념 상위 계층 concept ancestor hierarchy으로 표현된다.
- 이 중 대부분은 기존의 코딩 체계나 용어집에서 채택했지만, 일부는 OHDSI 용어팀에서 새롭게 조직했다.
- 모든 concept은 한 개의 도메인을 가지는데, 그런데 그 도메인은 concept으로 표현되는 그 사건이 CDM의 어떤 테이블에 저장될지를 결정한다.
- 다른 용어집에서 동등한 의미가 있는 개념은 그중 하나에 매칭되는데, 그것이 standard concept으로 지정된 개념이다. 다른 개념은 원천 개념이다.
- 매핑은 “Maps to”와 “Maps to value” 개념 관계 concept relationship를 통해 수행된다.
- 분류 개념 classification concept이라 불리는 추가 개념 계층이 있는데, 이는 비표준이지만, 원천 개념 source concept과 달리 계층 구조에 포함된다.
- 개념은 시간이 지남에 따라 수명 주기 life-cycle를 갖는다.

- 도메인 내의 개념은 계층으로 구성된다. 계층구조의 품질은 도메인마다 상이하며, 계층 구조 시스템의 완성을 위해 아직 진행 중이다.
- 실수나, 오류를 발견한 경우 커뮤니티에 참여할 것을 적극적으로 권장한다.

5.8 예제

전제조건

첫 연습에서는, ATHENA⁶ 또는 ATLAS⁷를 이용해서 표준용어집 내에서 개념을 찾아볼 필요가 있다.

Exercise 5.1. “위장관 출혈 gastrointestinal hemorrhage”에 대한 standard concept ID는 무엇인가?

Exercise 5.2. “위장관 출혈 gastrointestinal hemorrhage”에 대한 standard concept에 어떤 ICD-10CM 코드가 매핑되는가? 이 standard concept에 어떤 ICD-9CM 코드가 매핑되는가?

Exercise 5.3. “위장관 출혈 gastrointestinal hemorrhage”에 대한 standard concept에 해당하는 MedDRA 선호 용어는 무엇인가?

제시된 답변은 부록 E.2에서 찾을 수 있다.

⁶<http://athena.ohdsi.org/>

⁷<http://atlas-demo.ohdsi.org>

Chapter 6

추출 변환 적재 Extract Transform Load

Chapter leads: Clair Blacketer & Erica Voss

6.1 서론

원천 데이터에서 OMOP 공통 데이터 모델 Common Data Model(CDM)을 얻기 위해서는 추출 변환 적재 Extract Transform Load(ETL) 절차가 필요하다. 이 절차는 데이터를 CDM으로 변환하는 과정이며, 표준용어로의 매핑, SQL 코드를 이용한 자동화된 절차로 이루어지게 된다. ETL 절차는 원천 데이터가 갱신될 때마다 언제든지 재수행할 수 있게끔 반복할 수 있게 구축하는 것이 중요하다.

ETL을 진행하기 위해서는 다양한 작업이 필요하다. 몇 년 동안의 경험을 통해 우리는 4가지 주요 단계로 이루어진 모범사례를 개발하였다.

1. 데이터 전문가와 CDM 전문가가 함께 ETL을 설계할 것.
2. 의학 지식이 있는 사람이 용어 매핑을 할 것.
3. 엔지니어가 ETL을 수행할 것.
4. 모든 사람이 질 관리에 참여할 것.

이 장에서 우리는 각 단계를 세부적으로 살펴볼 것이다. 각 절차를 보조하기 위해 OHDSI 커뮤니티는 다양한 툴을 개발해 왔고, 이 툴에 대해서도 다룰 것이다. 마지막으로 CDM과 ETL의 유지에

관해 이야기하며 마무리할 것이다.

6.2 1단계: ETL 설계

ETL 설계와 ETL 수행을 명확하게 분리하는 것이 중요하다. ETL을 설계하는 것은 원천 데이터와 CDM 모두에 대한 넓은 지식이 필요하다. 반대로 ETL을 수행할 때는 ETL을 기술적인 측면에서 효율적으로 수행하는 방법에 대해 엔지니어에게 의존하게 된다. 만약 동시에 두 가지 모두를 진행하려 한다면, 전체적인 그림에 집중할 때보다 세부적인 사항에서 막히게 될 가능성이 높다.

ETL 설계를 위해 두 가지 밀접하게 연관된 툴을 개발하였다: White Rabbit과 Rabbit-in-a-Hat

6.2.1 White Rabbit

ETL 절차를 시작하기 위해서는 테이블, 필드, 내용을 포함한 데이터에 대한 이해가 필요하다. 하단의 링크에 White Rabbit에 대한 정보가 기록되어 있다. White Rabbit은 보건의료 종단 longitudinal 데이터베이스에서 OMOP CDM으로의 ETL 작업 준비를 도와주기 위한 소프트웨어이다. White Rabbit은 원천 데이터를 탐색하고 ETL 설계를 시작하기 위한 필수 정보에 대한 보고서를 생성해 준다. 모든 소스 코드, 설치 방법 및 설명서는 Github에서 확인할 수 있다.¹

범위와 목표

White Rabbit의 주요 기능은 원천 데이터에 대한 탐색을 수행하고, 테이블, 필드, 필드 값에 대한 세부적인 정보를 제공하는 것이다. 원천 데이터는 comma-separated(CSV) 텍스트 파일일 수도 있고, 데이터베이스 (MySQL, SQL Server, Oracle, PostgreSQL, Microsoft APS, Microsoft Access, Amazon RedShift)에 적재되어 있을 수도 있다. 탐색 과정에서 Rabbit-In-a-Hat 툴과 함께 쓴다면 ETL을 설계할 때 참고할 수 있는 보고서를 생성할 수 있다. White Rabbit은 다른 표준 데이터 프로파일링 소프트웨어와는 달리 개인 식별 정보 Personally Identifiable Information(PII)가 결과 데이터 파일에서 보이는 것을 방지한다.

절차 개요

원천 데이터를 탐색하기 위해 소프트웨어를 사용하는 일반적인 순서:

1. 결과를 내보낼 작업 폴더를 로컬 컴퓨터에 설정.
2. 데이터베이스 혹은 CSV 텍스트 파일과의 연결 및 연결 확인.
3. 탐색 대상 테이블 선택 및 탐색.
4. White Rabbit의 원천 데이터에 대한 정보 생성 및 내보내기.

¹<https://github.com/OHDSI/WhiteRabbit>.

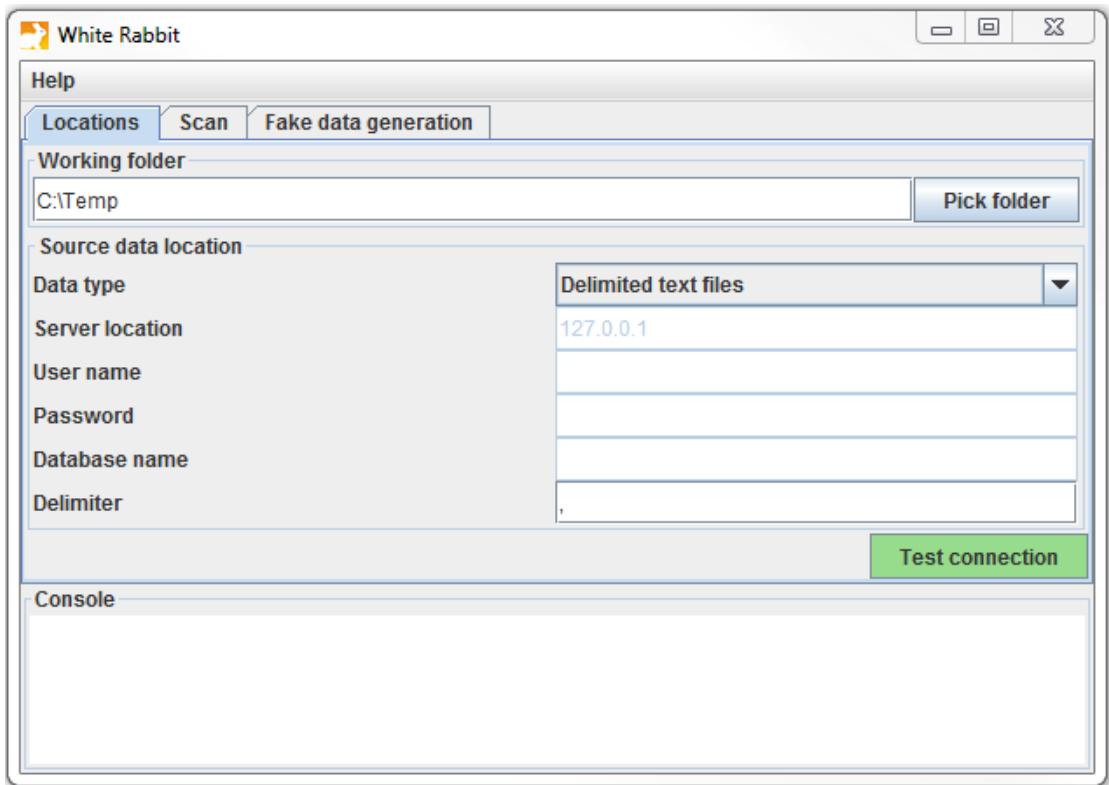


Figure 6.1: "Pick Folder" 버튼으로 White Rabbit 애플리케이션의 작업 폴더를 설정한다

작업 폴더 설정

White Rabbit 애플리케이션의 다운로드 및 설치 이후, 처음으로 할 일은 작업 폴더를 설정하는 것이다. White Rabbit이 생성하는 모든 파일은 설정한 로컬 폴더에 생성될 것이다. 그림 6.1에서 보이는 “Pick Folder” 버튼을 사용하여 탐색 문서가 저장될 로컬 환경을 탐색할 수 있다.

데이터베이스 연결

White Rabbit은 구분자로 구분된 텍스트 파일(CSV)과 다양한 데이터베이스 플랫폼을 지원한다. 다양한 필드에 대한 필요 항목의 설명을 보려면 마우스를 올려야 한다. 더욱 자세한 설명은 설명서에서 확인할 수 있다.

데이터베이스 테이블 탐색

데이터베이스에 연결한 이후에는 데이터베이스에 적재되어있는 테이블을 탐색할 수 있다. 탐색 과정은 ETL을 설계하는 데 도움이 되는 원천 데이터에 대한 정보를 담은 보고서를 생성할 수 있다. 그림 6.2에 보이는 Scan 탭의 “Add” (Ctrl + mouse click) 버튼을 눌러서 원천 데이터베이스의 각 테이블을 선택하거나, “Add all in DB” 누름으로써 모든 테이블을 자동으로 선택할 수 있다.

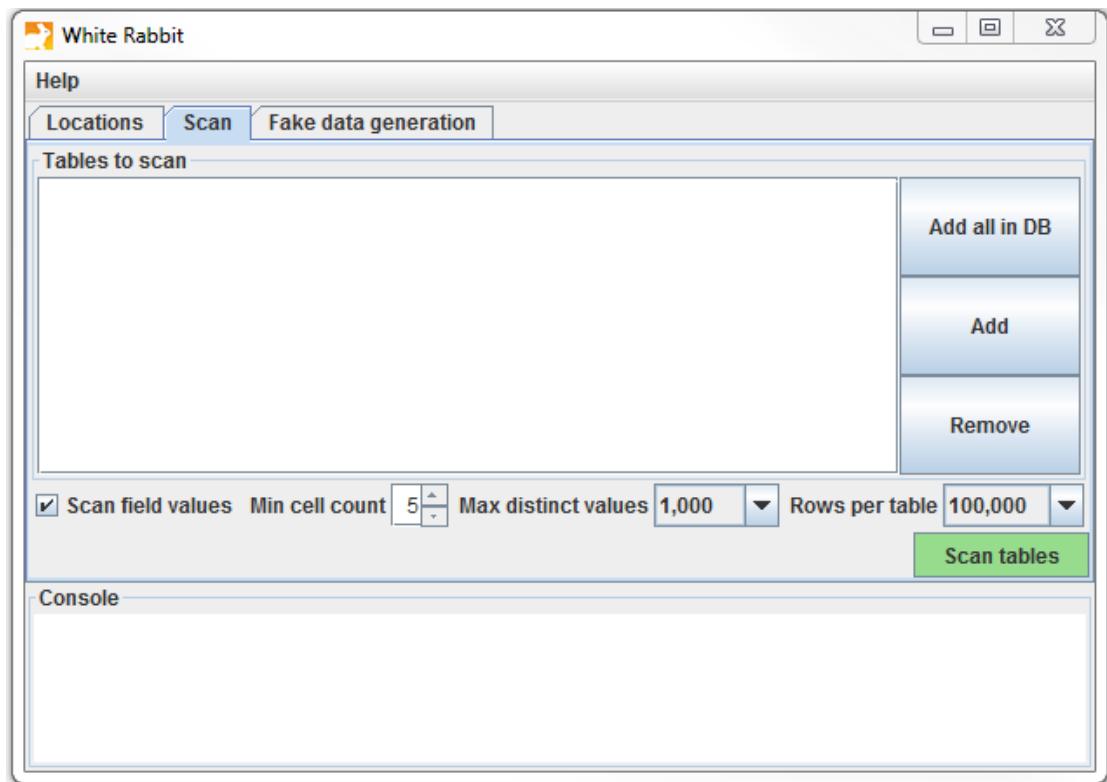


Figure 6.2: White Rabbit Scan 탭.

The screenshot shows a search report interface with a table of field statistics. The table has columns A through G. Rows 1 through 8 correspond to the fields of the dbo.allergies table, and rows 9 through 18 correspond to the fields of the dbo.careplans table. The 'Overview' tab is selected at the bottom.

A	B	C	D	E	F	G
1 Table	Field	Type	Max length	N rows	N rows checked	Fraction empty
2 dbo.allergies	start	date	10	3184	3184	0
3 dbo.allergies	stop	date	10	3184	3184	0.725188442
4 dbo.allergies	patient	varchar	36	3184	3184	0
5 dbo.allergies	encounter	varchar	36	3184	3184	0
6 dbo.allergies	code	varchar	9	3184	3184	0
7 dbo.allergies	description	varchar	24	3184	3184	0
8						
9 dbo.careplans	id	varchar	36	30199	30199	0
10 dbo.careplans	start	date	10	30199	30199	0
11 dbo.careplans	stop	date	10	30199	30199	0.057849598
12 dbo.careplans	patient	varchar	36	30199	30199	0
13 dbo.careplans	encounter	varchar	36	30199	30199	0
14 dbo.careplans	code	varchar	15	30199	30199	0
15 dbo.careplans	description	varchar	62	30199	30199	0
16 dbo.careplans	reasoncode	varchar	9	30199	30199	0.050796384
17 dbo.careplans	reasondescription	varchar	56	30199	30199	0.050796384
18						

Figure 6.3: 검색 리포트의 개요 탭 예시.

탐색에 사용될 몇 가지 옵션:

- “Scan field values”는 열에 어떠한 값이 나타나는지 보고 싶을 때 사용한다.
- “Min cell count”는 필드 값을 탐색할 때 쓰이는 옵션이다. 기본값은 5로 설정되어 있으며, 이는 원천 데이터에서 5번 이하로 나타나는 값은 보고서에 나타내지 않는 것을 의미한다. 각 데이터 사용 시 고유한 규칙에 따라 minimal cell count를 정해야 할 것이다.
- “Rows per table”는 필드 값을 탐색할 때 쓰이는 옵션이다. 기본값으로 White Rabbit은 테이블에서 무작위로 100,000개의 행을 선택하여 탐색할 것이다.

모든 옵션이 설정된 이후에는 “Scan tables”을 누르면 된다. 탐색이 완료된 이후에는 보고서가 작업 폴더에 생성될 것이다.

탐색 보고서의 이해

탐색이 완료된 이후에는 선택된 작업 폴더에 엑셀 파일이 생성될 것이며, 엑셀 파일에는 스캔한 각 테이블에 대한 하나의 탭과 개요 탭이 생성된다. 개요 탭은 탐색한 모든 테이블이며, 각 테이블의 필드, 각 필드의 데이터 타입, 필드의 최대 길이, 테이블의 행의 수, 탐색한 행의 수, 그리고 얼마나 많은 필드가 비어있는지 보여준다. 그림 6.3은 개요 탭의 예시를 보여준다.

각 테이블의 탭은 각각의 필드, 필드의 값, 그리고 값의 빈도를 나타낸다. 각 원천 테이블의 컬럼은 엑셀에서 두 개의 열 column으로 생성된다. 하나는 탐색 시 설정한 “Min cell count” 보다 자주

	A	B
1	Sex	Frequency
2	2	61491
3	1	35401
4	List truncated...	

Figure 6.4: 단일 열에 대한 예제 값.

나타난 값을 보여준다. 예를 들어, 성별 열이라면, ‘남성’ ’여성’ 등과 같이 열 내부에 있는 값들이 나올 것이다. 만약 고유한 값 목록 중 일부가 이러한 기준에 의해 보여지지 않는다면, 목록의 마지막 값에는 “List truncated” 가 나타날 것이다. 이는 하나 혹은 그 이상 값의 빈도가 “Min cell count” 보다 적어 사용자에게 보여지지 않음을 의미한다. 각각의 값 옆에는 빈도를 나타내는 두 번째 열이 있다 (표본에서 값이 발생하는 횟수). 이 두 열(고유한 값과 빈도수)은 작업 workbook에 프로파일링된 테이블의 모든 원천 변수에 대해 반복돼서 나타난다.

보고서는 원천 데이터에 무엇이 있는지를 강조함으로써 데이터를 이해하는 데 강력한 도움을 준다. 예를 들면, 그림 6.4에 나타난 결과가 탐색 된 테이블 열 중 하나인 “Sex”에 반환될 경우, 우리는 각각 61,491번과 35,401번 나타난 공통된 값(1과 2)이 있음을 알 수 있다. White Rabbit가 1을 남성으로, 2를 여성으로 정의하지는 않을 것이다. 데이터 소유자가 일반적으로 원천 시스템의 고유한 원천 코드에 대해 정의해야 한다. 하지만 이 두 가지 값 (1 & 2)은 데이터에 있는 유일한 값이 아니기 때문에 우리는 잘린 목록을 확인해야 한다. 이 값은 (“Min cell count” 정의에 따라) 매우 낮은 빈도로 나타나게 되고, 종종 부정확하거나 매우 의심스러운 값으로 표현된다. ETL 수행을 계획할 때 우리는 높은 빈도의 성별 개념으로써 1과 2만 다루는 것이 아니라, 열에 존재하는 낮은 빈도의 값도 고려해야 한다. 예를 들어 만약 낮은 빈도의 성별이 “NULL”일 경우 ETL 진행 시 이러한 데이터에 대해 어떻게 처리할 것인지 확실히 해야 한다.

6.2.2 Rabbit-In-a-Hat

White Rabbit과 함께 우리는 원천 데이터에 대한 분명한 그림을 그릴 수 있다. 또한, 우리는 CDM에 대한 전체 명세서를 알고 있다. 이제 우리는 하나에서 다른 하나로 넘어갈 로직을 정의해야 한다. 이 설계 활동은 원천 데이터와 CDM 모두에 대한 온전한 지식을 요구한다. White Rabbit 소프트웨어와 함께 사용되는 Rabbit-in-a-Hat 툴은 명확하게 이 분야의 전문가를 위해 개발되었다. 일반적으로 ETL 설계팀은 회의실에 앉아 Rabbit-in-a-Hat을 프로젝터 화면으로 같이 보면서 작업을 한다. 첫 번째로 테이블 간의 매핑은 협력적으로 결정될 수 있으며, 그 후에는 필드 간의 매핑이 설계되는 동시에 어떠한 값을 변환시킬지 로직을 정의할 수 있다.

범위와 목표

Rabbit-In-a-Hat은 White Rabbit의 탐색 문서를 읽고 시각화하기 위해 설계되었다. White Rabbit은 원천 데이터에 대한 정보를 생성하는 반면, Rabbit-In-a-Hat은 그 정보를 사용하고 그래픽 사용자 인터페이스를 통하여 사용자가 원천 데이터의 테이블과 열을 CDM으로 연결될 수 있게 해준다. Rabbit-In-a-Hat은 ETL 절차에 대한 문서를 생성해주지만 ETL을 위한 코드는 생성하지 않는다.

절차 개요

소프트웨어를 이용한 ETL 문서 생성을 위한 일반적인 순서:

1. White Rabbit이 완료한 탐색 결과
2. 탐색 결과 열기. 인터페이스가 원천 테이블과 CDM 테이블을 보여줌
3. 원천 테이블의 정보와 상응하는 CDM 테이블을 연결
4. CDM 테이블에 상응하는 각 원천 테이블에 대해서 세부적인 원천 열과 CDM 열의 연결을 정의
5. Rabbit-In-a-Hat 작업을 저장하고 MS 워드 문서로 내보내기

ETL 로직 작성

일단 Rabbit-In-a-Hat 내의 White Rabbit 탐색 보고서를 확인한다면, 원천 데이터를 OMOP CDM으로 변환하는 설계와 로직 작성을 시작할 준비가 되었다. 하나의 예시로써 하단의 장에서 Synthea² 데이터베이스의 일부 테이블의 변환을 보여줄 것이다.

ETL의 일반적인 흐름

CDM이 사람 중심의 모형이기 때문에 제일 먼저 PERSON 테이블 매핑으로 시작하는 것이 좋다. 모든 임상 사건과 관련 있는 테이블 (CONDITION_OCCURRENCE, DRUG_EXPOSURE, PROCEDURE_OCCURRENCE 기타 등)은 PERSON 테이블의 person_id를 참조하기에 PERSON 테이블에 대한 로직을 먼저 작성하는 것이 나중을 위해 좋다. PERSON 테이블을 변환한 다음에는 OBSERVATION_PERIOD를 변환하는 것이 좋은 선택이다. CDM 데이터베이스의 개인은 최소 하나 이상의 OBSERVATION_PERIOD를 가져야 하고, 일반적으로 한 사람에 대한 모든 사건은 이 관찰 기간 내에 들어온다. PERSON과 OBSERVATION_PERIOD 테이블이 완료되면 보통 PROVIDER, CARE_SITE, 그리고 LOCATION과 같은 테이블이 다음 대상이 된다. 임상 테이블 이전에 마지막으로 로직을 작성해야 하는 테이블은 VISIT_OCCURRENCE이다. 한 사람이 환자로서의 여정에서 대부분의 사건이 방문할 때 발생하기 때문에 종종 모든 ETL 과정에서 가장

²Synthea™ is a patient generator that aims to model real patients. Data are created based on parameters passed to the application. The structure of the data can be found here: <https://github.com/synthetichealth/synthea/wiki>.

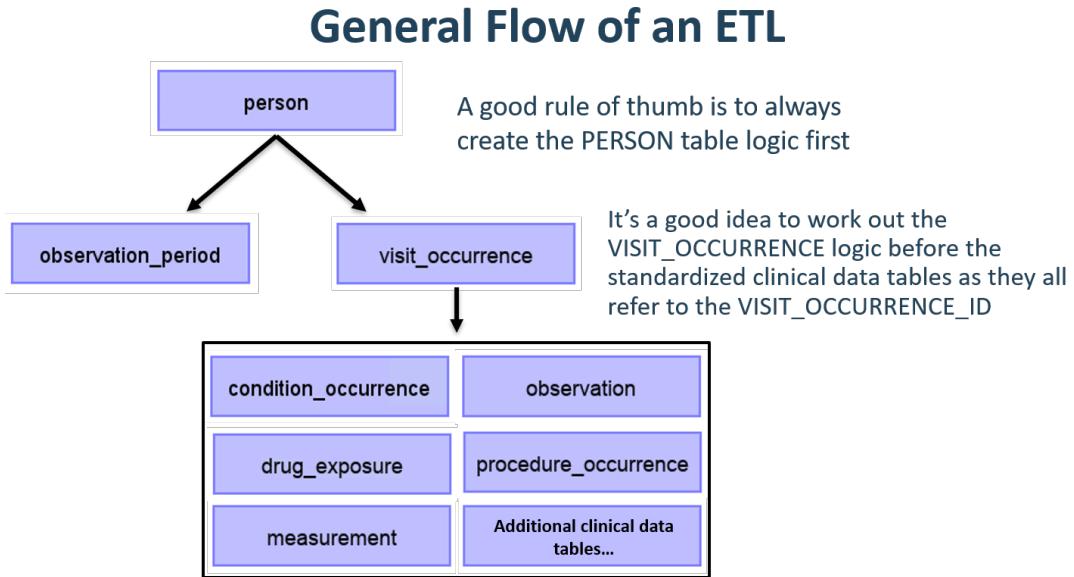


Figure 6.5: ETL의 일반적인 흐름과 먼저 매핑할 테이블.

복잡하고 중요한 부분이기도 하다. 일단, 이 테이블이 완료되면 어떤 CDM 테이블을 어떤 순서대로 매핑할지는 선택하기 나름이다.

CDM 변환 과정에서 종종 중간 테이블을 만들 필요가 있을 수 있다. 올바른 VISIT_OCCURRENCE_ID를 해당 사건에 부여하거나 아니면 원천 코드를 표준 코드로 매핑하는 경우일 수도 있다 (이 단계는 종종 매우 느리게 진행된다). 중간 테이블은 100% 허용되고 장려된다. 하지만 이러한 중간 테이블이 변환이 완료된 이후에도 남아있거나 사용하는 것은 추천하지 않는다.

매핑 예시: PERSON 테이블

Synthea 데이터 구조에서 환자 테이블은 20개의 열을 갖고 있지만 그림 6.6에서 보이는 것처럼 모든 열이 PERSON 테이블에 필요한 것은 아니다. 이런 일은 매우 흔한 일이고 문제가 되지 않는다. 이 예시에서는 환자 이름, 운전면허번호, 여권 번호 등 Synthea의 환자 테이블의 많은 데이터 포인트가 CDM PERSON 테이블에 사용되지 않는 것을 알 수 있다.

하단의 표 6.1은 Synthea의 환자 테이블이 CDM PERSON 테이블로 변환되는 로직을 보여준다. ‘Destination Field’는 CDM 데이터의 어디에 매핑되는지를 나타낸다. ‘Source field’는 원천 테이블 (예시에서는 환자 테이블)의 어느 열에서 CDM의 열로 변하는지 나타낸다. 마지막으로, ‘Logic & comments’는 로직에 대한 설명을 의미한다.

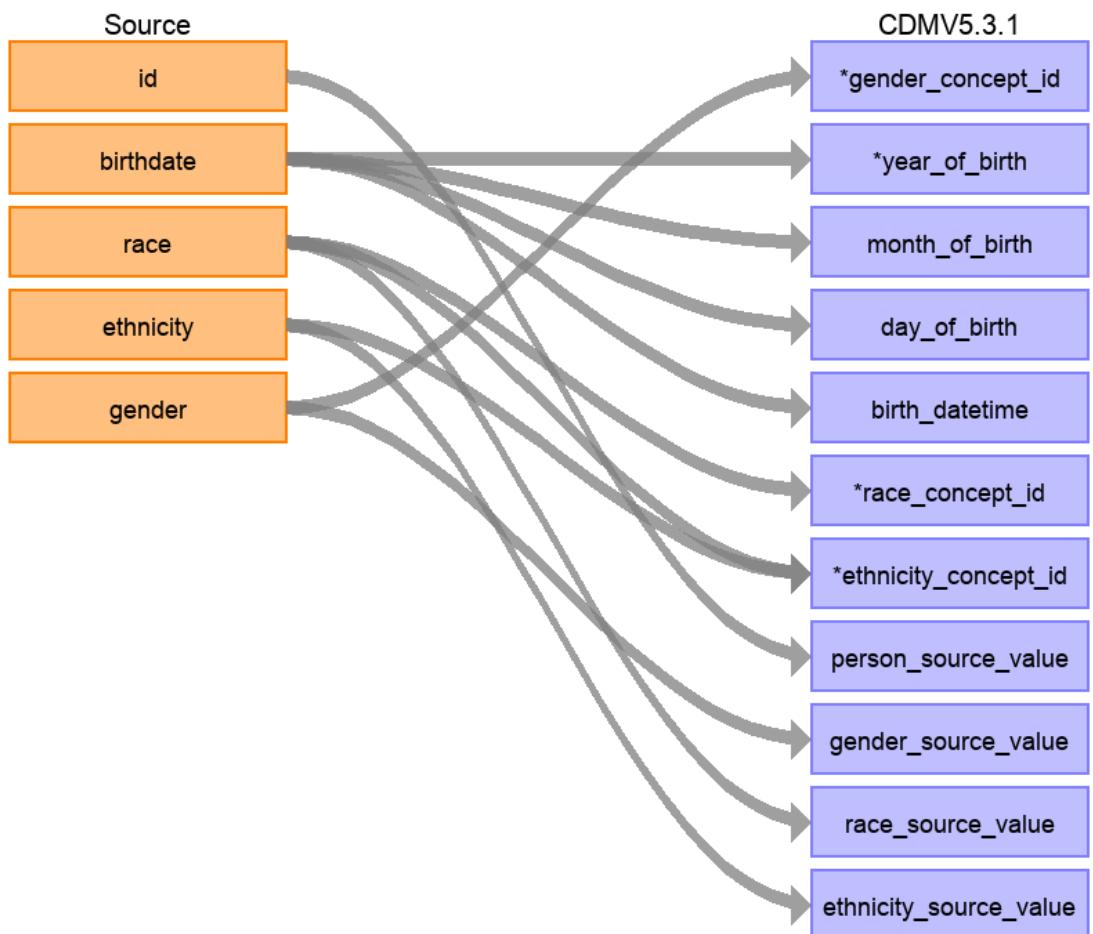


Figure 6.6: CDM PERSON 테이블에 Synta Patients 테이블 매핑.

Table 6.1: Synthea 환자 테이블을 CDM PERSON 테이블에 변환하기
위한 ETL 로직

Destination Field	Source field	Logic & comments
PERSON_ID		Autogenerate. The PERSON_ID will be generated at the time of implementation. This is because the id value from the source is a varchar value while the PERSON_ID is an integer. The id field from the source is set as the PERSON_SOURCE_VALUE to preserve that value and allow for error-checking if necessary.
GENDER_CONCEPT_ID	gender	When gender = 'M' then set GENDER_CONCEPT_ID to 8507, when gender = 'F' then set to 8532. Drop any rows with missing/unknown gender. These two concepts were chosen as they are the only two standard concepts in the gender domain. The choice to drop patients with unknown genders tends to be site-based, though it is recommended they are removed as people without a gender are excluded from analyses.
YEAR_OF_BIRTH	birthdate	Take year from birthdate
MONTH_OF_BIRTH	birthdate	Take month from birthdate
DAY_OF_BIRTH	birthdate	Take day from birthdate
BIRTH_DATETIME	birthdate	With midnight as time 00:00:00. Here, the source did not supply a time of birth so the choice was made to set it at midnight.
RACE_CONCEPT_ID	race	When race = 'WHITE' then set as 8527, when race = 'BLACK' then set as 8516, when race = 'ASIAN' then set as 8515, otherwise set as 0. These concepts were chosen because they are the standard concepts belonging to the race domain that most closely align with the race categories in the source.

Destination Field	Source field	Logic & comments
ETHNICITY_	race	When race = 'HISPANIC', or when ethnicity in ('CENTRAL_AMERICAN', 'DOMINICAN', 'MEXICAN', 'PUERTO_RICAN', 'SOUTH_AMERICAN') then set as 38003563, otherwise set as 0. This is a good example of how multiple source columns can contribute to one CDM column. In the CDM ethnicity is represented as either Hispanic or not Hispanic so values from both the source column race and source column ethnicity will determine this value.
LOCATION_ID		
PROVIDER_ID		
CARE_SITE_ID		
PERSON_SOURCE_	id	
VALUE		
GENDER_SOURCE_	gender	
VALUE		
GENDER_SOURCE_		
CONCEPT_ID		
RACE_SOURCE_	race	
VALUE		
RACE_SOURCE_		
CONCEPT_ID		
ETHNICITY_	ethnicity	In this case the ETHNICITY_SOURCE_VALUE will have more granularity than the ETHNICITY_CONCEPT_ID.
SOURCE_VALUE		
ETHNICITY_		
SOURCE_CONCEPT_ID		

Synthea 데이터의 CDM으로의 변환에 대한 더 자세한 설명은 전체 명세서를 참고하면 된다.³

6.3 2단계: 용어 매핑 생성

점점 더 많은 원천 코드가 OMOP 용어에 추가되고 있다. 이것은 CDM으로 변환된 데이터의 용어 체계가 이미 CDM에 포함되거나 매핑되었을 수도 있다는 것을 의미한다. OMOP 용어의 VOCABULARY 테이블을 통해 어떤 용어가 포함되었는지 확인할 수 있다. 비표준인 원천 용어 (예를 들어 KCD-7 codes)에서 표준용어 (예를 들어 SNOMED codes)로의 매핑을 확인하려면 CONCEPT_RELATIONSHIP 테이블 내의 relationship_id = “Maps to” 인 값을 찾으면 확인할 수 있다. 예를 들면 KCD-7 코드 ‘I21’ (“Acute Myocardial Infarction”)의 표준용어 ID를 확인하기 위해 다음과 같은 SQL을 사용할 수 있다:

```
SELECT concept_id_2 standard_concept_id
FROM concept_relationship
INNER JOIN concept source_concept
ON concept_id = concept_id_1
WHERE concept_code = 'I21'
AND vocabulary_id = 'KCD7'
AND relationship_id = 'Maps to';
```

STANDARD_CONCEPT_ID

312327

하지만 가끔은 원천 데이터가 OMOP 용어집에 없는 용어 시스템을 사용할 수도 있다. 이러한 경우에는 원천 용어 시스템을 표준 개념으로 변환하는 매핑을 정의하여야 한다. 하지만 원천 용어 시스템에 많은 수의 용어가 있으면 용어 매핑이 어려울 수도 있다. 이를 쉽게 진행하기 위한 몇 가지 참고사항이 있다.

- 가장 높은 빈도의 코드에 집중. 절대 쓰이지 않는 용어나 거의 안 쓰이는 용어는 실제 연구에서도 쓰이지 않기 때문에 큰 노력을 들여서 매핑을 진행할 필요가 없다.
- 가능하면 기존의 정보를 활용. 예를 들어 많은 국가 약물 코딩 시스템은 이미 ATC로 매핑되어 있다. 비록 ATC가 많은 목적에 대해 세부적으로 부합하지는 않지만, ATC와 RxNorm의 관계를 통해 어떤 RxNorm 코드가 사용되는지 추측할 수는 있다.
- Usagi를 사용한다.

³<https://ohdsi.github.io/ETL-Synthea/>

6.3.1 Usagi

Usagi는 용어 매핑 절차를 도와주는 툴이다. Usagi는 코드 설명의 단어 유사도에 기반하여 매핑을 추천할 수 있다. 만약 원천 코드가 외국어로만 확인 가능하다면, Google Translate⁴를 통해 종종 해당 용어의 훌륭한 영어 번역을 확인할 수 있다. Usagi의 자동 추천이 정확하지 않을 경우 사용자가 직접 적절한 목표 개념을 찾을 수 있다. 최종적으로 사용자는 어떤 매핑이 ETL에 사용될 수 있는지 지정할 수 있다. Usagi는 GitHub⁵를 통해 사용할 수 있다.

범위와 목표

매핑이 필요한 원천 용어를 Usagi로 불러올 수 있다 (만약 원천 용어 시스템이 영어가 아닐 경우, 추가 번역이 필요하다). 단어 유사도 접근법은 원천 용어와 OMOP 표준 용어 개념 concept을 연결하기 위해 필요하다. 하지만 이러한 용어 매핑은 수동적으로 검토해야 하고, Usagi는 이를 수행하기 위한 인터페이스를 제공한다. Usagi는 용어에 표준 개념만을 제안한다.

절차 개요

소프트웨어를 사용하기 위한 일반적인 순서:

1. 원천 시스템 (“원천 용어”)로부터 OMOP 표준 개념 standard concept으로의 매핑을 진행하고 싶은 용어를 업로드.
2. Usagi 단어 유사도 접근법을 이용하여 OMOP concept으로의 매핑을 진행.
3. Usagi 인터페이스를 활용하여 제안된 매핑을 확인하고 필요할 경우 개선. 용어 시스템과 의학 지식이 있는 사람이 리뷰를 진행하는 것이 바람직함.
4. 매핑 결과를 용어의 SOURCE_TO_CONCEPT_MAP으로 내보냄.

Usagi로 원천 코드 가져오기

원천 시스템에서 CSV나 엑셀(.xlsx) 파일로 원천 용어를 내보낸다. 이때 파일은 원천 용어 코드와 영어로 된 설명열이 있어야 하고, 추가적인 정보 역시 더할 수 있다 (예를 들어 약물 용량, 번역되었을 경우 원래 언어로의 코드 설명). 게다가 원천 용어에 대한 정보뿐만 아니라, 빈도 역시 어떤 용어를 먼저 매핑해야 할지 정하는 데 도움이 되기 때문에 포함하는 것이 좋다 (예를 들어 1,000 개의 원천 용어 중 100개만 실제 시스템에 정말로 사용되는 경우). 만약 원천 용어 설명을 영어로 번역해야 할 경우, Google Translate가 도움이 될 수 있다.

참고: 원천 용어는 도메인 (다시 말하면 약물 drug, 시술 procedure, 상태 condition, 관찰 observations) 별로 분류되어야 하며, 하나의 파일로 묶여서는 안 된다.

⁴<https://translate.google.com/>

⁵<https://github.com/OHDSI/Usagi>

The screenshot shows the 'Import codes from ICPC2SNOMED.csv' dialog box. The main table lists various medical codes and their English and Dutch terms. Below the table, there's a 'Column mapping' section where columns are mapped from the source to UMLS lookup and then to Dutch terms. On the right, there's a 'Filters' section with several checkboxes for filtering by user selected concepts, standard concepts, source terms, concept class, vocabulary, and domain.

Code	English term	Count	UMLS lookup	Dutch term
A99.00	General disease	5774012		Andere gegeneraliseerde/niet gespecificeerde ziekte(n)
K86.00	Hypertension uncomplicated	3987206		Essentiële hypertensie zonder orgaanbeschadiging
R44.00	Preventive Imunisations/Medications	3702922		Immunisatie/preventieve medicatie
T90.02	Diabetes mellitus type 2	2275799		Diabetes mellitus type 2
R05.00	Cough	1268829	4158493	Hoesten
R74.00	Upper respiratory infection acute	1061504		Acute infectie bovenste luchtwegen
A29.00	General symptom/complaint other	1035167		Andere algemene symptomen/klachten
L03.00	Low back symptom/complaint	998249		Lage rugpijn zonder uitsraling [ex. L86]
U71.00	Cystitis/urinary infection other	970719		Cystitis/urinegenital infections
A60.00	Results Tests/Procedures	903897		Uitslag onderzoek/verrichting
R97.00	Allergic rhinitis	892467	257007	Hooikoorts/allergische rhinitis
A97.00	No disease	868585		Geen ziekte

Column mapping

Source code column	Code
Source name column	English term
Source frequency column	Count
Auto concept ID column	UMLS lookup
Additional info column	Dutch term
Additional info column	

Filters

- Filter by user selected concepts / ATC code
- Filter by concept class:
- Filter standard concepts
- Filter by vocabulary:
- Include source terms
- Filter by domain: Condition

Cancel Import

Figure 6.7: Usagi 원천 용어 코드 입력화면.

파일로부터 원천 코드를 Usagi로 올린다 -> 코드 메뉴를 가져온다. 여기서 “Import codes ...”는 그림 6.7과 같이 보일 것이다. 이 그림에서 원천 코드 용어는 네덜란드어이고, 영어로 번역되어있다. Usagi는 표준용어로의 매핑을 위해 영어 번역을 이용할 것이다.

“Column mapping” 부분 (왼쪽 아래)은 Usagi가 불러온 테이블을 어떻게 사용할 것인지 정하는 단계이다. 마우스를 끌어다 놓으면, 각 열을 정의하는 팝업창이 나타날 것이다. Usagi 자체가 원천 용어를 용어 개념 Vocabulary concept 코드에 연결하는 정보로써 “Additional info” 열을 사용하지 않을 것이다. 하지만 이 추가적인 정보는 개인이 원천 코드 매핑을 검토하는 데 도움을 줄 수 있기에 포함되어야 한다.

마지막으로 “Filters” 부분 (아래 오른쪽)에서 Usagi로 매핑할 때의 몇 가지 제한을 설정할 수 있다. 예를 들어 그림 6.7에서 사용자는 Condition 도메인에만 원천 용어를 매핑하고 있다. 기본적으로 Usagi는 standard concept에만 매핑을 진행하지만, 만약 “Filter standard concepts” 옵션을 설정하지 않을 경우, Usagi는 분류 개념 classification concept 또한 검토할 것이다. 마우스를 다른 필터에 올려놓으면 해당 필터에 대한 추가적인 정보가 나타날 것이다.

한 가지 특별한 필터는 “Filter by automatically selected concepts / ATC code”이다. 만약 검색에 추가 조건을 걸어야 한다면, 자동 concept ID로 표시되는 열 (세미콜론으로 구분)에 CONCEPT_ID 목록이나 ATC 코드를 제공하면 된다. 예를 들어 약물의 경우 이미 각 약에 ATC 코드가 이미 할당되어 있을 수 있다. 비록 ATC 코드가 하나의 RxNorm 약물 코드로 인지되지 않더라도, 용어의 ATC 코드 한정으로 검색을 제한하는데 도와줄 수 있다. ATC 코드를 사용하려면 다음 절차를 따르면 된다.

6.3. 2단계: 용어 매핑 생성

95

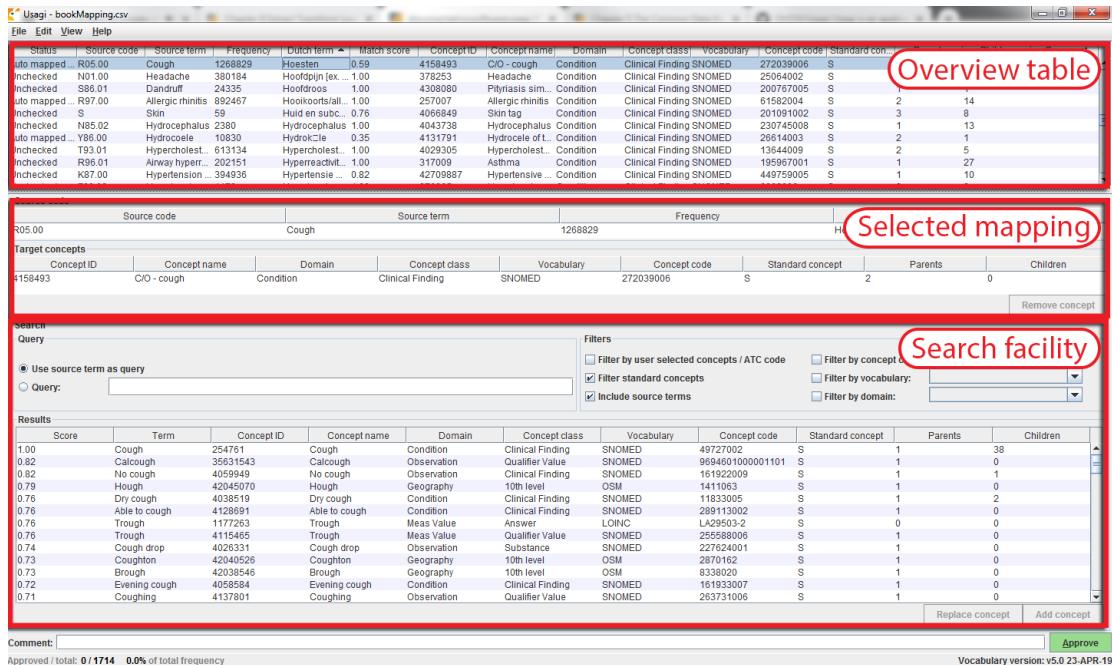


Figure 6.8: Usagi 소스코드 입력화면.

- 열 매핑 부분에서, “Auto concept ID column”을 “ATC column”으로 바꾼다.
- 열 매핑 부분에서, ATC 코드가 포함된 열을 “ATC column”으로 선택한다.
- “Filter by user selected concepts / ATC code” 필터를 누른다.

또한, ATC 코드 이외의 다른 것으로도 조건을 설정할 수 있다. 위의 그림 예시에서 보이듯이 우리는 UMLS의 부분 매핑을 이용하여 Usagi의 검색을 설정하였다. 이런 경우에는 “Auto concept ID column”을 사용하여야 한다.

일단 모든 설정을 마치고 나면, “Import” 버튼을 눌러서 파일을 불러와야 한다. 파일 불러오기를 할 때 단어 유사도 알고리즘을 이용하여 원천 용어를 매핑하기 때문에 대략 몇 분 정도 소요될 수 있다.

원천 용어의 OMOP Concept으로의 매핑 검토

일단 원천 코드의 파일을 불러오면, 매핑 절차가 시작된다. 그림 6.8에서 Usagi 화면이 3가지 주요 기능으로 구분된 것을 확인할 수 있다: 개요 테이블, 선택된 매핑 테이블, 검색 기능. 이때, 오른쪽 마우스 클릭을 하여 어떤 테이블에 대해서도 열을 선택하여 숨겨서 간단히 볼 수 있다.

제안된 매핑의 승인

“Overview Table”은 현재의 원천 용어의 매핑을 보여준다. 원천 용어를 불러온 직후, 검색 설정과 단어 유사도를 기반으로 자동으로 생성되어 제안된 매핑을 포함하고 있다. 그림 6.8에서 나타나듯이, 사용자가 검색 옵션을 Condition으로 설정했기에 네덜란드어 Condition 용어 코드의 영어 이름은 Condition 도메인의 표준용어로 매핑되는 것을 볼 수 있다. Usagi는 원천 코드 기술서의 concept 이름과 동의어를 비교함으로써 최적의 매칭을 수행한다. 사용자가 “Include source terms”를 선택하였기 때문에 Usagi는 용어의 특정 용어 코드로 매핑되는 모든 원천 용어 코드의 이름과 동의어까지 검토하게 된다. 만약 Usagi가 매핑을 진행할 수 없으면, CONCEPT_ID = 0으로 매핑될 것이다.

용어 시스템에 익숙한 사람이 원천 용어를 표준 용어로 매핑하는 것을 도와주는 것이 권장된다. 각 개인은 “Overview Table” 탭에서 각 코드에 대하여 Usagi가 권장하는 매핑을 받아들이거나 아니면 새로운 매핑을 선택하는 작업을 하게 된다. 예를 들어 그림 6.8에서 우리는 네덜란드어 “Hoesten”가 영어 “Cough”로 번역되는 것을 볼 수 있다. Usagi는 “Cough”를 사용하고 OMOP concept “4158493-C/O - cough”로 매핑을 한다. 이때의 매핑에 대하여 매칭 점수는 0.58 (매칭 점수는 일반적으로 0에서 1의 값을 가지며, 1이 더 신뢰할만한 매칭임) 이였고, 이는 Usagi가 이 네덜란드어 코드를 SNOMED로 매핑한 결과에 대한 확신을 가지기 어렵다는 것을 의미한다. 이 예시에서는 해당 매핑 결과에 동의하였고, 화면의 하단 우측의 “Approve” 버튼을 클릭함으로써 승인하였다.

새로운 매핑의 탐색

Usagi가 제시하는 매핑에 대하여 사용자가 새로운 매핑을 찾거나 아니면 매핑되는 concept가 없도록 (CONCEPT_ID = 0) 하는 경우도 있을 것이다. 그림 6.8의 예시를 통해 네덜란드어 “Hoesten”가 영어 “Cough”로 번역되는 것을 확인할 수 있다. Usagi의 제안은 UMLS에서 파생된 매핑으로 제한되기에, 그 결과가 적합하지 않을 수도 있다. 검색 기능을 통해서 실제 용어 자체 혹은 검색 상자 쿼리를 이용해서 다른 개념을 찾을 수 있다.

Usagi 매뉴얼 검색 상자를 이용할 때, Usagi는 구조화된 검색 쿼리를 지원하지 않고 fuzzy search를 한다는 것을 유념해야 한다. 그리고 현재까지는 AND나 OR과 같은 부울Boolean 연산자를 이용한 검색을 지원하지 않고 있다.

“Cough”에 대해서 더 나은 매핑을 찾는다고 가정해보자. 검색 기능의 오른편 쿼리 부분에 용어 검색을 할 때 결과를 정리해주는 기능을 제공하는 필터 부분이 있다. 그러면 우리는 표준 용어만을 찾아야 하며, 표준 용어에 매핑되는 코드의 이름과 동의어를 기반으로 검색할 수 있다.

이러한 검색 기준을 적용한다면 “254761-Cough”와 같은 용어 코드를 찾을 수 있으며, 이는 네덜란드어의 용어 코드 매핑에 적합한 용어일 수 있다. 이를 적용하기 위해 “Selected Source Code” 업데이트의 “Replace concept” 버튼을 누르고, “Approve” 버튼을 누르면 된다. 또한 “Add concept”

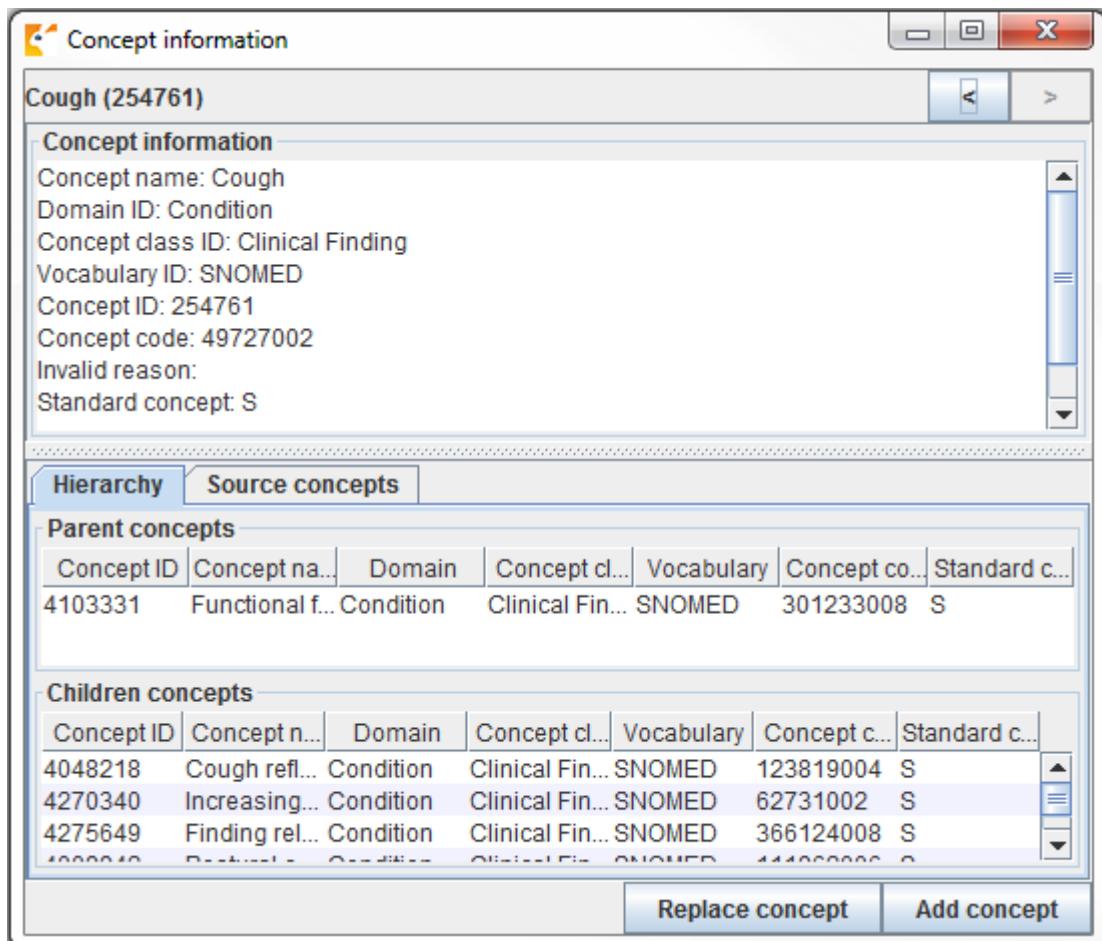


Figure 6.9: Usagi 개념 정보 패널.

버튼이 있는데, 이는 하나의 원천 용어 코드에 대한 다수의 OMOP standard concept 매핑을 할 수 있게 해준다 (예를 들어, 일부 원천 코드는 표준 용어와는 달리 다양한 질병을 함께 포함하고 있을 수 있다).

개념 정보

적절한 개념 concept을 찾아 매핑하려 할 때, concept의 “social life”를 고려하는 것은 중요하다. concept의 의미는 계층 구조에서의 위치에 따라 부분적으로 의존적일 수 있으며, 종종 계층적 지위와 거의 혹은 전혀 상관없고 대상 concept로도 적절하지 않은 “orphan concepts”도 있다. Usagi는 각 개념에 대해 얼마나 많은 부모, 자식 개념이 있는지 알려주기도 하고, ALT + C를 누르거나 위쪽 메뉴바의 view -> Concept를 누르면 더 자세한 정보를 볼 수 있게 해준다.

그림 6.9는 concept 정보 패널을 보여준다. concept의 일반적인 정보부터, 부모, 자식, 그리고 다른

원천 용어 코드와의 정보도 보여준다. 사용자는 이 패널을 이용해서 계층 구조를 탐색할 수 있고, 다른 목표 개념을 정할 수도 있다.

모든 용어 코드 검토가 끝날 때까지 용어 코드를 따라 이 절차를 진행하면 된다. 화면의 맨 위의 원천 용어 코드 목록에서, 열 머리글별로 용어 코드를 정렬할 수 있다. 때로 최고빈도부터 최저빈도의 용어 코드까지 살펴보는 것을 권장한다. 화면의 하단 왼쪽에는 매핑을 허용한 용어 코드의 개수, 그리고 그에 따라 얼마나 많은 용어 코드가 발생했는지를 확인할 수 있다.

또한 매핑 결정에 대한 설명을 추가하여 문서화 시에 사용할 수도 있다.

모범 사례

- 코딩 스키마에 경험 있는 사람이 해야 한다.
- “Overview Table”의 열을 열 이름을 눌러서 정렬할 수 있다. “Match Score”를 눌러서 정렬해보는 것도 좋다. Usagi가 가장 확실하게 제안하는 매핑 코드를 검토하면 많은 매핑 작업을 빠르게 끝날 수도 있다. 그리고 빈도가 높은 단어와 낮은 단어에 쓰이는 노력이 각각 다르므로 “Frequency”로 정렬해서 작업하는 것도 중요하다.
- 일부 코드를 CONCEPT_ID=0 (매핑 안 됨)으로 매핑하는 것도 허용되며, 어떤 코드는 좋은 매핑을 찾을 가치가 없어서 일 수도 있고 또는 적절한 매핑이 없어서 일 수도 있다.
- 특히 부모 계층과 자녀 계층에 대해서는 개념의 내용을 고려하는 것이 중요하다.

생성된 Usagi Map 내보내기

일단 Usagi를 통해 매핑을 생성하였으면, 이를 사용하기 가장 좋은 방법은 매핑을 내보낸 다음 용어의 SOURCE_TO_CONCEPT_MAP 테이블에 추가하는 것이다.

매핑 결과를 내보내기 위해서는, File -> Export source_to_concept_map으로 가면 된다. 이때 어느 SOURCE_VOCABULARY_ID를 이용할 것인지 묻는 팝업창이 나타나는데 짧은 식별자를 입력하면 된다. Usagi는 입력된 이 식별자를 SOURCE_TO_CONCEPT_MAP 테이블에서 특정 매핑을 식별할 수 있게 해주는 SOURCE_VOCABULARY_ID로 이용할 것이다.

SOURCE_VOCABULARY_ID를 선택한 후에는, 내보낼 CSV 파일의 이름과 파일 경로를 입력하게 된다. 내보내는 CSV 파일의 구조는 SOURCE_TO_CONCEPT_MAP 테이블과 동일하다. 이 매핑은 용어의 SOURCE_TO_CONCEPT 테이블로 추가될 수 있다. 그리고 앞선 단계에서 정의한 SOURCE_VOCABULARY_ID를 정의하는 VOCABULARY 테이블에 단일 행으로 추가하는 것 역시 가능하다. 마지막으로, “Approved” 상태인 매핑만을 CSV 파일로 내보내는 것이 중요하다; 매핑을 내보내기 위해서는 Usagi에서 매핑을 완료해야만 한다.

Usagi Map 업데이트

매핑은 종종 한 번에 끝나지 않는다. 원천 코드가 추가되는 식으로 데이터가 업데이트되거나 용어가 정기적으로 업데이트되면 매핑 또한 업데이트되어야 할 것이다.

원천 코드가 업데이트될 때는 다음과 같은 단계를 따르는 것이 좋다:

1. 새로운 원천 용어 코드 파일을 불러온다.
2. 파일을 고른다 -> 이전의 매핑을 적용하고, 예전의 Usagi 매핑 파일을 선택한다.
3. 이전의 매핑 파일에서 매핑되지 않았던 용어 코드를 식별하고, 새롭게 매핑한다.

용어가 업데이트되면 아래의 단계를 따른다:

1. Athena에서 새로운 용어 파일을 다운받는다.
2. Usagi 인덱스를 다시 생성한다 (Help -> Rebuild index).
3. 매핑 파일을 엽니다.
4. 새로운 용어 버전에 따라 표준 용어가 아닌 용어 코드를 식별하여 적절한 목표 개념을 찾는다.

6.4 3단계: ETL 수행

일단 ETL 설계와 코드 매핑이 완료되면, ETL 절차는 소프트웨어를 통해 수행할 수 있다. ETL이 설계될 때, CDM과 원천 데이터 둘 다에 대해 잘 아는 사람이 참여하기를 권장한 바 있다. 마찬가지로 ETL이 수행될 때도, 데이터 (특히 빅데이터)와 ETL 수행 경험이 있는 사람이 참여하는 것이 바람직하다. 즉, 기관 외부의 기술 전문가를 고용하거나 초청하여 ETL 수행을 시키는 것이 나을 수도 있다. 또한 이 작업은 한 번에 끝나는 작업이 아니라는 점을 참고하기 바란다. 그렇기에 앞으로는 ETL 수행 및 유지에 일정 시간 이상을 할애할 수 있는 사람이나 팀이 있는 것이 좋을 것이다 (6.7절에서 더 명확히 설명할 것이다).

수행은 각 기관에 따라 다양한 양상을 보이며 특히 정보 인프라, 데이터베이스의 크기, ETL의 복잡성, 기술 전문가의 능력 등의 요소에 따라 많이 달라진다. 많은 요소에 따라 달라지기 때문에 OHDSI는 ETL을 수행하기 위한 최선의 방법에 대한 공식적인 권고를 하고 있지 않다. 그동안 많은 그룹이 SQL builders, SAS, C#, Java, Kettle을 사용해왔다. 각각 장점과 단점이 있었고, 그 어느 것도 이러한 기술에 익숙한 사람이 없으면 수행이 불가능했다.

각각 다른 ETL의 예시 (복잡성에 따라 정렬된다):

- ETL-Synthea - SQL을 이용한 Synthea 데이터베이스 변환
- <https://github.com/OHDSI/etl-synthea>

- ETL-CDMBuilder - 다수의 데이터베이스를 변환하기 위해 고안된 .NET application
- <https://github.com/OHDSI/etl-cdmbuilder>
- ETL-LambdaBuilder - AWS lamda 기능을 이용한 빌더
- <https://github.com/OHDSI/etl-lambdabuilder>
- ETL-Korean-National Sample Cohort - R과 SQL을 이용한 데이터베이스 변환
- <https://github.com/OHDSI/ETL—Korean-NSC>

그동안 많은 시도가 있었지만, ‘최종적’인 사용자 친화적 ETL 툴을 개발하지는 못했다. 항상 이러한 툴은 ETL 작업의 80%까지는 잘 수행하지만, 남은 20%에 있어서는 원천 데이터베이스에 따라 인프라 단에서의 코드 작성이 필요하다.

일단 기술 전문가가 수행할 준비가 된다면, ETL 설계 문서를 공유해야만 한다. 문서에는 수행을 시작할만한 충분한 정보가 있어야 하지만, 개발자가 개발 과정 중에 ETL 설계자에게 언제나 질문할 수 있는 환경 역시 마련되어 있어야 한다. ETL 설계자에게는 로직이 명확해 보일지라도 CDM이나 데이터에 친숙하지 않은 수행자에게는 명확하지 않아 보일 수도 있다. 그렇기에 실행 단계는 팀 단위로 수행되어야 한다. 설계자와 수행자 모두 로직이 올바르게 작동한다고 동의할 때까지, 모두 CDM 생성과 검증을 같이 수행하는 것이 바람직하다.

6.5 4단계: 질 관리

ETL 절차 수행을 위해서 질 관리는 반복적으로 수행된다. 질 관리의 전형적인 패턴은 로직 작성 -> 로직 수행 -> 로직 검증 -> 로직 수정 및 작성이다. CDM을 검증하기 위한 많은 방법이 있지만, 아래의 단계는 몇 년간의 ETL 수행을 통해 OHDSI 내부에서 권고하는 단계이다.

- ETL 설계 문서, 컴퓨터 코드, 용어 매핑을 검토하라. 누구라도 실수를 할 수 있기 때문에, 항상 한 명 이상이 다른 사람이 어떤 작업을 수행하고 있는지 검토해야 한다.
- ETL 상의 가장 큰 문제점은 원천 데이터의 원천 용어 코드가 표준 용어에 어떻게 매핑되었는지와 관련된다. 특히 NDC처럼 날짜에 따라 유효 코드가 바뀌는 코드가 원천 용어 코드라면 이는 더욱 어려울 수 있다. 원천 용어가 항상 적절한 개념으로 변환되도록 매핑이 수행되는 부분을 두 번씩 봐야 한다.
- 원천 데이터와 목표 데이터의 표본으로써 특정한 한 사람의 모든 정보를 수작업으로 비교하라.
- 여러 개의 기록을 가진 한 사람의 데이터를 살펴보는 것이 큰 도움이 될 수 있다. 한 사람의 기록을 추적함으로써 CDM 데이터가 로직에 따라 기대했던 결과와 다른 경우를 발견해낼 수도 있다.
- 원천 데이터와 목표 데이터의 전체 수를 비교하라.
- 특정 문제를 어떻게 해결하는지 설명하기에 따라 몇 가지 차이점이 있을 수 있다. 예를 들면,

연구자에 따라 성별이 NULL로 기록된 사람을 어차피 연구에 포함되지 않기 때문에 삭제하기로 할 수도 있다. 그리고 CDM에서의 방문이나 원천 데이터에서의 방문이 다르게 구성될 수도 있다. 따라서 CDM과 원천 데이터의 총합을 비교할 때는 이러한 차이점이 발생할 수 있다는 것을 예상하고 설명할 수 있어야 한다.

- 해당 CDM 버전에서 원천 데이터에 대해 이미 수행된 기존의 연구를 재현해 보라.
- 비록 시간이 많이 들겠지만, 원천 데이터와 CDM 버전에 따른 차이점을 확인하기에 좋은 방법이다.
- ETL에서 다뤄야 하는 원천 데이터의 패턴을 따라 하기 위한 단위 검정 unit test을 작성하십시오. 예를 들어 만약 ETL 명세에서 성별 정보가 없는 환자를 없애야 한다고 명시한다면, 성별이 없는 환자에 대한 단위 검정을 작성하고 수행자가 처리하는 방안을 평가하십시오.
- 단위 검정은 ETL 변환의 정확도와 질을 평가하기 위한 편리한 방법이다. 보통은 변환하고자 하는 원천 데이터의 작은 표본을 사용한다. 데이터 세트의 각 사람이나 기록은 ETL 문서에 기록된 대로 로직의 특정 부분으로 검정해야 한다. 이 방법을 쓴다면 문제를 파악하거나 로직 실패를 확인하기에 좋다. 작은 사이즈의 데이터로 테스트사는 것은 컴퓨터 코드 구동도 훨씬 빠르고 여러 번 시행하기에도 좋고 에러를 빨리 확인하는데에도 좋다.

ETL의 관점에서 고차원의 질 관리 접근법도 있다. 데이터 질 관리에 대한 더 구체적인 노력은 OHDSI에서 진행되고 있으며, 15장을 확인해주기 바란다.

6.6 ETL 협약 convention과 테미스 THEMIS

많은 그룹이 데이터를 CDM으로 변환함에 따라 구체적인 협약 convention의 필요성이 명확해졌다. 예를 들어 만약 한 사람의 기록에서 출생연도가 없으면 ETL은 어떻게 할 것인가? CDM의 목적은 보건의료 데이터의 표준화이지만 만약 모든 그룹이 데이터의 특정 시나리오를 각각 다르게 다룬다면 네트워크를 통해 체계적으로 데이터를 다루는 것이 더욱 어려워질 것이다.

OHDSI 공동체는 CDM의 일관성을 증진하기 위해 협약 문서 작성은 시작하였다. OHDSI가 동의하는 이러한 협약의 정의에 대해서는 CDM wiki를 통해 확인할 수 있다.⁶ 각 CDM 테이블은 ETL 설계 시 참고할 수 있는 고유의 협약을 맺고 있다. ETL을 설계할 때 이 협약을 참고한다면 어떤 설계에 대한 결정 시에 커뮤니티와 동일한 일관성을 유지할 수 있도록 도와줄 것이다.

발생 가능한 모든 데이터 시나리오에 대해서 어떻게 다뤄야 할지에 대한 문서를 작성하는 것은 불가능하지만, OHDSI 워크 그룹을 통해 공통적인 시나리오를 문서화하는 것은 가능하다. THEMIS⁷는 협약을 모으고, 명시하고, 공동체에 조언을 나눈 다음, 마지막으로 CDM wiki에 완성된 문서를 공

⁶<https://github.com/OHDSI/CommonDataModel/wiki>.

⁷<https://github.com/OHDSI/Themis>

개하는 일을 하는 각 개인으로 구성되어 있다. Themis는 고대 그리스의 질서, 공정함, 법, 자연법, 관습을 관장하는 티타니스로 이 그룹의 이름으로 정했다. ETL을 수행할 때, 만약 특정 시나리오에 대해 어떻게 할지 모르겠다면, THEMIS 워크그룹은 OHDSI 포럼에 질문을 남기기를 권장한다.⁸ 대부분의 경우 질문은 커뮤니티의 다른 사람 역시 고민하고 있는 문제일 가능성이 높다. THEMIS는 이런 토론과 워크그룹 미팅과 대면 토론을 통하여 어떤 협약이 문서화될 필요가 있는지 알려준다.

6.7 CDM과 ETL의 유지

ETL을 설계하고, 매핑을 만들고, ETL을 수행하고, 질 관리 검증을 만들기는 절대 쉽지 않다. 안타깝게도 그게 다가 아니다. 첫 번째 CDM이 만들어진 후 지속해서 이어지는 ETL 유지 과정이 있다. 유지를 요구하는 몇몇 공통되는 계기는 다음과 같다: 원천 데이터의 변화, ETL 상의 오류, 새로운 OMOP 용어의 출시, CDM 자체의 변화 혹은 업데이트 등이 있다. 만약 이 중 하나라도 발생한다면, 다음 사항에 대한 업데이트가 필요하다 : ETL 문서, ETL을 수행한 소프트웨어 프로그래밍, 그리고 예시 검정과 질 관리

보건의료 데이터는 보통 계속 바뀐다: 새로운 데이터의 출시 (예를 들어 데이터에 새로운 열의 추가), 기존에 존재하지 않았던 새로운 환자 시나리오의 출현 (예를 들어 출생 전에 사망이 기록되어있는 새로운 환자), 데이터에 대한 이해도의 상승 (예를 들어 입원 아동 환자의 출생 기록이 청구 과정으로 인해 외래에서 발견). 원천 데이터의 모든 변경 사항에 대해서는 아니지만, 최소한 ETL 절차를 망가뜨리는 변경 사항은 해결해야만 할 것이다.

만약 오류가 발견된다면 역시 해결해야 할 것이다. 하지만 모든 오류가 중요하지는 않다. 예를 들어 COST 테이블에서 비용이 한 자릿수에서 반올림되었다고 가정해보자 (원천 데이터에서 \$3.82 가 CDM에서는 \$4.00이 됨). 만약 이 데이터를 사용하는 주요 연구자가 환자 약물 노출과 진단에 대한 특성을 주로 연구한다면, 이는 별로 중요하지 않으며 향후 해결하면 된다. 만약 이 데이터를 사용하는 주요 연구자 중 보건경제학자가 있다면 이는 즉시 해결해야 하는 주요 문제가 될 것이다.

OMOP 용어집 역시 원천 데이터처럼 지속해서 변화한다. 사실 용어집은 용어가 계속 업데이트됨에 따라 한 달 안에도 여러 개의 버전을 가질 수 있다. 각 CDM은 특정 용어집 개별 버전 기반으로 운영되며, 용어집 새 버전에서 작동할 때 원천 용어 코드가 표준 용어로의 매핑 정도에 따라 다른 결과를 만들 수도 있다. 용어집 간의 차이는 미미할 수도 있지만, 용어집이 개정될 때마다 CDM을 새로 만드는 것은 불필요하다. 하지만, 일 년에 한두 번 정도 새로 출시된 용어집은 기반으로 CDM 을 재가공하는 것은 유익할 수 있다. ETL 코드 자체를 새로 업데이트해야 할 정도로 새로운 버전의 용어집이 나오는 일은 매우 드물다.

⁸<http://forums.ohdsi.org/>

CDM 또는 ETL 업데이트를 하게끔 하는 마지막 계기는 공통 데이터 모델 자체의 업데이트이다. 공동체가 커짐에 따라 새로운 데이터의 필요성이 커지고, 이는 CDM에 새로운 데이터를 추가할 수 있는 방향으로 가게 된다. 이는 이전의 CDM에 없었던 데이터가 새로운 버전의 CDM에 들어갈 수 있는 것을 의미한다. CDM 구조의 변화는 잘 생기지 않지만, 충분히 가능한 일이다. 예를 들어 CDM은 DATE 필드 대신 DATETIME 필드를 필수 컬럼으로 정의하였고, 이는 ETL 절차에서 에러를 발생시킬 수 있는 일이다. 하지만, CDM 버전은 자주 출시되지 않으며, 각 기관은 데이터를 언제 업데이트할 지 결정할 수 있다.

6.8 ETL에 대한 마지막 생각

ETL 절차는 여러 가지 이유로 완전히 통달하기 어려운 절차이다. 각자가 서로 다른 고유한 원천 데이터를 다루기 때문에 “만능열쇠”를 만드는 것은 어렵다. 하지만, 수년간 시도에서 배운 몇 가지 교훈이 있다.

- 80/20 규칙. 피할 수만 있다면 너무 많은 시간을 원천 용어 코드를 OMOP standard concept 으로 수동 매핑하는데 할애하지 말기 바란다. 전체 데이터양의 대부분을 차지하는 원천 용어 코드 위주로 매핑하는 것이 이상적이다. 그것만으로도 시작하기에 충분할 것이고, 실제 사용 예시에 기반하여 나머지 남은 코드도 다룰 수 있다.
- 연구 품질에 맞지 않는 데이터를 잊어버리는 것은 괜찮다. 이런 기록은 분석을 시작하기 전에 결국은 벼려지게 되고, 우리는 대신 ETL 절차에서 미리 삭제할 뿐이다.
- CDM은 지속적인 유지를 필요로 한다. 단순히 ETL을 완료했다고, 두 번 다시 손대지 않는 것을 의미하는 것이 아니다. 원천 데이터는 변할 수도 있고, 코드에 오류가 있을 수도 있고, 새로운 용어가 나오거나 CDM 버전의 업데이트가 있을 수 있다. 이러한 변화에 대비하고 ETL 을 최신 상태로 유지하기 위해 자원을 할당할 필요가 있다.
- OHDSI CDM으로 시작하는 것을 돋고, 데이터베이스의 변환을 수행하거나 분석 툴을 사용하기 위해 Implementers Forum에 방문해주길 바란다.⁹

6.9 요약



- ETL에 접근하는 방법에 대해 일반적으로 합의된 절차가 있는데 다음과 같다.
- 데이터 전문가와 CDM 전문가가 함께 ETL을 설계한다.
- 의료 지식이 있는 사람이 용어 코드 매핑을 진행한다.

⁹<https://forums.ohdsi.org/c/implementers>

- 엔지니어가 ETL을 수행한다.
- 모든 사람이 질 관리에 참여한다.
- 이러한 단계를 돋기 위해 OHDSI 공동체에서 무료로 사용 가능한 툴을 개발하였다.
- 많은 ETL 예시가 있으며, 가이드로 삼을만한 협약이 있다.

6.10 예제

Exercise 6.1. 다음 ETL 절차를 올바른 단계로 정렬하십시오:

- A) 데이터 전문가와 CDM 전문가가 함께 ETL을 설계한다.
- B) 기술 전문가가 ETL을 수행한다.
- C) 의료 지식이 있는 사람이 코드 매핑을 진행한다.
- D) 모든 사람이 질 관리에 참여한다.

Exercise 6.2. 선택한 OHDSI 자원을 활용하여, 테이블 6.3의 PERSON 기록에서 나타나는 4가지 문제를 발견하십시오 (공간상 축약된 형태의 표):

Table 6.3: A PERSON table.

Column	Value
PERSON_ID	A123B456
GENDER_CONCEPT_ID	8532
YEAR_OF_BIRTH	NULL
MONTH_OF_BIRTH	NULL
DAY_OF_BIRTH	NULL
RACE_CONCEPT_ID	0
ETHNICITY_CONCEPT_ID	8527
PERSON_SOURCE_VALUE	A123B456
GENDER_SOURCE_VALUE	F
RACE_SOURCE_VALUE	WHITE
ETHNICITY_SOURCE_VALUE	NONE PROVIDED

Data Output Explain Messages Notifications Query History					
	id character varying (1000)	start date	stop date	patient character varying (1000)	encounterclass character varying (1000)
1	12	2004-09-26	2004-09-27	11	inpatient
2	13	2004-09-27	2004-09-30	11	inpatient

Figure 6.10: 원천 데이터 예시.

Exercise 6.3. VISIT_OCCURRENCE 기록을 만들어보자. Synthea에 대한 예시 로직이 다음과 같이 있다: PATIENT, START, END에 따라 오름차순으로 데이터를 정렬하십시오. 그다음 PERSON_ID 별로, 하나의 기록의 END 시간과 다음 기록의 START 시간의 차이가 1일 이하인 기록을 하나로 만들어 준다. 각 통합된 입원 환자 기록은 하나의 입원 환자 방문으로 간주하며, 다음과 같이 설정한다:

- MIN(START) as VISIT_START_DATE
- MAX(END) as VISIT_END_DATE
- “IP” as PLACE_OF_SERVICE_SOURCE_VALUE

만약 아래와 같은 그림 6.10의 방문 기록이 원천 데이터라고 가정한다면, CDM에서의 VISIT_OCCURRENCE 기록은 어떻게 보일 것인가?

제안된 답변은 부록 E.3에서 확인할 수 있다.

Part III

데이터 분석

Chapter 7

데이터 분석 이용 사례

Chapter lead: David Madigan

OHDSI는 실세계 헬스케어 데이터 (일반적으로 보험청구 또는 전자의무기록 데이터베이스)로부터 믿을만한 근거를 만들어내는 데 초점을 맞춰 왔다. OHDSI가 관심을 가져온 이용 사례는 크게 3 개의 카테고리로 구분된다:

- 임상적 특성 분석
- 인구 수준 추정
- 환자 수준 예측

본 장에서는 각 카테고리에 관해 설명한다. 모든 이용 사례에 있어서, 생성된 근거는 데이터 자체가 가지고 있는 한계를 이어받는다는 점을 유념하라. 이 한계에 대해서는 이 책의 “근거의 질” 장에서 다루고 있다. (14장~18장)

7.1 임상적 특성 분석

임상적 특성 분석 characterization은 다음과 같은 질문에 답변을 시도한다.

그들에게 무슨 일이 발생했는가?

우리는 데이터를 이용하여 코호트 또는 전체 데이터베이스 내 환자와 헬스케어의 특성에 대한 의문에 답할 수 있으며, 시간이 지남에 따라 이 특성이 어떻게 변화하는지 알 수 있다.

데이터는 다음과 같은 질문에 답을 제공할 수 있다:

- 심방세동으로 새 진단을 받은 환자 중 얼마나 많은 사람이 와파린을 처방받는가?
- 고관절 치환술을 받은 환자의 평균 연령은 어떻게 되는가?
- 65세 이상 환자 중 폐렴 발생률은 얼마나 되는가?

일반적인 임상적 특성 분석 질문은 다음과 같이 표현된다:

- 얼마나 많은 환자가...?
- 얼마나 자주...?
- 환자 중 어느 정도의 비율이...?
- ... 검사에 대한 결과값의 분포가 어떠한가?
- ... 질병을 앓는 환자의 당화혈색소(HbA1c) 수준이 어떠한가?
- ... 환자에 대한 검사 결과값이 어떠한가?
- 환자가 ...에 노출되는 평균 노출 기간은 얼마인가?
- ...의 시간 경과에 따른 추세는 무엇인가?
- 이 환자가 사용하는 다른 약이 무엇인가?
- 수반되는 치료법이 무엇인가?
- ...에 대한 충분한 증례가 있는가?
- ...에 대한 연구 X가 실행 가능한가?
- ...에 대한 인구통계학적 특징은 무엇인가?
- ...의 위험인자가 무엇인가? (특정 위험인자가 구분된다면, 이는 예측이 아닌 추정에 해당함)
- ...의 예측요인이 무엇인가?

그리고 원하는 결과값은 다음과 같다:

- 건수 또는 퍼센트 count or percentage
- 평균 averages
- 기술 통계 descriptive statistics
- 발생률 incidence rate
- 유병률 prevalence
- 코호트 cohort
- 규칙 기반 표현형 rule-based phenotype
- 약물 사용 drug utilization
- 질병의 자연 경과 disease natural history
- 순응도 adherence
- 동반 질환 프로파일 co-morbidity profile
- 치료 경로 treatment pathways

- 치료 요법 line of therapy

7.2 인구 수준 추정

제한된 범위내에서, 데이터는 헬스케어 개입 효과에 대한 인과적 추론을 지원함으로써 다음의 질문에 답할 수 있다.

인과적 영향 causal effect 이 어떠한가?

우리는 행동의 결과를 이해하기 위해 인과적 영향을 파악하고자 한다. 예를 들어, 우리가 어떤 치료법을 사용하기로 했다면, 이것이 앞으로 우리에게 무슨 변화를 일으킬까?

데이터는 다음과 같은 질문에 답을 줄 수 있다:

- 새로 심방세동으로 진단받은 환자에 있어서, 치료 시작한 지 1년 이내에 와파린 warfarin이나 다비가트 dabigatran보다 주요 출혈을 더 일으키는가?
- 메트포르민 metformin이나 설사에 미치는 인과 효과가 연령에 따라 다른가?

일반적인 인구 수준 효과 추정 population-level estimation 질문은 다음과 같이 표현된다:

- ...의 효과는 무엇인가?
- 내가 개입한다면 어떻게 될까...?
- 어떤 치료가 더 효과가 좋을까?
- X가 Y에 미치는 위험이 무엇일까?
- ... 이벤트 발생에 걸리는 시간 time-to-event가 얼마나 될까?

그리고 원하는 결과값은 다음과 같다:

- 상대 위험도 relative risk
- 발생 위험비 hazards ratio
- 대응위험도/교차비 odds ratio
- 평균 치치 효과 average treatment effect
- 인과적 영향 causal effect
- 관련성 association
- 상관성 correlation
- 안전성 감시 safety surveillance
- 비교 효과 comparative effectiveness

7.3 환자 수준 예측

데이터베이스에 쌓인 환자 건강 이력을 기반으로, 우리는 미래에 발생할 건강 이벤트에 대해 환자 수준의 예측할 수 있다.

나에게 무슨 일이 발생할까?

데이터는 다음과 같은 질문에 답을 제공할 수 있다:

- 주요 우울증으로 새롭게 진단받은 특정 환자가 진단받은 지 1년 안에 자살을 시도할 확률이 얼마나 되는가?
- 심방세동으로 새롭게 진단받은 특정 환자가, 와파린으로 치료를 시작한 지 1년 안에 혀혈성 뇌졸중을 겪을 확률이 얼마나 되는가?

일반적인 환자 수준 예측 patient-level prediction에 대한 질문은 다음과 같이 표현된다:

- 이 환자가 ... 할 가능성성이 얼마나 되는가?
- ...에 대한 후보자가 누구인가?

그리고 원하는 결과값은 다음과 같다:

- 개인에 대한 확률
- 예측 모델
- 높은/낮은 위험군
- 확률론적 표현형

인구 수준 추정과 환자 수준 예측은 어느 정도 중복된다. 예측을 위한 중요 이용사례로 다음 예제를 들 수 있다: 약물 A가 처방된 특정 환자의 임상 결과를 예측하는 것과 또한 약물 B가 처방된 사람에게서 그와 똑같은 임상 결과를 예측하는 것이다. 실제로 한 환자가 여러 약 중 하나 (약물 A라고 하자)를 처방받았고, 약물 A의 예상되는 결과가 실제로 일어나는지를 살펴봤다고 가정해 보자. 그 환자에게 약물 B는 처방되지 않았고 B 투여 이후의 임상 결과는 관찰된 적이 없으므로 그 예상된 임상 결과는 반 사실적 counterfactual이며, 예측은 할 수 있더라도 사실과 다를 수 있다. 이러한 각각의 예측 작업은 환자 수준 예측에 속한다. 그러나 두 결과의 차이 (또는 비율)는 단위 수준의 인과 효과이며, 예측모형이 아닌 인과적 영향 추정 방법론을 사용하여 추정하여야 한다.



사람들은 예측모델을 인과 모델로 잘못 해석하는 경향이 있다. 그러나 예측 모델은 상관성만을 보여줄 수 있으며 결코 인과성을 보여줄 수 없다. 예를 들자면, 당뇨병이 심근경색의 강한 예측변수이기 때문에, 당뇨병약을 사용하는 것은 심근경색의 강한 예측변수일 수 있다. 그러나

이것이 당뇨병약 복용을 중단하는 것이 심근경색을 막는다는 것을 의미하는 것은 아니다!

7.4 고혈압 이용 사례 예

당신은 고혈압의 1차 치료 요법으로서 ACE inhibitor(ACE 억제제) 단일요법과 thiazide(타이아자이드 이뇨제) 단일요법이 급성 심근경색과 혈관부종에 미치는 영향을 연구하는 데 관심이 있는 연구자이다. 당신은 OHDSI 연구에 기반하여 인구 수준 추정 연구 질문을 도출했지만, 먼저 관심이 있는 특정 치료에 대한 특성을 어떻게 분석할 것인지 해결해야 한다.

7.4.1 특성 분석 질문

급성 심근경색은 고혈압 환자에게 일어날 수 있는 심혈관계 합병증으로, 고혈압에 대한 효과적인 치료로 이 위험을 줄여야 한다. 혈관부종은 희귀하지만, 잠재적으로 심각한 ACE 억제제의 알려진 부작용이다. 당신은 관심 약제 (ACE 억제제와 티아자이드 이뇨제)에 노출된 코호트 (10장 참고)를 생성하는 것으로부터 연구를 시작한다. 노출된 환자의 인구통계학적 정보, 병적 상태, 병용 약물 등 기저 특성을 파악하기 위하여 임상적 특성 분석 (11장 참고)을 수행한다. 또한 이 노출 환자 내에서 분석하고자 하는 결과가 얼마나 발생하는지 추정하는 또 다른 특성 분석을 수행한다. 그리고, 당신은 ‘얼마나 자주 1) 급성 심근경색과 2) 혈관부종이 ACE 억제제와 티아자이드 이뇨제에 노출된 기간 동안 발생하는가?’를 묻게 된다. 이러한 특성 분석은 인구 수준 추정 연구 수행의 실현 가능성을 판단하게 하고, 두 치료군이 비교 가능한지를 평가하게 하며, 환자가 받는 치료에 예측되는 위험 요소를 파악할 수 있게 한다.

7.4.2 인구 수준 추정에 대한 질문

인구 수준 효과 추정 연구(12장 참고)는 ACE 억제제와 티아자이드 이뇨제가 급성 심근경색과 혈관부종에 미치는 상대 위험을 추정한다. 더 나아가, 분석에 대한 평가와 음성대조군 분석을 통해 우리가 평균 치료 효과에 대해 믿을 만한 추정치를 도출했는지 평가한다.

7.4.3 환자 수준 예측에 대한 질문

당신은 노출의 인과적 영향 여부와 상관없이, 가장 위험한 결과에 처한 환자를 알아내고자 할 수 있다. 이것은 환자 수준 예측 (13장 참고)의 문제이다. ACE 억제제를 처음 사용하는 환자 중 치료 시작한 지 1년 동안 급성 심근경색 발병 위험이 가장 높은 환자를 찾아내는 예측 모델을 개발한다고 생각해 보십시오. 이 모델을 통해 우리는 처음으로 ACE 처방을 받은 환자의 병력에서 관찰된 사건을 바탕으로, 향후 1년 동안 급성 심근경색을 겪을 가능성을 예측할 수 있다.

7.5 관찰 연구의 한계

OHDSI 데이터베이스가 답변을 제공할 수 없는 중요한 의료분야 질문이 많이 존재한다. 아래 질문이 이에 해당한다.

- 위약과 비교한 치료의 인과 효과. 때때로 치료군의 인과 효과를 비치료군과 비교하여 분석하는 것은 고려해 볼 수 있지만 위약군과 비교하려고 해서는 안 된다.
- 처방전 없이 살 수 있는 일반 의약품 (OTC)과 관련된 모든 것은 일반적인 청구 데이터에서는 확인할 수 없다.
- 임상 결과 또는 여러 변수의 빈도가 매우 낮은 경우. 많은 데이터 베이스에서 사망률, 행동 결과, 라이프 스타일 및 사회-경제적 지위와 같은 것은 매우 적은 숫자의 환자에 대해서만 기록되기 때문에 이를 유념해야 한다.
- 환자는 건강이 좋지 않을 때만 의료 시스템을 이용하는 경향이 있어서, 치료의 이점을 측정하기가 쉽지 않다.

7.5.1 잘못된 데이터

OHDSI 데이터베이스에 기록된 임상 데이터는 의료 현실과 차이가 있을 수 있다. 예를 들어, 환자가 심근경색을 경험한 적이 없어도 환자의 기록에 심근경색 코드가 포함되어 있을 수 있다. 마찬가지로 검사 값이 잘못되었거나 시술에 대한 잘못된 코드가 데이터베이스에 저장되었을 수도 있다. 15장과 16장은 이와 같은 문제를 다루고 있으며, 모범 사례를 통해 이러한 문제를 최대한 식별하고 수정하고자 한다. 그런데도, 잘못된 데이터는 필연적으로 어느 정도까지 존재할 수밖에 없으며, 분석의 타당성을 약화시킬 수 있다. 매우 많은 문헌이 데이터 오류를 처리하기 위한 통계적 추론 보정에 초점을 맞추고 있다. - 예를 들어 Fuller (2009) 참조

7.5.2 결측 데이터

OHDSI 데이터베이스에서의 결측은 감지하기 어려운 문제점을 낳는다. 데이터베이스에 기록되어야 하는 건강 이벤트 (예를 들어 처방, 검사 값 등)가 기록되지 않은 것, 그것이 “결측”이다. 통계 문헌은 “임의의 완전 결측”, “임의의 결측”, “임의가 아닌 결측”과 같은 결측 유형과 이러한 유형을 다루는 복잡한 방법론을 구별하고 있다. Perkins et al. (2017) 가 이 주제에 대한 유용한 입문서를 제공한다.

7.6 요약



- 관찰 연구의 이용 사례는 크게 3개의 카테고리로 구분된다.
- **임상적 특성 분석**은 “그들에게 무슨 일이 발생했는가?”라는 질문에 답하는 것을 목적으로 한다.
- **인구 수준 추정**은 “인과적 영향이 무엇인가?”라는 질문에 답하는 것을 목적으로 한다.
- **환자 수준 예측**은 “개별 환자에게 무엇이 일어날까?”라는 질문에 답하는 것을 목적으로 한다.
- 예측 모델은 인과 모델이 아니다. 강한 예측변수를 통제하는 것이 결과에 영향을 미칠 것이라고 믿을 근거가 없다.
- 관찰형 의료 데이터를 이용하여 연구할 수 없는 질문도 있다.

7.7 예제

Exercise 7.1. 다음 질문은 어떤 사용 사례 카테고리에 해당하는가?

1. 비스테로이드 약물에 최근 노출되었던 환자가 위장관 출혈을 겪을 비율을 계산하라.
2. 기저 특성을 기반으로 특정 환자가 차년도에 위장관 출혈을 겪을 확률을 계산하라.
3. 셀레콕시브 celecoxib와 비교하여 디클로페낙 diclofenac이 위장관 출혈에 미치는 위험을 추정하라.

Exercise 7.2. 디클로페낙 diclofenac이 위장관 출혈에 미치는 위험을 비노출 (위약)의 경우와 비교하여 추정하고자 한다. 이와 같은 연구가 헬스케어 관찰 데이터를 이용하여 수행 가능한가?

제안된 답변은 부록 E.4에서 확인할 수 있다.

Chapter 8

OHDSI 분석 툴

Chapter leads: Martijn Schuemie & Frank DeFalco

OHDSI는 관찰 환자 수준 데이터에 대한 다양한 데이터 분석 사용 사례를 지원하는 광범위한 오픈 소스 툴을 제공한다. 이러한 툴의 공통점은 공통 데이터 모델(CDM)을 사용하여 하나 이상의 데이터베이스와 상호 작용할 수 있다는 것이다. 또한, 이러한 툴은 다양한 사용 사례 use case에 대한 분석을 표준화한다. 처음부터 시작하는 것이 아니라 표준 템플릿을 작성함으로써 분석을 구현할 수 있다. 이렇게 하면 분석을 더 쉽게 수행할 수 있고, 재현성과 투명성을 향상할 수 있다. 예를 들어, 발생률을 계산하는 방법은 무한에 가까운 수가 있는 것처럼 보이지만, 이러한 방법은 몇 가지 선택 사항으로 OHDSI 툴에 지정할 수 있으며, 동일한 선택을 하는 사람은 동일한 방법으로 발생률을 계산할 것이다.

이 장에서는 먼저 분석을 실행하기 위해 선택할 수 있는 다양한 방법과 분석에서 어떤 전략을 사용할 수 있는지 설명한다. 그런 다음 다양한 OHDSI 툴과 다양한 사용 사례에 적합한 방법을 검토한다.

8.1 분석 구현

그림 8.1은 CDM을 사용하여 데이터베이스에 대한 연구를 구현하도록 선택할 수 있는 다양한 방법을 보여준다.

연구를 이행하는 데는 세 가지 주요 접근법이 있다. 첫 번째는 OHDSI가 제공하는 어떤 툴도 사용하지 않고 사용자가 직접 코드를 작성하는 것이다. R, SAS 또는 다른 언어로 새로운 분석 코드를 작성할 수 있다. 이는 최대의 유연성을 제공하며, 특정 분석이 우리의 툴에 의해 뒷받침되지 않는

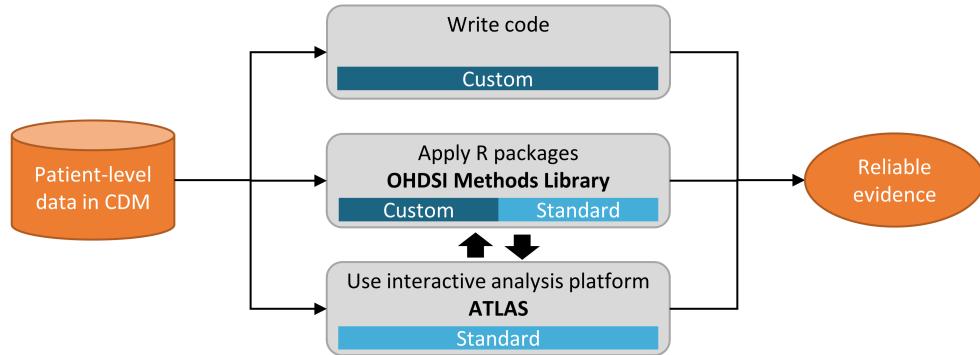


Figure 8.1: CDM의 데이터에 대한 분석을 구현하는 다양한 방법

경우 사실상 유일한 선택사항이 될 수 있다. 그러나 이러한 경로에는 많은 전문적 기술과 시간, 노력이 필요하며, 분석의 복잡성이 증가함에 따라 코드의 오류를 피하기 어려워진다.

두 번째 접근방식은 R을 이용하고, OHDSI Methods Library의 패키지를 이용하는 것이다. 최소한 9장에 설명된 SqlRender 및 DatabaseConnector 패키지를 사용하여 PostgreSQL, SQL Server, 그리고 Oracle과 같은 다양한 데이터베이스 플랫폼에서 동일한 코드를 실행할 수 있다. CohortMethod와 PatientLevelPrediction과 같은 다른 패키지는 자신의 코드로 호출할 수 있는 CDM에 대한 고급 분석을 위한 R 기능을 제공한다. 이것은 여전히 많은 기술적 전문지식이 필요하지만, Methods Library의 검증된 구성요소를 다시 사용함으로써 사용자가 모든 코드를 다 짜는 것보다 더 효율적이고 오류가 덜 발생할 수 있다.

세 번째 접근법은 프로그래머가 아닌 사람이 다양한 분석을 효율적으로 수행할 수 있도록 해주는 웹 기반 툴인 대화형 분석 플랫폼 ATLAS에 의존한다. ATLAS는 Method Libraries를 사용하지만, 분석을 설계하기 위한 간단한 그래픽 인터페이스를 제공하며 많은 경우 분석을 실행하는 데 필요한 R 코드를 생성한다. 그러나 ATLAS는 Methods Library에서 사용할 수 있는 모든 옵션을 다 지원하지는 않는다. 대부분의 연구가 ATLAS를 통해 수행될 수 있을 것으로 예상되지만, 일부 연구는 두 번째 접근방식이 제공하는 유연성을 필요로 할 수 있다.

ATLAS와 Methods Library는 독립적이지 않다. ATLAS에서 호출할 수 있는 더 복잡한 분석 중 일부는 Methods Library의 패키지에 대한 호출을 통해 실행된다. 마찬가지로 Methods Library에 사용되는 코호트는 ATLAS에서 설계되는 경우가 많다.

8.2 분석 전략

사용자 정의 코드를 사용하거나 Methods Library의 표준 분석 코드를 사용하여 CDM에 대한 분석을 구현하는 것 외에도, 그러한 분석 기법을 사용하여 근거를 생성하는 데에는 여러 가지 전략이

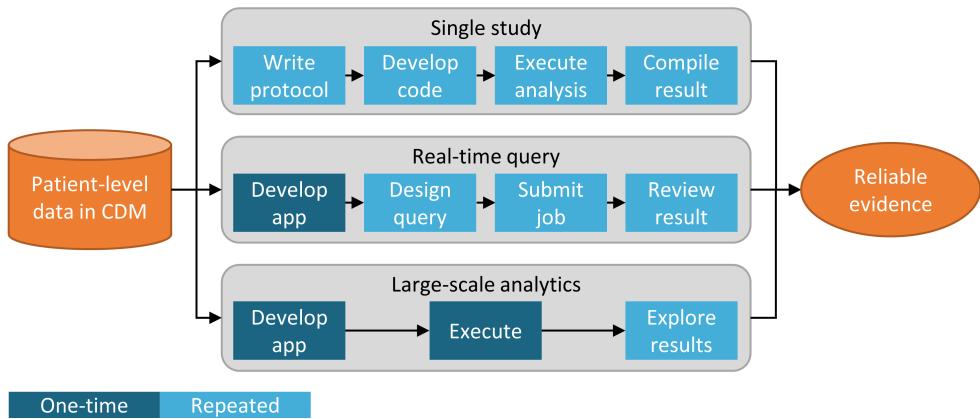


Figure 8.2: (임상적) 질문에 대한 근거를 생성하기 위한 전략

있다. 그림 8.2은 OHDSI에 채택된 세 가지 전략을 보여준다.

첫 번째 전략은 모든 분석을 하나의 개별적인 연구로 본다. 분석은 프로토콜에 미리 지정되어야 하고, 코드로 구현되어야 하며, 데이터에 대해 실행되어야 하며, 그 후에 결과를 컴파일하고 해석할 수 있어야 한다. 모든 질문에 대해 모든 단계를 반복해야 한다. 그러한 분석의 예로는 phenytoin과 비교하여 levetiracetam과 관련된 혈관부종angioedema의 위험에 대한 OHDSI 연구가 있다. (Duke et al., 2017) 이 연구에서, 프로토콜이 처음으로 작성되었고, OHDSI Methods Library를 이용한 분석 코드가 OHDSI 네트워크를 통해 개발되어 실행되었으며, 결과를 편집하여 저널 간행물에 배포하였다.

두 번째 전략은 사용자가 특정 종류의 질문에 실시간으로 또는 거의 실시간으로 답할 수 있는 애플리케이션을 개발한다. 애플리케이션이 개발되면 사용자는 애플리케이션과 상호 작용하여 쿼리를 정의하고 제출하고 결과를 볼 수 있다. 이 전략의 예로는 ATLAS의 코호트 정의 및 생성 툴이 있다. 이 툴은 사용자가 다양한 수준의 복잡한 코호트 정의를 내리고 원하는 데이터베이스에 대해 실행함으로써 얼마나 많은 환자가 다양한 포함 기준 및 제외 기준을 충족하는지 알 수 있게 한다.

세 번째 전략은 비슷하게 한 클래스의 질문 a class of questions에 초점을 맞추지만, 일단 작동하기 시작하면 그 클래스에 속한 모든 의문점에 대해 광범위하고 철저하게 모든 근거를 남김없이 생성하려고 시도한다. 사용자는 다양한 인터페이스를 통해 필요에 따라 (미리 생성된) 근거를 탐색할 수 있다. 한 예로 우울증 치료의 영향에 대한 OHDSI 연구가 있다. (Schuemie et al., 2018b) 이 연구에서 모든 우울증 치료는 4개의 큰 관찰 데이터베이스에서 큰 규모의 임상결과 집합에 대해 비교된다. 광범위한 연구 진단과 함께 경험적으로 보정된 17,718개의 위험비hazard ratio를 포함한 전체 결과는 대화형 웹 앱에서 이용할 수 있다.¹

¹<http://data.ohdsi.org/SystematicEvidence/>

The screenshot shows the ATLAS web application interface. On the left is a dark sidebar with various navigation links: Home, Data Sources, Search, Concept Sets, Cohort Definitions (which is selected), Characterizations, Cohort Pathways, Incidence Rates, Profiles, Estimation, Prediction, Jobs, Configuration, and Feedback. Below this sidebar, there is a note about Apache 2.0 open source software and the OHDSI logo.

The main content area has a title "Cohort #1770710" and a subtitle "New users of ACE inhibitors as first-line monotherapy for hypertension". It includes a toolbar with icons for Definition, Concept Sets, Generation, Reporting, Export, and Messages (with 3 notifications).

Cohort Entry Events: This section allows defining cohort entry events based on drug exposure criteria. It shows a dropdown for "a drug exposure of" set to "ACE inhibitors" and a condition "for the first time in the person's history". Buttons for "Add Initial Event", "Add attribute...", and "Delete Criteria" are available. There are also fields for continuous observation days before and after the event index date, and a limit on initial events per person.

Inclusion Criteria: This section lists two inclusion criteria: "1. Has hypertension diagnosis in 1 yr prior to treatment" and "2. Has no prior antihypertensive drug exposures in medical records". A "New inclusion criteria" button is present.

Figure 8.3: ATLAS 사용자 인터페이스

8.3 ATLAS

ATLAS는 CDM 형식으로 표준화된 환자 수준 관찰 데이터에 대한 분석 설계와 실행을 도와주는 OHDSI 커뮤니티에서 개발한 무료 웹 기반 툴이다. ATLAS는 OHDSI WebAPI와 함께 웹 애플리케이션으로 배포되며 일반적으로 Apache Tomcat에서 호스팅 된다. 실시간 분석을 수행하려면 CDM에 있는 환자 수준 데이터에 액세스해야 하므로 일반적으로 조직의 방화벽 내부에 설치된다. 그러나 공용 ATLAS²도 있으며, 이 ATLAS 인스턴스는 몇 개의 소규모 시뮬레이션 데이터 세트에만 액세스할 수 있지만, 여전히 테스트와 훈련을 포함한 여러 용도로 사용할 수 있다. ATLAS의 공개 인스턴스를 사용하여 효과 추정 또는 예측 연구를 완전히 정의하고, 연구를 실행하기 위한 R 코드를 자동으로 생성할 수도 있다. 이 코드는 ATLAS와 WebAPI를 설치할 필요 없이 사용 가능한 CDM이 있는 모든 환경에서 실행될 수 있다.

ATLAS 스크린샷은 그림 8.3에 제공된다. 왼쪽에는 ATLAS에서 제공하는 다양한 기능을 보여주는 내비게이션 바가 있다:

Data Sources 데이터 원천Data sources은 ATLAS 플랫폼 내에서 구성한 각 데이터 원본에 대해 기술적이고 표준화된 보고 기능을 제공한다. 이 기능은 대규모 분석 전략을 사용한다. 모든 기술 통계량은 사전에 계산된 것이다. 데이터 출처는 11장에서 논한다.

²<http://www.ohdsi.org/web/atlas>

Vocabulary Search ATLAS는 OMOP 표준용어집을 검색하고 탐색하여 그 용어 안에 존재하는 개념 concept과 데이터 소스에 대한 표준 분석에서 해당 concept을 적용하는 방법을 이해할 수 있는 기능을 제공한다. 이 특성은 5장에서 논한다.

Concept Sets 개념 집합 concept set은 표준화된 분석에서 사용할 concept set을 식별하는 데 사용할 수 있는 논리 표현식의 집합을 만들 수 있게 한다. concept set은 단순한 코드나 값 리스트보다 더 정교하게 만들어진다. concept set은 사용자가 용어 계층에 관련 concept을 포함하거나 배제할 수 있도록 하는 논리적 지표와 함께 표준화된 용어에서 나온 여러 concept으로 구성되어 있다. 용어를 검색하고, concept set을 식별하며, concept set을 해결하기 위해 사용할 논리를 명시하는 것은 분석 계획에서 흔히 접하는 모호한 의학 언어를 명확히 정의할 수 있게 하는 강력한 메커니즘을 제공한다. 이러한 concept set은 ATLAS 내에 저장한 다음 코호트 정의 또는 분석 규격의 일부로 분석 내내 사용할 수 있다.

Cohort Definitions 코호트 정의 Cohort Definition은 일정 기간 하나 이상의 기준을 충족하는 일련의 사람을 구성할 수 있게 하며, 이러한 코호트는 이후 모든 분석 시 입력의 기초basis of input가 된다. 이 특성은 10장에서 논한다.

Characterizations 특성은 당신이 정의한 하나 이상의 코호트를 보고 그 환자군에 대한 특성을 요약할 수 있는 분석 기능이다. 이 기능은 실시간 쿼리 전략을 사용하며, 11장에서 논한다.

Cohort Pathways 코호트 경로(Cohort pathways)는 하나 이상의 인구집단 내에서 발생하는 임상 사건의 순서를 살펴볼 수 있는 분석 툴이다. 이 기능은 실시간 쿼리 전략을 사용하며, 11장에서 논한다.

Incidence Rates 발생률은 관심 인구 (코호트) 내에서 임상 결과의 발생률을 추정할 수 있는 툴이다. 이 기능은 실시간 쿼리 전략을 사용하며, 11장에서 논한다.

Profiles 프로파일은 개별 환자에 대해 종적 관찰 데이터를 탐색하여 특정 개인 내에서 일어나는 일을 요약할 수 있는 툴이다. 이 기능은 실시간 쿼리 전략을 사용한다.

Population Level Estimation 추정은 비교 코호트 설계를 사용하여 인구 수준 효과 추정 연구를 정의할 수 있는 기능이며, 여기서 하나 이상의 대상 코호트와 비교 코호트 간의 비교를 통해 일련의 결과에 대해 탐색할 수 있다. 이 기능은 코딩이 필요하지 않으므로 실시간 쿼리 전략을 구현한다고 말할 수 있으며, 12장에서 논의한다.

Patient Level Prediction 예측은 주어진 대상 노출 군 내에서 임상 결과를 예측할 수 있는 환자 수준 예측 분석을 수행하기 위해 기계 학습 알고리즘을 적용할 수 있는 기능이다. 이 기능은 코딩이 필요하지 않음으로 실시간 쿼리 전략을 구현한다고 할 수 있으며, 13장에서 논한다.

Jobs WebAPI를 통해 실행 중인 프로세스의 상태를 탐색하려면 이 기능을 선택하라. 각각의 작업 job은 종종 코호트를 생성하거나 코호트 특성 보고서를 생성하는 것과 같은 장기 실행 과정일 수 있다.

Configuration 소스 구성 섹션에 구성된 데이터 소스를 검토하려면 구성 메뉴 항목을 선택하라.

Feedback 피드백 링크는 ATLAS의 이슈 로그로 이동 시켜 새로운 이슈를 기록하거나 기존 이슈를 검색할 수 있도록 해준다. 새로운 기능이나 개선사항에 대한 아이디어가 있다면, 이것은 개발 커뮤니티에 대한 참고 사항이기도 하다.

8.3.1 보안

ATLAS와 WebAPI는 전체 플랫폼 내의 기능 또는 데이터 소스에 대한 액세스를 제어하기 위한 세분화된 보안 모델을 제공한다. 이 보안 시스템은 Apache Shiro 라이브러리를 활용하여 구축된다. 보안 시스템에 대한 추가 정보는 온라인 WebAPI 보안 위키에서 찾을 수 있다.³

8.3.2 설명서

ATLAS에 대한 설명서는 ATLAS GitHub repository wiki.⁴에 있다. 이 위키에는 온라인 비디오 튜토리얼에 대한 링크뿐만 아니라 다양한 애플리케이션 기능에 대한 정보가 포함되어 있다.

8.3.3 설치 방법

ATLAS 설치는 OHDSI WebAPI와 함께 수행된다. 각 구성 요소의 설치 가이드는 ATLAS GitHub 저장소 설정 가이드⁵ 및 WebAPI GitHub 저장소 설치 가이드⁶에서 찾아볼 수 있다.

8.4 Methods Library

The OHDSI Methods Library는 그림 8.4에 표시된 오픈 소스 R 패키지의 모음이다.

패키지는 완전한 관찰 연구를 수행하기 위해 함께 사용할 수 있는 R 기능을 제공하며, CDM의 데이터에서 시작하여 결과 추정치와 이를 뒷받침하는 통계, 수치 및 표를 제공한다. 패키지는 CDM의 관찰 데이터와 직접 상호작용하며, 단순히 9장에서 설명한 대로 완전한 사용자 정의 분석에 대한 플랫폼 간 호환성을 제공하는 데 사용하거나, 인구 특성화를 위한 고급 표준화 분석 (11장 참조), 인구 수준 효과 추정 (12장 참조) 및 환자 수준 예측 (13장 참조) 을 제공할 수 있다. The Methods Library는 (이전 또는 진행 중인 연구에서 학습한) 투명성, 재현성, 그뿐만 아니라 “특정 맥락에서 방법론의 작동 특성operating characteristics 측정” 및 이어지는 “methods로부터 생성된 측정치의 경험적 교정empirical calibration”과 같은 관찰 데이터 및 관찰 연구 설계의 사용을 위한 모범 사례를 지원한다.

Method Library는 이미 발표된 많은 임상 연구 (Boland et al., 2017; Duke et al., 2017; Ramcharan

³<https://github.com/OHDSI/WebAPI/wiki/Security-Configuration>

⁴<https://github.com/OHDSI/ATLAS/wiki>

⁵<https://github.com/OHDSI/Atlas/wiki/Atlas-Setup-Guide>

⁶<https://github.com/OHDSI/WebAPI/wiki/WebAPI-Installation-Guide>

Prediction and estimation methods	<p>Cohort Method New-user cohort studies using large-scale regression for propensity and outcome models</p> <p>Patient Level Prediction Build and evaluate predictive models for user-specified outcomes, using a wide array of machine learning algorithms.</p>	<p>Self-Controlled Case Series Self-Controlled Case Series analysis using few or many predictors, includes splines for age and seasonality.</p> <p>Case-control Case-control studies, matching controls on age, gender, provider, and visit date. Allows nesting of the study in another cohort.</p>	<p>Self-Controlled Cohort A self-controlled cohort design, where time preceding exposure is used as control.</p> <p>Case-crossover Case-crossover design including the option to adjust for time-trends in exposures (so-called case-time-control).</p>
Method characterization	<p>Empirical Calibration Use negative control exposure-outcome pairs to profile and calibrate a particular analysis design.</p>	<p>Method Evaluation Use real data and established reference sets as well as simulations injected in real data to evaluate the performance of methods.</p>	<p>Evidence Synthesis Combining study diagnostics and results across multiple sites.</p>
Supporting packages	<p>Database Connector Connect directly to a wide range of database platforms, including SQL Server, Oracle, and PostgreSQL.</p> <p>ParallelLogger Support for parallel computation with logging to console, disk, or e-mail.</p>	<p>Sql Render Generate SQL on the fly for the various SQL dialects.</p>	<p>Cyclops Highly efficient implementation of regularized logistic, Poisson and Cox regression.</p> <p>Feature Extraction Automatically extract large sets of features for user-specified cohorts using data in the CDM.</p>

Figure 8.4: The OHDSI Methods Library의 패키지

et al., 2017; Weinstein et al., 2017; Wang et al., 2017; Ryan et al., 2017, 2018; Vashisht et al., 2018; Yuan et al., 2018; Johnston et al., 2019)와 방법론 연구에 사용되어 왔다. (Schuemie et al., 2014, 2016; Reps et al., 2018; Tian et al., 2018; Schuemie et al., 2018a,b; Reps et al., 2019) The Methods Library에서 방법론 구현의 타당성은 17장에 설명되어 있다.

8.4.1 대규모 분석 지원

모든 패키지에 통합된 한 가지 주요 특징은 많은 분석을 효율적으로 실행할 수 있는 능력이다. 예를 들어 인구 수준 추정을 수행할 때 CohortMethod 패키지는 다양한 분석 설정을 사용하여 많은 노출 exposure 및 결과 outcome에 대한 효과 크기 추정치 effect-size estimates를 계산할 수 있도록 하며, 패키지는 필요한 모든 중간 및 최종 데이터 세트를 계산하는 최적의 방법을 자동으로 선택한다. “공변량 추출 extraction of covariates”이나 하나의 대상군-비교군 쌍 target-comparator pair과 복수의 결과에 사용되는 “성향 모델 맞춤 fitting a propensity model”과 같이 재사용할 수 있는 단계는 한번만 실행된다. 가능한 경우 계산 자원의 사용을 극대화하기 위해 병렬처리 연산이 가능하다.

이러한 효율적 계산은 대규모 분석을 가능하게 하여 한꺼번에 많은 질문에 답할 수 있으며, 또한 제어 가설(예를 들어, 음성대조군negative controls을 포함해 방법론의 작동 특성을 측정하고 18장에 기술된 경험적 교정을 수행하는 데 필수적이다.

8.4.2 빅데이터 지원

The Methods Library는 또한 매우 큰 데이터베이스에 대해 실행하고 대량의 데이터를 포함하는 계산을 수행할 수 있도록 설계되었다. 이는 다음과 같은 세 가지 방법으로 달성되었다:

1. 대부분의 데이터 조작은 데이터베이스 서버에서 수행된다. 분석은 일반적으로 데이터베이스에 있는 전체 데이터의 극히 일부만을 필요로 하며 Methods Library는 SqlRender 및 DatabaseConnector 패키지를 통해 서버에서 고급 작업을 수행하여 관련 데이터를 사전 처리하고 추출할 수 있도록 한다.
2. 대용량 로컬 데이터 객체는 메모리 효율적인 방식으로 저장된다. 로컬 시스템으로 다운로드 되는 데이터의 경우 Method Library는 ff 패키지를 사용하여 대용량 데이터 객체를 저장하고 작업한다. 이것은 우리가 메모리를 직접적으로 사용하는 것보다 훨씬 더 큰 데이터로 작업할 수 있게 해준다.
3. 필요한 곳에 고성능 컴퓨팅을 적용한다. 예를 들어, Cyclops 패키지는 대량의 변수와 관측치로 인해 다른 방법으로는 할 수 없는 대규모 회귀 분석을 수행할 수 있는 매우 효율적인 회귀 엔진을 구현했으며 Methods Library 전체에서 이 엔진을 사용할 수 있다.

8.4.3 문서화

R은 패키지를 문서화하는 표준화된 방법을 제공한다. 각 패키지에는 패키지에 포함된 모든 기능과 데이터 세트를 문서화하는 패키지 설명서가 있다. 모든 패키지 매뉴얼은 the Methods Library 웹 사이트⁷를 통해 패키지 GitHub 온라인 저장소를 통해 사용할 수 있으며 CRAN을 통해 사용할 수 있는 패키지의 경우는 CRAN에서 찾을 수 있다. 또한, R 내에서 물음표를 사용하여 패키지 설명서를 참조할 수 있다. 예를 들어 DatabaseConnector 패키지를 로드한 후 `?connect` 명령을 입력하면 “연결connect” 기능에 대한 문서가 나타난다.

패키지 설명서 외에도 많은 패키지가 *vignette*를 제공한다. Vignettes는 특정 작업을 수행하기 위해 어떻게 패키지를 사용할 수 있는지 설명하는 긴 형식의 문서다. 예를 들어, 하나의 vignette⁸은 CohortMethod 패키지를 사용하여 여러 가지 분석을 효율적으로 수행하는 방법을 설명한다. 또한 Vignettes는 Methods Library 웹 사이트, 패키지 GitHub 저장소를 통해 찾을 수 있으며, CRAN을 통해 이용할 수 있는 패키지의 경우 CRAN에서 찾을 수 있다. Vignettes는 the Methods Library 웹 사이트를 통해 패키지 GitHub 온라인 저장소를 통해 사용할 수 있으며 CRAN을 통해 사용할 수 있는 패키지의 경우 CRAN에서 찾을 수 있다.

8.4.4 시스템 요구 사항

시스템 요구 사항을 논의할 때 두 가지 컴퓨팅 환경을 고려해야 한다: 데이터베이스 서버 및 분석 워크스테이션

데이터베이스 서버는 의료 데이터를 CDM 형식으로 보관해야 한다. Method Library는 전통적인 데이터베이스 시스템 (PostgreSQL, Microsoft SQL Server, 그리고 Oracle), 병렬 데이터 웨어하우스 (Microsoft APS, IBM Netezza, 그리고 Amazon RedShift) 및 빅데이터 플랫폼 (Impala를 통한 Hadoop, 그리고 Google BigQuery)을 포함한 광범위한 데이터베이스 관리 시스템을 지원한다.

분석 워크스테이션은 Methods Library가 설치되어 실행되는 곳이다. 이것은 누군가의 랩탑과 같은 로컬 시스템이나 RStudio Server를 실행하는 원격 서버일 수 있다. 모든 경우에, R은 RStudio와 함께 설치되어야 한다. Methods Library는 또한 Java가 설치되어야 한다. 분석 워크스테이션은 데이터베이스 서버에 연결할 수 있어야 하며, 특히 이 사이의 방화벽은 데이터베이스 서버 접근 포트를 워크스테이션에 개방해야 한다. 일부 분석은 계산 집약적일 수 있으므로 여러 개의 코어와 충분한 메모리를 갖는 것이 분석 속도를 높이는 데 도움이 될 수 있다. 적어도 4개의 코어와 16GB의 메모리를 가질 것을 추천한다.

⁷<https://ohdsi.github.io/MethodsLibrary>

⁸<https://ohdsi.github.io/CohortMethod/articles/MultipleAnalyses.html>



Figure 8.5: CRAN으로부터 R 다운로드

8.4.5 설치 방법

다음은 OHDSI R 패키지를 실행하는 데 필요한 환경을 설치하는 단계다. 다음 네 가지를 설치해야 한다:

1. **R**은 통계 컴퓨팅 환경이다. 그것은 주로 명령어 인터페이스인 기본 사용자 인터페이스와 함께 제공된다.
2. **RTools**는 Windows에서 소스로부터 R 패키지를 만드는 데 필요한 프로그램의 모음이다.
3. **RStudio**는 R을 사용하기 쉽게 하는 통합 개발 환경Integrated Development Environment(IDE)이다. 여기에는 코드 편집기, 디버깅 및 시각화 툴이 포함되어 있다. 사용하기 편한 유저인터페이스를 원한다면 사용하기를 권한다.
4. **Java**는 OHDSI R 패키지의 일부 구성 요소 (예를 들어, 데이터베이스에 연결하는 데 필요한 구성 요소)를 실행하는 데 필요한 컴퓨팅 환경이다.

아래에서는 Windows 환경에 이러한 각 항목을 설치하는 방법에 관해 설명한다.



Windows에서 R과 Java는 32-bit 및 64-bit 아키텍처를 모두 제공한다. 두 아키텍처에 R을 설치하는 경우, 반드시 두 아키텍처에 모두 Java를 설치해야 한다. R은 64-bit 버전만 설치하는 것을 추천한다.

R 설치하기

1. <https://cran.r-project.org/>으로 이동하여, “Download R for Windows”를 클릭 후 “base”를 클릭한 다음 그림 8.5에 표시된 다운로드 링크를 클릭하라.
2. 다운로드가 완료된 후 설치 프로그램을 실행하라. 다음 두 가지 예외를 제외하고 모든 곳에서 기본 옵션을 사용하라. 첫째, 프로그램 파일 폴더에 설치하지 않는 것이 좋다. 대신 R을 그림 8.6과 같이 C 드라이브의 하위 폴더로 만들라. 둘째, R과 Java 간의 아키텍처 차이로 인한

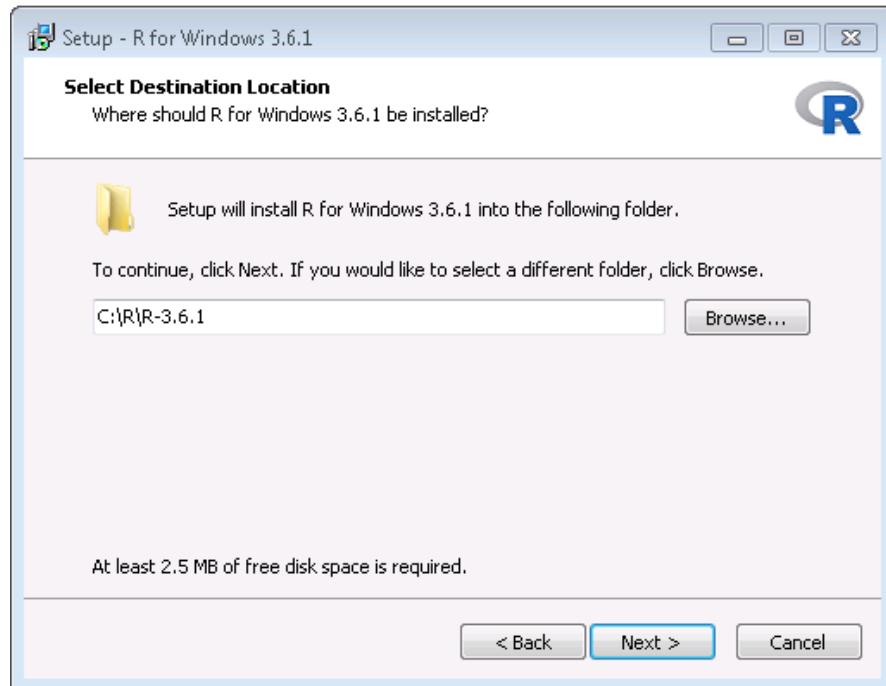


Figure 8.6: R의 대상 폴더 설정하기.

문제를 방지하려면 그림 8.7과 같이 32-bit 아키텍처를 비활성화하라.

완료되면 시작 메뉴에서 R을 선택할 수 있어야 한다.

RTools 설치하기

1. <https://cran.r-project.org/>으로 이동하여 “Windows용 R 다운로드”를 클릭한 다음 “Rtools”를 클릭하고 다운로드할 최신 버전의 RTools를 선택하라.
2. 다운로드가 완료된 후 설치 프로그램을 실행하라. 어디에서나 기본 옵션을 선택하라.

RStudio 설치하기

1. <https://www.rstudio.com/>으로 이동하여, “Download RStudio”을 선택 (또는 “RStudio”에서 “Download” 버튼을 선택) 하고, 무료 버전을 선택한 후, 그림 8.8과 같이 Windows용 설치 프로그램을 다운로드하라.
2. 다운로드한 후, 설치 관리자를 시작하고, 모든 곳에서 기본 옵션을 선택하라.

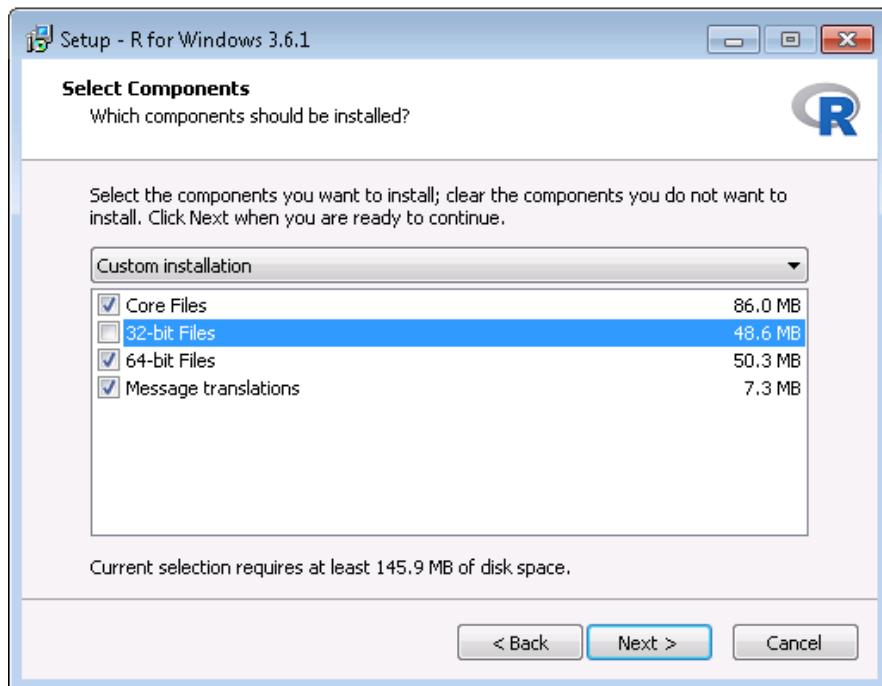


Figure 8.7: 32-bit 버전의 R을 사용하지 않도록 설정하기.

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.2.1335 - Windows 7+ (64-bit)	126.9 MB	2019-04-08	d0e2470f1
RStudio 1.2.1335 - Mac OS X 10.12+ (64-bit)	121.1 MB	2019-04-08	6c570b0e2
RStudio 1.2.1335 - Ubuntu 14/Debian 8 (64-bit)	92.2 MB	2019-04-08	c1b07d051

Figure 8.8: RStudio 다운로드

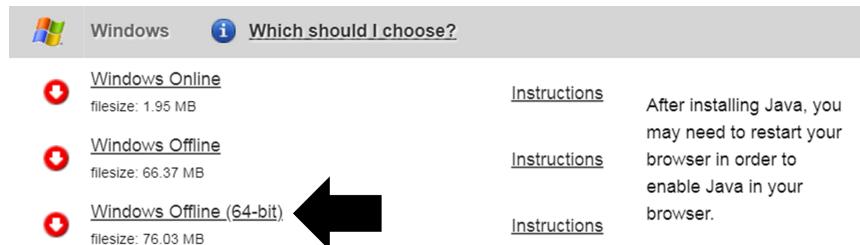


Figure 8.9: Java 다운로드

Java 설치하기

1. <https://java.com/en/download/manual.jsp>으로 이동하여, 그림 8.9와 같이 Windows 64-bit installer를 선택하라. 32-bit 버전의 R을 설치한 경우 반드시 다른 32-bit 버전의 Java도 설치해야 한다.
2. 다운로드한 후 설치 프로그램을 실행하라.

설치 검수하기

이제 시작할 준비를 해야 하지만, 그 전에 확실히 해야 한다. RStudio를 시작하고 및 아래의 내용을 입력하자.

```
install.packages("SqlRender")
library(SqlRender)
translate("SELECT TOP 10 * FROM person;", "postgresql")

## [1] "SELECT * FROM person LIMIT 10;"
```

이 기능은 Java를 사용하기 때문에, 만약 모든 것이 잘 된다면, R과 Java가 모두 올바르게 설치되었다는 것을 알 수 있다!

또 다른 테스트는 소스 패키지를 제대로 구축할 수 있는지 확인하는 것이다. 다음 R 코드를 실행하여 OHDSI GitHub 저장소에서 CohortMethod 패키지를 설치하라:

```
install.packages("drat")
drat::addRepo("OHDSI")
install.packages("CohortMethod")
```

8.5 설치 전략

ATLAS 및 Method Library를 포함한 전체 OHDSI 툴 스택을 기관에 모두 설치하는 것은 어려운 작업이다. 의존성 높은 구성 요소를 많이 고려해야 하고, 설정해야 할 환경이 많다. 이 때문에 두 이니셔티브 (Broadsea와 AWS(Amazon Web Services))는 일부 가상화 형태를 이용해 전체 스택을 하나의 패키지로 설치할 수 있는 통합 설치 전략을 개발했다.

8.5.1 Broadsea

Broadsea⁹는 Docker 컨테이너 기술을 사용한다. ¹⁰ OHDSI 툴은 라이브러리간 의존성과 함께 Docker Image라는 단일 휴대용 이진 파일로 패키징된다. 그러면 이 이미지는 Docker 엔진 서비스에서 실행되고, 모든 소프트웨어가 설치되어 실행 준비가 된 가상 시스템virtual machine을 생성할 수 있다. Docker 엔진은 Microsoft Windows, MacOS, Linux를 포함한 대부분의 운영 체제에 사용할 수 있다. Broadsea Docker 이미지에는 Methods Library와 ATLAS를 포함한 주요 OHDSI 툴이 포함되어 있다.

8.5.2 Amazon AWS

Amazon은 버튼 클릭 한 번으로 OHDSI 환경을 AWS 클라우드 컴퓨팅 환경에서 바로 인스턴스화할 수 있는 두 가지 환경, 즉 OHDSI-in-a-Box¹¹와 OHDSIonAWS.¹²을 준비했다.

OHDSI-in-a-Box는 특별히 학습 환경으로 만들어졌으며, OHDSI 커뮤니티에서 제공하는 대부분의 튜토리얼에 사용된다. 그것은 많은 OHDSI 툴, 샘플 데이터 세트, RStudio 및 기타 지원 소프트웨어를 저렴한 단일 Windows 가상 머신에 포함했다. PostgreSQL 데이터베이스는 CDM을 저장하고 ATLAS의 중간 결과를 저장하는 데 사용된다. OMOP CDM 데이터 매핑과 ETL 툴도 OHDSI-in-a-Box에 포함되어 있다. OHDSI-in-a-Box 아키텍처는 그림 8.10에 나타나 있다.

OHDSIonAWS는 기관이 그들의 데이터 분석을 수행하는 데 사용할 수 있는 엔터프라이즈급, 다중 사용자, 확장할 수 있는 내결합성 OHDSI 환경을 위한 참조 아키텍처이다. 여기에는 몇 가지 샘플 데이터 세트가 포함되어 있으며 기관의 실제 의료 데이터를 자동으로 적재할 수도 있다. 데이터는 OHDSI 툴에 의해 지원되는 Amazon Redshift 데이터베이스 플랫폼에 배치된다. ATLAS의 중간 결과는 PostgreSQL 데이터베이스에 저장된다. 프런트 엔드에서 사용자는 웹 인터페이스(leveraging RStudio Server)를 통해 ATLAS와 RStudio에 접근할 수 있다. RStudio에는 OHDSI Methods Library가 이미 설치되어 있으며, 데이터베이스에 연결하는 데 사용할 수 있다. OHDSIonAWS를

⁹<https://github.com/OHDSI/Broadsea>

¹⁰<https://www.docker.com/>

¹¹<https://github.com/OHDSI/OHDSI-in-a-Box>

¹²<https://github.com/OHDSI/OHDSIonAWS>

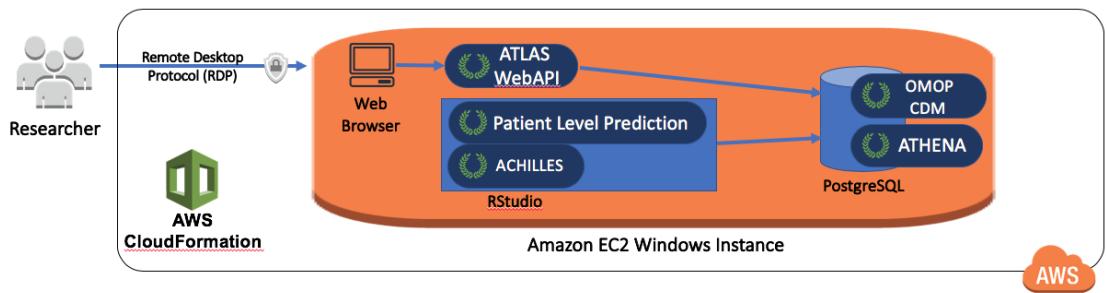


Figure 8.10: OHDSI-in-a-Box을 위한 Amazon Web Services 아키텍처

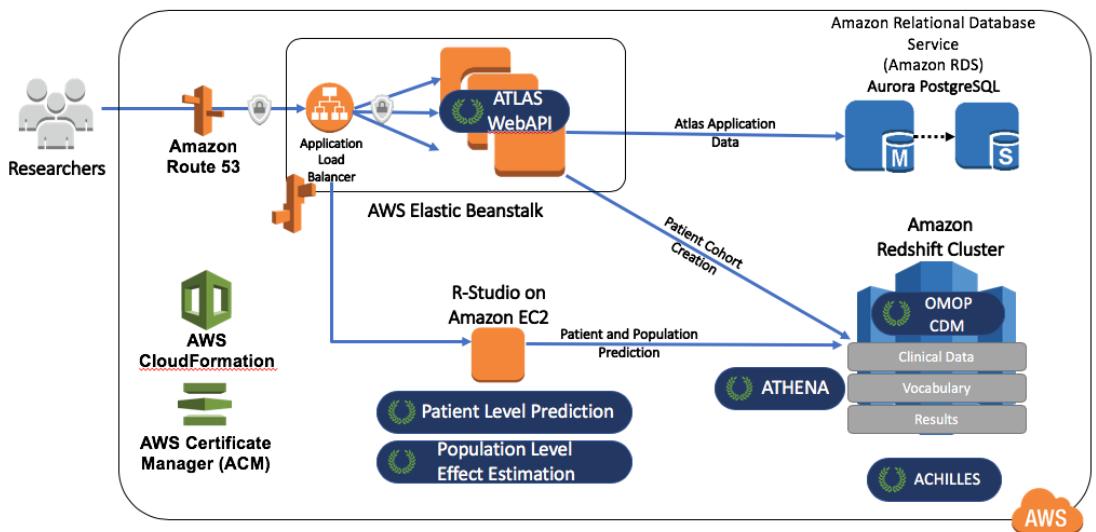


Figure 8.11: OHDSIonAWS를 위한 Amazon Web Services 아키텍처

배포하는 자동화 툴은 오픈 소스로서 기관의 관리 툴과 모범 사례를 포함하도록 사용자 정의할 수 있다. OHDSIonAWS에 대한 아키텍처는 그림 8.11에 설명되어 있다.

8.6 요약



- 다음과 같은 방식으로 CDM 데이터에 대한 분석을 수행할 수 있다.
 - * 사용자가 직접 분석코드 작성
 - * OHDSI Method Library에서 R 패키지를 사용하여 코드 작성
 - * 분석 코드 작성 없이 대화형 분석 플랫폼 ATLAS를 사용
- OHDSI 툴은 다양한 분석 전략을 사용한다.
 - * 단일 연구

- * 실시간 쿼리
- * 대규모 분석
- 대부분의 OHDSI 분석 툴이 다음에 내장되어 있다.
 - * 대화형 분석 플랫폼 ATLAS
 - * OHDSI Methods Library R 패키지
- OHDSI 툴의 구축을 촉진하는 몇 가지 전략이 존재한다.

Chapter 9

SQL과 R

Chapter leads: Martijn Schuemie & Peter Rijnbeek

공통 데이터 모델 Common Data Model(CDM)은 모든 데이터가 필드가 있는 테이블의 레코드로 표시되는 관계형 데이터베이스 모델이다. 이는 일반적으로 PostgreSQL, Oracle, Microsoft SQL Server와 같은 소프트웨어 플랫폼을 사용하여 데이터가 관계형 데이터베이스에 저장된다는 것을 의미한다. 사용자는 ATLAS와 Methods Library 같은 다양한 OHDSI 도구를 통해 데이터베이스에 간접적으로 쿼리를 작성하여 분석을 수행하지만, 적절한 접근 권한이 있으면 이 도구를 사용하지 않고 직접 데이터베이스에 쿼리를 이용해 분석할 수도 있다. 직접 쿼리를 작성하는 주된 이유는 기존 도구가 지원하지 않는 분석을 수행하기 위한 것이다. 그러나 OHDSI 도구는 사용자가 데이터를 적절하게 분석을 할 수 있도록 (전문가가 오랜 시간 고려하여 만든 지침을 따르도록) 설계되어 있고, 직접 쿼리를 작성하다보면 실수를 범할 위험이 더 커진다. 직접 쿼리를 작성할 때는 질의하는 것은 그런 지침을 제공하지 않는다. 관계형 데이터베이스에 쿼리를 작성하는 표준 언어는 Structured Query Language(SQL)이며, 이는 데이터에 대한 변경뿐만 아니라 데이터 추출용 쿼리를 위해 사용할 수 있다. SQL의 기본 명령어는 실제로 표준이고 소프트웨어 플랫폼 전반에 걸쳐 같은 의미가 있지만, 각각의 SQL 플랫폼마다 미묘한 변경되는 고유한 문법을 가지고 있다. 예를 들면, SQL Server에서 PERSON 테이블에서 상위 10개의 행을 검색하려면 다음과 같은 쿼리를 입력한다:

```
SELECT TOP 10 * FROM person;
```

PostgreSQL의 동일한 쿼리는 다음과 같다:

```
SELECT * FROM person LIMIT 10;
```

OHDSI에서는 플랫폼이 사용하는 고유한 문법에 구애받지 않고 모든 OHDSI 데이터베이스에서 동일한 SQL 언어를 사용하고자 한다. 이러한 이유로 OHDSI는 이 장의 뒤에서 논의하게 될, 하나의 표준 문법을 다른 여러 개의 문법으로 번역해줄 수 있는 패키지인 SqlRender를 개발하였다. 이 표준 언어 - **OHDSI SQL** - 는 주로 SQL Server SQL 언어의 하위 집합이다. 이 장에서 제공되는 SQL 문에는 모두 OHDSI SQL을 사용한다.

각 데이터베이스 플랫폼에는 SQL을 사용하여 데이터베이스를 쿼리하기 위한 자체 소프트웨어 도구가 제공된다. OHDSI는 여러 데이터베이스 플랫폼에 연결할 수 있는 하나의 R 패키지인 DatabaseConnector를 개발하였다. DatabaseConnector도 이 장의 뒤에서 논의할 것이다.

따라서 OHDSI 도구를 사용하지 않고도 CDM에 맞게 쿼리를 작성할 수 있지만 DatabaseConnector 및 SqlRender 패키지를 사용하는 것을 권장한다. 이를 통해 한 기관에서 개발한 쿼리를 수정하지 않고도 다른 기관에서 사용할 수 있다. 즉, SQL server를 사용하는 기관이 작성한 쿼리를 PostgreSQL을 사용하는 기관에서도 적용할 수 있다. R 자체는 통계 분석 및 대화식 그래프 생성과 같이 데이터베이스에서 추출된 데이터를 추가로 분석하는 기능도 직접 제공한다.

이 장에서는 독자가 SQL에 대한 기본 지식을 가지고 있다고 가정한다. 먼저 SqlRender 및 DatabaseConnector 사용 방법을 안내한다. 독자가 이 패키지를 사용할 생각이 없다면 이 절은 건너뛰어도 된다. 9.3절에서는 SQL(OHDSI SQL)을 사용하여 CDM에 쿼리하는 방법에 관해 설명한다. 그 다음 절에서는 CDM 쿼리 작성 시 OHDSI 표준 용어를 사용하는 방법을 강조한다. 공개적으로 이용 가능한 CDM에 대해 일반적으로 사용되는 쿼리 모음인 QueryLibrary를 특히 자세히 살펴본다. 발생률을 추정하는 예제 연구로 이 장을 마무리하고 SqlRender 및 DatabaseConnector를 사용하여 이 연구를 구현한다.

9.1 SqlRender

SqlRender 패키지는 Comprehensive R Archive Network (CRAN)에서 이용할 수 있으므로 다음을 사용하여 설치할 수 있다:

```
install.packages("SqlRender")
```

SqlRender는 전통적인 데이터베이스 시스템 (PostgreSQL, Microsoft SQL Server, SQLite, and Oracle), 병렬 데이터 웨어하우스 (Microsoft APS, IBM Netezza, and Amazon RedShift), 빅데이터 플랫폼 (Hadoop through Impala, and Google BigQuery) 을 포함한 다수의 기술 플랫폼을

지원한다.

9.1.1 SQL의 매개 변수화

패키지의 기능 중 하나는 SQL문에 매개 변수를 지원하는 것이다. 때때로 일부 매개 변수에 기반하여 SQL문을 조금씩 변형할 필요가 있다. 일반적인 SQL 문법에서 매개변수 이용은 매우 복잡하며, 매개변수화 지원은 SqlRender 의 매우 강력한 기능이다. SqlRender는 매개 변수를 허용하기 위해 SQL 코드 내에서 간단한 마크업 구문을 제공한다. 매개 변수를 기반으로 SQL을 렌더링하는 것은 `render()` 함수를 사용하여 수행한다.

매개 변수값 대체하기

© 문자는 렌더링 시 실제 매개 변수값과 교환해야 하는 매개 변수 이름을 나타내는 데 사용할 수 있다. 다음 예에서 `a`라고 불리는 변수가 SQL에서 언급되어 있다. 렌더 함수를 호출할 때 이 매개 변수의 값이 정의된다:

```
sql <- "SELECT * FROM concept WHERE concept_id = @a;"  
render(sql, a = 123)
```

```
## [1] "SELECT * FROM concept WHERE concept_id = 123;"
```

대부분의 SQL 문법과 달리 테이블 또는 필드 이름을 값으로 매개 변수화하기 매우 쉽다!

```
sql <- "SELECT * FROM @x WHERE person_id = @a;"  
render(sql, x = "observation", a = 123)
```

```
## [1] "SELECT * FROM observation WHERE person_id = 123;"
```

매개 변수값은 숫자, 문자열, 부울 및 쉼표를 기준으로 항목을 나눈 리스트로 변환된 벡터일 수도 있다.

```
sql <- "SELECT * FROM concept WHERE concept_id IN (@a);"  
render(sql, a = c(123, 234, 345))
```

```
## [1] "SELECT * FROM concept WHERE concept_id IN (123,234,345);"
```

If-Then-Else

때로는 하나 이상의 매개 변수값에 따라 코드 블록을 켜거나 끌 필요가 있다. 이 작업은 {Condition} ? {if true} : {if false} 구문을 사용한다. 조건이 참 또는 1로 평가되면, if true 블록이 사용되고 그렇지 않으면, if false 블록이 표시된다 (있는 경우). (역주 저: 매개변수화 기능은 전반적으로 정말 놀라운 기능이다!)

```
sql <- "SELECT * FROM cohort {@x} ? {WHERE subject_id = 1}"
render(sql, x = FALSE)
```

```
## [1] "SELECT * FROM cohort "
```

```
render(sql, x = TRUE)
```

```
## [1] "SELECT * FROM cohort WHERE subject_id = 1"
```

간단한 비교도 지원된다:

```
sql <- "SELECT * FROM cohort {@x == 1} ? {WHERE subject_id = 1};"
render(sql, x = 1)
```

```
## [1] "SELECT * FROM cohort WHERE subject_id = 1;"
```

```
render(sql, x = 2)
```

```
## [1] "SELECT * FROM cohort ;"
```

IN 연산자도 지원된다:

```
sql <- "SELECT * FROM cohort {@x IN (1,2,3)} ? {WHERE subject_id = 1};"
render(sql, x = 2)
```

```
## [1] "SELECT * FROM cohort WHERE subject_id = 1;"
```

9.1.2 다른 SQL 언어로의 변환

SqlRender 패키지의 또 다른 기능은 OHDSI SQL에서 다른 SQL 언어로 변환하는 것이다. 예를 들면 다음과 같다:

```
sql <- "SELECT TOP 10 * FROM person;"  
translate(sql, targetDialect = "postgresql")
```

```
## [1] "SELECT * FROM person LIMIT 10;"
```

`targetDialect` 매개 변수는 다음과 같은 값을 가질 수 있다: “oracle”, “postgresql”, “pdw”, “redshift”, “impala”, “netezza”, “bigquery”, “sqlite”, “sql server”.



패키지에는 제한된 변환 규칙 세트만 구현되었기 때문에 SQL의 함수 및 구성을 모두 적절하게 번역하지 못할 수 있고, 일부 SQL의 특징은 모든 언어에서 동일하지 않다. OHDSI SQL이 독자적인 새로운 문법으로 개발된 주된 이유이다. OHDSI SQL은 SQL Server 구문을 유지하였다.

최선의 노력에도 불구하고, 지원되는 모든 플랫폼에서 오류 없이 실행될 OHDSI SQL을 작성할 때 고려해야 할 사항이 몇 가지 있다. 다음은 이러한 고려 사항에 대해 자세히 설명한다.

변환에 의해 지원되는 기능 및 구조

다음과 같은 SQL Server 함수는 테스트 되었으며 다양한 언어로 올바르게 변환되는 것으로 확인되었다:

Table 9.1: Functions supported by translate.

Function	Function	Function
ABS	EXP	RAND
ACOS	FLOOR	RANK
ASIN	GETDATE	RIGHT
ATAN	HASHBYTES*	ROUND
AVG	ISNULL	ROW_NUMBER
CAST	ISNUMERIC	RTRIM
CEILING	LEFT	SIN
CHARINDEX	LEN	SQRT
CONCAT	LOG	SQUARE
COS	LOG10	STDEV
COUNT	LOWER	SUM
COUNT_BIG	LTRIM	TAN

Function	Function	Function
DATEADD	MAX	UPPER
DATEDIFF	MIN	VAR
DATEFROMPARTS	MONTH	YEAR
DATETIMEFROMPARTS	NEWID	
DAY	PI	
EOMONTH	POWER	

* Oracle은 특별 권한이 필요하다. SQLite에는 해당하는 것이 없다.

마찬가지로 많은 SQL 구문 구조가 지원된다. 다음은 우리가 잘 번역할 수 있는 표현식의 전체 목록이다:

```
-- Simple selects:
SELECT * FROM table;

-- Selects with joins:
SELECT * FROM table_1 INNER JOIN table_2 ON a = b;

-- Nested queries:
SELECT * FROM (SELECT * FROM table_1) tmp WHERE a = b;

-- Limiting to top rows:
SELECT TOP 10 * FROM table;

-- Selecting into a new table:
SELECT * INTO new_table FROM table;

-- Creating tables:
CREATE TABLE table (field INT);

-- Inserting verbatim values:
INSERT INTO other_table (field_1) VALUES (1);

-- Inserting from SELECT:
```

```

INSERT INTO other_table (field_1) SELECT value FROM table;

-- Simple drop commands:
DROP TABLE table;

-- Drop table if it exists:
IF OBJECT_ID('ACHILLES_analysis', 'U') IS NOT NULL
    DROP TABLE ACHILLES_analysis;

-- Drop temp table if it exists:
IF OBJECT_ID('tempdb..#cohorts', 'U') IS NOT NULL
    DROP TABLE #cohorts;

-- Common table expressions:
WITH cte AS (SELECT * FROM table) SELECT * FROM cte;

-- OVER clauses:
SELECT ROW_NUMBER() OVER (PARTITION BY a ORDER BY b)
    AS "Row Number" FROM table;

-- CASE WHEN clauses:
SELECT CASE WHEN a=1 THEN a ELSE 0 END AS value FROM table;

-- UNIONs:
SELECT * FROM a UNION SELECT * FROM b;

-- INTERSECTIONS:
SELECT * FROM a INTERSECT SELECT * FROM b;

-- EXCEPT:
SELECT * FROM a EXCEPT SELECT * FROM b;

```

문자열 연결

문자열 연결은 SQL Server가 다른 언어보다 덜 구체적인 영역이다. SQL Server에서는 `SELECT first_name + ' ' + last_name AS full_name FROM table`과 같이 작성하지만 Postgres와 Oracle에서는 `SELECT first_name || ' ' || last_name AS full_name FROM table`이라고

작성한다. SqlRender는 연결되는 값이 문자열인지 추측하려고 한다. 위의 예에서 명시적인 문자열(작은따옴표로 둘인 공백) 이 있으므로 번역은 정확할 것이다. 그러나 `SELECT first_name + last_name AS full_name FROM table`과 같이 작성한다면 SqlRender는 두 필드가 문자열이라는 단서가 없으며, 잘못된 더하기 기호를 남겼다. 값이 문자열이라는 또 다른 단서는 “VARCHAR”에 대한 명시적 형 변환이므로 `SELECT last_name + CAST(age AS VARCHAR(3)) AS full_name FROM table`도 올바르게 변환된다. 모호성을 피하려면 `CONCAT()` 함수를 사용하여 두 개 이상의 문자열을 연결하는 것이 가장 좋다.

테이블 별칭과 AS 키워드

많은 SQL 언어는 테이블 별칭을 정의할 때 `AS` 키워드를 사용할 수 있지만, 키워드 없이도 잘 동작 한다. 예를 들어, 이 두 SQL 문은 SQL Server, PostgreSQL, RedShift 등에 적합하다:

```
-- Using AS keyword
SELECT *
FROM my_table AS table_1
INNER JOIN (
    SELECT * FROM other_table
) AS table_2
ON table_1.person_id = table_2.person_id;

-- Not using AS keyword
SELECT *
FROM my_table table_1
INNER JOIN (
    SELECT * FROM other_table
) table_2
ON table_1.person_id = table_2.person_id;
```

그러나 Oracle에서는 `AS` 키워드를 사용하면 오류가 발생한다. 위의 예제 중 첫 번째 쿼리는 실패한다. 따라서 테이블 별칭을 지정할 때 `AS` 키워드를 사용하지 않는 것이 좋다. (참고로 Oracle에서 `AS`를 사용할 수 없는 테이블 별칭과 이 `AS`를 사용해야 하는 필드 별칭을 쉽게 구별할 수 없기 때문에 SqlRender가 이것을 처리하도록 만들 수 없다)

임시 테이블

임시 테이블은 중간 결과를 저장하는 데 매우 유용할 수 있으며 올바르게 사용하면 쿼리 성능을 크게 향상할 수 있다. 대부분의 데이터베이스 플랫폼에서 임시 테이블이라는 매우 좋은 기능을 가지고 있다. 현재 사용자에게만 보이며 세션이 끝나면 자동으로 삭제되고 사용자에게 쓰기 권한이 없어도 생성할 수 있다. 불행히도, Oracle에서는 임시테이블은 기본적으로 영구적인 테이블이며, 데이터의 내부는 현재 사용자에게만 보인다는 차이점만 있다. 이것이 Oracle에서 SqlRender가 다음과 같이 임시 테이블을 에뮬레이션하려고 시도하는 이유이다.

1. 테이블 이름에 임의의 문자열을 추가하여 다른 사용자의 테이블이 충돌하지 않도록 한다.
2. 사용자가 임시 테이블이 작성될 스키마를 지정할 수 있도록 허용한다.

예를 들면:

```
sql <- "SELECT * FROM #children;"  
translate(sql, targetDialect = "oracle", oracleTempSchema = "temp_schema")  
  
## [1] "SELECT * FROM temp_schema.d0ernfshchildren ;"
```

사용자는 temp_schema에 대한 쓰기 권한이 있어야 한다.

또한 Oracle은 테이블 이름이 30자로 제한되어 있다. 세션 아이디를 추가한 후 이름이 너무 길어지기 때문에 **임시 테이블 이름은 최대 22자까지만 허용된다.**

그뿐만 아니라 Oracle의 임시 테이블은 자동 삭제되지 않으므로 Oracle에 임시 테이블 스키마가 쌓이는 것을 방지하기 위해 모든 임시 테이블을 사용한 후에는 명시적으로 TRUNCATE 및 DROP을 해야 한다.

암묵적인 형 변환

SQL Server가 다른 언어보다 덜 명시적인 몇 가지 점 중 하나는 암묵적인 형 변환을 허용한다는 것이다. 예를 들어 이 코드는 SQL Server에서 작동한다:

```
CREATE TABLE #temp (txt VARCHAR);  
  
INSERT INTO #temp  
SELECT '1';  
  
SELECT * FROM #temp WHERE txt = 1;
```

비록 txt는 VARCHAR 필드이고 이것을 정수와 비교하고 있다. 즉, 이는 논리적으로 비교할 수 없는 (문자열과 정수의 비교) 비교이지만, SQL Server는 비교를 허용하기 위해 두 가지 중 하나를 자동으로 올바른 타입으로 변환한다 (보통 정수를 문자열로 ‘암묵적으로’ 변환한다). 이와 대조적으로, PostgreSQL과 같은 다른 언어는 VARCHAR과 INT를 비교하려고 할 때 오류를 일으킬 것이다.

따라서 형 변환은 항상 명시적으로 해야 한다. 위의 마지막에 있는 예는

```
SELECT * FROM #temp WHERE txt = CAST(1 AS VARCHAR);
```

또는 아래와 같이 대체되어야 한다.

```
SELECT * FROM #temp WHERE CAST(txt AS INT) = 1;
```

문자열 비교의 대소문자 구분

SQL Server와 같은 일부 DBMS 플랫폼은 대소문자를 구분하지 않는 비교를 수행하는 반면, PostgreSQL과 같은 다른 플랫폼은 대소문자를 구분한다. 따라서 항상 대소문자를 구분하는 비교를 가정하고 명확하게 모르는 경우 명시적으로 대소문자를 구분하지 않도록 하는 명령을 추가하여 비교하기 추천한다. 예를 들어,

```
SELECT * FROM concept WHERE concep_class_id = 'Clinical Finding'
```

대신, 다음과 같이 사용하는 것이 좋다.

```
SELECT * FROM concept WHERE LOWER(concep_class_id) = 'clinical finding'
```

스키마와 데이터베이스

SQL Server에서 테이블은 스키마 안에 있으며 스키마는 데이터베이스 안에 있다. 예를 들면, cdm_data.dbo.person은 cdm_data 데이터베이스의 dbo 스키마 안에 있는 person 테이블을 말한다. 다른 언어에서는 비슷한 계층 구조가 종종 존재하더라도 매우 다르게 사용된다. SQL Server에는 일반적으로 데이터베이스 당 하나의 스키마 (dbo라고 함), 가 있으며 사용자는 다른 데이터베이스의 데이터를 쉽게 사용할 수 있다. Postgres와 같은 다른 플랫폼에서는 단일 세션에서 데이터베이스 간 데이터를 사용할 수 없지만, 데이터베이스 안에는 많은 스키마를 가지고 있다. SQL Server의 데이터베이스는 PostgreSQL에서 스키마라고 할 수 있다.

따라서 SQL Server의 데이터베이스와 스키마를 단일 매개변수로 연결할 것을 권장한다. 이 매개 변수는 일반적으로 `@databaseSchema` 라고 한다. 예를 들면 우리는 매개 변수화된 SQL을 가질 수 있다.

```
SELECT * FROM @databaseSchema.person
```

SQL Server에서 `databaseSchema = "cdm_data.dbo"`값에 데이터베이스와 스키마 이름을 모두 포함할 수 있다. 다른 플랫폼에서는 같은 코드를 사용할 수 있지만, 스키마 매개 변수값은 다음과 같이 지정한다: `databaseSchema = "cdm_data"`

이것이 실패하는 한 가지 상황은 에러를 발생시키는 `USE cdm_data.dbo;`, 즉 `USE` 명령어를 사용했기 때문이다. 따라서 `USE` 명령어를 사용하지 말고 항상 테이블이 있는 데이터베이스 및 스키마를 지정하는 것이 바람직하다.

매개 변수화된 SQL 디버깅하기

매개 변수화된 SQL을 디버깅하는 것은 약간 복잡할 수 있다. 렌더링 된 SQL만 데이터베이스 서버에 대해 테스트할 수 있지만 매개 변수화된 (사전 렌더링 된) SQL에서 코드를 변경해야 한다.

`SqlRender` 패키지에는 대화형으로 SQL 소스를 편집하여 SQL을 렌더링하거나 반대로 번역할 수 있는 Shiny 앱이 포함되어 있다. 이 앱은 다음과 같이 시작한다:

```
launchSqlRenderDeveloper()
```

그리면 그림 9.1에 표시된 앱으로 기본 브라우저가 열린다. 이 앱은 웹에서도 공개적으로 사용할 수 있다.¹

앱에서 OHDSI SQL을 입력하고 대상 언어를 선택하고 SQL에 매개 변수값을 제공하면 자동으로 번역된 SQL이 하단에 나타난다.

9.2 DatabaseConnector

`DatabaseConnector` 는 Java의 JDBC 드라이버를 사용하여 다양한 데이터베이스 플랫폼에 연결하기 위한 R 패키지이다. `DatabaseConnector` 패키지는 CRAN (종합 R 아카이브 네트워크)에서 사용할 수 있으므로 다음을 사용하여 설치할 수 있다:

¹<http://data.ohdsi.org/SqlDeveloper/>

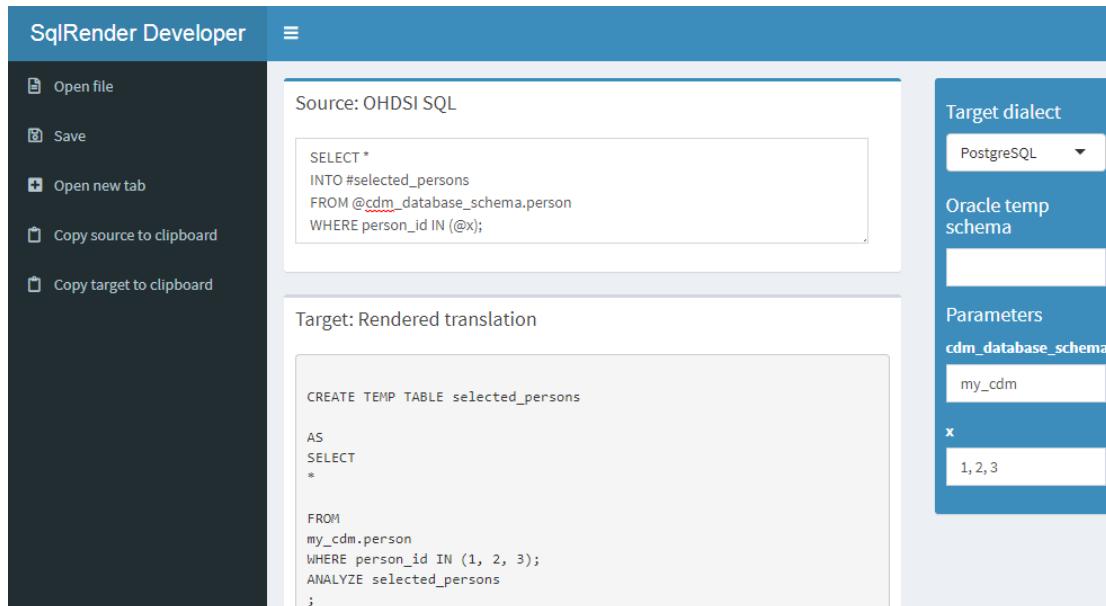


Figure 9.1: The SqlDeveloper Shiny 앱.

```
install.packages("DatabaseConnector")
```

DatabaseConnector는 기존 데이터베이스 시스템 (PostgreSQL, Microsoft SQL Server, SQLite 및 Oracle), 병렬 데이터웨어 하우스 (Microsoft APS, IBM Netezza 및 Amazon RedShift) 및 빅데이터 플랫폼 (Hadoop through Impala 및 Google BigQuery) 을 포함한 다양한 기술 플랫폼을 지원한다. 패키지에는 이미 대부분의 드라이버가 포함되어 있지만, 라이센스 문제로 인해 BigQuery, Netezza 및 Impala 용 드라이버는 포함되어 있지 않아서 사용자가 구해야 한다. 이러한 드라이버를 다운로드 하는 방법에 대한 지침을 보려면 ?jdbcDrivers 를 입력한다. 다운로드한 후 connect, dbConnect, and createConnectionDetails 함수의 pathToDriver 인수로 사용할 수 있다.

9.2.1 연결 생성하기

데이터베이스에 연결하려면 데이터베이스 플랫폼, 서버의 위치, 사용자 이름 및 비밀번호와 같은 많은 세부 사항을 지정해야 한다. connect 함수를 호출하여 다음 세부 사항을 직접 지정할 수 있다:

```
conn <- connect(dbms = "postgresql",
                 server = "localhost/postgres",
                 user = "joe",
```

```
    password = "secret",
    schema = "cdm")
```

```
## Connecting using PostgreSQL driver
```

각 플랫폼에 필요한 세부 사항에 대한 정보는 `?connect` 를 참조하라. 나중에 작업을 마치고 연결 끊는 것을 잊지 말라:

```
disconnect(conn)
```

서버 이름을 제공하는 대신 JDBC connecting string을 사용하는 것이 더 편리할 경우 이를 제공할 수도 있다는 점에 유의하라:

```
connString <- "jdbc:postgresql://localhost:5432/postgres"
conn <- connect(dbms = "postgresql",
                connectionString = connString,
                user = "joe",
                password = "secret",
                schema = "cdm")
```

```
## Connecting using PostgreSQL driver
```

때때로 먼저 세부 사항을 지정하고 나중에 연결할 때까지 연결을 연기해야 할 수 있다. 예를 들어, 함수 내에서 연결이 설정되고 세부 사항은 인수로 전달해야 하는 경우에 편리할 수 있다. 이를 목적으로 `createConnectionDetails` 함수를 사용할 수 있다:

```
details <- createConnectionDetails(dbms = "postgresql",
                                      server = "localhost/postgres",
                                      user = "joe",
                                      password = "secret",
                                      schema = "cdm")
conn <- connect(details)
```

```
## Connecting using PostgreSQL driver
```

9.2.2 쿼리하기

데이터베이스 쿼리를 위한 주요 함수는 `querySql`과 `executeSql`이다. 이러한 함수의 차이점은 `querySql`은 데이터베이스가 데이터를 반환할 것으로 예상하며, 한 번에 하나의 SQL 문만 처리할 수 있다는 것이다. 이와 대조적으로 `executeSql`은 데이터를 반환할 것을 예상하지 않으며, 단일 SQL 문자열에서 복수의 SQL 문을 수용한다.

몇 가지 예시:

```
querySql(conn, "SELECT TOP 3 * FROM person")
```

```
##   person_id gender_concept_id year_of_birth
## 1         1                 8507        1975
## 2         2                 8507        1976
## 3         3                 8507        1977
```

```
executeSql(conn, "TRUNCATE TABLE foo; DROP TABLE foo;")
```

두 함수 모두 광범위한 오류 보고 기능을 제공한다: 서버에서 오류가 발생하면 오류 메시지와 문제가 되는 SQL 부분이 텍스트 파일에 기록되어 더 나은 디버깅을 돋는다. 기본적으로 `executeSql` 함수도 실행된 SQL 문의 백분율을 나타내는 진행 표시줄을 보여준다. 이러한 속성이 필요하지 않은 경우 패키지는 `lowLevelQuerySql`과 `lowLevelExecuteSql` 함수도 제공한다.

9.2.3 Ffdf 객체를 사용하여 쿼리하기

데이터베이스에서 가져올 데이터가 너무 커서 종종 메모리에 들어갈 수 없는 경우도 있다. 8.4.2 절에서 언급했듯이, 그러한 경우 `ff` 패키지를 사용하여 R 데이터 객체를 디스크에 저장하고 메모리에서 사용하듯이 사용할 수 있다. `DatabaseConnector`는 객체에 데이터를 직접 다운로드할 수 있다:

```
x <- querySql.ffdf(conn, "SELECT * FROM person")
```

`x`는 이제 `ffdf` 객체이다.

9.2.4 같은 SQL을 사용하여 다른 플랫폼 쿼리하기

SqlRender 패키지의 render 및 translate 함수를 먼저 호출하는 다음과 같은 편의 함수를 사용할 수 있다: renderTranslateExecuteSql, renderTranslateQuerySql, renderTranslateQuerySql.ffd. 예를 들면:

```
x <- renderTranslateQuerySql(conn,
                               sql = "SELECT TOP 10 * FROM @schema.person",
                               schema = "cdm_sympuf")
```

SQL Server 관련 ‘TOP 10’ 구문은 PostgreSQL에서 예를 들어 ‘LIMIT 10’으로 변환되며 SQL 매개변수 @schema는 제공된 값 ‘cdm_sympuf’로 인스턴스화 되는 것에 주의해야 한다.

9.2.5 테이블 삽입하기

executeSql 함수를 사용하여 SQL 문을 전송하여 데이터베이스에 데이터를 삽입할 수도 있지만, insertTable 함수를 사용하는 것이 더 편리하고 빠르다 (일부 최적화로 인해):

```
data(mtcars)
insertTable(conn, "mtcars", mtcars, createTable = TRUE)
```

이 예는 mtcars 데이터 프레임을 자동으로 서버의 ‘mtcars’라는 테이블로 업로드하고 생성한다.

9.3 CDM 쿼리 실행

다음 예시에서는 CDM이 적용된 데이터베이스를 쿼리 실행하기 위해 OHDSI SQL을 사용한다. 이러한 쿼리는 CDM의 데이터를 찾을 수 있는 데이터베이스 스키마를 나타내기 위해 @cdm을 사용한다.

데이터베이스에 얼마나 많은 사람이 있는지 쿼리 실행하는 것부터 시작할 수 있다:

```
SELECT COUNT(*) AS person_count FROM @cdm.person;
```

PERSON_COUNT
26299001

그렇지 않으면 observation period의 평균에 관심이 있을 수도 있다:

```
SELECT AVG(DATEDIFF(DAY,
                      observation_period_start_date,
                      observation_period_end_date) / 365.25) AS num_years
FROM @cdm.observation_period;
```

NUM_YEARS
1.980803

테이블을 조인하여 추가 통계를 생성할 수 있다. 조인은 일반적으로 테이블의 특정 필드가 동일한 값을 갖도록 하여 여러 테이블의 필드를 결합한다. 예를 들어 두 테이블 모두 가지고 있는 PERSON_ID 필드로 PERSON 테이블과 OBSERVATION_PERIOD 테이블을 조인할 수 있다. 즉, 조인의 결과는 두 테이블의 모든 필드를 갖는 새로운 테이블과 같은 집합이지만, 모든 행에서 두 테이블의 PERSON_ID는 동일한 값을 가져야 한다. 예를 들어 PERSON 테이블의 YEAR_OF_BIRTH 필드와 함께 OBSERVATION_PERIOD 테이블의 OBSERVATION_PERIOD_END_DATE 필드를 사용하여 관찰 종료 시 환자의 최고 나이를 계산할 수 있다:

```
SELECT MAX(YEAR(observation_period_end_date) -
           year_of_birth) AS max_age
FROM @cdm.person
INNER JOIN @cdm.observation_period
ON person.person_id = observation_period.person_id;
```

MAX_AGE
90

관찰 시작 당시 연령 분포를 결정하려면 훨씬 더 복잡한 쿼리가 필요하다. 이 쿼리에서는 먼저 PERSON 테이블과 OBSERVATION_PERIOD을 조인하여 관찰 당시 연령을 계산한다. 또한 연령을 기준으로 이 조인된 집합의 순서를 정렬하고 order_nr로 저장한다. 이 조인의 결과를 여러 번 사용하고 싶기 때문에 “ages”라고 하는 common table expression(CTE) (WITH ... AS를 사용하여 정의된)으로 정의한다. 즉, 연령을 기준 테이블인 것처럼 나타낼 수 있다. “ages”의 행 수를 세어 “n”을 생성하고 각 사분위 수에 대해 order_nr이 분수 시간 “n” 보다 작은 최소 연령을 찾는다. 예를

들어, 중앙값을 찾기 위해 $order_nr < .50 * n$ 인 최소 연령을 사용한다. 최소 및 최대 연령은 별도로 계산된다:

```
WITH ages
AS (
    SELECT age,
           ROW_NUMBER() OVER (
               ORDER BY age
           ) order_nr
    FROM (
        SELECT YEAR(observation_period_start_date) - year_of_birth AS age
        FROM @cdm.person
        INNER JOIN @cdm.observation_period
            ON person.person_id = observation_period.person_id
    ) age_computed
)
SELECT MIN(age) AS min_age,
       MIN(CASE
           WHEN order_nr < .25 * n
               THEN 9999
           ELSE age
           END) AS q25_age,
       MIN(CASE
           WHEN order_nr < .50 * n
               THEN 9999
           ELSE age
           END) AS median_age,
       MIN(CASE
           WHEN order_nr < .75 * n
               THEN 9999
           ELSE age
           END) AS q75_age,
       MAX(age) AS max_age
FROM ages
CROSS JOIN (
    SELECT COUNT(*) AS n
    FROM ages
```

```
) population_size;
```

MIN_AGE	Q25_AGE	MEDIAN_AGE	Q75_AGE	MAX_AGE
0	6	17	34	90

SQL을 사용하는 대신 R에서 더 복잡한 계산을 수행할 수도 있다. 예를 들어, 이 코드를 사용하여 동일한 결과를 얻을 수 있다:

```
sql <- "SELECT YEAR(observation_period_start_date) -
        year_of_birth AS age
FROM @cdm.person
INNER JOIN @cdm.observation_period
        ON person.person_id = observation_period.person_id;"
age <- renderTranslateQuerySql(conn, sql, cdm = "cdm")
quantile(age[, 1], c(0, 0.25, 0.5, 0.75, 1))

##    0%   25%   50%   75% 100%
##    0     6    17    34    90
```

서버에서 연령을 계산하고 모든 연령을 다운로드한 다음 연령 분포를 계산한다. 그러나 이를 위해서는 데이터베이스 서버에서 수백만 행의 데이터를 다운로드해야 하므로 효율성이 떨어진다. 계산이 SQL에서 가장 잘 수행되는지 R에서 가장 잘 수행되는지를 사례별로 결정해야 한다.

쿼리는 CDM의 source value를 사용할 수도 있다. 예를 들어, 다음을 사용하여 가장 빈번한 상위 10 개의 condition source code를 검색할 수 있다:

```
SELECT TOP 10 condition_source_value,
       COUNT(*) AS code_count
FROM @cdm.condition_occurrence
GROUP BY condition_source_value
ORDER BY -COUNT(*);
```

CONDITION_SOURCE_VALUE	CODE_COUNT
4019	49094668
25000	36149139

CONDITION_SOURCE_VALUE	CODE_COUNT
78099	28908399
319	25798284
31401	22547122
317	22453999
311	19626574
496	19570098
I10	19453451
3180	18973883

여기서 CONDITION_OCCURRENCE 테이블의 행을 CONDITION_SOURCE_VALUE 필드의 값으로 그룹화하고 각 그룹의 행 수를 세었다. 우리는 CONDITION_SOURCE_VALUE, count, 그리고 count의 역순을 검색했다.

9.4 쿼리 작성 시 OMOP 용어 사용하기

많은 작업에서 용어 vocabulary는 유용하다. Vocabulary 테이블은 CDM의 일부이므로 SQL 쿼리를 사용하여 이용할 수 있다. Vocabulary에 대한 쿼리가 CDM에 대한 쿼리와 어떻게 결합할 수 있는지 보여준다. CDM의 많은 필드에는 CONCEPT 테이블을 사용하여 확인할 수 있는 concept ID가 포함되어 있다. 예를 들어, 데이터베이스에서 성별에 따라 계층화된 인원수를 세려고 할 때, GENDER_CONCEPT_ID를 개념 이름으로 찾아 바꾸어 사용하는 것이 더 편리할 것이다:

```
SELECT COUNT(*) AS subject_count,
       concept_name
  FROM @cdm.person
 INNER JOIN @cdm.concept
    ON person.gender_concept_id = concept.concept_id
 GROUP BY concept_name;
```

SUBJECT_COUNT	CONCEPT_NAME
14927548	FEMALE
11371453	MALE

용어의 매우 강력한 특징은 계층구조에 있다. 특정 개념과 그에 속하는 모든 하위 개념을 찾는 쿼리를 사용하는 경우가 빈번하다. 예를 들어, ibuprofen 성분이 들어 있는 모든 약의 처방 개수를 세고 싶다고 상상해보라:

```
SELECT COUNT(*) AS prescription_count
FROM @cdm.drug_exposure
INNER JOIN @cdm.concept_ancestor
  ON drug_concept_id = descendant_concept_id
INNER JOIN @cdm.concept ingredient
  ON ancestor_concept_id = ingredient.concept_id
WHERE LOWER(ingredient.concept_name) = 'ibuprofen'
  AND ingredient.concept_class_id = 'Ingredient'
  AND ingredient.standard_concept = 'S';
```

PRESCRIPTION_COUNT
26871214

9.5 QueryLibrary

QueryLibrary는 CDM에 대해 일반적으로 사용되는 SQL 쿼리의 라이브러리이다. 그림 9.2에 표시된 응용 프로그램² 및 R 패키지로 제공된다.³

라이브러리의 목적은 새로운 사용자가 CDM에 쿼리하는 방법을 배우도록 돋는 것이다. 라이브러리의 쿼리는 OHDSI 커뮤니티에서 검토하고 승인하였다. 쿼리 라이브러리는 주로 교육 목적으로 사용되지만 숙련된 사용자에게 유용한 자원이기도 하다.

QueryLibrary는 SqlRender를 사용하여 선택한 SQL 언어로 쿼리를 출력한다. 사용자는 CDM 데이터베이스 스키마, vocabulary 데이터베이스 스키마 (별도의 경우) 및 Oracle 임시 스키마 (필요한 경우)를 지정할 수 있으므로 이러한 설정으로 쿼리가 자동으로 렌더링 된다.

²<http://data.ohdsi.org/QueryLibrary>

³<https://github.com/OHDSI/QueryLibrary>

Select a query

Group	Name
["drug exposure"]	All
drug exposure	DEX01 Counts of persons with any number of exposures to a certain drug
drug exposure	DEX02 Counts of persons taking a drug, by age, gender, and year of exposure
drug exposure	DEX03 Distribution of age, stratified by drug
drug exposure	DEX04 Distribution of gender in persons taking a drug
drug exposure	DEX05 Counts of drug records for a particular drug
drug exposure	DEX06 Counts of distinct drugs in the database
drug exposure	DEX07 Maximum number of drug exposure events per person over some time period

Query Description

DEX01: Counts of persons with any number of exposures to a certain drug

Description

This query is used to count the persons with at least one exposures to a certain drug (drug_concept_id). See vocabulary queries for obtaining valid drug_concept_id values. The input to the query is a value (or a comma-separated list of values) of a drug_concept_id. If the input is omitted, all drugs in the data table are summarized.

Query

The following is a sample run of the query. The input parameters are highlighted in blue.

```

SELECT
    c.concept_name,
    drug_concept_id,
    COUNT(person_id) AS num_persons
FROM cdm.drug_exposure
INNER JOIN cdm.concept c
ON drug_concept_id = c.concept_id
WHERE domain_id='Drug'
  
```

Figure 9.2: QueryLibrary: CDM에 대한 SQL 조회 라이브러리.

9.6 간단한 연구 구성하기

9.6.1 문제 정의

혈관 부종Angioedema은 ACE inhibitor(ACEi)의 잘 알려진 부작용이다. (Slater et al., 1988) ACEi 치료 첫 주에 혈관 부종의 발생률이 주당 3,000명의 환자당 1건인 것으로 추정하였다. 여기서 우리는 이 결론을 모방하고 나이와 성별에 따라 계층화한다. 간단하게 하기 위해서 우리는 하나의 ACEi: lisinopril에 중점을 둔다. 따라서 우리는 질문에 대답한다.

Lisinopril 치료 개시 후 첫 주에 나이와 성별에 따라 계층화되는 혈관 부종의 비율은 얼마인가?

9.6.2 노출

노출 Exposure은 lisinopril에 대한 첫 번째 노출로 정의한다. 먼저 이전에 lisinopril에 노출되지 않았음을 의미한다. 첫 노출 전에 365일의 연속 관찰 기간이 필요하다.

9.6.3 결과

입원 또는 응급실 방문 중 혈관 부종 진단 코드의 발생으로 혈관 부종을 정의한다.

9.6.4 위험 노출 기간 Time-at-risk

환자가 일주일 동안 노출되었는지와 관계없이 이 치료 시작 후 첫 주에 발생률을 계산한다.

9.7 SQL과 R을 사용하여 연구 구현

OHDSI 툴 규약에 구속되지 않지만 동일한 원칙을 따르는 것은 도움이 된다. 이 경우 OHDSI 툴의 작동 방식과 유사하게 SQL을 사용하여 코호트 테이블을 채운다. 코호트 테이블은 CDM에 정의되어 있으며 사전 정의된 필드 집합도 있다. 먼저 쓰기 접근 권한이 있는 데이터베이스 스키마에 COHORT 테이블을 만들어야 하는데, 이는 CDM 데이터를 저장하는 데이터베이스 스키마와 동일하지 않을 수 있다.

```
library(DatabaseConnector)
conn <- connect(dbms = "postgresql",
                 server = "localhost/postgres",
                 user = "joe",
```

```
password = "secret")  
cdmDbSchema <- "cdm"  
cohortDbSchema <- "scratch"  
cohortTable <- "my_cohorts"  
  
sql <- "  
CREATE TABLE @cohort_db_schema.@cohort_table (  
    cohort_definition_id INT,  
    cohort_start_date DATE,  
    cohort_end_date DATE,  
    subject_id BIGINT  
);  
"  
renderTranslateExecuteSql(conn, sql,  
    cohort_db_schema = cohortDbSchema,  
    cohort_table = cohortTable)
```

여기서는 데이터베이스 스키마 및 테이블 이름을 매개 변수화하여 다른 환경에 쉽게 적용할 수 있다. 결과는 데이터베이스 서버의 빈 테이블이다.

9.7.1 노출 코호트

다음으로 노출 코호트를 만들어 COHORT 테이블에 삽입한다:

```
sql <- "  
INSERT INTO @cohort_db_schema.@cohort_table (  
    cohort_definition_id,  
    cohort_start_date,  
    cohort_end_date,  
    subject_id  
)  
SELECT 1 AS cohort_definition_id,  
    cohort_start_date,  
    cohort_end_date,  
    subject_id  
FROM (  
    SELECT drug_era_start_date AS cohort_start_date,
```

```

drug_era_end_date AS cohort_end_date,
person_id AS subject_id
FROM (
  SELECT drug_era_start_date,
    drug_era_end_date,
    person_id,
    ROW_NUMBER() OVER (
      PARTITION BY person_id
      ORDER BY drug_era_start_date
    ) order_nr
  FROM @cdm_db_schema.drug_era
  WHERE drug_concept_id = 1308216 -- Lisinopril
) ordered_exposures
WHERE order_nr = 1
) first_era
INNER JOIN @cdm_db_schema.observation_period
ON subject_id = person_id
AND observation_period_start_date < cohort_start_date
AND observation_period_end_date > cohort_start_date
WHERE DATEDIFF(DAY,
  observation_period_start_date,
  cohort_start_date) >= 365;
"

```

```

renderTranslateExecuteSql(conn, sql,
  cohort_db_schema = cohortDbSchema,
  cohort_table = cohortTable,
  cdm_db_schema = cdmDbSchema)

```

여기에서는 DRUG_EXPOSURE 테이블에서 자동으로 파생되는 CDM의 표준 테이블인 DRUG ERA 테이블을 사용한다. DRUG ERA 테이블에는 성분 수준에서 연속 노출 기간이 기록되어 있다. 따라서 lisinopril을 검색할 수 있으며, 이는 lisinopril을 함유한 모든 약물에 대한 노출을 자동으로 식별할 수 있다. 사람당 첫 번째 약물 노출을 취한 다음 OBSERVATION_PERIOD 테이블과 조인한다. 청구 데이터의 경우 한 사람이 여러 개의 관찰 기간을 가질 수 있으므로 약물 노출이 포함된 기간과 겹치는 OBSERVATION PERIOD를 조인해야 한다. 그런 다음 OBSERVATION_PERIOD_START_DATE와 COHORT_START_DATE 사이에 적어도 365일이 필요함을

명시한다.

9.7.2 결과 코호트

마지막으로, 우리는 결과outcome 코호트를 만들어야 한다:

```
sql <- "
INSERT INTO @cohort_db_schema.@cohort_table (
    cohort_definition_id,
    cohort_start_date,
    cohort_end_date,
    subject_id
)
SELECT 2 AS cohort_definition_id,
    cohort_start_date,
    cohort_end_date,
    subject_id
FROM (
    SELECT DISTINCT person_id AS subject_id,
        condition_start_date AS cohort_start_date,
        condition_end_date AS cohort_end_date
    FROM @cdm_db_schema.condition_occurrence
    INNER JOIN @cdm_db_schema.concept_ancestor
        ON condition_concept_id = descendant_concept_id
    WHERE ancestor_concept_id = 432791 -- Angioedema
) distinct_occurrence
INNER JOIN @cdm_db_schema.visit_occurrence
    ON subject_id = person_id
    AND visit_start_date <= cohort_start_date
    AND visit_end_date >= cohort_start_date
WHERE visit_concept_id IN (262, 9203,
    9201) -- Inpatient or ER;
"

renderTranslateExecuteSql(conn, sql,
    cohort_db_schema = cohortDbSchema,
    cohort_table = cohortTable,
```

```
cdm_db_schema = cdmDbSchema)
```

CONDITION OCCURRECE 테이블과 CONCEPT ANCESTOR 테이블을 조인하여 모든 혈관 부종과 그 자손을 찾는다. 같은 날에 여러 혈관 부종의 진단이 여러 혈관 부종 발생이 아닌 동일한 사건일 가능성이 높기 때문에 DISTINCT를 사용하여 하루에 하나의 행만 선택하도록 한다. 이러한 발생을 VISIT_OCCURRENCE 테이블과 조인하여 입원이나 응급실 환경에서 진단되었는지 확인한다.

9.7.3 발생률 계산

코호트가 준비되었으므로 연령과 성별에 따라 계층화되는 발생률을 계산할 수 있다:

```
sql <- "
WITH tar AS (
    SELECT concept_name AS gender,
        FLOOR((YEAR(cohort_start_date) -
            year_of_birth) / 10) AS age,
        subject_id,
        cohort_start_date,
        CASE WHEN DATEADD(DAY, 7, cohort_start_date) >
            observation_period_end_date
        THEN observation_period_end_date
        ELSE DATEADD(DAY, 7, cohort_start_date)
        END AS cohort_end_date
    FROM @cohort_db_schema.@cohort_table
    INNER JOIN @cdm_db_schema.observation_period
        ON subject_id = observation_period.person_id
        AND observation_period_start_date < cohort_start_date
        AND observation_period_end_date > cohort_start_date
    INNER JOIN @cdm_db_schema.person
        ON subject_id = person.person_id
    INNER JOIN @cdm_db_schema.concept
        ON gender_concept_id = concept_id
    WHERE cohort_definition_id = 1 -- Exposure
)
SELECT days.gender,
```

```

days.age,
days,
CASE WHEN events IS NULL THEN 0 ELSE events END AS events
FROM (
  SELECT gender,
  age,
  SUM(DATEDIFF(DAY, cohort_start_date,
  cohort_end_date)) AS days
  FROM tar
  GROUP BY gender,
  age
) days
LEFT JOIN (
  SELECT gender,
  age,
  COUNT(*) AS events
  FROM tar
  INNER JOIN @cohort_db_schema.@cohort_table angioedema
  ON tar.subject_id = angioedema.subject_id
  AND tar.cohort_start_date <= angioedema.cohort_start_date
  AND tar.cohort_end_date >= angioedema.cohort_start_date
  WHERE cohort_definition_id = 2 -- Outcome
  GROUP BY gender,
  age
) events
ON days.gender = events.gender
AND days.age = events.age;
"
results <- renderTranslateQuerySql(conn, sql,
  cohort_db_schema = cohortDbSchema,
  cohort_table = cohortTable,
  cdm_db_schema = cdmDbSchema,
  snakeCaseToCamelCase = TRUE)

```

우선 적절한 위험 노출 기간으로 모든 노출을 포함하는 CTE인 “tar”를 만든다. OBSERVA-

TION_PERIOD_END_DATE에서 위험 관찰 기간을 단축한다는 점에 유의한다. 또한 10년 단위로 나이를 계산하고 성별을 파악한다. CTE를 사용하면 동일한 중간 결과 집합을 쿼리에서 여러 번 사용할 수 있다는 장점이 있다. 이 경우 위험 관찰 기간 동안 발생하는 혈관 부종 사건의 수와 총 위험 관찰 기간의 양을 계산하는 데 사용된다.

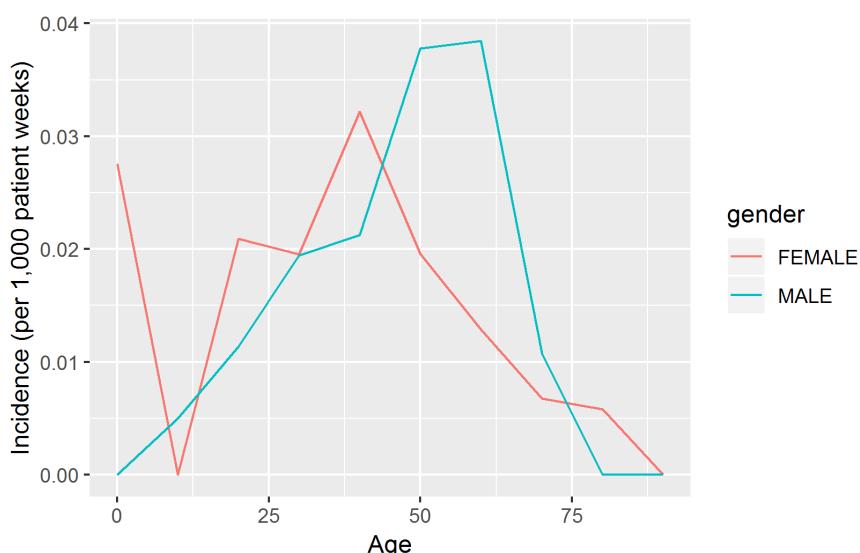
SQL에서는 필드 이름에 snake_case(대소문자를 구분하지 않는)를 사용하는 반면 R에서는 camelCase(대소문자를 구분하는)를 사용하는 경향이 있기 때문 snakeCaseToCamelCase = TRUE로 한다. results 데이터 프레임 열 이름은 이제 camelCase이다.

ggplot2 패키지의 도움을 받아 다음과 같은 결과를 쉽게 표시할 수 있다:

```
# Compute incidence rate (IR) :
results$ir <- 1000 * results$events / results$days / 7

# Fix age scale:
results$age <- results$age * 10

library(ggplot2)
ggplot(results, aes(x = age, y = ir, group = gender, color = gender)) +
  geom_line() +
  xlab("Age") +
  ylab("Incidence (per 1,000 patient weeks)")
```



9.7.4 마무리하기

생성한 테이블을 정리하고 연결을 닫는 것을 잊지 말자.

```
sql <- "
TRUNCATE TABLE @cohort_db_schema.@cohort_table;
DROP TABLE @cohort_db_schema.@cohort_table;
"

renderTranslateExecuteSql(conn, sql,
    cohort_db_schema = cohortDbSchema,
    cohort_table = cohortTable)

disconnect(conn)
```

9.7.5 호환성

OHDSI SQL을 DatabaseConnector 및 SQLRender와 함께 사용하기 때문에 여기서 검토한 코드는 OHDSI가 지원하는 모든 데이터베이스 플랫폼에서 실행된다.

여기에서는 예시 목적으로 수작업으로 만든 SQL 쿼리를 사용하여 코호트를 만들었다는 점에 유의하라. ATLAS에서 코호트 정의를 구성하고 ATALS에서 생성된 SQL을 사용하여 코호트를 생성하는 것이 훨씬 더 편리했을 것이다. ATLAS는 OHDSI SQL을 생성하기 때문에, SqlRender 및 DatabaseConnector와 함께 쉽게 사용할 수 있다.

9.8 요약



- **SQL**은 공통 데이터 모델을 따르는 데이터베이스를 포함하여 데이터베이스를 조회하기 위한 표준 언어이다.
- 데이터베이스 플랫폼마다 SQL 언어가 다르며 각 플랫폼별 쿼리를 작성하기 위해서는 다른 많은 작업이 필요하다.
- **SqlRender** 및 **DatabaseConnector** R 패키지는 CDM에서 데이터를 쿼리하는 통합된 방법을 제공하므로 동일한 분석 코드를 수정 없이 다른 환경에서 실행할 수 있다.
- R과 SQL을 함께 사용하면 OHDSI 툴에서 지원하지 않는 사용자 맞춤 분석 연구를 구현할 수 있다.

- **QueryLibrary** 는 CDM-용으로 재사용 가능한 SQL 쿼리 모음을 제공한다.

9.9 예제

전제조건

이 연습문제에서는 8.4.5절에서 설명된 대로 R, R-Studio, Java가 설치되었다고 가정한다. 또한 다음을 사용하여 설치할 수 있는 SqlRender, DatabaseConnector 및 Eunomia 패키지도 필요하다:

```
install.packages(c("SqlRender", "DatabaseConnector", "devtools"))
devtools::install_github("ohdsi/Eunomia", ref = "v1.0.0")
```

Eunomia 패키지는 로컬 R 세션 내에서 실행될 CDM의 시뮬레이션 된 다른 데이터 세트를 제공한다. 연결 세부 사항은 다음을 사용하여 얻을 수 있다:

```
connectionDetails <- Eunomia::getEunomiaConnectionDetails()
```

CDM 데이터베이스 스키마는 “main”이다.

Exercise 9.1. SQL과 R을 사용하여 데이터베이스에 몇 사람이 있는지 계산하십시오.

Exercise 9.2. SQL과 R을 사용하여 celecoxib을 적어도 한 번 이상 처방 한 사람을 계산하십시오.

Exercise 9.3. SQL과 R을 사용하여 celecoxib에 노출되는 동안 얼마나 많은 위장 출혈gastrointestinal hemorrhage이 있는지 진단한다. (힌트: 위장 출혈의 개념 ID는 192671이다.)

제안된 답변은 부록 E.5에서 찾을 수 있다.

Chapter 10

코호트 만들기

Chapter lead: Kristin Kostka

실세계 데이터 *Real world data*라고도 불리는 관찰 건강 정보 Observational health data는 다양한 출처에서 꾸준하게 수집되는 환자의 건강 상태나 환자에게 제공되는 의료서비스에 관한 정보이다. CDM 데이터 유지를 위해 노력하는 OHDSI 공동 연구자는 전자 의무 기록(EHR), 보험 청구 자료, 결제 자료, 제품 및 질병 정보 등을 포함하는 다양한 출처의 데이터를 활용하며, 환자 개인이 가정 내에서, 혹은 핸드폰 등의 다른 기기를 통해 생성한 건강 정보를 활용하기도 한다. 이러한 데이터는 연구 목적으로 수집된 데이터가 아니기 때문에 우리가 보고자 하는 임상 정보를 명료하게 담고 있지 못할 수도 있다.

예를 들어, 건강 보험 청구 자료 데이터베이스는 특정 질병 (예를 들어 혈관성 부종)이 있는 환자에게 제공된 의료 서비스를 파악하고 그 치료비용을 의료기관에 상환해주기 위해 설립되었기 때문에, 청구목적에 맞는 정보만 부분적으로 담고 있다. 우리가 그런 데이터를 연구 목적으로 사용하기를 원한다면, 우리가 데이터를 사용할 때 실제로 관심 있는 것을 추론해야 하며, 타당한 추론을 가능케 하는 적절한 코호트를 설정해서 연구를 진행해야 한다. 그러므로 만약 우리가 보험 청구 자료 데이터베이스에서 새로 발생한 혈관성 부종만을 확인하고 싶다면, 코호트를 만들 때 이미 치료하고 있는 혈관성 부종 환자를 제외하기 위해서 응급실로 방문한 혈관성 부종 환자만을 코호트에 포함한다는 논리를 세워야 할 것이다. 전자 의무 기록에 담긴 임상 정보를 사용할 경우에도 비슷하다. 데이터를 일차적인 목적으로 사용하는 것이기 때문에 데이터베이스가 설립된 일차적인 목적을 인식하고 있어야 한다. 연구를 설계할 때마다 다양한 데이터베이스 환경에서 우리가 설정한 코호트가 어떻게 존재하고 있는지에 대한 뉘앙스를 항상 생각해야 한다.

이 장에서는 코호트를 생성하고 공유하는 것이 가지는 의미와 코호트를 개발하는 방법, 그리고 ATLAS와 SQL을 이용해 당신만의 코호트를 생성하는 방법에 관해서 설명할 것이다.

10.1 코호트란 무엇인가?

OHDSI 연구에서는 특정 기간 안에 하나 이상의 포함 기준에 속하는 사람의 집단을 코호트라고 정의한다. 때로 표현형 *phenotype*이라는 용어로 코호트를 대신 사용하기도 한다. 코호트는 OHDSI 분석 도구를 사용하거나 연구를 시작하기 위한 첫 단계로 사용된다. 예를 들어 고혈압 치료약물인 ACE inhibitor를 복용하기 시작한 사람 중에서 혈관성 부종이 일어날 위험을 예측하기 위한 연구를 진행할 때, 우리는 다음 두 가지 코호트를 지정해야 한다: 결과 코호트 outcome cohort(혈관성 부종이 발생한 사람), 그리고 대상 코호트 target cohort(ACE inhibitor를 복용하기 시작한 사람). OHDSI에서 사용되는 코호트라는 개념이 가지는 중요한 특성은, 연구 내에서 지정된 각각의 코호트는 연구 내의 다른 코호트와 상호 독립적이기 때문에 (그 연구만이 아닌 다른 연구에서도) 재사용이 가능하다는 것이다. 앞서 제시된 혈관성 부종 코호트를 예로 들어 보면, 이 코호트는 관찰되는 인구 내의 모든 혈관성 부종 발생 예를 담게 되며, 이는 혈관성 부종 코호트에는 ACE inhibitor를 사용한 사람과 사용하지 않은 사람 모두가 포함된다는 것을 의미한다. 우리의 분석 도구는 두 코호트의 교집합을 분석할 것이다. 이것이 가지는 장점은, ACE inhibitor를 복용함으로써 발생하는 다른 결과를 분석할 때에도 이번에 만든 동일한 혈관성 부종 코호트를 재사용할 수 있다는 것이다.



코호트는 특정 시간 내에 하나 이상의 포함 기준을 만족시키는 사람의 집합이다.

OHDSI에서 사용되는 코호트의 정의가 다른 분야에서 사용되는 코호트의 정의와 다를 수 있다는 것을 인지하는 것이 중요하다. 예를 들어 동료 심사 peer review를 거친 많은 과학 논문에서, 코호트는 특정 임상 코드 집합 (예를 들어 ICD-9/ICD-10, NDC, HCPCS 등)과 동일한 의미로 사용되었다. 코드 집합은 코호트를 설정하는 데 중요한 부분을 담당하지만, 코호트는 코드 집합에 의해서만 정의되는 것은 아니다. 코호트는 기준에 맞도록 코드 집합을 사용하는 특정 논리가 필요하다. (예를 들어 그 환자에게 첫 번째로 발생된 ICD-9/ICD-10 코드인가?). 잘 정의된 코호트는 환자가 어떻게 코호트에 포함되고 제외되는지를 구체적으로 설명한다.

OHDSI가 코호트를 정의하는 방식에는 다음과 같은 독특한 특징이 있다:

- 한 사람은 여러 개의 코호트에 속할 수도 있다.
- 한 사람이 동일한 코호트 여러 다른 기간에 걸쳐 속할 수도 있다.
- 한 사람이 같은 기간에 동일한 코호트에 여러 번 속하지 않을 수도 있다
- 코호트에는 0명 혹은 그 이상의 구성원을 가질 수도 있다.

코호트를 만드는 방법에는 두 가지 주요 방법이 있다:

1. **규칙 기반 코호트 정의**는 언제 환자가 코호트 내에 속하는지에 관한 명확한 포함 규칙을 가진다. 이 포함 규칙을 정하는 것은 코호트를 디자인하는 사람의 전문적인 지식에 상당히 의존한다.
2. **확률적 코호트 정의**는 확률 모델을 사용하여 환자가 코호트에 속할 확률 (0~100%)을 계산한다. 이 확률은 역치 값을 사용하여 ‘예-아니오’ 분류로 전환할 수도 있고, 그대로 사용할 수도 있다. 확률 모델은 일반적으로 예측 가능한 관련 환자 특성을 자동으로 식별하기 위해 일부 기계학습모델 (예를 들어 로지스틱 회귀)의 학습을 통해 만들어진다.

다음으로 이 두 가지의 방법에 대해서 구체적으로 알아보겠다.

10.2 규칙 기반 코호트 정의

규칙 기반 코호트 정의는 특정 기간 내에 (예를 들어 “지난 6개월 이내 해당 질병이 발생한 사람”) 하나 혹은 그 이상의 포함 기준 (예를 들어 “혈관 부종을 앓는 환자”)을 명확히 제시함으로써 시작한다.

이러한 기준을 만드는 데 사용되는 표준 구성 요소는 다음과 같다:

- **도메인:** CDM 도메인 (예를 들어 “Procedure Occurrence”, “Drug Exposure”)은 데이터가 저장되는 곳인데, 도메인의 종류에 따라 어떤 유형의 임상 정보가 담길지, 어떤 개념이 담길지가 결정된다. 도메인에 관한 세부사항은 4.2.4절에서 확인할 수 있다.
- **개념 모음 Concept set:** 우리가 관심을 가지는 임상적 개념을 대변하는 하나 이상의 표준화된 개념의 모음을 의미한다. concept set은 표준 용어 (임상에서 쓰이는 용어는 국가나 병원, 사람에 따라 동일한 개념도 조금씩 다른 용어로 사용되는데 이를 표준 용어로 매핑함)로 구성되어 있기 때문에 다양한 관찰 의료 데이터에서 상호 운용이 가능하다. concept set에 관하여 10.3절에 자세한 설명이 있다.
- **도메인별 속성:** 관심 있는 임상 실체와 연관된 추가적인 속성 (예를 들어 DRUG_EXPOSURE의 DAYS_SUPPLY, MEASUREMENT의 VALUE_AS_NUMBER와 RANGE_HIGH)
- **시간의 설정:** 선정 기준과 임상사건 발생 간의 시간 간격 (예를 들어 노출 시작 또는 노출 시작 후 365일 이내에 특정 조건이 발생해야 함)

코호트 정의를 작성할 때, 코호트 속성을 나타내는 도메인을 빌딩 블록 (그림 10.1 참조)과 유사하게 생각하면 도움이 될 수 있다. 각 도메인에서 허용 가능한 구성 요소에 대해 혼란스럽다면 언제든지



Figure 10.1: 코호트 정의를 위한 빌딩 블록

공통 데이터 모델 4장을 참조한다.

코호트 정의를 작성할 때, 다음과 같은 질문에 답할 수 있어야 한다:

- 코호트 진입 시간을 정의하는 초기 이벤트는 무엇인가?
- 초기 이벤트에는 어떤 포함 기준이 적용되는가?
- 코호트 종료 시간을 정의하는 것은 무엇인가?

코호트 진입 이벤트 cohort entry event: 코호트 진입 이벤트 (초기 이벤트)는 환자가 코호트에 진입하는 코호트 기준 시점 cohort index date으로 정의된다. 코호트 진입 이벤트는 약물 노출 Drug exposure, 질병 상태 conditions, 절차 procedures, 측정 measurements 및 방문 visits과 같은 CDM에 기록된 모든 사건일 수 있다. 초기 이벤트는 데이터가 저장되는 CDM 도메인 (예를 들어 PROCEDURES_OCCURRENCE, DRUG_EXPOSURE 등), 임상 활동을 식별하기 위해 구축된 concept set (예를 들어 질병 상태에 대한 SNOMED 코드, 약물에 대한 RxNorm 코드) 및 기타 특정 속성 (예를 들어 발생 연령, 첫 진단 / 절차 등, 지정된 시작 및 종료 날짜, 방문 유형 등)에 의해 정의된다. 진입 이벤트를 가진 사람의 집합을 초기 사건 코호트 initial event cohort라고 한다.

포함 기준 inclusion criteria: 포함 기준은 초기 이벤트 코호트에 적용되어 코호트에 진입할 사람을 추가로 제한한다. 각 포함 기준을 만들 때는 데이터가 저장되는 CDM 도메인, concept set, 도메인별 속성 (예를 들어 days supply, 방문 유형) 및 코호트 색인 날짜에 관한 시간 논리를 결정해야 한다. **적격 코호트 qualifying cohort**는 초기 이벤트 코호트에서 모든 포함 기준을 충족하는 사람의 집합으로 정의한다.

코호트 종료 기준 cohort exit criteria: 코호트 종료 이벤트는 한 사람이 더 이상 코호트 자격 요건을 갖추지 못했을 때를 의미한다. 코호트 종료는 관찰 기간이 끝났을 때, 초기 진입 이벤트로부터 일정한 시간이 지났을 때 혹은 마지막 이벤트가 발생했을 때 등 여러 방법으로 정의할 수 있다. 코호트 종료 기준에 따라 한 사람의 오랜 시간에 걸친 기록 중에서 특정한 기간이 선정기준에 맞아 코호트에 한 번 포함된 후에 또 다른 기간이 코호트 선정 기간에 맞아 다시 코호트에 포함되는 등 한 사람의 관찰이 하나의 코호트에 여러 번 속할 수 있다.



OHDSI 도구에는 포함 기준과 제외 기준이 구분되지 않는다. 모든 기준은 포함 기준으로 설정해야 한다. 예를 들어 ‘이전 고혈압 환자 제외’라는 제외 기준을 ‘이전 고혈압 발생이 0인 사람 포함’이라는 포함 기준으로 설정해야 한다.

10.3 개념 모음

개념 모음 concept set을 구성하는 개념은 다양한 다른 분석에서 재사용이 가능하다. concept set은 관찰 연구에서 종종 사용되는 표준화된 컴퓨터 코드라고 생각해도 된다. concept set은 다음 특성을 포함하고 있다:

- **Exclude:** concept set으로부터 해당 개념과 해당 개념의 하위 개념을 제외한다.
- **Descendants:** 이 개념뿐만 아니라 모든 하위 항목 개념을 고려한다.
- **Mapped:** 표준화되지 않은 개념도 검색한다.

예를 들어 표 10.1과 같이 concept set은 두 개의 개념을 포함할 수 있다. 여기서 우리는 4329847 (“심근경색 Myocardial infarction”)과 그 모든 하위 개념을 포함했고, 314666 (“과거 심근경색 Old myocardial infarction”)과 그 모든 하위 개념은 제외했다.

Table 10.1: concept set의 예시

Concept Id	Concept Name	Excluded	Descendants	Mapped
4329847	Myocardial infarction	NO	YES	NO
314666	Old myocardial infarction	YES	YES	NO

그림 10.2에서 볼 수 있다시피, “심근경색”과 그 모든 하위 개념을 포함할 것이고, 하위 개념 중에서 “과거 심근경색”과 그 모든 하위 개념은 제외할 것이다. 결과적으로 거의 100개 정도의 표준 개념 standard concept을 포함한 concept set이 만들어졌다. 이 standard concept은 다양한 데이터베이스에서 사용되는 수백 개의 원천 코드 (예를 들어 ICD-9, ICD-10)를 반영한다.

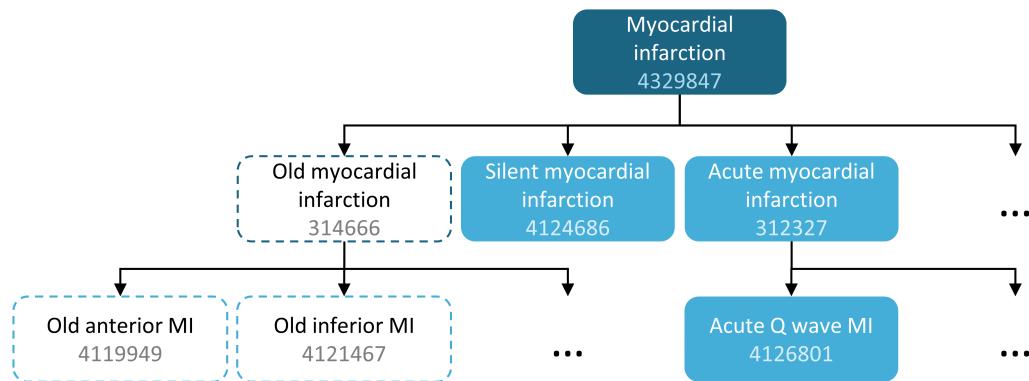


Figure 10.2: "심근경색"과 그 하위 개념을 포함하지만 "과거 심근경색"과 그 하위 개념은 제외하는 concept set

10.4 확률적 코호트 정의

규칙 기반 코호트 정의는 코호트 정의를 수행할 때 널리 사용되는 방법이다. 그러나 코호트를 만들기 위해 전문가끼리 합의를 이루는 것은 매우 많은 시간이 소요되는 일이다. 확률적 코호트 정의는 코호트 속성의 효율적인 선택을 위한 대안적인 기계 구동 방식이다. 이 접근법에서, 레이블이 붙은 증례를 이용하여지도 기계학습이 코호트 설계 알고리즘을 학습한다. 이 알고리즘은 전반적인 연구의 정확도의 균형 (민감도/특이도 간의 균형 trade-off)를 향상시킬 수 있도록 추가 수정을 거칠 수 있다.

이 접근 방법을 CDM의 데이터에 적용한 예는 아프로디테 (APHRODITE: Automated PHenotype Routine for Observational Definition, Identification, Training and Evaluation) R 패키지이다. 이 패키지는 불완전하게 레이블이 붙은 데이터로부터 학습하는 능력을 결합한 코호트 구축 프레임워크를 제공한다. (Banda et al., 2017)

10.5 코호트 정의 유효성

코호트를 구축할 때, 다음 중 더 중요한 것이 무엇인지 고려하는 것이 필요하다: 코호트 조건에 해당하는 환자를 모두 찾는 것이 더 중요한가? 아니면 확신이 가는 환자만 찾는 것이 더 중요한가?

코호트를 구축할 때 당신의 전략은 전문가가 질병을 얼마나 엄격하게 정의하는지에 의존할 것이다. 얻을 수 있는 모든 것을 사용하거나, 최소 공통분모를 사용하거나 이 둘을 결충하는 코호트 정의를 작성할 수 있다. 관심 코호트를 적절하게 연구하기 위해 얼마나 엄격한 임계값을 사용할지는 궁극적으로 연구자의 재량에 달려 있다.

이 장의 시작 부분에서 언급했듯이 코호트 정의는 데이터로부터 무엇인가 관찰하고자 하는 것을

유추하려는 시도이다. 그러면 그러한 시도에서 코호트를 얼마나 잘 정의했는지 의문을 품게 된다. 일반적으로, 규칙 기반의 코호트 정의나 확률적 알고리즘의 검증은 작성한 코호트를 ‘절대 표준 gold standard’ 참고 값 (즉 수작업으로 차트를 검토한 것)과 비교함으로써 검증할 수 있다. 이에 대해서는 16장 (“임상적 타당성”)에서 자세히 설명한다.

10.5.1 OHDSI 절대 표준 표현형 라이브러리

커뮤니티를 지원하기 위해서 OHDSI 절대 표준 표현형 라이브러리 OHDSI Gold Standard Phenotype Library(GSPL) 그룹이 형성되었다. GSPL 그룹의 목표는 규칙 기반 및 확률적 방법으로 커뮤니티 기반의 코호트 라이브러리를 개발하는 것이다. GSPL은 OHDSI 커뮤니티의 멤버가 각자의 연구를 위해 커뮤니티가 검증한 코호트를 찾아서 실행시킬 수 있게 하였다. 이 ‘절대 표준 gold standard’ 코호트는 라이브러리 안에 들어 있다. GSPL과 관련된 추가적인 정보를 얻으려면 OHDSI 워크그룹 페이지에 문의한다. 이전에 소개되었던 APHRODITE (Banda et al., 2017) 와 PheEvaluator tool (Swerdel et al., 2019) 뿐만 아니라 OHDSI 네트워크에서 EHR과 유전 정보를 공유하기 위해 만들어진 Electronic Medical Records and Genomics (eMERGE) Phenotype Library (Hripcsak et al., 2019) 도 해당 작업 그룹에서 다루고 있다. 당신이 코호트를 설계하는 데 관심이 많다면, 이 워크 그룹에 참여한다.

10.6 고혈압 환자 코호트 작성하기

규칙 기반의 접근 방법으로 코호트를 작성해보자. 이번 예제에서는, 고혈압의 초기 치료를 위해 *ACE inhibitors* 단일 치료를 시작한 환자를 찾을 것이다.

이 연습을 진행하면서 표준 소모 도표 attrition chart와 비슷한 코호트를 작성하게 될 것이다. 그럼 10.3은 우리가 어떤 논리로 코호트를 작성할지 보여준다.

ATLAS 유저 인터페이스를 사용해서 코호트를 작성해도 되고, 쿼리를 직접 작성해도 된다. 이 장에서는 두 가지 방법 모두를 간단히 소개하겠다.

10.7 ATLAS를 이용해 코호트 작성하기

ATLAS를 시작하기 위해  Cohort Definitions 버튼을 클릭한다. 다음으로 ‘New cohort’ 버튼을 클릭한다. 다음 화면에서 비어 있는 코호트를 확인할 수 있을 것이다. 그럼 10.4에서 당신이 현재 보고 있는 화면을 확인한다.

먼저 “New Cohort Definition”로 지정된 코호트 이름을 다른 이름으로 바꿔 지어 주기를 추천한다.

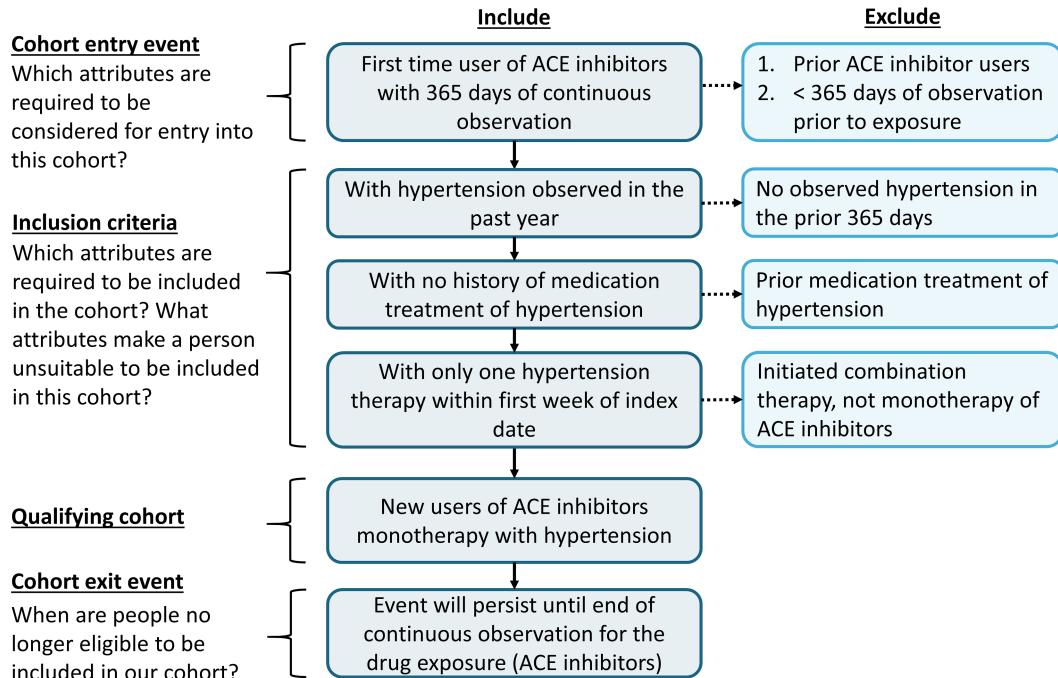


Figure 10.3: 만들고자 하는 코호트의 논리적 구성도

New Cohort Definition

New Cohort Definition

Definition Concept Sets Generation Reporting Export

enter a cohort definition description here

Cohort Entry Events

Events having any of the following criteria:

with continuous observation of at least days before and days after event index date

Limit initial events to: earliest event per person.

+ Add Initial Event

Restrict initial events

Inclusion Criteria

New inclusion criteria

Limit qualifying events to: earliest event per person.

Figure 10.4: 새로운 코호트 정의

The screenshot shows the ATLAS software interface for cohort definition. At the top, it says 'Cohort #1771427' and 'EXAMPLE: new users of ACE inhibitors as first-line mono-therapy for hypertension'. Below this are tabs for 'Definition', 'Concept Sets', 'Generation', 'Reporting', and 'Export'. A large input field says 'enter a cohort definition description here'. Under 'Cohort Entry Events', it says 'Events having any of the following criteria:' followed by a dropdown for 'with continuous observation of at least [0] days before and [0] days after event index'. Below this is a button 'Restrict initial events'. To the right, a dropdown menu is open with options like 'Add Initial Event', 'Add Condition Era', 'Add Condition Occurrence', 'Add Death', 'Add Device Exposure', 'Add Dose Era', 'Add Drug Era', 'Add Drug Exposure', and 'Find patients with specific diagnosis era'.

Figure 10.5: 초기 이벤트 추가하기

'New users of ACE inhibitors as first-line monotherapy for hypertension'라고 지으면 적당할 것이다.



ATLAS는 동일한 이름을 가진 두 개의 코호트를 허용하지 않는다. 기존에 있던 이름을 사용 하려고 하면 에러 메시지가 뜰 것이다.

이름을 정했으면,

을 눌러서 코호트를 저장하라.

10.7.1 초기 이벤트 기준 Initial Event Criteria

이제 우리는 초기 코호트 이벤트를 정의해야 한다. “Add initial event”를 클릭한다. 어떤 도메인 내에서 기준을 설정할지 결정해야 한다. 초기 코호트 이벤트를 정의하기 위해 어떤 도메인이 필요한지 어떻게 알 수 있을까? 함께 알아보자.

그림 10.5에서 볼 수 있듯이 ATLAS는 각 기준 아래에 설명을 제공한다. 우리가 만약 특정 질병을 진단받은 환자를 찾으려 한다면 CONDITION_OCCURRENCE 도메인에서 기준을 만들어야 한다. 특정 약물이나 특정 계열의 약물을 복용한 환자를 찾고 싶다면 DRUG_EXPOSURE 도메인에서 기준을 만들어야 한다. 우리는 고혈압의 초 치료로 ACE inhibitors 단독요법을 시행한 환자를 찾고 싶기 때문에 DRUG_EXPOSURE 도메인에서 기준을 만들어야 한다. 그런데 고혈압을 진단받은

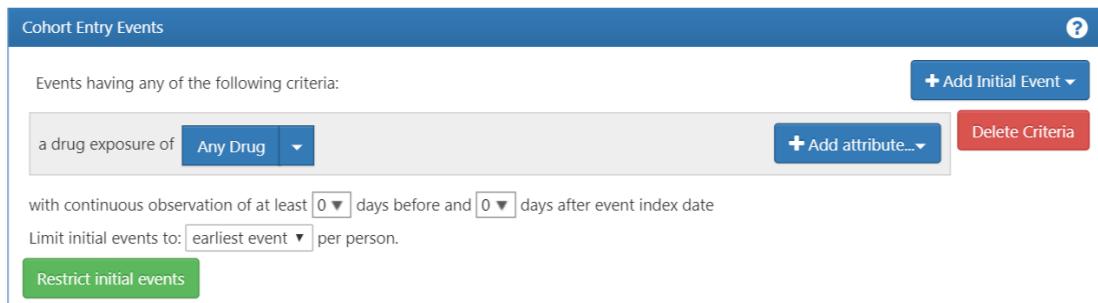


Figure 10.6: 약물 복용에 관하여 정의하기

환자도 찾아야 하지 않는가? 맞다! 고혈압 진단과 관련해서는 다른 기준을 만들 것이다. 하지만 고혈압 약물을 복용하기 시작한 날짜가 코호트 시작 날짜, 즉 시작 이벤트 initial event가 될 것이다. 고혈압 진단은 소위 추가적 적격 기준 *additional qualifying criteria*이 된다. 이에 관해서는 뒤에서 다시 설명하겠다. 이제 'Add Drug Exposure'를 클릭한다.

화면은 당신이 선택한 기준에 따라 업데이트되겠지만, 아직 끝난 것은 아니다. 그림 10.6에서 볼 수 있다시피 ATLAS는 우리가 어떤 약물을 찾고자 하는지 아직 모른다. ATLAS에게 어떤 concept set이 ACE inhibitors와 연관이 있는지 알려주어야 한다.

10.7.2 concept set 정의하기

ACE inhibitors를 정의하기 위한 대화 상자를 열기 위해 ▾을 클릭한다.

시나리오 1: 당신은 아직 concept set을 만들지 않았다

아직 당신의 코호트에 추가할 concept set을 만들지 않았다면, 이것을 먼저 진행해야 한다. 'Concept set' 탭의 'New Concept Set'을 클릭하여 코호트를 작성하는 데 쓰일 concept set을 만들 수 있다. concept set의 이름을 'Unnamed Concept Set'에서 새로 만들어 주어야 한다. 이제 Search 모듈을 통해 ACE inhibitors를 나타내는 개념을 찾아보자. (그림 10.7)

필요한 용어를 찾았다면, 을 클릭함으로써 그 개념을 선택할 수 있다. 그림 10.7의 좌상단의 왼쪽을 향하는 화살표 버튼을 클릭하여 코호트 작성 페이지로 돌아갈 수 있다. 적절한 용어를 찾기 위한 방법은 5장 ("OMOP 표준 용어")을 참고한다.

그림 10.8에서 우리가 선택한 concept set의 구성을 확인할 수 있다. 우리는 모든 ACE inhibitors 성분을 선택했으며, 하위 개념도 포함했다. 'Included concepts'를 클릭하여 포함된 21,536개의 모든 개념을 확인할 수 있고, 'Included Source Codes'를 클릭하여 모든 원천 코드를 확인할 수도 있다.

◀ EXAMPLE: new users of ACE inhibitors as first-line mono-therapy for hypertension ➤ ACE Inhibitors

Search

Search Import

ace inhibitors 

Advanced Options

Column visibility Copy CSV Show 15 entries Filter: Previous [1] Next

Vocabulary

- ATC (6)
- Multilex (1)
- VA Class (1)
- LOINC (1)

Class

- ATC 4th (4)

	Id	Code	Name	Class	RC	DRC	Domain	Vocabulary
21601784	C09AA	ACE inhibitors, plain	ATC 4th	0	507,772	Drug	ATC	
21601783	C09A	ACE INHIBITORS, PLAIN	ATC 3rd	0	507,772	Drug	ATC	
21601802	C09BA	ACE inhibitors and diuretics	ATC 4th	0	10,982	Drug	ATC	
21601801	C09B	ACE INHIBITORS, COMBINATIONS	ATC 3rd	0	10,982	Drug	ATC	

Figure 10.7: ACE Inhibitors 용어 찾기

Concept Set Expression Included Concepts 21536 Included Source Codes Export Import

Name: ACE Inhibitors

Show 25 entries Search: []

Showing 1 to 15 of 15 entries Previous [1] Next

	Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	Exclude	Descendants	Mapped
1335471	18867	benazepril	Drug	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1340128	1998	Captopril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19050216	21102	Cilazapril	Drug	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1341927	3827	Enalapril	Drug	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1342001	3829	Enalaprilat	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1363749	50166	Fosinopril	Drug	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19122327	60245	imidapril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1308216	29046	Lisinopril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1310756	30131	moexipril	Drug	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1373225	54552	Perindopril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1331235	35208	quinapril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1334456	35296	Ramipril	Drug	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19040051	36908	spirapril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1342439	38454	trandolapril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19102107	39990	zofenopril	Drug	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Classification Non-Standard Standard

Figure 10.8: ACE inhibitor를 포함한 약물의 concept set

Import Concept Set From Repository...

Show 10 entries Filter Repository Concept Sets: ace inhibitors

Id	Title	Created	Modified	Author
1794480	[OHDSI EU 2019] Excluded concepts of ACE inhibitors or Thiazide diuretics	03/28/2019 11:04 AM	03/28/2019 11:04 AM	anonymous
963	ACE Inhibitors			anonymous
3268	COPY OF: ACE Inhibitors			anonymous
99283	Ace Inhibitors			anonymous
142965	PheKB ACE-I ACE inhibitors			anonymous

Showing 1 to 5 of 5 entries (filtered from 11,667 total entries)

New Concept Set

Previous 1 Next

Figure 10.9: ATLAS 저장소에서 concept set 가져오기

시나리오 2: 당신은 이미 concept set을 만들었다

만약 당신이 이미 concept set을 만들었고, ATLAS에 저장했다면, 'Import Concept Set'을 클릭한다. 그러면 그림 10.9에서 볼 수 있다시피 ATLAS의 concept set 저장소에서 당신의 concept set을 찾을 수 있는 대화창이 뜬다. 이번 예시에서는 사용자가 ATLAS에 저장되어 있던 concept set을 이용한다고 가정하자. 사용자는 검색 창에 'ACE inhibitors'를 검색하였고, 검색 내용이 이름에 포함된 concept set을 볼 수 있을 것이다. 사용자는 해당하는 concept set을 클릭하여 선택할 수 있다 (참고로 당신이 concept set을 선택하면 대화창은 사라진다). Any Drug 칸이 당신이 선택한 concept set의 이름으로 바뀌어 있다면 성공한 것이다.

10.7.3 추가적 초기 이벤트 기준 Additional Initial Event Criteria

이제 코호트에 concept set을 만들어 붙였지만, 아직 끝난 것이 아니다. 우리는 ACE inhibitors를 태어나서 처음 복용한 사람을 찾고 있다. 이는 ACE inhibitors를 처음 복용한 환자 기록을 찾는 것을 의미한다. 이를 지정하기 위해 당신은 '+Add attribute'를 클릭하여 'Add first exposure criteria'를 선택해야 한다. 당신이 만든 기준에 다른 특성도 지정할 수 있다는 것을 참고한다. 약물을 복용한 날짜나 나이, 성별 혹은 약물과 관련한 다른 특성을 지정할 수 있다. 각 도메인에 따라 선택할 수 있는 특성이 다르다.

선택했으면, 창은 자동으로 닫힌다. 선택된 특성은 초기 기준과 같은 칸 안에서 볼 수 있을 것이다 (그림 10.10 참조).



현재 ATLAS 디자인은 활용하기에 약간 혼란스러울 수 있다. 생긴 모양과는 다르게 버튼 X는 'NO'를 의미하는 것이 아니다. 이는 사용자에게 해당 기준을 삭제할 수 있도록 만들어진

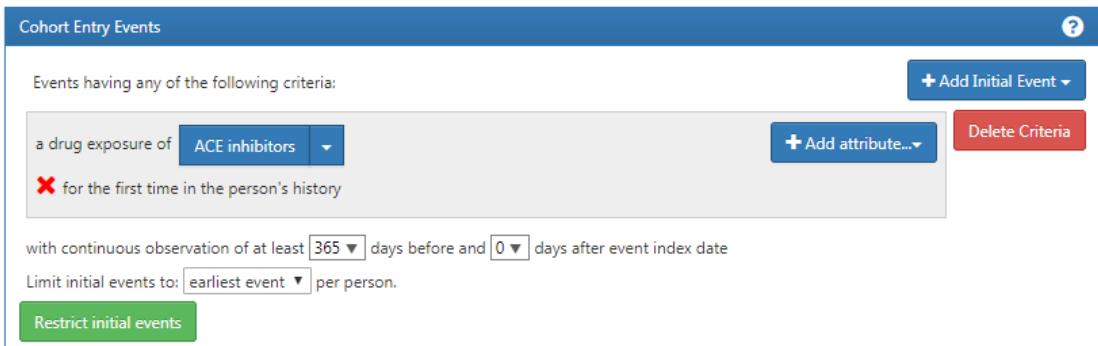


Figure 10.10: Index date 이전에 필요로 하는 관찰 기간 설정하기.

버튼이다. 만약 당신이 **X**를 클릭한다면, 해당 기준은 사라질 것이다. 그러므로 당신의 기준을 사라지지 않은 채 그대로 보존시키고 싶다면, 옆에 **X** 버튼을 그대로 놔두어야 한다.

이제 만족스러운 초기 이벤트를 설정했다. 환자가 처음으로 약물을 복용했다는 사실을 보증하기 위해, 환자의 그 이전 기록을 확인할 수 있는 충분한 기간을 설정해 주면 좋을 것이다. 짧은 관찰 기간을 가진 환자는 우리가 확인할 수 없는 다른 곳에서 약물을 복용하였을 수도 있다. 우리가 이것을 강제적으로 막을 수는 없지만, 기준일자 index date 이전에 관찰 기간을 설정함으로써 최소한 해당 관찰 기간 동안에는 약물 복용이 이루어지지 않았음을 보증할 수 있다. 이를 위해 관찰 기간을 설정하는 부분이 있으며, 구체적인 관찰 기간을 직접 설정할 수도 있다. 우리는 초기 이벤트 이전에 365일 동안 관찰된 환자를 필요로 한다. 그럼 10.10처럼 관찰 기간을 다음과 같이 설정하라: *with continuous observation of 365 days before*. 당신 연구팀의 재량껏 관찰 기간을 설정하면 된다. 다른 코호트에서는 관찰 기간을 다르게 설정해서 다양한 시도를 해볼 수 있다. 이는 환자의 과거력에 관한 기간이며, 기준일자 이후의 시간은 포함하지 않는다. 그러므로 우리는 0 dates after index date라고 설정해야 한다. 우리는 생에 처음 ACE inhibitors를 복용한 환자를 찾고 싶어서 *limit initial events to the “earliest event” per person* (한 환자에서 발생한 여러 번의 ACE inhibitor 복용 중, 첫 번째 복용을 초기 이벤트로 설정하는 것)으로 설정한다.

지금껏 설정한 논리를 한눈에 보기 위해서 환자의 타임라인을 설정해볼 수 있다.

그림 10.11에서 각 행은 코호트에 들어올 자격을 갖출 수 있는 환자 개개인을 나타낸다. 그리고 진한 별은 환자가 특정 기준을 만족했던 시간을 나타낸다. 추가 기준이 설정될수록 진한 별 대신 연한 별이 그려진 것을 볼 수 있다 (즉, 추가 기준에 의해서 코호트에 포함되지 못하고 탈락). 이는 환자가 조건을 모두 만족하는 이벤트도 가지고 있지만, 그렇지 않은 이벤트도 가지고 있음을 의미한다. 마지막 기준을 그런 그림을 보면 ACE inhibitors를 처음으로 복용하였으며, 복용 이전에 최소 365

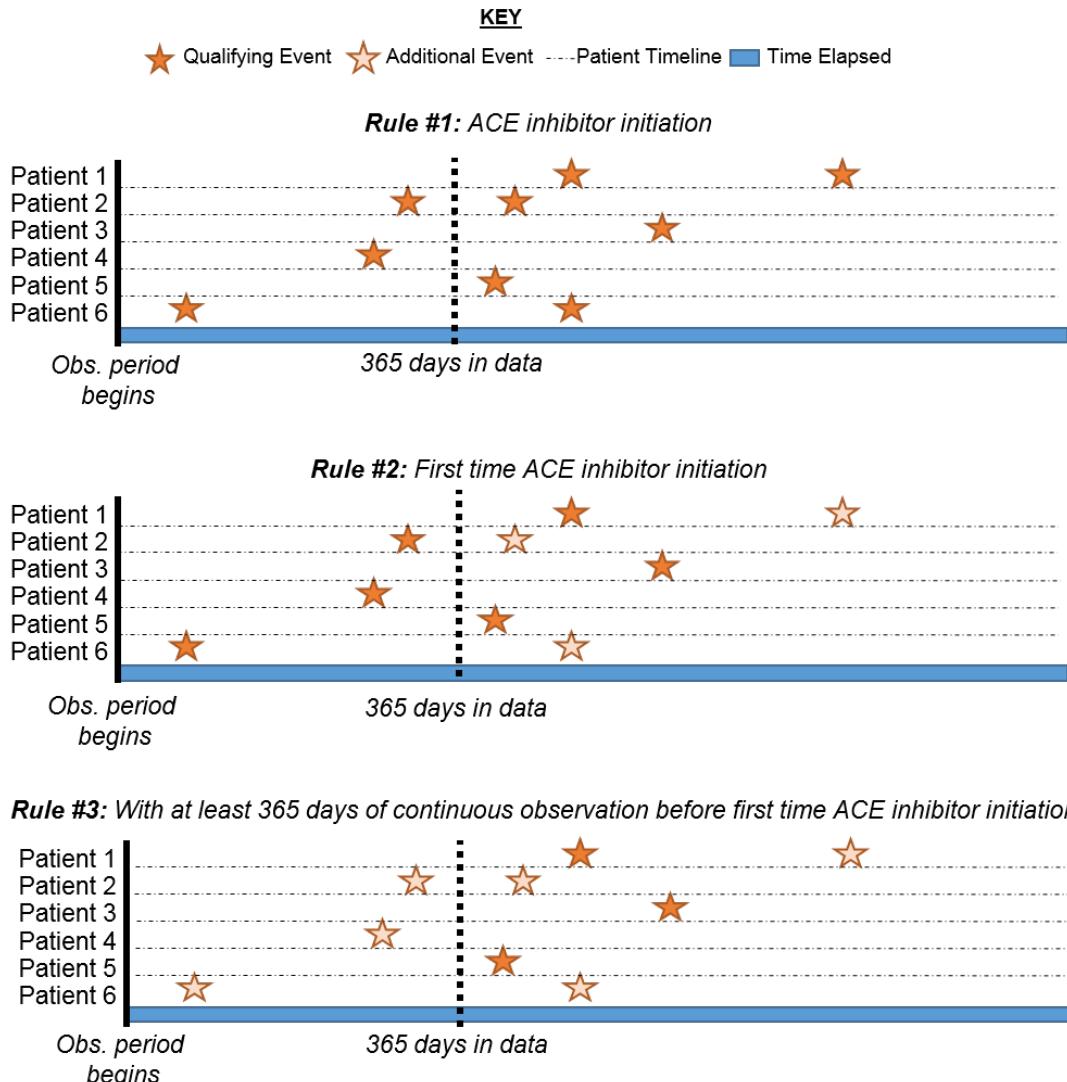


Figure 10.11: 기준이 적용됨에 따라 환자가 코호트에 적합한지 살펴보기

일의 관찰 기간을 가진 환자 (환자 1번, 환자 3번, 환자 5번의 진한 별은 관찰에 포함; 환자 1번의 연한 별은 관찰에서 탈락)를 확인할 수 있다. 당신의 코호트를 설계할 때 OHDSI 포럼에 참여하는 연구자의 의견을 참고하면 더 좋을 것이다.

10.7.4 포함 기준 Inclusion Criteria

코호트 진입 이벤트를 설정했으면, 다음 두 옵션을 통해 추가적 이벤트를 설정할 수 있다: ‘Restrict initial events’, 그리고 ‘New inclusion criteria’. 이 두 옵션 사이에는 ATLAS가 사용자에게 어떤 임시 정보를 제공하는가의 차이가 있다. 만약 당신이 기준을 추가하기 위해 ‘Restrict initial events’를 사용한다면, ATLAS에서 조건에 맞는 대상 환자 수를 셀 때, 모든 기준을 충족시키는 사람의 숫자만을 얻게 될 것이다. ‘New inclusion criteria’를 통해 기준을 추가한다면, 추가 포함 기준을 적용하여 손실된 환자 수를 보여주는 감소 차트를 확인할 수 있을 것이다. 당신이 추가한 기준에 의해 얼마나 큰 손실이 발생하는지 단계별로 보여주는 감소 차트를 확인하는 것은 중요하기 때문에 ‘New inclusion criteria’를 통해 기준을 추가하는 것을 권장한다. 이를 통해 코호트에 포함되는 환자 수를 급격하게 감소시키는 기준이 무엇인지 확인할 수 있다. 당신은 해당 기준을 완화하여 더욱 큰 코호트를 얻을 수 있다. 이것은 궁극적으로 이 코호트를 설계하는 전문가의 재량에 달려있다.

이제 ‘New inclusion criteria’를 통해 기준을 추가해보자. 이는 위에서 코호트 기준을 설정한 것과 동일한 방법으로 하면 된다. 특정 기준을 만들어서 넣은 다음, 특정 속성을 추가할 수 있을 것이다. 우리가 첫 번째로 추가할 기준은 다음과 같다: *ACE inhibitors* 약물을 복용한 시점 이후 0~365일 이내에 최소 1회 고혈압이 발생한 사람. ‘New inclusion criteria’를 클릭한 다음, 그 기준을 설명해 줄 수 있는 이름을 정한다. 그래야 나중에 이 코호트를 다시 보았을 때 자신이 무엇을 만들었는지 헷갈리지 않을 것이다.

이 새로운 기준에 이름을 달고 난 다음, “+Add criteria to group” 버튼을 클릭하여 여러 규칙을 담은 기준을 설계한다. 이 버튼은 “Add Initial Event”와 비슷한데, 다만 “+Add criteria to group” 버튼은 초기 이벤트를 설계하고 수정하는 버튼이 아니다. 우리는 여기서 여러 개의 기준을 추가할 수 있다. 예를 들어 질병의 발생을 확인하는 여러 가지 방법을 가지고 있다고 가정하자 (예를 들어 CONDITION_OCCURRENCE, 혹은 DRUG_EXPOSURE, 혹은 MEASUREMENT를 사용한 방법). 모두 다른 도메인이고 각각 다른 기준이 필요하겠지만 특정 조건을 찾는 하나의 기준으로 그룹화할 수 있다. 이 경우에는, 우리는 고혈압의 진단을 찾고 싶기 때문에 “Add condition occurrence”를 선택한다. 여기에 적절한 concept set을 붙이는 등 초기 이벤트를 설정할 때와 비슷하게 하면 된다. 또한, ACE inhibitor를 처음 복용한 날 index date로 이후 0~365일의 기간을 설정한다. 그림 10.12와 같이 작성될 수 있을 것이다.

아마도 환자를 탐색할 또 다른 기준을 추가하고 싶을 것이다: *with exactly 0 occurrences of hypertension*

The screenshot shows the ATLAS Inclusion Criteria builder. A single criterion is listed:

1. has hypertension diagnosis in 1 yr prior to treatment

Details of the criterion:

- Condition: has hypertension diagnosis in 1 yr prior to treatment
- Description: enter an inclusion rule description
- Having: all of the following criteria:
- With: at least 1 using all occurrences of: Hypertensive disorder
- Where: event starts between 365 days Before and 0 days After index start date (add additional constraint)
- Options: restrict to the same visit occurrence, allow events from outside observation period

Buttons: Copy, Delete, + Add criteria to group...

Limit qualifying events to: earliest event per person.

Figure 10.12: 추가적 포함 기준 1

sion drugs ALL days before and 1 day before index start date (ACE inhibitor 투여 이전에 어떠한 고혈압 약물도 복용하지 않은 사람). (역자주: xx before and yy 구문은 항상 혼란스럽지만, from xx to yy로 고쳐서 생각하면 이해하기 쉽다. 즉 해당 조건의 시작과 끝을 지정하는 용법이다. 앞선 예라면 “과거 전체로부터 시작해서 (ACE inhibitor가 최초 투여된) 기준 날짜 index date 바로 하루 전까지 고혈압치료제를 한 번도 복용하지 않은 경우”가 된다) 먼저 “New inclusion criteria”를 클릭해 당신의 기준을 설정한 다음, “+Add criteria to group”을 클릭한다. 이는 DRUG_EXPOSURE의 영역이니 “Add Drug Exposure”를 클릭한 다음, 고혈압 약물의 concept set을 붙인다. 그리고, index date로부터 ALL days before and 0 days after라는 시간을 설정해준다 (역자 주: “ALL days before and 0 days after”는 “ALL days before and 0 days before”와 같은 의미이며 기준 날짜 index date를 포함하여 그날까지의 의미이다. 그림에는 “ALL days before and 1 days before”로 표현했는데 과거 전체로부터 기준 날짜 index date 하루 전까지의 의미이다. 본인이 원하는 기준이 무엇인지에 따라 구분하여 사용하라). exactly 0 occurrence를 선택하였는지 다시 한번 확인하고 그림 10.13과 같이 잘 만들어졌는지 확인한다.

“having no occurrences”(발생하지 않았다)라는 말이 왜 “exactly 0 occurrences”(발생 횟수 0회)라고 쓰이는지 혼란스러울 수 있다. 이는 ATLAS가 사용하는 규칙이다. ATLAS는 오직 포함 기준만을 사용하고, 제외 기준을 사용하지 않는다. 만약 당신이 어떤 특성을 가진 환자를 제외하고 싶다면 해당 특성을 0회 가지는 환자를 포함한다는 말로 대체하여야 한다. 처음에는 헷갈릴 수 있지만 계속 사용하다 보면 이러한 논리가 익숙해질 것이다.

마지막으로 목표 환자군 설정을 위한 기준을 하나 더 추가해야 한다: *with exactly 1 occurrence of hypertension drugs between 0 days before and 7 days after index start date AND can only start one HT drug (an ACE inhibitor) – index date 이후 0~7일 동안 정확히 1회의 항고혈압제 치방이 발생*

The screenshot shows the ATLAS Inclusion Criteria interface. On the left, under 'New inclusion criteria', there is a list of three items:

1. has hypertension diagnosis in 1 yr prior to treatment
2. Has no prior antihypertensive drug exposures in medical history

To the right of this list is a detailed configuration panel for the second item. It includes fields for 'having all' or 'any' occurrences of criteria, a dropdown for 'using all' or 'using distinct' occurrences, and a 'with' section where 'exactly 0' occurrences of 'Hypertension drugs' are selected. Below this, a 'where' section specifies 'event starts between All days Before and 1 days Before' and includes an 'index start date' button and 'add additional constraint' link. There are also checkboxes for 'restrict to the same visit occurrence' and 'allow events from outside observation period'. At the bottom of the panel, there are 'Copy' and 'Delete' buttons.

At the very bottom of the interface, there is a note: 'Limit qualifying events to: earliest event per person.'

Figure 10.13: 추가적 포함 기준 1

This screenshot shows the same ATLAS Inclusion Criteria interface as Figure 10.13, but with a third item added to the 'New inclusion criteria' list:

1. has hypertension diagnosis in 1 yr prior to treatment
2. Has no prior antihypertensive drug exposures in medical history
3. Is only taking ACE as monotherapy, with no concomitant combination treatments

The configuration panel for the third item is identical to the one in Figure 10.13, showing 'exactly 1' occurrence of 'Hypertension drugs' between '0 days Before' and '7 days After' the index start date. The 'Copy' and 'Delete' buttons are visible at the top right of the panel.

At the bottom, the note 'Limit qualifying events to: earliest event per person.' is present.

Figure 10.14: 추가적 포함 기준 3

했으며, 반드시 ACE inhibitor로 고혈압 약물치료를 시작해야 한다. 먼저 “New inclusion criteria”를 클릭해 당신의 기준을 설정한 다음, “+Add criteria to group”을 클릭한다. DRUG_ERA 를 사용하여 “Add Drug Era”를 클릭한 다음, 고혈압 약물의 concept set을 붙였다. (역자 주: Drug era는 9.7.1에 간략히 설명되어 있는데 약물 노출 테이블에서 계산된 것으로 연속으로 치방된 동일한 성분의 여러 약물 노출을 합쳐서 하나의 기간으로 표현한 것이다. 동일한 성분의 약물 노출 간에 30 일 이상의 공백이 있으면 다른 drug era로 계산된다. 이 점은 condition era도 마찬가지이다) 그리고 index date 이후 0~7일이라는 시간을 설정해준다. 그림 10.14를 통해 진행된 모습을 확인한다.

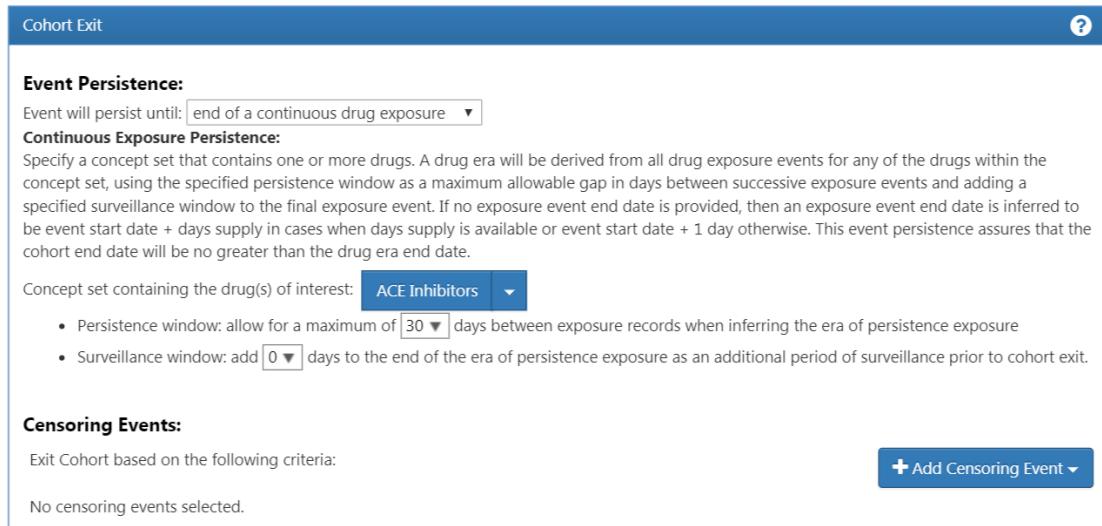


Figure 10.15: 코호트 종료 기준

10.7.5 코호트 종료 기준 Cohort Exit Criteria

이제 모든 적절한 포함 기준을 추가했다. 다음으로 코호트 종료 기준을 정해야 한다. 사람이 더 이상 이 코호트에 포함될 자격이 없어질 때는 언제일지 생각해보아야 할 것이다. 우리는 이 코호트에서 약물을 처음 복용한 사람을 추적한다. 즉, 약물 복용을 중단한 시점에 환자는 코호트에서 나오게 하면 된다. 약물 복용이 중단되는 동안에는 해당 환자에게 무슨 일이 일어나는지 확인할 수 없기 때문이다. 또한 약물 복용 사이에 허용되는 공백 기간을 지정하기 위해 persistence 창에서 기준을 설정할 수 있다. 이 연구에서 전문가는 약물 복용 사이에 최대 30일의 공백 기간은 허용된다고 결론지었다.

왜 공백 기간이 허용되는가? 데이터 세트에서 실제로 이루어지는 일의 일부만 관찰할 수 있을 뿐이다. 특히 환자의 약물 복용에 관한 정보는 처방전의 기록으로 확인한다. 그리고 처방전을 통해 하루 치 이상의 약을 처방하기 때문에 기록이 비어 있는 시간 동안에도 환자가 약을 복용하고 있다는 합리적 추론이 가능하다.

Event will persist “end of a continuous drug exposure”를 선택하고, persistence 창에 “allow for a maximum of 30 days”를 추가한 다음 ‘ACE inhibitor’ concept 모음을 추가로 지정해 주면 된다. 그럼 10.15를 통해 이를 확인한다.

이 코호트의 경우 다른 중도 절단 사건 Censoring event는 선택되지 않았다. 하지만 Censoring event를 추가해야 하는 다른 코호트를 만들어야 한다면 다른 속성을 추가했던 것과 비슷하게 진행하면 된다. 이제 코호트를 성공적으로 만들었다. 반드시 버튼을 눌러야 한다. 축하한다! 코호트를

만드는 것은 OHDSI가 제공하는 도구를 이용할 때 가장 중요한 부분이다. 이제 ‘Export’ 탭을 클릭하면 ATLAS에 당신이 정의한 코호트가 SQL 코드와 JSON 파일로 저장되어 다른 연구자와 공유할 수 있다.

10.8 SQL을 사용하여 코호트 구현하기

여기서는 동일한 코호트를 SQL과 R을 이용하여 작성하는 방법을 설명할 것이다. 9장에서 설명하였듯이 OHDSI는 SqlRender, DatabaseConnector라는 두 개의 R 패키지를 제공하는데, 이는 SQL의 코드가 다양한 플랫폼에서 실행될 수 있게끔 SQL문을 자동으로 번역해준다.

자세한 설명을 위해 SQL 코드를 여러 개의 단계로 나눌 것이고, 각 단계에서는 다음 단계에 필요한 임시 테이블이 생성될 것이다. 이런 설명 방법이 가장 효율적이지는 않겠지만 매우 긴 단일 명령문을 읽는 것보단 쉬울 것이다.

10.8.1 데이터베이스에 연결하기

먼저 R이 어떻게 서버에 접속하는지 알려주어야 한다. `createConnectionDetails`라는 기능을 가진 DatabaseConnector 패키지를 사용할 것이다. 다양한 데이터베이스 관리 시스템(DBMS)에 연결하는 데 필요한 설정을 확인하려면 `?createConnectionDetails` 과 같이 입력한다. 예를 들어 아래의 코드를 이용해 PostgreSQL에 연결할 수 있다:

```
library(CohortMethod)

connDetails <- createConnectionDetails(dbms = "postgresql",
                                         server = "localhost/ohdsi",
                                         user = "joe",
                                         password = "supersecret")

cdmDbSchema <- "my_cdm_data"
cohortDbSchema <- "scratch"
cohortTable <- "my_cohorts"
```

마지막 3줄은 변수 `cdmDbSchema`, `cohortDbSchema`, 그리고 `cohortTable`을 정의한다. 나중에 이 변수를 R에게 CDM 포맷의 데이터가 어디에 있으며, 우리가 만든 코호트가 어디에 생성되어야 하는지 알려주기 위해 사용할 것이다. Microsoft SQL Server에서는 `cdmDbSchema <- "my_cdm_data.dbo"`의 예시와 같이 데이터베이스와 스키마 모두를 지정해 주어야 함을 참고한다.

10.8.2 개념 결정하기

가독성을 위해 R에 필요한 개념 ID를 정의하고 SQL에 전달한다:

```
aceI <- c(1308216, 1310756, 1331235, 1334456, 1335471, 1340128, 1341927,
         1342439, 1363749, 1373225)

hypertension <- 316866

allHtDrugs <- c(904542, 907013, 932745, 942350, 956874, 970250, 974166,
                 978555, 991382, 1305447, 1307046, 1307863, 1308216,
                 1308842, 1309068, 1309799, 1310756, 1313200, 1314002,
                 1314577, 1317640, 1317967, 1318137, 1318853, 1319880,
                 1319998, 1322081, 1326012, 1327978, 1328165, 1331235,
                 1332418, 1334456, 1335471, 1338005, 1340128, 1341238,
                 1341927, 1342439, 1344965, 1345858, 1346686, 1346823,
                 1347384, 1350489, 1351557, 1353766, 1353776, 1363053,
                 1363749, 1367500, 1373225, 1373928, 1386957, 1395058,
                 1398937, 40226742, 40235485)
```

10.8.3 약물을 처음 복용한 환자 찾기

먼저 각 환자에 대한 ACE inhibitor의 첫 복용을 찾을 것이다:

```
conn <- connect(connectionDetails)

sql <- "SELECT person_id AS subject_id,
           MIN(drug_exposure_start_date) AS cohort_start_date
      INTO #first_use
     FROM @cdm_db_schema.drug_exposure
    INNER JOIN @cdm_db_schema.concept_ancestor
        ON descendant_concept_id = drug_concept_id
   WHERE ancestor_concept_id IN (@ace_i)
  GROUP BY person_id;"

renderTranslateExecuteSql(conn,
                        sql,
```

```
cdm_db_schema = cdmDbSchema,
ace_i = aceI)
```

DRUG_EXPOSURE 테이블을 CONCEPT_ANCESTOR 테이블에 조인함으로써 ACE inhibitor 를 포함하는 모든 약물을 찾았다는 것을 참고한다.

10.8.4 약물 복용 이전 최소 365일 동안 관찰될 수 있었던 환자

OBSERVATION_PERIOD 테이블을 조인하여 약물 복용 이전 최소 365일 동안 관찰될 수 있었던 환자를 선택해야 한다:

```
sql <- "SELECT subject_id,
cohort_start_date
INTO #has_prior_obs
FROM #first_use
INNER JOIN @cdm_db_schema.observation_period
ON subject_id = person_id
AND observation_period_start_date <= cohort_start_date
AND observation_period_end_date >= cohort_start_date
WHERE DATEADD(DAY, 365, observation_period_start_date) < cohort_start_date;"

renderTranslateExecuteSql(conn, sql, cdm_db_schema = cdmDbSchema)
```

10.8.5 이전에 고혈압을 진단받은 환자

기준 날짜 index date로부터 365일 이내에 고혈압 진단을 받은 환자여야 한다:

```
sql <- "SELECT DISTINCT subject_id,
cohort_start_date
INTO #has_ht
FROM #has_prior_obs
INNER JOIN @cdm_db_schema.condition_occurrence
ON subject_id = person_id
AND condition_start_date <= cohort_start_date
AND condition_start_date >= DATEADD(DAY, -365, cohort_start_date)
INNER JOIN @cdm_db_schema.concept_ancestor
```

```

ON descendant_concept_id = condition_concept_id
WHERE ancestor_concept_id = @hypertension;"
```

```

renderTranslateExecuteSql(conn,
    sql,
    cdm_db_schema = cdmDbSchema,
    hypertension = hypertension)

```

SELECT DISTINCT를 사용하여 과거에 여러 번의 고혈압 진단을 받은 환자가 여러 번의 코호트 진입을 하지 않도록 했다.

10.8.6 사전에 받은 치료가 없어야 함

이전에 어떠한 고혈압 약물이라도 복용해서는 안 된다:

```

sql <- "SELECT subject_id,
    cohort_start_date
INTO #no_prior_ht_drugs
FROM #has_ht
LEFT JOIN (
    SELECT *
    FROM @cdm_db_schema.drug_exposure
    INNER JOIN @cdm_db_schema.concept_ancestor
        ON descendant_concept_id = drug_concept_id
    WHERE ancestor_concept_id IN (@all_ht_drugs)
) ht_drugs
    ON subject_id = person_id
    AND drug_exposure_start_date < cohort_start_date
WHERE person_id IS NULL;"
```

```

renderTranslateExecuteSql(conn,
    sql,
    cdm_db_schema = cdmDbSchema,
    all_ht_drugs = allHtDrugs)

```

Left join을 사용했으며, DRUG_EXPOSURE 테이블의 person_id 행이 NULL인 (일치하는 기록이 없음을 의미) 행만 찾을 수 있도록 했다는 점에 유의한다.

10.8.7 단독 요법

코호트에 진입하고 첫 1주일 동안은 고혈압 치료 약물에 단 한 번만 노출되도록 설정할 필요가 있다 (역자 주: 아래 코드는 입원환자에게는 정확히 적용되지 않을 수 있다. 만일 입원하여 하루 단위로 고혈압 처방이 이루어진다면 기준 날짜 index date로부터 1주일 이내에 여러 번의 고혈압 처방 start date가 생성되며 아래 코드에 의하면 해당 환자의 두 번째 처방으로 인해 그 환자는 코호트에서 탈락한다. 이러한 점을 피하려면 drug_exposure 테이블 대신에 drug_era table을 이용하면 될 것이다. drug_era 테이블에서는 30일 이내에 처방된 같은 동일 성분명의 약물 노출은 서로 합쳐서 하나의 노출로 만들어 준다. 정확히는 약 처방일 + 처방된 기간 (day supply) + 30을 판단 기준으로 한다. 예를 들어 14일 처방을 냈을 경우 처방 낸 날 + 14 + 30 이내에 같은 성분명의 약물이 다시 처방되면 같은 약물 처방으로 간주하여 하나의 drug_dra로 그 두 처방 (혹은 이후 계속되는 동일 성분명 처방 전체)을 묶어준다. 10.8.8 코호트 종료에서 drug era로 묶는 코드가 제시되고 있다):

```
sql <- "SELECT subject_id,
  cohort_start_date
INTO #monotherapy
FROM #no_prior_ht_drugs
INNER JOIN @cdm_db_schema.drug_exposure
  ON subject_id = person_id
    AND drug_exposure_start_date >= cohort_start_date
    AND drug_exposure_start_date <= DATEADD(DAY, 7, cohort_start_date)
INNER JOIN @cdm_db_schema.concept_ancestor
  ON descendant_concept_id = drug_concept_id
WHERE ancestor_concept_id IN (@all_ht_drugs)
GROUP BY subject_id,
  cohort_start_date
HAVING COUNT(*) = 1;"

renderTranslateExecuteSql(conn,
  sql,
  cdm_db_schema = cdmDbSchema,
  all_ht_drugs = allHtDrugs)
```

10.8.8 코호트 종료 Cohort Exit

이제 코호트 종료 일자를 제외하고 코호트를 완전히 지정했다. 코호트는 노출이 중단되면 종료되도록 정의되며, 노출 사이에 최대 30일의 간격까지는 허용된다. 즉, 약물의 복용 시작뿐만 아니라 ACE

inhibitor의 후속 복용에 대해서도 고려한다는 말이다. SQL을 통해 약물의 후속 복용을 고려하여 약물 복용 기간을 정의하는 것은 매우 복잡하다. 운이 좋게도 약물 복용 기간을 효율적으로 만들 수 있는 표준 코드가 작성되었다. 이 코드는 Chris Knoll이 작성했으며 OHDSI 내에서 종종 마법이라고 불리는 코드이기도 하다. 먼저 병합하려는 모든 약물 복용을 포함하는 임시 테이블을 만든다:

```
sql <- "
SELECT person_id,
       CAST(1 AS INT) AS concept_id,
       drug_exposure_start_date AS exposure_start_date,
       drug_exposure_end_date AS exposure_end_date
  INTO #exposure
     FROM @cdm_db_schema.drug_exposure
    INNER JOIN @cdm_db_schema.concept_ancestor
      ON descendant_concept_id = drug_concept_id
 WHERE ancestor_concept_id IN (@ace_i);"

renderTranslateExecuteSql(conn,
                         sql,
                         cdm_db_schema = cdmDbSchema,
                         ace_i = aceI)
```

그린 다음 순차적 복용을 병합하기 위한 표준 코드를 실행한다:

```
sql <- "
SELECT ends.person_id AS subject_id,
       ends.concept_id AS cohort_definition_id,
       MIN(exposure_start_date) AS cohort_start_date,
       ends.era_end_date AS cohort_end_date
  INTO #exposure_era
     FROM (
       SELECT exposure.person_id,
              exposure.concept_id,
              exposure.exposure_start_date,
              MIN(events.end_date) AS era_end_date
         FROM #exposure exposure
        JOIN (
          --cteEndDates
            SELECT person_id,
```

```
concept_id,
DATEADD(DAY, - 1 * @max_gap, event_date) AS end_date
FROM (
    SELECT person_id,
    concept_id,
    event_date,
    event_type,
    MAX(start_ordinal) OVER (
        PARTITION BY person_id ,concept_id ORDER BY event_date,
        event_type ROWS UNBOUNDED PRECEDING
    ) AS start_ordinal,
    ROW_NUMBER() OVER (
        PARTITION BY person_id, concept_id ORDER BY event_date,
        event_type
    ) AS overall_ord
    FROM (
-- select the start dates, assigning a row number to each
        SELECT person_id,
        concept_id,
        exposure_start_date AS event_date,
        0 AS event_type,
        ROW_NUMBER() OVER (
            PARTITION BY person_id, concept_id ORDER BY exposure_start_date
        ) AS start_ordinal
        FROM #exposure exposure
        UNION ALL
-- add the end dates with NULL as the row number, padding the end dates by
-- @max_gap to allow a grace period for overlapping ranges.

        SELECT person_id,
        concept_id,
        DATEADD(day, @max_gap, exposure_end_date),
        1 AS event_type,
        NULL
        FROM #exposure exposure
)
```

```

    ) rawdata
  ) events
WHERE 2 * events.start_ordinal - events.overall_ord = 0
) events
ON exposure.person_id = events.person_id
  AND exposure.concept_id = events.concept_id
  AND events.end_date >= exposure.exposure_end_date
GROUP BY exposure.person_id,
  exposure.concept_id,
  exposure.exposure_start_date
) ends
GROUP BY ends.person_id,
  concept_id,
  ends.era_end_date;"
```

`renderTranslateExecuteSql(conn,`

```

  sql,
  cdm_db_schema = cdmDbSchema,
  max_gap = 30)
```

이 코드는 모든 후속 복용을 병합하며, `max_gap`의 변수를 통해 약물 복용 사이에 허용되는 최대기간을 정의할 수 있다. 그 결과로 작성된 약물 복용 기간은 임시 테이블 `#exposure_era`라고 불리는 임시 테이블에 기록된다.

다음으로 ACE inhibitor 복용 기간을 우리의 기존 코호트에 조인하기만 하면, ACE inhibitor 복용 종료 날짜를 코호트 종료 날짜로써 사용할 수 있게 된다.

```

sql <- "SELECT ee.subject_id,
  CAST(1 AS INT) AS cohort_definition_id,
  ee.cohort_start_date,
  ee.cohort_end_date
INTO @cohort_db_schema.@cohort_table
FROM #monotherapy mt
INNER JOIN #exposure_era ee
  ON mt.subject_id = ee.subject_id
  AND mt.cohort_start_date = ee.cohort_start_date;"
```

```
renderTranslateExecuteSql(conn,  
                          sql,  
                          cohort_db_schema = cohortDbSchema,  
                          cohort_table = cohortTable)
```

이제 정의한 최종 코호트를 스키마와 테이블에 저장해야 한다. 코호트 정의 ID를 1로 설정하여 동일한 테이블에 저장된 다른 코호트와 구별할 것이다.

10.8.9 정리하기

마지막으로, 작성된 임시 테이블을 정리하고 데이터베이스 서버와의 연결을 끊는 것이 좋다:

```
sql <- "TRUNCATE TABLE #first_use;  
DROP TABLE #first_use;  
  
TRUNCATE TABLE #has_prior_obs;  
DROP TABLE #has_prior_obs;  
  
TRUNCATE TABLE #has_ht;  
DROP TABLE #has_ht;  
  
TRUNCATE TABLE #no_prior_ht_drugs;  
DROP TABLE #no_prior_ht_drugs;  
  
TRUNCATE TABLE #monotherapy;  
DROP TABLE #monotherapy;  
  
TRUNCATE TABLE #exposure;  
DROP TABLE #exposure;  
  
TRUNCATE TABLE #exposure_era;  
DROP TABLE #exposure_era;"  
  
renderTranslateExecuteSql(conn, sql)  
  
disconnect(conn)
```

10.9 요약



- 코호트는 일정 기간 동안 하나 이상의 포함 기준을 만족시키는 사람의 집합이다.
- 코호트 정의는 특정 코호트를 식별하는 데 사용되는 논리에 대한 설명이다.
- 코호트는 OHDSI 분석 도구 전체에서 사용 (및 재사용) 될 수 있다.
- 코호트를 작성하기 위한 두 가지 주요 접근 방법이 있다: 규칙 기반 정의, 확률적 정의
- 규칙 기반의 코호트 정의는 ATLAS나 SQL을 통해 작성할 수 있다.

10.10 예제

전제조건

첫 번째 예제로, ATLAS에 접근이 필요하다. <http://atlas-demo.ohdsi.org>를 통해 접속하거나, 다른 접속 방법을 이용해도 된다.

Exercise 10.1. ATLAS를 이용하여 아래의 기준에 따라 코호트를 작성하라:

- diclofenac을 복용하기 시작한 환자
- 16세 이상의 환자
- 약물 복용 이전 최소 365일간 관찰이 되어 있던 환자
- 이전에 Non-Steroidal Anti-Inflammatory Drug(NSAID)를 복용하지 않은 환자
- 이전에 암을 진단받지 않은 환자
- 약물 복용 중단을 코호트 종료로 정의 (30일 이하의 약물 미복용 기간은 허용)

전제조건

두 번째 예제를 수행하기 위해서 8.4.5절에서 설명된 것처럼 R과 R-Studio, 그리고 JAVA가 설치되어 있다고 가정한다. 또한 아래의 코드를 사용하여 SqlRender, DatabaseConnector, 그리고 Eunomia 패키지를 설치하라:

```
install.packages(c("SqlRender", "DatabaseConnector", "devtools"))
devtools::install_github("ohdsi/Eunomia", ref = "v1.0.0")
```

Eunomia 패키지는 로컬 R 세션에서 수행될 CDM 데이터를 제공한다. 아래의 코드를 이용하여 연결할 수 있다.

```
connectionDetails <- Eunomia::getEunomiaConnectionDetails()
```

CDM 데이터베이스 스키마는 ‘main’ 이다.

Exercise 10.2. 다음 기준에 따르도록 현재 존재하는 코호트 테이블 안에서 급성 심근경색 Acute Myocardial Infarction 코호트를 SQL과 R을 이용하여 만들어 보자:

- 심근 경색을 진단받은 사람 (개념 4329847 ‘심근경색 Myocardial infarction’과 그 하위 개념에서 개념 314666 ‘과거 심근경색 Old myocardial infarction’과 그 모든 하위 개념을 제외하기)
- 입원환자 혹은 응급실 방문 환자만 선택 (개념 9201 ‘Inpatient Visit’, 9203 ‘Emergency Room Visit’, 262 ‘Emergency Room and Inpatient Visit’)

제안된 답변은 부록 E.6에서 찾을 수 있다.

Chapter 11

임상적 특성 분석

Chapter leads: Anthony Sena & Daniel Prieto-Alhambra

관찰형 보건의료 데이터는 다양한 특성을 바탕으로 인구집단의 변화를 이해할 수 있는 귀중한 자원이다. 기술통계를 통해 인구집단의 특성을 확인하는 것은 건강과 질병에 영향을 주는 요인에 대한 가설 설정을 위한 중요한 첫 번째 단계이다. 이번 장에서는 특성 분석 characterization을 위한 방법에 대해 살펴보기로 한다:

- **데이터베이스 수준의 특성 분석 Database-level characterization:** 상위 수준 top-level의 요약 통계량을 제공하여 데이터베이스 전체에 대한 이해를 돋운다.
- **코호트 특성 분석 Cohort characterization:** 의무기록 집합 aggregation 수준에서 인구집단을 기술한다.
- **치료 경로 Treatment pathways:** 특정 기간 동안 한 사람에 대하여 행해진 중재 intervention 순서를 기술한다.
- **발생 Incidence:** 위험 노출 기간 time at risk(TAR) 동안의 임상 결과 outcome의 발생률을 계산한다.

데이터베이스 수준의 특성 분석을 제외하고 나머지 방법은 기준 날짜 index date라고 하는 시점과 관련된 인구 집단에 대해 설명하는 데 목적이 있다. 이와 같은 관심 집단은 코호트라고 정의되며 10장에 기술되어 있다. 코호트는 관심 집단의 개개인에 대한 기준 날짜를 정의한다. 기준 날짜를 기준으로 기준 날짜 이전의 시간을 **기저 시간 baseline time**이라 정의한다. 기준 날짜를 포함한 그 이후의 시간은 **기준 후 시간 post-index time**이라고 부른다.

특성 분석은 질병의 자연 경과, 치료 이용, 진료 질 개선 등과 같은 것에 활용될 수 있다. 이번 장에

서는 특성 분석 방법에 관해 기술하며, ATLAS와 R을 이용한 고혈압 환자군에 대한 특성 분석을 해 볼 것이다.

11.1 데이터베이스 수준의 특성 분석 Database Level Characterization

관심 집단에 대한 특성 분석을 시행하기 전, 사용하고자 하는 데이터베이스의 특성을 이해하는 것이 선행되어야 한다. 데이터베이스 수준의 특성 분석 Database Level Characterization은 전체 데이터베이스에 대한 시간의 흐름에 따른 경향과 분포 측면에서 데이터 전체를 설명하기 위해 사용한다. 데이터베이스의 정량적 분석은 일반적으로 다음과 같은 질문을 포함한다:

- 이 데이터베이스의 총 사람 수는 몇인가?
- 환자의 연령 분포는 어떠한가?
- 환자의 관찰 기간은 얼마나 오래되었는가?
- 시간이 지남에 따라 기록/처방된 {치료, 질병, 처치 등}을 받은 사람의 비율은 어떠한가?

데이터베이스 수준의 기술 통계는 연구자가 어떠한 데이터에서 손실이 있을 수 있는지와 같이 확인할 수 없는 부분을 이해하는 데 도움을 주며 15장 데이터 품질을 설명할 때 자세히 다룬다.

11.2 코호트 특성 분석 Cohort Characterization

코호트 특성 분석 Cohort Characterization은 기준 날짜와 기준 날짜 이후 코호트 구성원의 특성을 기술한다. OHDSI는 상태 condition, 약물 drug, 치료 재료 device, 시술 procedure, 임상 관찰 clinical observation 등 개인의 의무기록에 존재하는 모든 것에 대한 기술 통계량을 바탕으로 특성 분석에 접근한다. 또한, 기준 날짜 시점에서 코호트 구성원에 대한 사회인구학적 내용에 대해 요약해준다. 이와 같은 접근 방식을 통해 관심 코호트에 대한 완벽한 요약을 제공한다. 특히, 이러한 접근을 통해 데이터의 변화에 대한 안목을 가지고 코호트에 대한 전체적인 탐색을 가능하게 하는 한편, 잠재적인 결측값을 찾을 수 있도록 한다.

코호트 특성 분석 방법은 이미 치료를 받은 사람에게서 치료의 적응증 유발률과 금기를 추정하는 개인 수준의 개인 수준의 약물 사용 연구 person-level drug utilization studies(DUS)에 이용될 수 있다. 코호트 특성 분석의 보급은 Strengthening the Reporting of Observational Studies in Epidemiology(STROBE) 가이드라인에서 자세히 제시하고 있는 관찰 연구에 권장되는 모범사례이다. (von Elm et al., 2008)

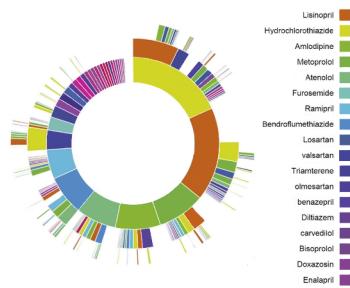


Figure 11.1: 고혈압 환자의 OHDSI 치료 경로 "sunburst" 시각화 사례

11.3 치료 경로 Treatment Pathways

인구 집단의 특성을 분석하는 또 하나의 방법은 기준 날짜 이후 post-index 시간 동안의 치료 순서를 기술하는 것이다. 예를 들어, 이전 연구 (Hripcsak et al., 2016)에서 OHDSI의 공통 데이터 모델(CDM)을 활용해 제2형 당뇨, 고혈압, 우울증에 대한 치료 경로의 특성을 분석하기 위한 기술 통계를 고안하였다. 이러한 분석 방법을 표준화함으로써, Hripcsak과 그 연구팀은 관심 집단의 특성 분석을 OHDSI 네트워크상에서 같은 통계 방법으로 실행하였다.

치료 경로 Treatment Pathways 분석은 가장 처음 처방/조제된 약물 이후 특정 상태에 있는 환자에 대한 치료행위 events를 요약하기 위해 시행한다. 이 연구에서는 제2형 당뇨, 고혈압, 우울증의 진단 이후의 치료행위를 기술하였다. 개개인이 받은 치료행위는 이후 요약 통계량 모음으로 합쳐지고 각각의 질병과 각각의 데이터베이스별로 시각화하였다.

그림 11.1에서 제시된 사례는 고혈압 치료를 시작한 환자집단을 나타낸다. 가운데 위치한 첫 번째 고리는 1차 요법 first-line therapy에 대한 환자의 비율을 나타낸다. 이 사례의 경우 Hydrochlorothiazide는 고혈압 환자군의 1차 요법으로 가장 흔하게 사용되는 약물이라는 것을 알 수 있다. Hydrochlorothiazide에서 파생된 상자는 해당 코호트 대상자에서 기록된 두 번째(2nd), 세 번째(3rd) 요법을 의미한다.

치료 경로 분석은 인구집단 내의 치료 이용 현황에 대한 중요한 근거를 제공한다. 이 분석을 통해 가장 빈번히 사용되는 1차 요법을 기술할 수 있고, 치료가 중단/변경/확대된 사람의 비율을 알 수 있다. 경로 분석을 통해 metformin이 당뇨에서 가장 일반적으로 처방된 1차 치료제임을 밝혔고, 이를 통해 미국 내분비학회의 당뇨 치료 알고리즘의 일차 요법이 일반적으로 잘 적용되고 있음을 확인했다. 더불어 당뇨 환자의 10%, 고혈압 환자의 24%, 그리고 우울증 환자의 11%는 다른 데이터베이스와 비교했을 때 자신과 같은 치료경로를 가진 사람이 단 한 명도 없는 고유한 치료 경로를 따르고 있는 것으로 나타났다.

고전적인 약물 사용 연구(DUS) 용어에서, 치료 경로 분석은 특정 집단에서 하나 이상의 약물 사용

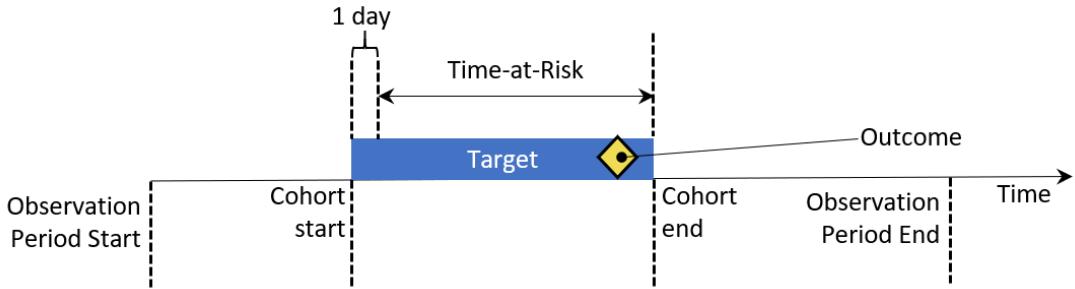


Figure 11.2: 개인 수준에서의 발생률 계산의 구성 성분. 본 예시에서 위험 노출 기간은 코호트가 시작하고 하루 이후부터 코호트가 끝나는 시점으로 정의됨.

률과 같은 집단 수준의 약물 사용 연구 population-level DUS 뿐 아니라 치료 방법의 지속률, 서로 다른 치료 간의 전환율과 같은 개인 수준의 약물 사용 연구 person-level DUS가 포함된다.

11.4 발생 Incidence

발생률 incidence rate과 발생비 incidence proportion은 공중 보건에서 위험 노출 기간 time-at-risk(TAR) 동안 인구집단 내 새로운 질병 outcome의 발생 평가에 사용되는 통계적 지표이다. 그림 11.2는 한 사람에게서 발생률 계산에 필요한 구성요소를 보여주고 있다:

그림 11.2에서 한 사람은 데이터 내에서 관찰이 시작되고 observation period start 끝나는 시점 observation period end이 표시된 기간을 갖는다. 그다음, 그 사람은 몇몇 연구 기준 eligibility criteria에 의해 코호트에 들어가는 시점 cohort start과 나오는 시점 cohort end을 갖게 된다. 위험 노출 기간은 우리가 질병 outcome의 발생을 보고자 하는 기간을 의미한다. 만일 질병이 위험 노출 기간 내에 발생한다면, 그 질병 한 건이 발생한 것으로 계산한다.

발생 incidence를 계산하기 위해서는 두 가지 측정법이 있다:

$$\text{Incidence Proportion} = \frac{\# \text{ persons in cohort with new outcome during TAR}}{\# \text{ persons in cohort with TAR}}$$

발생 분율 incidence proportion은 위험 노출 기간 동안 집단 내 새로운 outcome의 발생을 측정하는 방법이다. 다시 말해, 관심 집단에서 정해진 시간의 틀 안에서 발생한 outcome의 비율이다. (역자 주: incidence proportion = cumulative incidence (누적 발생률): 코호트에서 관찰 기간 내 새롭게 발생하는 환자 수 / 관찰 시작 시점의 코호트 인구수)

$$\text{Incidence Rate} = \frac{\# \text{ persons in cohort with new outcome during TAR}}{\text{person time at risk contributed by persons in cohort}}$$

발생률 incidence rate은 인구집단에서 누적되는 위험 노출 기간의 새로 발생한 outcome의 횟수를 측정한 것이다. 만일 한 환자가 위험 노출 기간 내에서 outcome을 경험했을 때, 그 환자가 전체의 인-시 person-time에 outcome이 발생하기까지의 시간만큼 기여했다고 산정한다. 누적된 위험 노출 기간은 **인-시**라고 하고, 단위는 일(인-일), 월(인-월), 혹은 연(인-년)으로 표현된다. (역자 주: incidence Rate = cumulative incidence (누적 발생률): 코호트에서 관찰 기간 내 새롭게 발생하는 환자 수 / 코호트 내 총 위험 노출 인-시)

치료 요법에 대하여 계산할 때, 정해진 치료 요법의 사용 비율 혹은 발생률을 계산하는 것은 전형적인 인구 집단 수준의 약물 사용 연구 (population-level DUS)라 할 수 있다.

11.5 고혈압 환자의 임상적 특성 분석

세계보건기구(WHO)의 고혈압에 대한 보고서 (Who, 2013) 에 따르면, 고혈압의 초기 발견과 적절한 치료 및 양호한 혈압 조절은 건강과 경제적인 면에서 상당한 이득이 있다. WHO의 보고서에는 고혈압의 개요 및 여러 국가의 질병 부담의 특성을 보여주고 있다. WHO는 지역과 사회 경제적 계급 및 성별 고혈압에 대한 기술통계를 제공하고 있다.

WHO가 수행한 고혈압 환자의 특성 분석과 동일한 분석을 관찰형 데이터를 이용해 수행 할 수 있다. 이 장의 다음 절에서는 고혈압 환자 집단의 구성을 이해하기 위한 데이터베이스 탐색을 ATLAS와 R을 이용해 시행하는 방법을 살펴볼 것이다. 또한, 같은 도구를 사용해 고혈압 환자의 자연 경과와 치료 패턴을 기술하려고 한다.

11.6 ATLAS를 활용한 데이터베이스의 특성 분석

여기서는 ATLAS에 탑재된 데이터 탐색 모듈을 사용하여 ACHILLES로 생성된 데이터베이스의 통계치를 살펴보고, 고혈압 환자와 관련된 데이터베이스 수준의 특성을 찾아내는 방법을 설명하고자 한다. 시작하기 위해 ATLAS의 왼쪽 바에 있는  Data Sources을 클릭하자. ATLAS의 첫 번째 드롭다운 목록에서 데이터 탐색 database to explore을 선택한다. 그리고, 데이터베이스 아래의 드롭다운 목록을 통해 보고서 탐색을 시작할 수 있다. 고혈압 환자에 대한 데이터베이스 수준의 특성 분석을 위해 두 번째 드롭다운 목록인 report 드롭다운 목록에서 Condition Occurrence를 선택하면 해당 데이터베이스의 모든 질병에 대한 트리 맵 시각화 결과가 표시된다:

특정 관심 질환을 검색하기 위해 Table 탭을 클릭하면 환자 수, 유병률, 환자별 기록 건수를 포함하

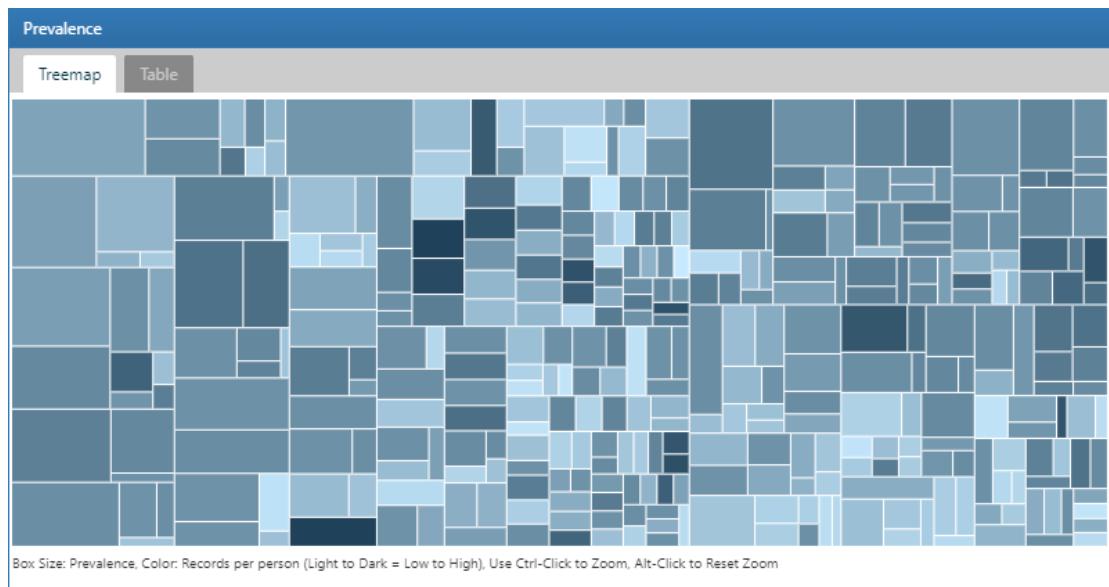


Figure 11.3: Atlas 데이터 원천: Condition Occurrence Treemap

는 데이터베이스의 전체 condition 목록이 나타난다. 상단의 filter 상자를 이용해 “hypertension”을 포함하는 concept name만을 걸러낼 수 있다:

하나의 행을 클릭하면 해당 condition에 대한 자세한 드릴다운 보고서를 확인할 수 있다. 이 경우 “essential hypertension”을 선택한 결과이며, 선택된 condition의 시간에 따른 경향과 성별, 월별 유병률, 기록 유형 (주상병 혹은 부상병 등) 그리고 최초 진단 시 나이의 경향을 알 수 있다:

지금까지 고혈압이라는 concept에 대한 데이터의 특성과 시간에 따른 경향을 살펴보았다. 데이터 베이스 수준의 특성 분석을 통해 고혈압 환자의 치료에 사용된 약물에 관해서도 확인할 수 있다. 이는 RxNorm 성분명에 요약된 약물 특성 검토를 위해 Drug Era report를 사용한 것 외에는 위와 동일하게 진행하면 된다. 관심 있는 항목에 대한 데이터베이스 특성 탐색을 마쳤다면, 이제는 특성화하고자 하는 고혈압 환자의 특성 분석을 위한 코호트를 설계하는 단계로 나아갈 준비를 마친 것이다.

11.7 ATLAS를 이용한 코호트 특성 분석

이 장에서는 ATLAS를 이용한 대규모의 코호트 특성 분석 방법 cohort characterization에 관해 설명할 것이다. ATLAS의 왼쪽 바에 있는 **Characterizations**를 클릭하면, 새로운 특성 분석을 만들 수 있다. 분석을 명명하고 버튼을 눌러 저장한다.

Prevalence			
Treemap		Table	
		Column visibility	Copy
		CSV	Show 15 ▾ entries
		Filter: hypertension	
Showing 1 to 15 of 47 entries (filtered from 15,907 total entries)		Previous 1 2 3 4 Next	
Concept	Name	Person Count	Records per person
Id			
320128	Essential hypertension	17,814,076	12.30% 5.80
312648	Benign essential hypertension	11,014,877	7.61% 4.35
317898	Malignant essential hypertension	1,021,441	0.70% 2.22
381290	Ocular hypertension	521,264	0.36% 2.40
441922	Transient hypertension of pregnancy	209,317	0.14% 2.45
44782429	Chronic kidney disease due to hypertension	170,534	0.12% 3.60
137940	Transient hypertension of pregnancy - delivered	153,806	0.11% 1.07
321080	Hypertension complicating pregnancy, childbirth and the puerperium	148,728	0.10% 2.15
314423	Benign essential hypertension complicating pregnancy, childbirth and the puerperium - not delivered	132,245	0.09% 3.94
44782690	Chronic kidney disease stage 5 due to hypertension	119,375	0.08% 5.20
44783618	Heritable pulmonary arterial hypertension	104,737	0.07% 3.61
319826	Secondary hypertension	96,356	0.07% 2.14
4167493	Pregnancy-induced hypertension	91,675	0.06% 2.60
321074	Pre-existing hypertension complicating pregnancy, childbirth and puerperium	74,311	0.05% 2.99
192680	Portal hypertension	71,240	0.05% 3.11
Showing 1 to 15 of 47 entries (filtered from 15,907 total entries)		Previous 1 2 3 4 Next	

Figure 11.4: Atlas 데이터 원천: concept name에서 "고혈압 hypertension"이라는 단어를 이용해 걸러낸 결과

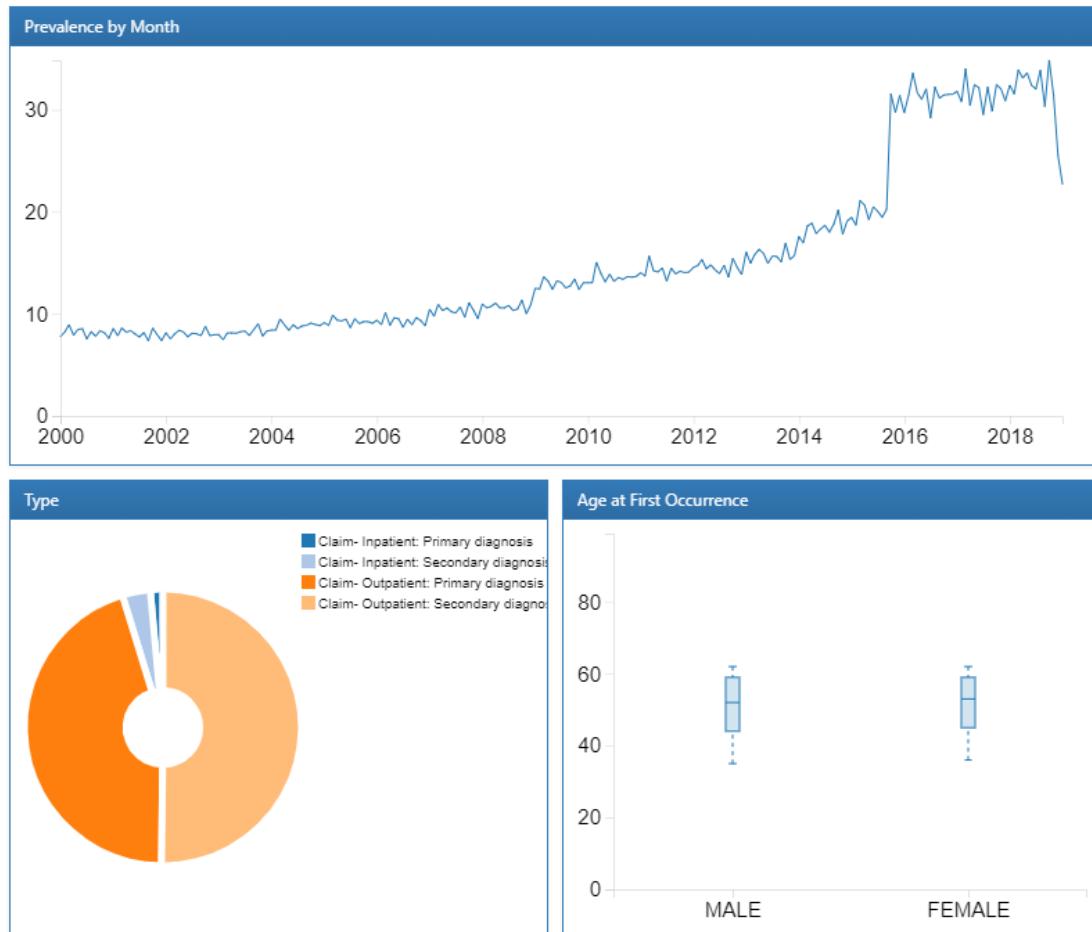


Figure 11.5: Atlas 데이터 원천: Essential hypertension 드릴다운 보고서

Cohort characterization is defined as the process of generating cohort level descriptive summary statistics from person level covariate data. Summary statistics of these person level covariates may be count, mean, sd, var, min, max, median, range, and quantiles. In addition, covariates during a period may be stratified into temporal units of time for time-series analysis such as fixed intervals of time relative to cohort_start_date (e.g. every 7 days, every 30 days etc.), or in absolute calendar intervals such as calendar-week, calendar-month, calendar-quarter, calendar-year.

Cohort definitions

Import

Show 10 entries Search:

ID	Name	Edit cohort	Remove
10447	Patients initiating first-line therapy for hypertension with >1 yr follow-up	Edit cohort	Remove
10448	Patients initiating first-line therapy for hypertension with >3 yr follow-up	Edit cohort	Remove

Showing 1 to 2 of 2 entries Previous 1 Next

Figure 11.6: 특성 디자인 탭 - 코호트 정의 선택

11.7.1 설계

특성 분석은 하나 이상의 코호트와 분석하고자 하는 하나 이상의 속성이 필요하다. 예시의 경우, 두 개의 코호트를 사용할 것이다. 첫 번째 코호트는 고혈압 치료를 시작한 환자로 기준 날짜 index date 이전 1년간 한 번이라도 고혈압 진단을 받은 환자로 정의하자. 또한 이 코호트에 속한 사람은 고혈압 치료제를 복용하기 시작한 후 최소 1년의 관찰 기간을 갖는다고 정의하자 (부록 B.6). 두 번째 코호트는 첫 번째 코호트와 모든 조건이 동일하지만, 최소관찰 기간을 1년 대신 3년을 갖는다고 한다 (부록 B.7).

코호트 정의

해당 코호트는 10장에서 이미 만들어져 있다고 가정한다. Import 을 클릭하고 그림 11.6에서 보이는 바와 같이 코호트를 선택한다. 그다음, 두 코호트 특성 분석을 위해 사용할 속성을 설정한다.

속성 선택

ATLAS에서 OMOP CDM에서 모델링 된 임상 도메인에서 특성 분석을 수행하기 위해 미리 만들어 둔 100개 이상의 특성 분석 도구 feature analysis (역자 주: CHADS2 score, Charlson comorbidity index 등 복잡한 쿼리나 계산을 해야만 구할 수 있는 각종 계산 값)가 정의되어 있다. 각각의 미리 정의해 둔 특성 분석 도구는 선택된 분석 대상 코호트에 대한 임상 관찰을 집계하고 요약하는 기능을 수행한다. 이와 같은 계산은 코호트의 index date 그리고 index date 후 특성을 설명하기 위해 수천

Feature analyses

Import

Show 25 ▾ entries Search:

ID	Name	Description	Actions
43	Drug Era Short Term	One covariate per drug in the drug_era table overlapping with any part of the short window.	Remove
49	Charlson Index	The Charlson comorbidity index (Romano adaptation) using all conditions prior to the window end.	Remove
67	Condition Occurrence Long Term	One covariate per condition in the condition_occurrence table starting in the long term window.	Remove
71	Demographics Age Group	Age of the subject on the index date (in 5 year age groups)	Remove
72	Demographics Race	Race of the subject.	Remove
73	Demographics Prior Observation Time	Number of continuous days of observation time preceding the index date.	Remove
74	Demographics Gender	Gender of the subject.	Remove
76	Condition Occurrence Medium Term	One covariate per condition in the condition_occurrence table starting in the medium term window.	Remove
77	Demographics Age	Age of the subject on the index date (in years).	Remove
79	Demographics Time In Cohort	Number of days of observation time during cohort period.	Remove
80	Demographics Index Year	Year of the index date.	Remove
81	Demographics Post Observation Time	Number of continuous days of observation time following the index date.	Remove
87	Procedure Occurrence Any Time Prior	One covariate per procedure in the procedure_occurrence table any time prior to index.	Remove
103	Visit Count Long Term	The number of visits observed in the long term window.	Remove

Figure 11.7: 특성 디자인 탭 - 속성 선택.

가지의 속성을 제공한다. ATLAS는 OHDSI에서 제공하는 FeatureExtraction R package를 이용해 개별 코호트에 대한 특성 분석을 시행하며, 이 FeatureExtraction이라는 R package를 사용하는 방법에 대하여 다음 절에서 더 자세하게 다룰 것이다.

분석하고자 하는 속성을 선택하기 위해 **Import** 를 클릭한다. 아래에는 코호트의 특성 분석을 위해 사용할 속성의 리스트가 있다:

위 그림은 각의 코호트에서 분석할 속성에 어떤 것이 있는지를 설명과 함께 목록을 나타낸 것이다. “Demographics”라고 시작하는 속성은 각 환자의 인구 통계학적인 정보를 코호트의 시작 시점에서 계산한다. 도메인 이름으로 시작한 속성 (예를 들어, Visit, Procedure, Condition, Drug 등)은 해당 도메인에 기록된 모든 관찰 값의 특성을 나타낸 것이다. 각각의 도메인의 특성은 코호트 시작 지점에 앞서 네 가지의 시간대의 옵션이 있다:

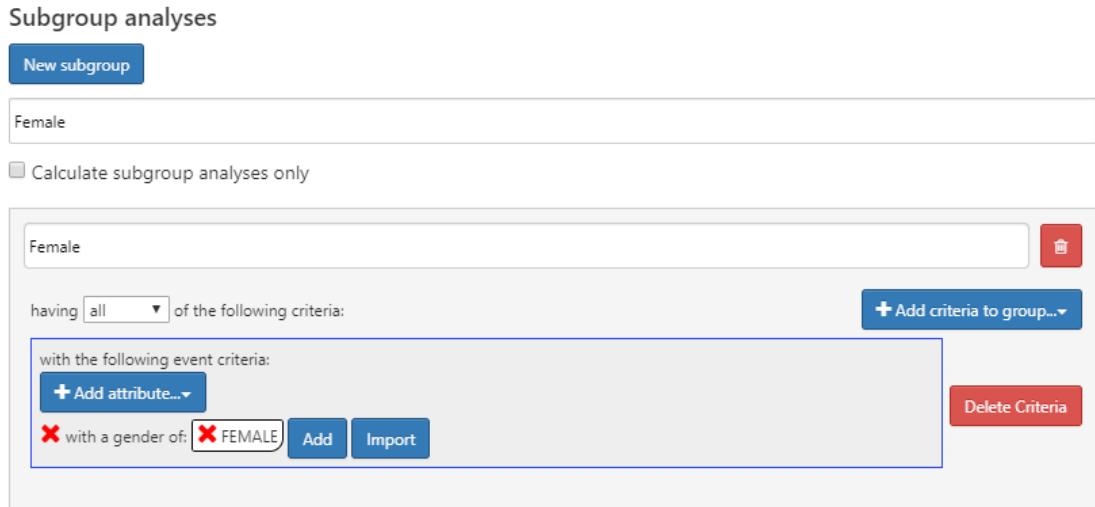


Figure 11.8: 여성에 대한 하위 그룹 분석을 위한 특성 분석 디자인.

- **Any time prior:** 환자의 관찰 기간 내에서 코호트 시작 지점 전의 모든 시간대를 사용
- **Long term:** 코호트 시작 지점을 포함하여 365일 이전까지
- **Medium term:** 코호트 시작 지점을 포함하여 180일 이전까지
- **short term:** 코호트 시작 지점을 포함하여 30일 이전까지

하위 집단 분석

만일 성별에 따라 특성에 차이가 있는지가 알고 싶다면 어떻게 해야 할까? 이때 우리는 “하위 집단 분석 subgroup analyses”을 이용할 수 있다. 이는 특성 분석 안에서 새로운 관심 하위 집단에 대한 정의를 할 수 있도록 해준다. 하위 집단을 만들기 위해 하위 그룹에 대한 기준을 클릭해 더하면 된다. 이 단계는 코호트 정의에 사용되는 기준과 유사하다. 이 사례에서 우리는 코호트 내의 여성을 확인할 수 있는 기준 모음을 정의할 것이다:



ATLAS의 하위 그룹 분석은 충화 strata와 같지 않다. 충화는 상호배제하는 반면, 하위그룹은 선택된 기준 criteria에 따라 동일한 사람이 하위그룹에 포함될 수 있다.

11.7.2 실행

특성 분석의 설계가 끝났다면, 우리의 환경 내에서 사용할 수 있는 하나 이상의 데이터베이스에 대하여 설계한 특성 분석을 시행할 수 있다. Execution 탭으로 이동하여 Generate 버튼을 클릭하면 선택된 데이터베이스에서 분석이 시작된다.

The screenshot shows the 'Executions' tab selected in a software interface. There are two main sections for 'SYNPUF 1K' and 'SYNPUF 5%'. Each section has a 'Generate' button, a 'View latest result' button, and a 'All executions (3)' button.

Figure 11.9: 설계한 특성 분석의 실행 - CDM source 선택.

CONDITION / Condition Occurrence Long Term / stratified by Female

		Concept ID	Patients initiating first-line therapy for hypertension with >1 yr follow-up				Patients initiating first-line therapy for hypertension with >3 yr follow-up				Std diff ▾		
Covariate	Explore		Count	Pct	Female		Count	Pct	Female				
					Count	Pct			Count	Pct			
Tachycardia	Explore ▾	444070	17,322	1.04%	9,042	1.18%	6,547	0.78%	3,530	0.90%	-0.0193		
Cardiomegaly	Explore ▾	314658	20,958	1.26%	8,007	1.04%	9,016	1.08%	3,465	0.89%	-0.0121		
Cardiac arrhythmia	Explore ▾	44784217	30,474	1.83%	13,221	1.72%	14,540	1.74%	6,318	1.62%	-0.0052		

Showing 1 to 3 of 3 entries (filtered from 206 total entries)

Previous 1 Next

Figure 11.10: 특성 분석 결과 - condition occurrence long term

분석이 완료되면, “All Executions” 버튼을 클릭하고 실행 목록에서 “View Reports”를 선택하면 보고서를 볼 수 있다. 이와 별도로 “view latest result”를 클릭하면 가장 최근에 시행된 분석의 결과를 확인할 수 있다.

11.7.3 결과

설계 시 선택한 각 코호트의 여러 속성은 표 형식으로 나타난다. 그림 11.10에서 보이는 것처럼 코호트 시작일로부터 365일 이전에 두 코호트에 존재하는 모든 질환을 요약하여 확인할 수 있다. 각 변수는 개별 코호트와 코호트 내에서 정의한 여성 하위그룹에 대한 수 count와 백분율 percentage 을 보여준다.

어떤 심혈관 질환이 모집단에서 관찰되는지 이해하고자 심 부정맥 cardiac arrhythmia 과거력을 갖는 환자의 비율이 얼마인지를 알아보기 위해 검색창에서 필터를 사용하였다. 심 부정맥 concept 옆의 Explore 링크를 이용하면 그림 11.11에 보이는 것과 같이 하나의 코호트에 대하여 더 자세한 내용이 들어 있는 새로운 창을 띠울 수 있다.

Exploring condition_occurrence during day -365 through 0 days relative to index: Cardiac arrhythmia						
Cohort: Patients initiating first-line therapy for hypertension with >1 yr follow-up						
Relationship type	Distance	Concept name	All stratas		Female	
			Count	Pct	Count	Pct
Explore Ancestor	4	Disorder by body site	32	0.00%	17	0.00%
Explore Ancestor	4	Finding of trunk structure	991	0.06%	605	0.08%
Explore Ancestor	3	Disorder of trunk	23	0.00%	14	0.00%
Explore Ancestor	3	Disorder of thorax	241	0.01%	104	0.01%
Explore Ancestor	3	Disorder of body system	4,135	0.25%	1,992	0.26%
Explore Ancestor	2	Disorder of cardiovascular system	12,979	0.78%	6,073	0.79%
Explore Ancestor	2	Disorder of mediastinum	138	0.01%	62	0.01%
Explore Ancestor	2	Disorder of body cavity	24	0.00%	10	0.00%
Explore Ancestor	1	Heart disease	4,691	0.28%	1,869	0.24%
Explore Selected	0	Cardiac arrhythmia	30,474	1.83%	13,221	1.72%

Showing 1 to 10 of 62 entries

Previous 1 2 3 4 5 6 7 Next

Figure 11.11: 특성 분석 결과 - 하나의 concept에 대하여 자세한 내용이 들어 있는 새로운 창.

CONDITION / Condition Occurrence Long Term / stratified by Female

			Patients initiating first-line therapy for hypertension with >1 yr follow-up				Patients initiating first-line therapy for hypertension with >3 yr follow-up				Std diff ▼	
Covariate	Explore	Concept ID	Female				Female					
			Count	Pct	Count	Pct	Count	Pct	Count	Pct		
Edema	Explore ▾	433595	32,243	1.94%	20,200	2.63%	15,173	1.81%	9,684	2.48%	-0.0066	

Showing 1 to 1 of 1 entries (filtered from 206 total entries)

Previous 1 Next

Figure 11.12: 특성 분석 결과 - 금기 상태 contraindicated condition를 탐색한 결과.

Exploring condition_occurrence during day -365 through 0 days relative to index: Edema									
Cohort: Patients initiating first-line therapy for hypertension with >1 yr follow-up									
			All stratas				Female		
Relationship type	Distance	Concept name	Count	Pct	Count	Pct	Count	Pct	Count
Explore Descendant	-2	Angioedema	2,605	0.16%	1,506	0.20%			

Showing 1 to 1 of 1 entries (filtered from 56 total entries)

Previous 1 Next

Figure 11.13: 특성 분석 결과 - 금기 상태에 대한 자세한 결과.

분석하고자 하는 코호트에 대하여 모든 condition에 대한 concept을 분석하고 나면, 탐색 옵션을 통해 선택된 모든 concept의 모든 상, 하위 관계에 있는 concept를 확인할 수 있다. 본 연구의 경우 심 부정맥에 관하여 확인했다. 이와 같은 탐색 기능은 concept의 계층 구조를 탐색 할 수 있도록 하며, 고혈압 환자에서 나타날 수 있는 다른 심장 질환을 확인할 수 있도록 한다. 요약된 결과처럼 개수와 백분율로써 화면에 출력된다.

같은 특성 분석은 혈관 부종과 같은 항고혈압제제의 부작용에 의한 상태를 찾는 분석에도 사용할 수 있다. 방법은 위의 분석과 같이 그림 11.12에 나온 것처럼 '부종 edema'를 검색하여 진행할 수 있다:

다시 한번, 고혈압 환자에서 혈관 부종의 발병률을 알아보기 위해 부종의 특성을 보는 탐색 기능을 사용할 것이다:

항고혈압제제를 시작하기 1년 전부터 혈관 부종의 기록이 있었던 사람의 비율을 확인할 수 있었다.

도메인 변수는 이분 변수 binary variable를 사용해 계산되는 반면 (즉, 과거력 유/무), 코호트 시작 시점의 연령과 같은 일부 변수는 연속적인 값 continuous variable을 갖는다. 위의 예시와 같이 두

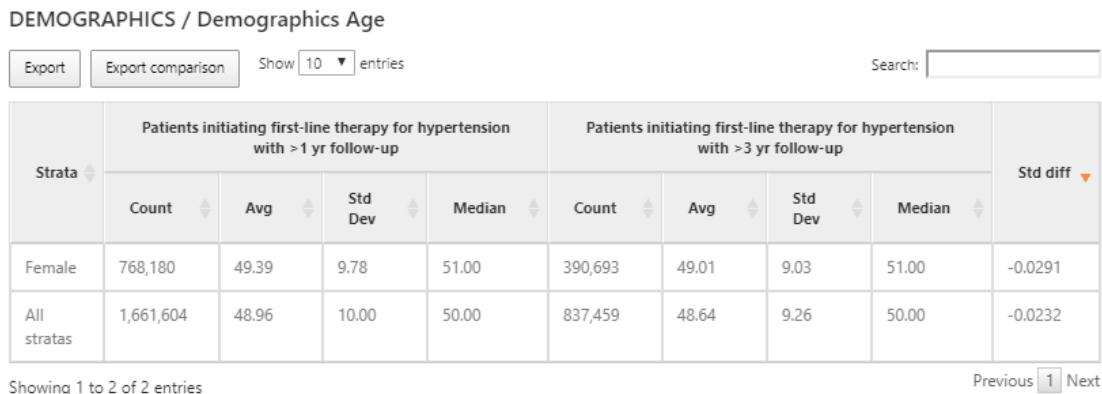


Figure 11.14: 각 코호트와 하위 집단의 연령 관련 특성 분석 결과.

코호트에서 연령에 대한 특성 분석 결과는 사람의 총 수, 연령의 평균, 연령의 중앙값 그리고 표준 편차를 통해 확인할 수 있었다.

11.7.4 사용자-정의 속성 Defining Custom Features

사전에 정의되어 제공되는 특성뿐 아니라 ATLAS에서는 사용자가 필요에 따라 특성을 주문 제작하고 정의할 수 있는 기능을 지원한다. 왼쪽 메뉴에서 **characterization**을 클릭하고, **Feature Analysis** 탭을 클릭한 후, **New Feature Analysis** 버튼을 클릭하면 사용자가 정의한 속성 분석을 할 수 있다. 사용자가 원하는 속성 분석을 정의하고, 버튼을 통해 저장할 수 있다.

예를 들어, 코호트 시작 이후에 ACE inhibitors의 drug era를 갖는 각각의 코호트에 속하는 사람 수를 알아본다고 하자:

위에서 정의한 기준은 코호트 시작 날짜에 적용된다고 가정하자. 이전에 정의한 기준을 저장했다면, 이전 절에서 디자인한 특성 분석에 이를 적용할 수 있다. 이는 위해 characterization을 열고, Feature Analysis로 이동해 보자. 버튼을 누르고 메뉴에서 new custom features를 선택하자. 그러면 분석할 특특성 목록에 사용자 정의 속성이 올라간 것이 보일 것이다. 앞 절에서 설명한 것과 같이 사용자 정의 속성에 대한 분석을 데이터베이스에 적용하여 시행할 수 있다:

11.8 R을 이용한 코호트 특성 분석

코호트의 특성 분석은 R을 통해서도 가능하다. R에서 고혈압 코호트에서 기저 특성(변수)을 생성하기 위해 OHDSI의 FeatureExtraction이라고 하는 R package를 사용하는 방법을 알아보기로 한다. FeatureExtraction은 세 가지 방법으로 변수를 구성할 수 있는 기능을 제공한다. 그 방법은 다음과 같다:

Design

Criteria Custom

Analysis type:

Prevalence

Add Criteria feature

Ace inhibitor exposure after index Delete

having all of the following criteria:

+ Add criteria to group... ▾

with at least 1 using all occurrences of:
a drug era of ACE inhibitors + Add attribute...

where event starts between 1 days After and All days After index start date add additional constraint Delete Criteria

allow events from outside observation period

Figure 11.15: ATLAS를 이용한 사용자가 정의하는 속성.

DRUG / Ace inhibitor exposure after index / stratified by Female

				Patients initiating first-line therapy for hypertension with >1 yr follow-up				Patients initiating first-line therapy for hypertension with >3 yr follow-up				Std diff
Covariate	Explore	Concept ID	Female				Female					
			Count	Pct	Count	Pct	Count	Pct	Count	Pct		
Ace inhibitor exposure after index	Explore ▾	0	686,034	41.29%	289,215	17.41%	426,280	50.90%	182,219	21.76%	0.1001	

Showing 1 to 1 of 1 entries

Previous 1 Next

Figure 11.16: 사용자 정의 속성을 적용하여 특성 분석을 진행한 결과 창

- 기본 설정된 변수 모음을 선택
- 사전 지정된 분석 모음 중에서 선택
- 사용자 정의 분석 모음 생성

FeatureExtraction은 개인 수준 특성 person-level feature과 통합된 특성의 두 가지 방법으로 변수를 만든다. 개인 수준의 특성은 기계 학습에 적용할 때 유용하다. 이 절에서는 관심 코호트를 설명하는 기저 변수를 생성할 때 유용한 통합된 특성의 사용 방법을 집중적으로 설명할 것이다. 더불어 사전 지정된 분석과 사용자가 정의하는 분석의 두 가지 변수를 구성 방법에 대해 알아볼 것이다. (기본 설정 모음에 대해서는 독자의 연습을 위해 남겨두도록 하겠다)

11.8.1 코호트 생성 Cohort Instantiation

특성 분석을 위해 우선 코호트를 예시를 들어 설명하겠다. 코호트 생성 예시는 10장에서 실습했었다. 본 실습에서는 고혈압의 1차 약물치료를 시작한 사람 중 1년간 관찰된 사람을 사용할 것이다 (부록 B.6). 부록 B의 다른 코호트는 독자의 연습을 위해 남겨두었다. cohort definition ID가 1인 scratch.my_cohorts라는 테이블에 실습에 사용할 코호트가 있다고 가정하자.

11.8.2 데이터 추출 Data Extraction

먼저 R이 서버에 접속할 수 있도록 해야 한다. FeatureExtraction은 DatabaseConnector package의 `createConnectionDetails` 함수를 이용해 서버와 연결할 수 있다. `?createConnectionDetails`를 입력하면 다양한 데이터베이스 관리 시스템(DBMS)이 요구하는 설정값이 어떤 것이 있는지 확인할 수 있다. 예를 들어, PostgreSQL 데이터베이스와 연결해야 하는 경우 다음과 같이 연결 설정을 해야 한다:

```
library(FeatureExtraction)

connDetails <- createConnectionDetails(dbms = "postgresql",
                                         server = "localhost/ohdsi",
                                         user = "joe",
                                         password = "supersecret")

cdmDbSchema <- "my_cdm_data"
cohortsDbSchema <- "scratch"
cohortsDbTable <- "my_cohorts"
cdmVersion <- "5"
```

마지막 네 줄의 코드는 `cdmDbSchema`, `cohortsDbSchema`, `cohortsDbTable` 변수를 정의하고 CDM

버전 또한 정의하기 위함이다. 이러한 정의는 나중에 R에게 CDM 형식의 데이터가 있는 위치, 관심 코호트가 만들어진 위치, 어떤 버전의 CDM이 사용되었는지를 확인할 수 있도록 해준다. Microsoft SQL Server에서 주의할 점은, 데이터베이스 스키마는 데이터베이스와 스키마 정보를 둘 다 필요로 한다는 것이다. 예를 들면 다음과 같다. `cdmDbSchema <- "my_cdm_data.dbo"`.

11.8.3 미리 만든 분석 정의 사용 Using Prespecified Analyses

`createCovariateSettings` 함수는 사전에 정의된 대규모 변수의 모음을 선택할 수 있도록 한다. `?createCovariateSettings`를 입력하면 사용 가능한 옵션을 확인할 수 있다. 예를 들어:

```
settings <- createCovariateSettings(
  useDemographicsGender = TRUE,
  useDemographicsAgeGroup = TRUE,
  useConditionOccurrenceAnyTimePrior = TRUE)
```

다음과 같이 입력하면 성별, 연령 (5년 단위의 연령 그룹), 그리고 코호트 시작 날짜를 포함한 이전의 모든 condition_occurrence 테이블에서 관찰된 각각의 concept에 대하여 이분 변수 binary variable를 생성할 수 있다.

많은 사전 정의 분석은 단기 short term, 중기 medium term, 혹은 장기 long term 구간 time window을 지정할 수 있다. 기본적으로 구간은 다음과 같이 정의되어 있다:

- **Long term** : 코호트 시작 날짜를 포함한 365일 이전까지의 구간.
- **Medium term** : 코호트 시작 날짜를 포함한 180일 이전까지의 구간.
- **short term** : 코호트 시작 날짜를 포함한 30일 이전까지의 구간.

그러나 아래의 예시와 같이 사용자가 시간 구간을 변경 할 수 있다:

```
settings <- createCovariateSettings(useConditionEraLongTerm = TRUE,
                                       useConditionEraShortTerm = TRUE,
                                       useDrugEraLongTerm = TRUE,
                                       useDrugEraShortTerm = TRUE,
                                       longTermStartDays = -180,
                                       shortTermStartDays = -14,
                                       endDays = -1)
```

이 코드에서 새로 정의된 장기 구간은 코호트 시작 날짜를 포함하지 않고 180일 이전까지의 구간을 나타내고, 단기 구간은 코호트 시작 날짜를 포함하지 않고 14일 이전까지의 구간을 나타낸다. 또한,

변수 구성에서 필수적으로 들어가야 할 것과 빼져야 할 concept ID를 지정할 수 있다:

```
settings <- createCovariateSettings(useConditionEraLongTerm = TRUE,
                                      useConditionEraShortTerm = TRUE,
                                      useDrugEraLongTerm = TRUE,
                                      useDrugEraShortTerm = TRUE,
                                      longTermStartDays = -180,
                                      shortTermStartDays = -14,
                                      endDays = -1,
                                      excludedCovariateConceptIds = 1124300,
                                      addDescendantsToExclude = TRUE,
                                      aggregated = TRUE)
```



`aggregated = TRUE`로 바꾸면 위에 표시된 모든 사례에 대하여 Feature Extraction으로 하여
금 모든 요약 통계치를 표시하도록 한다. 이 지표를 제외하면 코호트 내의 각의 사람에 대한
변수값이 계산될 것이다.

11.8.4 통합된 변수의 생성 Creating Aggregated Covariates

다음의 코드는 코호트에 대한 통합 공변량을 생성하도록 한다:

```
covariateSettings <- createDefaultCovariateSettings()

covariateData2 <- getDbCovariateData(
  connectionDetails = connectionDetails,
  cdmDatabaseSchema = cdmDatabaseSchema,
  cohortDatabaseSchema = resultsDatabaseSchema,
  cohortTable = "cohorts_of_interest",
  cohortId = 1,
  covariateSettings = covariateSettings,
  aggregated = TRUE)

summary(covariateData2)
```

그 결과값은 다음과 비슷하게 보일 것이다:

```
## CovariateData Object Summary
```



Figure 11.17: 한 환자의 경로 분석의 내용.

```
##  
## Number of Covariates: 41330  
## Number of Non-Zero Covariate Values: 41330
```

11.8.5 결과 형식 Output Format

통합된 공변량 데이터 covariateData에서 주요한 두 가지 구성요소는 이분 binary 혹은 연속적 continuous 변수에 대한 공변량 covariates과 공변량 양 covariatesContinuous이다:

```
covariateData2$covariates  
covariateData2$covariatesContinuous
```

11.8.6 사용자 정의 공변량 Custom Covariates

FeatureExtraction은 또한 공변량을 사용자가 정의하고 사용할 수 있도록 사용자 정의 공변량 기능을 제공한다. 이는 고급 주제로 다음 링크를 통해 사용자 문서에서 자세히 볼 수 있다: <http://ohdsi.github.io/FeatureExtraction/>.

11.9 ATLAS에서 코호트 경로 Cohort Pathways in ATLAS

경로 분석은 하나 이상의 관심 코호트에서 치료 과정의 순서를 이해하기 위해 시행한다. 분석 방법은 Hripcsak의 연구 (Hripcsak et al., 2016) 의 디자인을 기반으로 한다. 이 방법은 ATLAS의 Cohort Pathways 기능으로 일반화되고 체계화되었다.

Cohort pathways는 하나 이상의 분석하고자 하는 대상 코호트 target cohort의 코호트 시작 날짜 이후 발생한 사건을 요약하는 분석 기능 제공을 목표로 한다. 이를 위해 분석 대상 모음에서 관심 임상 사건을 식별하기 위한 사건 코호트 event cohort 모음을 정의하고 생성해야 한다.

대상 코호트에서 한 환자를 어떻게 찾는지를 보면 다음과 같다:

Cohort Pathway is defined as the process of generating an aggregated sequence of transitions between the Event Cohorts among those people in the Target Cohorts.

Target Cohorts

Each of the Target Cohorts will be analyzed for the pathways through the event cohorts.

Import

ID	Name	Edit cohort	Remove
10447	Patients initiating first-line therapy for hypertension with >1 yr follow-up	Edit cohort	Remove
10448	Patients initiating first-line therapy for hypertension with >3 yr follow-up	Edit cohort	Remove

Showing 1 to 2 of 2 entries Previous 1 Next

Figure 11.18: 선택된 대상 코호트를 이용한 경로 분석.

그림 11.17에서 대상 코호트에 속한 환자의 정의된 시작과 마지막 날짜를 볼 수 있다. 숫자가 매겨진 조각난 선은 해당 환자에게서 해당 구간 동안 사건 코호트가 발생한 위치를 표현한 것이다. 사건 코호트를 통해 CDM에서 표현되는 관심 있는 모든 임상적 사건을 설명 할 수 있기 때문에 단일 도메인 혹은 개별 concept에 대한 경로를 만드는 데 제한을 받지 않는다.

경로 분석을 시작하기 위해서는 ATLAS 왼쪽 바에 있는 Cohort Pathways를 눌러 새로운 코호트 경로 분석 연구를 생성한다. 분석을 명명하고 저장 버튼을 눌러 저장한다.

11.9.1 디자인

사용할 코호트는 고혈압 1차 치료를 시작하고 환자 1년간 관찰된 환자 코호트와 3년간 추적 관찰된 환자 코호트로 이전 분석에서 사용된 코호트를 지속해서 사용할 것이다 (부록 B.6과 B.7). import 버튼을 사용해 두 개의 코호트를 불러오자.

그다음, 각각 분석하고자 하는 일차 고혈압 치료 약제에 대한 코호트를 사건 코호트로 정의한다. 이를 위해 ACE inhibitor 사용에 대한 코호트를 생성하고 코호트 마지막 날짜를 약물의 마지막 노출이 끝나는 날짜로 정의한다. 여덟 개의 다른 고혈압 약제에 대하여 같은 방식으로 코호트를 생성하고, 코호트 생성에 필요한 여러 정의는 부록 B.8-B.16에서 찾아볼 수 있다. 완료한 후 **Import** 버튼을 눌러 경로 분석 디자인의 Event Cohort 부분에 삽입한다:

모든 과정을 완료하고 나면 다음과 같이 된다. 그 이후에 몇 가지 추가 설정을 해야 한다:

- **Combination window:** 이 세팅을 사용하면 사건이 겹쳐질 경우 사건을 조합 combination

Event Cohorts

Each Event Cohort defines the step in a pathway that may occur for a person in the Target Cohort.

Import			
Show	10 ▾ entries	Search:	
ID	Name		
9174	ACE inhibitor use	Edit cohort	Remove
9175	Angiotensin receptor blocker (ARB) use	Edit cohort	Remove
9176	Thiazide or thiazide-like diuretic use	Edit cohort	Remove
9177	dihydropyridine Calcium Channel Blocker (dCCB) use	Edit cohort	Remove
9178	non-dihydropyridine Calcium Channel Blocker (ndCCB) use	Edit cohort	Remove
9179	beta blocker use	Edit cohort	Remove
9180	Diuretic-loop use	Edit cohort	Remove
9181	Diuretic-potassium sparing use	Edit cohort	Remove
9182	alpha-1 blocker use	Edit cohort	Remove

Showing 1 to 9 of 9 entries

Previous [1](#) Next

Figure 11.19: 고혈압 일차 치료 시작의 경로 분석을 위한 사건 코호트.

으로 간주할 수 있도록 하는 시간 구간 window of time 설정할 수 있도록 해주며, 일 days 단위로 정의할 수 있도록 한다. 예를 들어, 만약 두 개의 사건 코호트 (사건 코호트 1과 사건 코호트 2)로 정의한 두 개의 약물이 그 조합 구간 combination window안에 노출 시간이 겹치면 경로 알고리즘은 두 사건 코호트를 합쳐서 “사건 코호트 1 + 사건 코호트 2”로 만든다.

- **Minimum cell count:** 사건 코호트가 이 수보다 적으면 개인정보 보호를 위해 결과에서 제외된다.
- **Max path length:** 최대의 연속 사건 sequential event의 수 (pathway 몇 단계까지 볼 것인지 설정)

11.9.2 실행 Executions

경로 분석의 디자인이 완료된 후, 하나 이상의 데이터베이스 내에서 분석 실행 execution할 수 있다. 방법은 ATLAS의 코호트 특성 분석에서 진행한 것과 같은 방식으로 진행된다. 실행이 끝나면 분석 결과를 확인할 수 있다.

11.9.3 결과 시각화

경로 분석에 관한 결과는 3개의 부분으로 나누어진다. 범례 legend 부분은 대상 코호트의 총 환자 수를 나타내는데, 이는 하나 이상의 사건이 있는 환자 수를 나타낸다. 아래의 요약은 sunburst plot의 가운데 부분을 각 코호트에 색을 지정해 표현하였다.

sunburst plot은 시간에 따라 한 사람이 겪는 다양한 사건 경로를 시각화한다. 그래프의 가운데 부분은 코호트의 시작 부분을 나타내고 첫 번째 색으로 구분된 고리는 전체 코호트에서 각 사건 코호트의 환자 비율을 보여준다. 본 예시에서 가장 가운데의 원은 고혈압 환자가 처음으로 시작한 일차 약제를 의미한다. 그리고 sunburst plot의 첫 번째 고리는 사건 코호트에서 정의한 것과 같이 시작한 일차 약제의 유형별 비율을 나타낸다 (예를 들어, ACE inhibitor, Angiotensin receptor blocker 등).

두 번째 모음의 고리는 환자의 두 번째 사건 코호트를 의미한다. 특정 사건의 과정에서, 어떤 환자는 두 번째 사건 코호트를 가지지 않을 수 있는데, 그럴 경우 그 비중은 고리에서 (연한) 회색으로 표현되어 있다 (역자 주: 즉 첫 번째 약만 복용한 경우).

sunburst plot의 세션을 클릭하면 오른편에 세부 경로가 나타난다.

이를 통해 대상 코호트에서 가장 많은 비율을 차지하는 환자가 일차 약제로 ACE inhibitor를 시작한 환자임을 알 수 있고, 가장 작은 비율을 차지하는 약제는 thiazide나 thiazide diuretics임을 알 수 있었다.

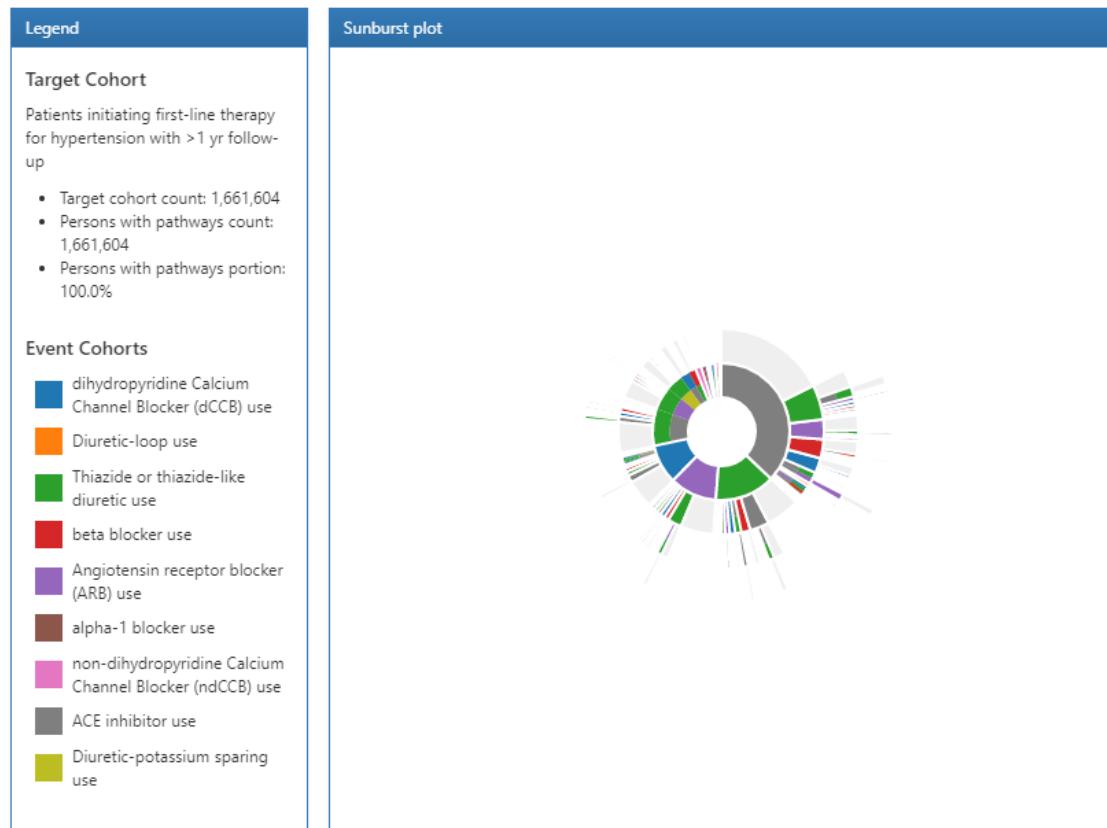


Figure 11.20: 경로 분석 결과의 범례와 sunburst 그래프를 통한 시각화.

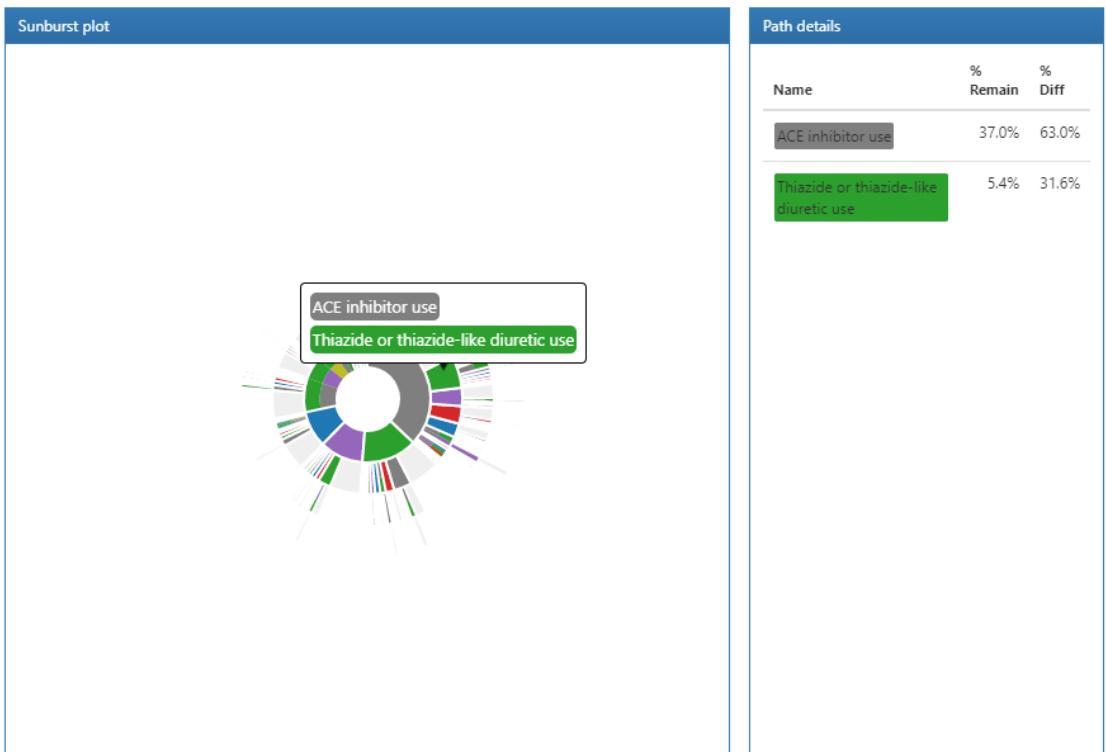


Figure 11.21: 더 자세한 경로 분석의 결과 창.

11.10 ATLAS를 이용한 발생률 분석

발생률 계산 시 대상 코호트의 사람 중 위험 노출 기간 동안 결과 코호트를 경험한 환자에 대하여 설명하도록 한다. 예를 들어, ACE inhibitor(ACEi)를 시작한 사람과 Thiazides와 tiazide-like diuretics(THZ)를 시작한 사람 중 혈관 부종과 급성 심근경색 outcome 발생을 분석하는 발생률 분석을 디자인했다고 하자. 이를 위해 약물에 노출된 사람의 위험 노출 기간 동안 이 outcome을 평가해야 한다. 더불어, Angiotensin receptor blockers(ARBs)에 대한 약물 노출 결과를 추가하기 위해 대상 코호트 (ACEi와 THZ)에 속해 있는 동안 ARBs 약물 사용의 발생을 outcome으로 추가한다. 이 outcome은 대상 코호트에 속해있는 동안 얼마나 ARBs 사용이 발생했는지를 측정해 줄 수 있다.

발생률 분석을 시작하기 위해서는 ATLAS의 왼쪽 바에서 **Incidence Rates** 버튼을 누른다. 분석의 이름을 적고 **Save** 버튼을 눌러 저장한다.

Figure 11.22: 발생률 분석에 사용할 대상 코호트와 결과 코호트의 정의.

Time At Risk

Time at risk defines the time window relative to the cohort start or end date with an offset to consider the person 'at risk' of the outcome.

- Time at risk starts with plus days.
- Time at risk ends with plus days.

No study window defined.

Figure 11.23: 발생률 분석에 사용할 대상 코호트와 결과 코호트의 정의.

11.10.1 디자인

이미 이전 10장에서 ATLAS로 예제에 사용할 코호트를 만들었다고 가정해 보자. 부록에서 예제에 사용할 대상 코호트 (부록 B.2, B.5)의 모든 정의와 결과 (부록 B.4, B.3, B.9)를 확인할 수 있다.

definition 탭을 클릭해서 *New users of ACE inhibitors* 코호트와 *New users of Thiazide or Thiazide-like diuretics* 코호트를 선택한다. 선택한 코호트가 분석 디자인에 추가되었는지 확인하기 위해 대화 상자를 닫아야 한다. 그다음 대화 상자에서 클릭해 결과 코호트를 추가한다. *acute myocardial infarction events*, *angioedema events*와 *Angiotensin receptor blocker (ARB) use* 코호트 등이 결과 코호트로 선택되어야 한다. 분석 디자인에 결과 코호트가 추가되었는지 확인하기 위해 대화 상자를 닫아야 한다.

이후 분석에 필요한 위험 노출 기간을 정의해야 한다. 위에서 보이는 바와 같이 위험 노출 기간은 코호트 시작과 마지막 날짜에 기반하여 정해진다. 예제에서는 위험 노출 기간의 시작일을 대상 코호트의 코호트 시작일보다 1일 후로 정의했다. 위험 노출 기간의 마지막 날짜는 코호트 마지막 날짜로 정의했다. 이 경우, ACEi와 THZ 코호트는 코호트의 정의에 따라 약물 노출이 끝나면 코호트 또한 종료된다.

ATLAS는 또한 분석의 선택 사항의 일부분으로 대상 코호트의 층화 기능을 제공한다:

분석을 위해서 New Stratify Criteria 버튼을 누르고 11장에서 설명한 것과 같은 단계로 진행한다. 분석 디자인이 완성되면 하나 이상의 데이터베이스를 이용해 실행할 수 있다.

Stratify Criteria: You can provide optional stratification criteria to the analysis that will divide the population into unique groups based on their satisfied criteria.

Figure 11.24: 여성에 대한 충화를 정의한 발생률.

Figure 11.25: 발생률 분석의 실행.

11.10.2 실행

Generation 탭을 누르고 버튼을 클릭하면, 분석에 사용할 데이터베이스 목록을 볼 수 있다:

하나 이상의 데이터베이스를 선택하고 Generate 버튼을 누르면 주어진 분석 디자인의 대상과 결과의 모든 조합에 대하여 발생 분석이 실행된다.

11.10.3 결과 보기

Generation 탭의 상단의 화면은 결과를 확인할 때 대상과 결과를 선택할 수 있도록 해준다. 바로 아래에는 분석에 사용된 각 데이터베이스에 대한 요약과 발생에 관한 내용이 표시된다.

대상 코호트 중 ACEi 사용자와 outcome 코호트 중에서 Acute Myocardial Infarction(AMI)을 드롭다운 목록에서 선택해 보자. 그리고 버튼을 누르면 발생 분석의 결과가 나온다:

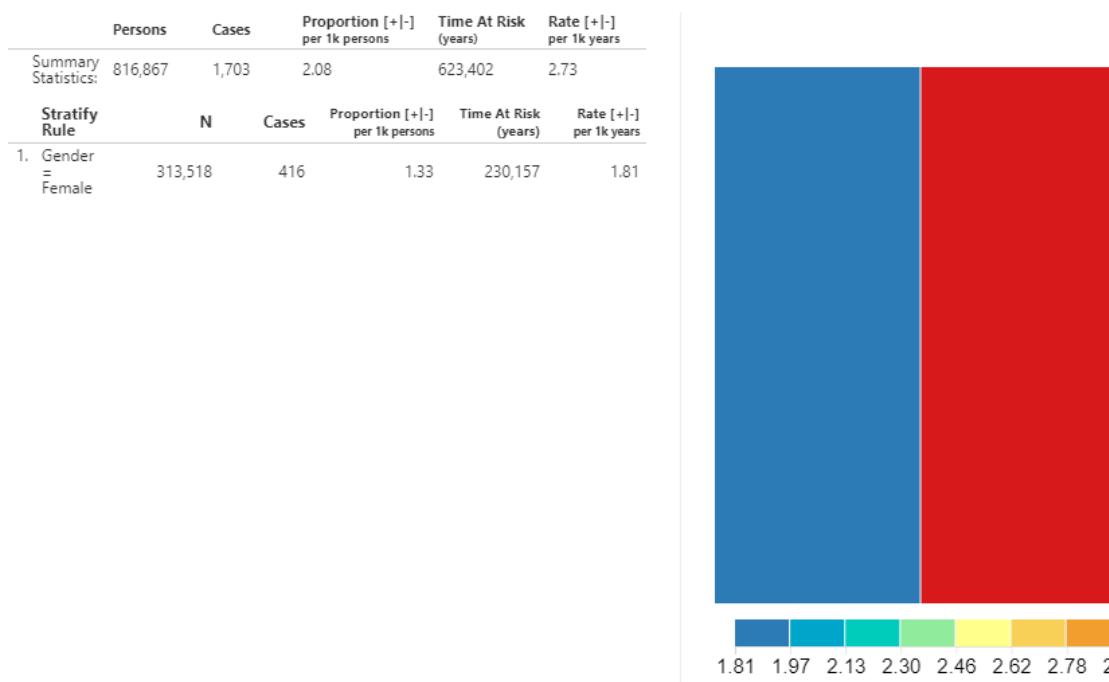


Figure 11.26: ACEi 의 새로운 사용자에서 AMI의 발생률을 분석한 결과.

결과를 통해 데이터베이스 내 해당 코호트에서 위험 노출 기간 동안 발생한 모든 중례에 해당하는 전체 환자 수를 요약해 보여준다. 발생 비율은 1,000명당 발생 건수로 표시한다. 위험 노출 기간은 대상 코호트에서 연 단위로 계산된다. 발생률은 1,000인-년당 발생 건수로 표현된다.

또한 분석 디자인에서 충화를 위해 정의 한 계층에 대한 발생률을 확인할 수 있다. 각 계층에 대해 위에 언급한 바와 동일한 방식으로 계산되었다. 더불어, 트리 맵 시각화 방법을 통해 각 계층의 비율을 상자 공간으로 표현했다. 아래 눈금에 표시된 대로 색상은 발생률을 나타낸다.

ACEi 환자군 내에서 ARBs를 새롭게 사용하기 시작한 신규 환자군의 발생을 확인하기 위한 정보도 얻을 수 있다. 상단의 드롭다운을 사용하여 ARBs의 사용으로 outcome을 바꾸고, 버튼을 누르면 자세한 내용이 나타난다.

위에 보이는 것처럼, 지금까지 사용한 방법과 동일한 방식으로 계산되었지만, 입력값이 (ARB 사용) 건강에 대한 outcome에서 약물 사용에 대한 평가로 바뀌었기 때문에 해석이 달라진다.

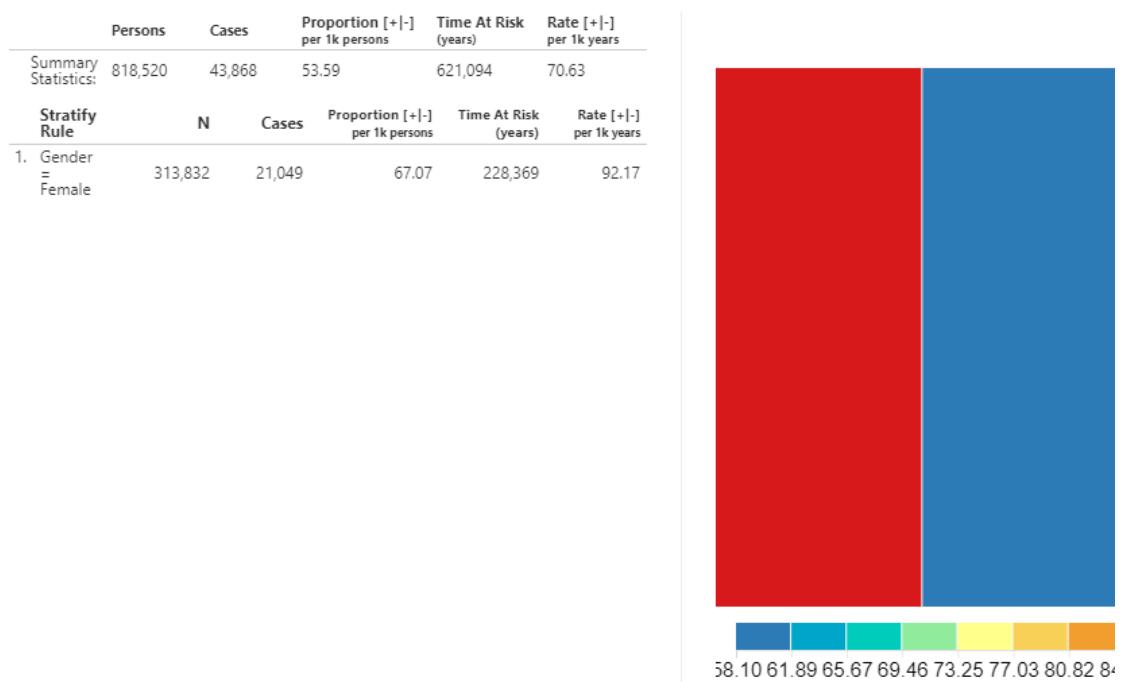


Figure 11.27: 발생률 - ACEi 노출 기간 중 ARBs 치료를 새로 시작한 ACEi 사용 환자군.

11.11 요약



- OHDSI는 모든 데이터베이스 혹은 관심 코호트에 대한 특성을 분석하는 도구를 제공한다.
- 코호트의 특성 분석은 index date 이전 (**baseline**)과 index date 후 (**post-index**) 시간 동안의 관심 코호트를 설명해 준다.
- ATLAS의 특성 분석 모듈과 OHDSI 연구방법론 라이브러리 Methods Library는 다양한 관찰 기간에 대한 기저 특성에 대해 계산 할 수 있는 기능을 제공한다.
- ATLAS의 경로 pathway와 발생률 incidence모듈은 index date 이후 시간 동안의 기술 통계를 제공한다.

11.12 예제

전제조건

본 예제를 위해 ATLAS에 접근할 수 있어야 한다. 다음의 ATLAS를 사용하거나 <http://atlas-demo.ohdsi.org>, 혹은 개별적으로 구축하여 접속 가능한 ATLAS가 있다면 사용해도 좋다.

Exercise 11.1. 실제 임상에서 celecoxib가 얼마나 사용되는지를 알고 싶다. 시작하기에 앞서 데이터베이스에 약물에 대한 데이터베이스의 데이터를 이해해야 한다. ATLAS의 Data Sources를 이용해 celecoxib에 대한 정보를 찾아보자.

Exercise 11.2. celecoxib 사용자의 질병 자연 경과에 대해 더 알고 싶다. celecoxib의 새로운 사용자에 대한 간단한 코호트를 만들어 보자. 이때, 365일의 휴약 washout 기간을 설정하자 (어떻게 해야 하는지 자세히 알고 싶다면 10장을 참고한다). 그리고 ATLAS에서 이 코호트의 characterization을 생성하고, 동반 상병 질환과 약물 노출을 찾아보자.

Exercise 11.3. celecoxib 치료를 받는 사람이 이후 기간에 상관없이 위장관 출혈 gastrointestinal(GI) bleeds가 얼마나 자주 발생하는지를 알고 싶다. 우선 GI bleed의 사건 코호트를 생성해야 한다. 해당 코호트의 정의를 위해 192671 (“Gastrointestinal hemorrhage”)으로 정의된 concept이나 그 하위 concept을 사용하자. 이전 예제에서 정의한 약물 노출 코호트를 이용해, celecoxib을 시작한 이후 GI bleed의 발생률을 계산하자.

답변은 부록 E.7에서 찾을 수 있다.

Chapter 12

인구 수준 추정

Chapter leads: Martijn Schuemie, David Madigan, Marc Suchard & Patrick Ryan

관찰형 보건의료 데이터 (예를 들어 보험청구자료, 전자 의무 기록) 는 환자의 삶을 의미 있게 향상 할 수 있는 치료 효과에 대한 실세계 증거를 생성할 기회를 제공한다. 이 장에서는 인구 수준 효과 추정 population-level effect estimation, 즉 특정 건강 결과에 대한 노출 (예를 들어, 약물 노출 또는 시술과 같은 의료개입)의 평균 인과적 영향 효과에 대한 추정에 초점을 맞춘다. 두 가지의 다른 추정 업무를 고려한다.

- **직접 효과 추정 direct effect estimation:** 위험인자 비노출에 비교하여 위험인자 노출의 질병 결과 발생 위험에 대한 영향 추정.
- **비교 효과 추정 comparative effect estimation:** 다른 노출 comparator exposure과 비교 하여 표적 노출 target exposure의 질병 발생 위험에 대한 영향 추정.

두 가지의 경우에서, 인구 수준의 효과는 사실적 효과와 대조된다. 다시 말하면, 반 사실적인 counterfactual 결과를 가진 노출된 환자에게 무슨 일이 일어났는가? 노출이 일어나지 않았다면 (직접적) 혹은 다른 노출이 일어났다면 (상대적) 무슨 일이 일어났을까? 어떤 환자라도 하나의 사실적인 결과 만 노출할 수 있기 때문에 (인과 추론의 근본적인 문제), 다양한 효과 추정 설계는 여러 분석 장치를 사용하여 반 사실적인 결과를 조명한다. (역자 주: 반사실 counterfactual이란 이론상의 가정으로서 A란 사람에게 B란 시점에 C란 약물을 투여하고 D란 질병 발생 유무를 측정한 후에, 타임머신을 타고 다시 시간을 거슬러 올라 B란 시점으로 돌아간 후에, 그 동일한 A에게 C를 투여하지 않고 관찰하여 D란 질병 발생 유무를 측정하는 것을 말한다. 이렇게 한다면 각종 비뚤림과 교란인자를 완전히 통제할 수 있다. 이론상으로만 가능하다)

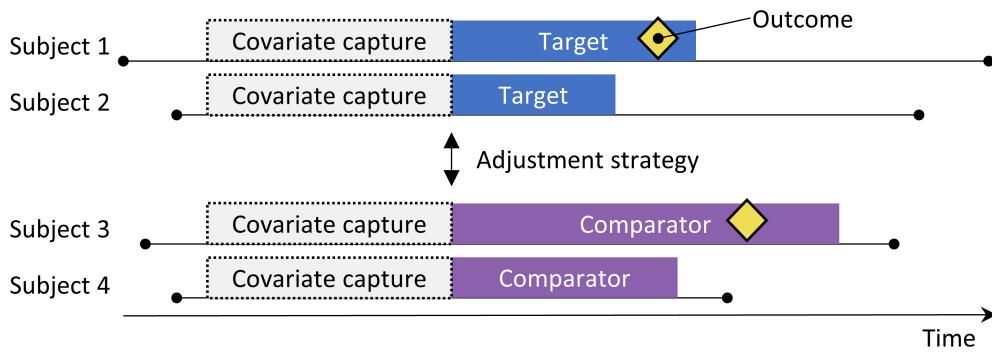


Figure 12.1: 새로운 사용자의 코호트 디자인 new-user cohort design: 표적 치료 target treatment를 시작하기 위해 관찰된 대상은 대조 대상 치료 comparator treatment를 시작한 대상과 비교된다. 두 치료군 간의 차이를 조정하기 위해 충화 stratification, 매칭 matching, 성향 점수에 의한 가중치 부여 weighting by propensity score, 결과 모델에 기저 특징 baseline characteristics 보정 추가와 같은 다양한 보정법 adjustment strategy을 사용할 수 있다. 성향 모델 propensity model 또는 결과 모델 outcome model에 포함된 특징은 치료 시작 전에 결정된다. (역자 주: new-user란 대상 위험에 생애 처음 노출된 환자를 말한다)

인구 수준 효과 추정의 사용 사례 use-cases는 치료 선택, 안전 감시 safety surveillance, 비교 연구 comparative effectiveness를 포함한다. 방법은 특정 가설을 한 번에 하나씩 테스트 (예를 들어 부작용 실마리정보 평가 signal evaluation) 하거나 다중 가설을 한 번에 탐색 (예를 들어 부작용 실마리정보 감지 signal detection) 할 수 있다. 모든 경우에 있어, 목적은 고품질의 인과 관계 추정을 산출하는 것이다.

이 장에서는 우선 OHDSI Methods Library에 R 패키지로 구현된 다양한 **인구 수준 추정 Population-Level Estimation** 연구설계를 설명한다. 예제 평가 연구의 설계를 자세히 설명한 다음, ATLAS 및 R을 사용하여 설계를 구현하는 방법에 대한 단계별 가이드를 또한 설명한다. 마지막으로 연구 진단 및 효과 크기 추정을 포함하여 연구에서 생성된 다양한 결과를 검토한다.

12.1 코호트 방법론 설계

코호트 방법론은 무작위 임상 시험을 모방하려고 한다. (Hernan and Robins, 2016) 하나의 치료를 시작한 환자(표적군 target cohort)는 다른 치료를 시작한 환자(대조군 comparator)와 비교되고, 치료를 받은 후 특정 기간 (예를 들어 치료를 받는 기간) 추적 관찰된다. 표 12.1에서 강조하는 5 가지 사항을 선택함으로써 코호트 연구에서 연구자가 얻기 원하는 답에 대한 질문을 지정할 수 있다.

Table 12.1: 코호트 비교연구 설계에서 주된 고려사항.

선택	설명
표적 코호트 Target cohort	표적 치료를 받은 환자군 cohort
대조 코호트 Comparator cohort	대조 치료를 받은 환자군 cohort
결과 코호트 Outcome cohort	관심 결과 outcome of interest가 발생한 환자군 cohort
위험노출기간 Time-at-risk	(보통 표적 및 대조 코호트의 시작과 종료 시점에 기반하여) 결과 발생을 모델
	표적군과 대조군의 차이를 보정하여 효과 effect를 추정할 때 사용하는 통

모델 선택은 결과 모델의 유형을 지정한다. 예를 들어, 결과가 발생했는지를 평가하고 교차비 odds ratio를 산출하는 로지스틱 회귀분석 logistic regression을 사용할 수 있다. 로지스틱 회귀분석은 위험 노출 기간(TAR)이 표적군 target cohort과 대조군 comparator cohort 양쪽 모두에서 같거나 무관하다고 가정한다. 대안으로, 포아송 회귀분석 poisson regression을 선택할 수 있는데, 이는 일정한 발생률 incidence rate을 가정하고, 발생률 비율 incidence rate ratio을 추정한다. 콕스 회귀분석 Cox regression을 종종 사용하기도 하는데, 이는 표적군과 대조군 사이의 비례 proportional hazard을 가정하며, 위험 비율 hazard ratio을 추정하려고 대상 질병이 처음 발생할 때까지의 시간 time-to-first-outcome을 고려한다.



New-user cohort method는 본질적으로 하나의 치료를 다른 치료에 비교하여 비교 효과를 추정하는 방법이다. 치료 노출 군의 비교 대상인 치료 비노출 군을 정의하기 힘들기 때문에, 치료와 비 치료를 비교하기 위해 이 방법을 사용하긴 어렵다. 직접 효과 추정에 이 방법을 사용하려는 경우, 결과에 영향을 미치지 않을 동일한 적응증이 적용되는 비교 대상을 관심 노출 군 exposure of interest으로 선택하는 방법이 선호된다 (역자 주: 예를 들면 ACE inhibitor 노출 군과 고혈압 치료제 비노출 군을 비교하기는 어렵다. 왜냐하면, 그 두 군은 본질적으로 기저 특성이 다르기 때문이다. 사과와 생선을 비교한다고 가정해 보라. 대신 ACE inhibitor 노출 군과 ARB 노출 군을 비교해 볼 수는 있다. 그 둘 다 기저 특성이 비슷할 것이기 때문이다. 부사 사과와 흥옥 사과를 비교 (within-class comparison)한다고 상상해 보라). 하지만, 이러한 비교 대상을 항상 사용할 수 있는 것은 아니다.

주요 관심사는 표적 치료를 받는 군이 대조 치료를 받는 군과 전체적으로 systemically 다를 수 있다는 것이다. 예를 들어, 연구 대상 치료를 받는 표적 코호트 target cohort의 평균 연령이 60세이지만, 해당 치료를 받지 않은 대조 코호트 comparator cohort의 평균 연령이 40세라고 가정해보자. 연령과 관련된 건강 결과 (예를 들어 뇌졸중)는 양 군 간에 상당히 차이가 날 것이다. 이런 정보에 대해 정확히 숙지하지 못한 연구자는 해당 치료가 뇌졸중과 유의미한 인과관계를 보인다고 결론을

내릴 수 있다. 따라서, 해당 치료를 받지 않았다면 표적군의 환자가 뇌졸중에 걸리지 않으리라 생각할 수 있다. 이러한 결과는 전적으로 잘못되었다. 단순히 표적군의 연령이 높아서 뇌졸중을 많이 경험할 수 있기 때문이다. 표적군이 해당 치료를 받지 않았더라도, 뇌졸중의 발병률은 비슷할 수 있다. 여기서 나이는 “교란변수 confounder”이다. 관찰형 연구에서 교란변수를 통제하는 한 가지 방법은 성향 점수 propensity score를 이용하는 것이다.

12.1.1 성향 점수

무작위 배정 시험 randomized trial에서 (가상의) 동전 던지기를 통해 환자를 각각의 그룹에 무작위로 배정한다. 이렇게 하면 설계상 치료군과 대조군에 속한 환자가 대상치료를 받을 확률은 나이와 같은 환자의 기본 특성과 관련이 없게 된다. 동전에는 환자에 대한 정보가 없으며, 우리는 환자가 대상에 노출될 정확한 확률을 확실하게 알 수 있다. 결과적으로 임상시험에서 환자 수가 증가함에 따라 신뢰도가 증가해 두 환자군은 본질에서 어떠한 환자 특성이라도 다를 수 없다. 이 보장된 균형은 무작위 배정 시험이 측정한 특성 (예를 들어 나이)뿐 아니라 유전적 특성과 같이 무작위 시험이 측정하지 못한 특성에도 모두 적용된다.

주어진 환자의 **성향 점수 Propensity score(PS)**는 환자가 대조 치료군과 비교하여 표적 치료를 받을 확률이다. (Rosenbaum and Rubin, 1983) 균형 잡힌 two-arm 무작위 임상시험에서, 모든 환자의 성향 점수는 0.5이다. 성향 점수 조정된 관찰 연구에서, 우리는 치료개시 시점과 치료개시 전 (환자가 실제로 받은 치료와 관계없이)에 관찰할 수 있는 것에 근거해 표적 치료를 받을 환자의 확률을 추정한다. 이것은 간단한 예측 모델링 응용프로그램이다. 환자가 표적 치료를 받았는지의 여부를 예측하는 적합한 모델 (예를 들어 로지스틱 회귀분석)을 만들고, 이 모델을 사용하여 각 환자에 대한 예측 확률을 생성한다. 표준 무작위 임상시험과 달리, 다른 환자는 표적 치료를 받을 확률이 다르다. 성향 점수는 여러 가지 방법으로 사용할 수 있다. 예를 들어, 표적 피험자를 유사한 PS를 가진 comparator 피험자에게 매칭(PS matching)하거나, 성향 점수를 기반으로 연구 집단을 충화 PS stratification하거나, 성향 점수에서 파생된 Inverse Probability of Treatment Weighting(IPTW)을 사용하여 피험자에게 가중치를 적용하여 사용할 수 있다. 매칭할 때, 각 대상에 대하여 한 명의 비교 대상을 선택하거나, 일-대-다 비율 짹짓기 variable-ratio matching를 활용하여 대상당 두 명 이상의 비교 대상을 허용할 수 있다. (Rassen et al., 2012)

예를 들어 일대일 PS 매칭을 사용한다고 가정해보자. Jan이라는 환자가 표적 치료를 받을 선별 확률 prior probability이 0.4이고, 실제로 표적 치료 target treatment를 받고, Jun이라고 하는 또 다른 환자는 표적 치료를 받을 선별 확률이 0.4이지만, 사실상 대조 치료 comparator treatment를 받았다면, 적어도 측정된 교란변수에 대해 Jan과 Jun의 결과 비교는 작은 무작위 시험과 같다. 이 비교는 Jan과 Jun의 인과적인 대조를 무작위 시험으로 산출한 결과만큼 양호하게 추정할 것이다. 추정은 다음과 같이 진행된다: 표적치료를 받은 모든 환자에 대해, 대조 치료를 받았지만, 표적을

받는 선형적 확률이 동일한 하나 이상의 일치하는 환자를 찾는다. 그들의 짹지어진 환자군 matched group 안에서 표적 환자 target group의 결과와 비교 그룹 comparator group의 결과를 비교한다.

성향점수 방법은 측정된 교란변수 measured confounder를 제어한다. 사실, 측정된 특성 하에서 치료배정 treatment assignment이 “강하게 무시할 수 있는” 경우라면, 성향 점수는 인과 관계의 비 편향적 추정을 산출할 것이다. “강력하게 무시할 수 있는” 조건이란 측정되지 않은 교란변수가 없고, 측정된 교란변수는 적절하게 조정된다는 것을 의미한다. 불행히도, 이것은 검증할만한 가정은 아니다. 18장에서 이에 대한 추가적인 논의를 볼 수 있다.

12.1.2 변수 선택

전통적으로 성향 점수는 연구자가 임의로 선택된 특성 manually selected characteristics을 기반으로 계산되었다. OHDSI 도구가 그러한 관행을 지원할 수는 있지만, 많은 일반적 특성 (즉, 연구의 특정 노출 및 결과에 따라 선택되지 않은 특성)을 포함하는 것을 선호한다. (Tian et al., 2018) 이러한 특성에는 인구학적인 특성뿐만 아니라 치료 개시일 전과 개시일에 관찰된 모든 진단, 약물 노출, 측정 및 의료절차가 포함된다. 모델은 전형적으로 10,000 – 100,000가지의 독특한 특성을 포함하며, 이러한 모델은 Cyclops 패키지에서 구현되는 large-scale regularized regression (Suchard et al., 2013) 을 사용하여 적합한다. 본질적으로 우리는 치료 배정의 예측을 위하여 어떠한 환자 특성이 알고리즘에 사용되어야 하는지 데이터가 스스로 결정하도록 한다.



치료로 이어지는 진단과 같은 많은 관련 데이터 포인트가 해당 날짜에 기록되기 때문에 일반적으로 공변량을 정의할 때 치료 개시일의 변수를 포함한다. 이날에 연구의 주제가 되는 표적 치료와 대조 치료 자체도 기록되는데, 이러한 치료는 우리가 예측하려는 바로 그것이기 때문에 성향 모델에 포함해서는 안 된다. 따라서 공변량 집합에서 표적 치료와 대조치료는 반드시 제외해야 한다.

일부 연구자는 “올바른” 인과 구조를 반영하기 위해 임상적 전문지식에 의존하지 않는 데이터 기반 접근방식 data-driven approach을 통한 공변량 선택이 소위 도구적 변수 instrumental variable와 충돌자 collider를 잘못 포함해 분산을 증가시키고 잠재적으로 비뚤림을 만들어낼 위험이 있다고 주장해왔다. (Hernan et al., 2002) 하지만 이러한 우려가 실제 시나리오에서 큰 영향을 미칠 가능성은 적다. (Schneeweiss, 2018) 게다가, 의학에서 진정한 인과 관계는 거의 알려지지 않아서 피하며, 서로 다른 연구자에게 특정 연구 주제에 대해 ‘올바른’ 공변량을 선택해 달라고 요청한다면, 각 연구자는 서로 다른 공변량 리스트를 주문할 것이 분명하고, 전체 과정은 재현 불가능해질 것이다. 무엇보다도, 성향 점수 모델의 검사, 모든 공변량의 균형 balance 평가, 음성 대조군을 통한 평가 등을 통해 도구적 변수 및 충돌자에 의해 발생하는 대부분의 문제를 진단할 수 있다.

12.1.3 캘리퍼

성향 점수가 0에서 1까지의 연속성을 갖기 때문에 정확한 일치는 거의 불가능하다. 그 대신, 매칭 프로세스는 대상 환자의 성향 점수와 일치하는 환자를 “캘리퍼 caliper”라고 알려진 내성 범위 내에서 찾는다. 이전 연구 (Austin, 2011)에 따라, 우리는 로직 척도에서 0.2 표준편차의 기본 default 캘리퍼를 사용한다.

12.1.4 오버랩: 선호 점수

성향 매칭 방법은 일치하는 환자가 필요하다! 따라서 주요 진단은 두 그룹의 성향 점수 분포를 보여준다. 해석을 용이하게 하기 위해 OHDSI 도구는 “선호 점수 preference score”라는 성향 점수의 변형을 그린다. (Walker et al., 2013) 선호 점수는 표적 치료와 대조 치료, 두 가지 치료법의 “market share”를 조정하다. 예를 들면, 10%의 환자가 표적 치료를 받고 (90%가 대조 치료를 받는 경우), 선호 점수가 0.5인 환자는 표적 치료를 받을 확률이 10%이다. 수학적으로 선호 점수는 다음과 같다.

$$\ln \left(\frac{F}{1-F} \right) = \ln \left(\frac{S}{1-S} \right) - \ln \left(\frac{P}{1-P} \right)$$

여기서 F 는 선호 점수, S 는 성향 점수, 그리고 P 는 표적치료를 받은 환자의 비율이다.

Walker et al. (2013)는 “경험적 평형 empirical equipoise”의 개념을 논의한다. 적어도 노출의 절반이 0.3과 0.7 사이의 선호 점수를 갖는 환자에게 노출 쌍 exposure pair이 경험적 평형을 이루는 것으로 받아들인다.

12.1.5 균형

좋은 방침 good practice은 성향 점수 보정이 균형 잡힌 환자 그룹을 만드는 데 성공했는지 항상 확인하는 것이다. 그림 12.19는 균형을 점검하기 위한 표준 OHDSI 출력물을 보여준다. 각 환자의 특성에 대해 성향 점수 보정 전과 후에 두 노출 그룹 간의 평균 차이를 표준화한다. 일반적으로 조정 후 표준화된 차이 standardized mean difference의 상한이 0.1 미만가 되도록 권장한다. (Rubin, 2001)

12.2 자가 통제 코호트 연구 설계 Self-Controlled Cohort Design

자가 통제 코호트 self-controlled cohort(SCC) 설계 (Ryan et al., 2013a)는 노출 직전의 결과 비율을 기준으로 노출되는 동안의 결과 비율을 비교한다. 표 12.2에 제시된 4가지 선택 사항은 SCC 질문을 정의한다.

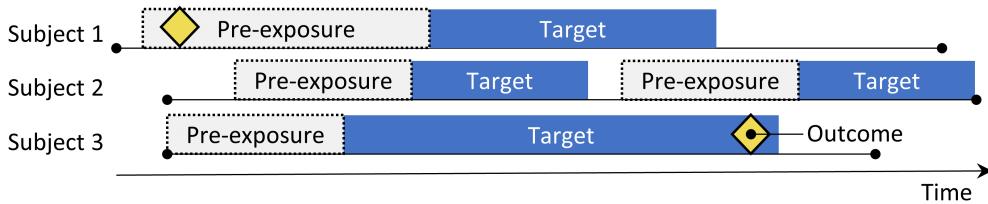


Figure 12.2: 자가 통제 코호트 연구 디자인. 목표물에 노출되는 동안의 결과 비율은 사전 노출 시간의 결과 비율과 비교된다.

Table 12.2: 자가 통제 코호트 연구에서의 주요 설계 선택

선택	설명
표적 코호트 Target cohort	표적 치료를 받은 환자군 cohort
결과 코호트 Outcome cohort	관심 결과 outcome of interest가 발생한 환자군 cohort
위험노출기간 Time-at-risk	(보통 표적 코호트의 시작과 종료 시점에 기준으로) 결과 발생을 살펴볼 기간의 범위
통제 기간 Control time	통제 기간 control time 으로 사용하는 기간

노출 그룹을 구성하는 동일한 피험자가 대조 그룹 control group으로 사용되기 때문에 사람 간 between-person의 차이를 조정할 필요가 없다. 그러나 이 방법은 다른 기간 간의 기준의 위험도 차이 등 다른 차이점에 대해 취약하다.

12.3 환자-대조군 연구 설계 Case-Control Design

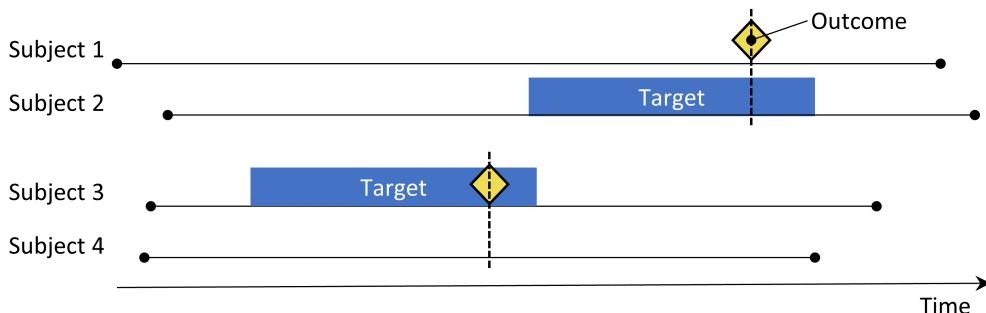


Figure 12.3: 환자-대조군 설계. 결과가 있는 대상 ("케이스")은 노출 상태 측면에서 결과 ("컨트롤")이 없는 대상과 비교된다. 나이와 성별 등 다양한 특성에 케이스와 컨트롤이 매칭되는 경우가 많다.

환자-대조군 연구 (Vandenbroucke and Pearce, 2012) 는 “특정 질병 결과가 있는 사람이 질병이 없는 사람보다 특정 치료 agent에 더 자주 노출되는가?”라는 질문을 고려한다. 따라서, 주요 아이디어는 환자 cases (다시 말하면 outcome을 경험한 피험자)를 대조군 controls (다시 말하면 관심 결과를 경험하지 않은 피험자)에 비교하는 것이다. 표 12.3에 선택 사항은 환자-대조군 case-control 질문을 정의한다.

Table 12.3: 환자-대조군 연구에서의 주요 설계 선택

선택	설명
결과 코호트 Outcome cohort	관심 결과 outcome of interest가 발생한 환자군 cohort
Control cohort	control 군. 일반적으로는 결과 코호트 특정 선택 로직을 통해 결과 코호트 기반으로 자동으로 유도된다.
표적 코호트 Target cohort	표적 치료를 받은 환자군 cohort
네스팅 코호트 Nesting cohort	선택사항, case 와 control 을 유도할 부분집합 subpopulation 코호트
위험노출기간 Time-at-risk	(보통 코호트의 index date 기준) 노출 상태를 판단할 기간

Table 12.4: 환자-교차 연구에서의 주요 설계 선택

선택	설명
결과 코호트 Outcome cohort	관심 결과 outcome of interest가 발생한 환자군 cohort
표적 코호트 Target cohort	표적 치료를 받은 환자군 cohort
위험노출기간 Time-at-risk	(보통 코호트의 index date 기준) 노출 상태를 판단할 기간
Control 기간	control 기간으로 사용할 기간

종종 우리는 나이와 성별 등 환자군의 특성을 매칭하여 대조군을 설정한다. 또 달리 많이 사용되는 방법은, 특정 질병이 있는 환자군처럼, 특정 하위 집단 subgroup 환자군 안에서 nested analysis를 이용한다.

12.4 환자-교차 연구 설계 Case-Crossover Design

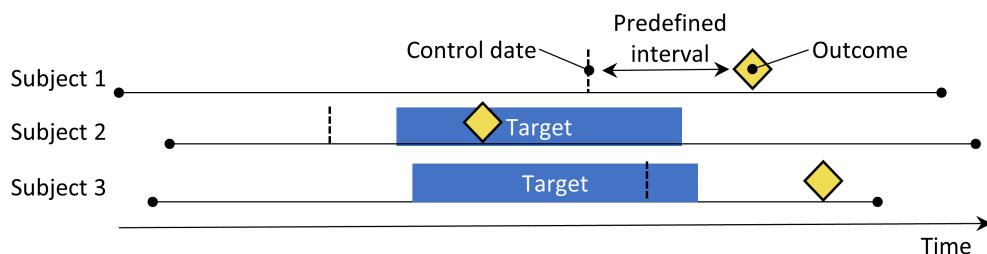


Figure 12.4: 환자-교차 연구 설계. 결과 근처의 시간을 결과 날짜 이전의 미리 정의된 간격으로 설정된 제어 날짜와 비교한다.

환자-교차 연구 case-crossover study (Maclure, 1991) 설계는 결과 outcome 이전의 정해진 기간 노출률 rate of exposure의 차이가 나는지 평가하는 방법이다. 이것은 결과 발생 시점의 특이한 사항이 있는지 확인하는 방법이다. 표 12.4는 환자-연구 연구 정의를 위한 선택지를 보여준다.

환자군은 그 자체로 대조군으로 사용된다. 자가 통제 코호트 연구 설계처럼, 환자 간 차이에 의한 교란변수를 통제할 수 있도록 환자군이 잘 선택되어야 한다. 한 가지 우려는, 결과 일시가 항상 대조군 일시보다 뒤에 오기 때문에, 전반적인 노출 빈도가 시간이 갈수록 높아져 양성 편향 (만약

Table 12.5: 자기 대조 환자군 연구에서의 주요 설계 선택

선택	설명
표적 코호트 Target cohort	표적 치료를 받은 환자군 cohort
결과 코호트 Outcome cohort	관심 결과 outcome of interest가 발생한 환자군 cohort
위험노출기간 Time-at-risk	(보통 표적 코호트의 시작과 종료 시점에 기준으로) 결과 발생을 살펴볼 기간의 범위
모델	시간에 따라 변하는 교란변수 time-varying confounder의 보정을 포함하여 효과 effect를 추정할 때 사용할 통계 모델

노출빈도가 시간이 갈수록 줄어든다면 음성 편향이)이 일어날 수 있다는 점이다. 이를 통제하기 위하여 환자-교차 연구 연구 설계에 나이, 성별을 이용한 짹짓기를 통한 대조군을 추가하여 노출률을 보정하는 환자-시간-대조군 연구 설계 case-time-control design (Suissa, 1995) 가 개발되었다.

12.5 자기 대조 환자군 연구 설계 Self-Controlled Case Series Design

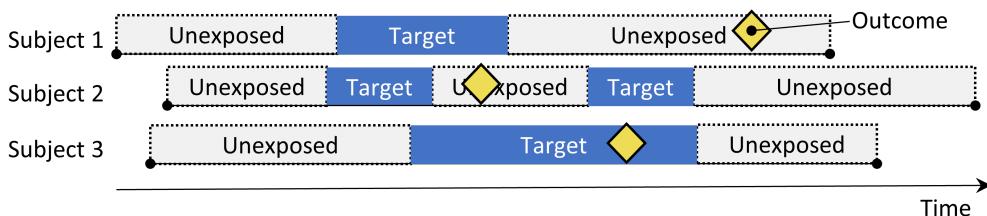


Figure 12.5: 자기 대조 환자군 연구설계. 노출 중 결과 발생의 비율은 노출되지 않는 중 발생한 결과의 비율과 비교된다.

자기 대조 환자군 연구 Self-Controlled Case Series(SCCS) 설계 (Farrington, 1995; Whitaker et al., 2006) 는 전체 비노출 기간 (노출 이전, 노출 사이, 노출 후)과 노출 기간의 결과 발생의 비율을 비교한다. 즉 Poisson regression conditioned on the person이라고 할 수 있다. 따라서, 그것은 “환자에게 결과가 발생하였을 때, 비노출 기간보다 노출 기간에 발생할 가능성이 더 높은가?”이다. 표 12.5의 선택사항은 SCCS 질문을 정의한다.

다른 자가 통제 설계 self-controlled design와 마찬가지로, SCCS는 사람 간의 교란 변수 confounding due to between-person difference는 잘 바로잡지만, 시간의 변화에 따른 교란변수 confounding due to time-varying effect의 영향에는 취약하다. 이를 위해 몇 가지 보정을 시도할 수 있는데, 예를 들면 나이와 계절을 바로잡는 것이다. SCCS의 특별한 변형은 관심 대상의 노출뿐만 아니라 데이터베이스에 기록된 약물에 대한 다른 모든 노출 (Simpson et al., 2013)에 잠재적으로 수천 개의 추가변수를 모델에 추가하는 것을 포함한다. 정규화 하이퍼-파라미터를 선택하기 위해 교차검증 cross-validation 을 사용하는 L1-regularization이 관심 대상 노출을 제외한 모든 노출 계수에 적용된다.

SCCS의 기본 가정 중 하나는 관찰 기간 종료가 결과 날짜 outcome date와 독립적이라는 것이다. 몇 가지 결과의 경우, 예를 들어 뇌졸중과 같은 치명적인 fatal 질병의 경우, 이러한 가정이 위반될 수 있다. 이러한 종속성을 수정하는 SCCS의 확장이 개발되었다. (Farrington et al., 2011)

12.6 고혈압 연구 설계하기

12.6.1 문제 정의

ACE 억제제(ACEi)는 고혈압이나 협심증 심장 질환 환자, 특히 울혈성 심부전, 당뇨병 또는 만성 신장 질환과 같은 다른 합병증이 있는 환자에게 널리 사용된다. (Zaman et al., 2002) 일반적으로 입술, 혀, 입, 후두, 인두 또는 눈 주위 부위가 부어오르는 심각한 중증도의 때로는 생명을 위협하는 혈관부종 부작용은 ACE 억제제 사용과 관련이 있다. (Sabroe and Black, 1997) 그러나 이러한 약물의 사용과 관련된 혈관부종에 대한 절대 및 상대 위험에 대한 정보는 제한적이다. 기존의 증거는 주로 다른 집단에 대한 일반화가 불가능한 특정 코호트 (예를 들어 주로 남성 퇴역 군인이나 메디케이드 (역자 주: Medicaid는 미국의 65세 미만 저소득층과 장애인을 위한 의료 보조 제도) 수혜자)에 대한 조사 또는 불안정한 위험 추정치를 제공하는 경위가 거의 없는 조사를 기반으로 한다. (Powers et al., 2012) 많은 관찰 연구에서 혈관부종의 위험에 대해서 ACEi와 베타 차단제를 비교하였지만 (Magid et al., 2010; Toh et al., 2012), 베타 차단제는 더 이상 고혈압의 1차 치료제로 권장되지 않는다. (Whelton et al., 2018) 사용 가능한 대체 치료제는 thiazide 또는 thiazide-like 이뇨제일 수 있으며, 이는 혈관부종의 위험 증가 없이 급성 심근경색과 같은 고혈압 관련 위험을 관리하는데 ACEi만큼 효과적이다.

다음은 비교 추정 질문을 다루기 위한 인구 수준 평가 프레임워크를 관찰 보건 데이터 observational healthcare data에 적용하는 방법을 보여준다:

Thiazide 및 thiazide-like 이뇨제를 새로 사용하는 환자에 비교해 ACEi를 새로 사용하는 환자의 혈관부종의 위험도는 어떻게 되는가?

Thiazide 및 thiazide-like 이뇨제를 새로 사용하는 환자에 비교해 ACEi를 새로 사용하는 환자의 급성 심근경색의 위험도는 어떻게 되는가?

이는 비교 효과 추정 comparative effect estimation 질문이기 때문에 12.1장에서 설명한 대로 Cohort Method를 적용할 것이다.

12.6.2 표적군 및 대조군

첫 번째 관찰된 고혈압 치료가 ACEi 또는 THZ 계열의 활성 성분을 단독요법으로 사용하는 경우를 새 사용자로 간주한다. 이 중 치료 시작 후 7일 동안 다른 항고혈압제를 시작하지 않은 경우를 단독요법으로 정의한다. 환자가 첫 번째 노출 전 데이터베이스에서 적어도 1년 동안 지속해서 관찰되고, 치료 시작 전 또는 그 이전에 기록된 고혈압 진단이 있는 경우로 정의했다.

12.6.3 결과

입원 또는 응급실 방문 중에 혈관부종 기록이 있고, 그 이전 일주일간 혈관부종 발생이 없었던 경우를 혈관부종으로 정의하였다. 입원 또는 응급실 방문 중에 심근경색 기록이 있고, 그 이전 180일간 심근경색 발생 기록이 없었던 경우를 심근경색으로 정의하였다.

12.6.4 위험 노출 기간 Time-At-Risk(TAR)

30일까지의 차이 gap를 인정하여, 치료 시작 다음 날부터 시작하여 연속적인 약물 노출이 (30일 이상) 중단될 때까지를 위험 노출 기간 Time-At-Risk(TAR)로 정의하였다.

12.6.5 모델

인구학적 특징, 상태, 약물, 절차, 측정, 관찰 결과, 다양한 병존 질환을 포함하는 공변량 기본 모음을 사용하여 적합한 성향 점수 모델을 구하는데, 공변량에서 ACEi와 THZ를 제외한다. 여기에서 다비율 짹짓기를 수행하고, 성향 점수 짹짓기 된 모음에 대해 조건화된 콕스 회귀분석을 실시한다.

12.6.6 연구 요약

Table 12.6: 코호트 비교 연구를 위한 주요 설계 선택.

선택	설명
표적 코호트 Target cohort	고혈압 치료로 ACE 억제제 초치료를 시작한 환자 cohort
대조 코호트 Comparator cohort	고혈압 치료로 thiazide 계 이뇨제 초치료를 시작한 환자 cohort
결과 코호트 Outcome cohort	혈관부종 또는 심근경색이 발생한 코호트
위험노출기간 Time-at-risk	치료 시작 다음날부터 치료 중단까지
모델	일-대-다 매칭 후 콕스 비례위험모형 사용

12.6.7 대조군 질문

우리 연구 디자인이 실제와 일치하는 추정치를 산출하는지 평가하기 위해 진짜 효과 크기가 알려진 곳에 일련의 통제 질문을 추가로 포함한다. 통제 질문은 위험비 hazard ratio는 1인 음성 대조군 negative control과 1보다 큰 위험 비율을 갖는 양성 대조군 positive control으로 나눌 수 있다. 우리는 몇 가지 이유에서 실제 음성 대조군을 사용하고, 음성 대조군에 근거해 양성 대조군을 만든다. 대조군을 설정하고 사용하는 방법은 18장에서 자세히 다룬다.

12.7 ATLAS를 사용한 연구 구현하기

여기서는 위에서 수행한 고혈압 연구를 ATLAS의 추정 기능 Estimation function을 사용하여 어떻게 구현하는지 보여준다. ATLAS의 왼쪽 바에서 Estimation 를 클릭하고 새로운 평가 연구를 작성하고, 이 연구에 쉽게 인식할 수 있는 이름을 붙이자. 연구 설계는 를 클릭하여 언제든지 저장할 수 있다.

추정 설계 기능 estimation design function에는 세 가지 섹션이 있다: 비교 comparisons, 분석 설정 analysis settings, 평가 설정 evaluation settings. 다중 비교 및 다중 분석 설정을 할 수 있으며, ATLAS는 이러한 모든 조합을 각각의 분석으로 수행한다. 여기서는 각 섹션에 대해 설명한다:

12.7.1 대조 코호트 설정

한 연구에는 하나 이상의 대조 대상이 있을 수 있다. “Add Comparison”을 클릭하면 새 대화 상자가 열린다. 표적 target 및 대조 comparator 코호트를 선택하려면 을 클릭하면 된다. “Add Outcome”을 클릭하면, 위에서 정의한 두 개의 결과 코호트를 추가할 수 있다. 10장에서 설명한 대로 이미 코호트가 생성된 것으로 가정한다. 부록에서 표적군 (부록 B.2), 대조군 (부록 B.5), 결과 (부록 B.4과 부록 B.3) 코호트에 대해 자세히 볼 수 있다. 완료되면 그림 12.6에서와 같은 창이 생성될 것이다.

하나의 표적-대조 쌍 target-comparator pair에 대해 여러 개의 결과를 선택할 수 있다는 점에 주목하자. 각 결과는 독립적으로 처리되며 별도의 분석이 이루어진다.

음성 대조군 결과

음성 대조군 결과 Negative Control Outcome은 표적군 또는 대조군에 의해 야기된 것으로 생각되지 않는 결과이며 (역자 주: 즉, 위험 노출에 독립적으로 발생한 결과, 예를 들면 고혈압 약물 노출에 따른 항문 용종 발생 유무는 좋은 음성 대조군이 될 수 있다), 따라서 실제 위험 비는 1과 동일해야 한다. 이상적으로는 각 결과 코호트에 대해 적절한 코호트 정의를 가진다고 가정한다.

The screenshot shows the 'Comparison' dialog in ATLAS. It has three main sections: 'Choose your target cohort:' containing 'New users of ACE inhibitors as first-line monotherapy for hypertension' with a blue save icon and a red delete icon; 'Choose your comparator cohort:' containing 'New users of Thiazide-like diuretics as first-line monotherapy for hypertension' with a blue save icon and a red delete icon; and 'Choose your outcome cohorts:' with a 'Add Outcome' button. Below these is a table showing two entries:

ID	Name	Edit cohort	Remove
1770712	Angioedema outcome	Edit cohort	Remove
1770713	Acute myocardial infarction outcome	Edit cohort	Remove

At the bottom, it says 'Showing 1 to 2 of 2 entries' and has 'Previous' and 'Next' buttons.

Figure 12.6: 대조 dialog

그러나, 우리는 일반적으로 음성 대조 결과 당 하나의 concept set과 이를 결과 코호트로 변환하는 표준 논리만 가진다. 여기서는 18장에서 설명한 대로 concept set이 이미 생성되었다고 가정하고 간단하게 선택할 수 있다. 음성 통제 concept set에는 음성 통제 당 하나의 concept만을 포함해야 하며, **하위 개념은 포함하지 않아야** 한다. 그림 12.7은 본 연구에 사용된 음성 대조군 concept set을 보여준다.

포함할 개념

포함할 개념 concepts to include 선택 시, 우리는 성향점수 모델 등에 어떠한 공변량이 생성되기를 원하는지 선택할 수 있다. 공변량을 지정하면, 모든 다른 공변량 (선택하지 않은)은 제외된다. 보통 regularized regression을 이용해 환자의 모든 기저 공변량에 대해 균형을 맞출 수 있는 모델이 만들 어지기를 바란다. 만약 특정 공변량만을 선택하기 원한다면, 그것은 다른 연구자가 직접 공변량을 골라 수행한 다른 연구를 따라 해 보기를 원해서일 것이다. 가끔 공변량이 특정한 비교 (비교 시 이미 알고 있는 교란변수) 또는 분석 (특정 공변량 선택시 그 결과 차이에 대한 평가)에 관련되어 있기 때문에, 그러한 공변량을 비교 comparison 섹션 또는 분석 analysis 섹션에서 정의할 수 있다.

배제할 개념

추가하는 개념보다는 배제할 개념 concepts to exclude을 지정하는 경우가 많다. 배제할 concept 을 지정하면, 배제할 concept을 제외한 모든 공변량을 사용함을 뜻한다. 기본 공변량 집합 default

Negative controls for ACEi and THZ

Concept Set Expression Included Concepts (75) Included Source Codes Explore Evidence Export Compare

Show 25 entries Search: _____

Showing 1 to 25 of 75 entries Previous 1 2 3 Next

	Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	<input type="checkbox"/> Exclude	<input checked="" type="checkbox"/> Descendants	<input checked="" type="checkbox"/> Mapped
72748	74779009	Strain of rotator cuff capsule	Condition	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
73241	197210001	Anal and rectal polyp	Condition	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
73560	55260003	Calcaneal spur	Condition	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
75911	65358001	Acquired hallux valgus	Condition	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
76786	63643000	Derangement of knee	Condition	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure 12.7: 음성 대조군 concept set

set of covariate 설정을 이용하면, 치료 시작 시의 모든 약물, 시술을 이용하기 때문에, 표적 치료 및 대조 치료에 해당하거나 이것과 직접적으로 관련된 concept을 배제해야 한다. 예를 들어, 만약 표적 치료가 약물 정맥 주입 치료라면, 우리는 약물뿐 아니라 정맥 주입 시술 역시도 성향 점수 모델에서 제외해야 한다. 이 예제에서 우리는 ACEi와 THZ를 배제했다. 그림 12.8에서 ACEi와 THZ, 그리고 하위 concept을 포함하여 배제할 concept set을 구성하는 것을 볼 수 있다.

음성 대조군과 배제할 concept을 지정한 후, 비교 섹션의 아래쪽 절반 창은 그림 12.9과 같이 보일 것이다.

12.7.2 효과 추정 분석 설정

비교 창을 닫은 후 “Add Analysis Settings” 을 클릭할 수 있다. “Analysis Name”이라는 상자에 다음에 기억하고 분류하기 쉽도록 분석별 고유한 이름을 지정할 수 있다. 예를 들어 “Propensity score matching”이라고 이름을 지을 수 있다.

연구 집단

분석에 포함할 피험자 집단과 같은 연구 집단 study population을 지정하는데 다양한 옵션이 있다. 코호트 정의 cohort definition 도구에서 표적 및 대조 코호트를 설계할 때 사용할 수 있는 옵션과 대부분 겹친다. 코호트 정의 도구 대신에 Estimation 옵션을 사용하는 한 가지 이유는 재사용성 re-usability이다. 표적, 대조 및 결과 코호트를 완전히 독립적으로 정의한 후에 이들 사이의 종속관계를 추가할 수 있다. 예를 들어, 치료 개시 전에 결과가 있었던 사람을 제외하기를 원한다면, 표적 및 대조 코호트 정의 내에서 그렇게 설정할 수도 있지만, 그렇게 한다면 모든 결과 outcome에 대해 별도의 코호트를 작성해야 한다! 대신에, Estimation 설정에서 이런 결과를 가진 사람을 제거하도록

Concept Set #1798551		Concepts to exclude for ACEi and THZ					Optimize	
Concept Set Expression	Included Concepts (38225)	Included Source Codes	Explore Evidence	Export	Compare			
Show 25 ▾ entries								Search: <input type="text"/>
Showing 1 to 14 of 14 entries								Previous [1] Next
Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	<input type="checkbox"/> Exclude	<input checked="" type="checkbox"/> Descendants	<input type="checkbox"/> Mapped	
1342439	38454	trandolapril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1334456	35296	Ramipril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1331235	35208	quinapril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1373225	54552	Perindopril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
1310756	30131	moexipril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Figure 12.8: 배제할 concept을 정의하는 concept set

Negative control concept set:

Negative controls for ACEi and THZ

<input checked="" type="radio"/> Covariate selection		
Concepts to include when constructing the covariates to be used in this study. (Leave blank if you want to include every concept).*	<input type="button" value=""/>	
* Concepts defined here are combined with those defined in the Analysis settings section.		
Concepts to exclude when constructing the covariates to be used in this study.*	<input type="button" value=""/>	
* Concepts defined here are combined with those defined in the Analysis settings section.		

Figure 12.9: 음성 대조군에 대한 concept set과 배제할 concept을 보여주는 비교 창

선택할 수 있다. 이렇게 함으로써 이제 우리는 (음성 대조군 결과뿐만 아니라) 두 가지 관심 결과에 대해 표적 및 대조 코호트를 재사용할 수 있다.

연구 시작 및 종료일 study start and end dates은 분석을 특정 기간으로 제한하는데 사용할 수 있다. 연구 종료일 또한 위험 노출 기간 risk window를 잘라낼 수 있게 되어 연구 종료일 이후의 결과는 고려하지 않게 할 수 있다. 연구 시작일을 선택하는 한 가지 이유는 연구 중인 약물 중 하나가 새로운 것이며, 전에는 존재하지 않을 수 있기 때문이다. “**두 노출이 모두 관찰되는 기간으로 분석을 제한하라 Restrict the analysis to the period when both exposures are present in the data?**”는 옵션을 “예 yes”라고 설정하면, 새 약물이 데이터베이스에 존재하는 시점을자동으로 연구 시작일로 조정할 수 있다. 연구 시작일과 종료일을 조정하는 또 다른 이유는 시기에 따라 (예를 들어 새로운 약물 부작용이 알려지면서) 임상 업무의 변화가 있고, 우리는 보통 특정 방식으로 임상이 이루어질 때만 관심이 있기 때문이다. (역자 주: 날짜는 상대 날짜가 아닌 절대 날짜임을 기억하라. 2019-12-31이라고 지정하면 실제 2019년 12월 31일을 의미한다)

“**환자별로 첫 번째 위험 노출만 포함하겠는가? Should only the first exposure per subject be included?**” 옵션을 사용하여 환자별로 첫 번째 위험 노출만으로 코호트를 제한할 수 있다. 이 옵션은 이번 예제에서처럼 코호트 정의에서 이미 수행한 경우가 많다. 유사하게, 코호트 정의에 “**코호트에 포함될 사람이 기준 날짜 전에 최소 연속적 관측 시기 The minimum required continuous observation time prior to index date for a person to be included in the cohort**” 옵션이 설정된 경우가 많아, 여기에 0으로 남겨둘 수 있다. 이러한 옵션은 기준 날짜 이전에 관찰된 시간 (OBSERVATION_PERIOD 테이블에서 정의된)을 가짐으로써 성향 점수를 계산할 수 있는 환자에 대한 충분한 정보가 있음을 보장하고, 환자가 이전에 노출된 적 없는, 치료에 대한 진정한 새로운 사용자 new user임을 보장하기 위해 자주 사용한다.

“**만일 피험자가 여러 코호트에 중복되어 포함된다면, 중복포함을 막기 위해 새로운 위험 노출 기간 시작 시 중도 절단할 것인가? If a subject is in multiple cohorts, should time-at-risk be censored when the new time-at-risk starts to prevent overlap?**” 하는 옵션과 함께 “**표적 및 대조 코호트에 모두 포함된 피험자를 제거하겠는가? Remove subjects that are in both the target and comparator cohort?**” 옵션은 피험자가 표적과 대조 코호트 양쪽 모두에 포함되어 있을 때 어떻게 할지 정의한다. “**표적 및 대조 코호트에 모두 포함된 피험자를 제거하겠는가?**” 옵션에 대해서는 세 가지 선택 사항이 있다:

- “**Keep All**”은 양 코호트의 모든 환자를 보존한다는 뜻이다. 이 옵션은 환자와 결과 쌍 개수를 중복으로 셀 수 있다.
- “**Keep First**”은 한 환자가 양 코호트에 모두 들어있을 경우, 두 코호트 중 먼저 들어간 코호트의 환자만 인정한다는 뜻이다.

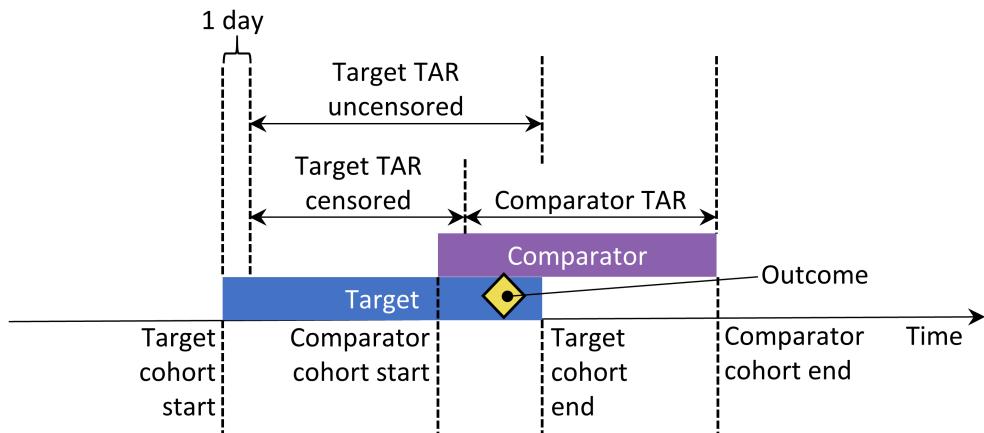


Figure 12.10: 두 코호트에 모두 포함된 피험자의 위험 노출 기간 TAR은 치료 시작 다음 날부터 시작하여 노출 끝에서 멈춘다고 가정한다.

- “Remove All”은 양 코호트에 모두 들어간 환자를 모두 제외하는 것이다.

“Keep all” 또는 “keep first” 옵션이 선택되면, 우리는 연구 대상자가 양쪽 코호트 모두에 속하는 시기를 절단하기를 바랄 수 있다. (그림 12.10) 기본적으로 위험 노출 기간은 코호트 시작일과 종료일을 기준으로 정의된다. 이번 예에서, 위험 노출 기간은 코호트 시작일 다음 날부터 시작되어 코호트 종료일에 끝난다. 절단하지 않는다면, 두 코호트의 위험 노출 기간이 겹칠 수 있다. 이 겹치는 동안 발생하는 모든 결과가 (그림과 같이) 두 번 계산되기 때문에 “keep all”을 선택하면 특히 문제가 된다. 만약 절단하기를 선택하면 첫 번째 코호트의 위험 노출 기간은 두 번째 코호트의 위험 노출 기간이 시작될 때 종료된다 (역자 주: 즉, Keep First를 선택하면 그림 12.10에서 Target TAR censored가 위험 노출 기간으로 설정됨).

최초의 결과 outcome 발생 이후 연속적으로 결과가 추가 발생하는 경우가 종종 있어서, 위험 노출 기간이 시작하기 전 결과가 발생한 피험자를 제거 remove subjects that have the outcome prior to the risk window start 할 수도 있다. 예를 들어, 누군가에게 심부전과 같은 만성 질병이 최초로 발생한 후, 두 번째 발생이 있을 수 있는데, 이는 심부전이 새로 다시 발생했다기보다는, 이전의 심부전이 완전히 치료되지 않은 상태를 의미할 가능성이 높다. 한편으로는 어떤 결과 outcome은 일시적일 수도 있다. 예를 들어 상부 호흡기 감염 upper respiratory infection과 같은 급성 질병이 한 환자에 여러 번 발생한다면, 이는 실제로 독립적인 질병이 시간 간격으로 두고 발생함을 의미할 수도 있다. 이전 결과를 확인할 때 며칠 전까지 검토해야 할지 how many days we should look back when identifying prior outcomes를 선택함으로써, 이전에 결과가 있는 사람을 제거하는 방법을 선택할 수 있다.

예제 연구에 대한 우리의 선택은 그림 12.11과 같다. 표적 및 대조 ‘코호트 정의’ 시 이미 첫 번째

Study Population

Study start date - a calendar date specifying the minimum date that a cohort index can appear (leave blank to use all time):

Study end date - a calendar date specifying the maximum date that a cohort index can appear (leave blank to use all time). **Important:** the study end date is also used to truncate risk windows, meaning no outcomes beyond the study end date will be considered.

Restrict the study to the period when both exposures are present in the data? (E.g. when both drugs are on the market)
 No ▾

Should only the first exposure per subject be included?
 No ▾

The minimum required continuous observation time (in days) prior to index date for a person to be included in the cohort.

Remove subjects that are in both the target and comparator cohort?
 Remove All ▾

If a subject is in multiple cohorts, should time-at-risk be censored when the new time-at-risk start to prevent overlap?
 No ▾

Remove subjects that have the outcome prior to the risk window start?
 Yes ▾

How many days should we look back when identifying prior outcomes?

If either the target or the comparator cohort is larger than this number it will be sampled to this size. (0 for this value indicates no maximum size)

Figure 12.11: Study population 설정.

노출로 한정하고 치료 개시 전에 관찰 시기가 필요하기 때문에 Estimation에서 이러한 기준을 다시 적용하지 않았다.

공변량 설정

여기서 사용할 공변량을 지정한다. 이러한 공변량은 일반적으로 성향 점수 모델에서 사용되지만, 결과 모델 (이 경우 콕스 비례위험모형 Cox proportional hazards model)에도 포함될 수 있다. **공변량 설정의 세부 사항 click to view details** 을 클릭하면, 사용할 공변량 셋을 선택할 수 있다. 하지만, 인구학적 정보, 모든 진단명, 약물, 시술, 검사 등에 대한 공변량으로 구성된 기본 집합을 그대로 사용하길 권장한다.

포함하거나 제외할 개념을 지정하여 공변량 셋을 수정할 수 있다. 이러한 설정은 비교 설정의 12.7.1 절에 있는 설정과 동일하다. 두 곳에서 이 설정이 가능한 이유는, 때때로 이번 예제처럼 비교하고자

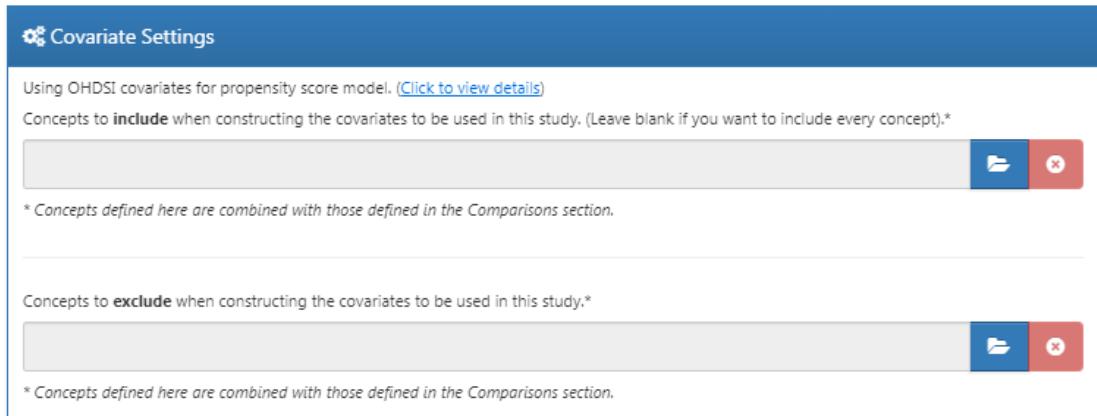


Figure 12.12: 공변량 설정.

하는 약물을 공변량에서 제외해야 하는 등 특별한 비교가 가능하게 하기 위함이다. 특별한 분석 설정을 이용해서 특별한 비교 분석을 수행하면 OHDSI 도구는 이러한 모음을 병합해서 적용한다.

그림 12.12는 이 연구에 대한 선택을 보여준다. 그림 12.9의 비교 설정에서 제외 개념에 하위 개념을 포함하도록 설정했다는 점에 주의하자.

위험 노출 기간

위험 노출 기간은 표적 및 대조 코호트의 시작일과 종료일을 기준으로 정의된다. 이 예제에서는, 치료 시작일을 코호트 시작일로, 약물 노출이 30일 이상 중지되면 코호트 종료일이 되도록 설정하였다. 코호트 시작 후 1일 (즉, 치료 시작 후 1일)을 위험 노출 기간의 시작으로 설정하였다. 치료 시작과 함께 발생한 결과가 이론상 치료에 의해 발생한 것이라고 믿기 어려울 때, 코호트 시작 이후에 위험 노출 기간이 시작하도록 설정한다.

코호트 종료일을 위험 노출 기간 종료일로 설정하여, 약물 노출이 중지된 시점으로 설정하였다. 예를 들어, 치료 종료 후에 발생한 event가 노출로 인한 것으로 판단될 경우, 위험 노출 기간의 종료일을 나중으로 설정할 수 있다. 극단적인 경우, 위험 노출 기간 종료를 코호트 종료일 후에 아주 나중(예를 들어 99999일)로 설정할 수 있다. 이는 관찰 종료까지 피험자를 추적 관찰하는 것을 의미한다. 이러한 연구 설계를 때로는 배정된대로 *intent-to-treat* 설계라고도 한다.

기준 날짜 이후 절단 또는 결과 발생 전까지의 **위험 노출 일 days at risk**가 0일인 환자는 분석할 정보가 없기 때문에, **최소 관찰 기간 minimum days at risk**은 보통 1일로 설정한다. 노출과 결과 발생에 대한 지연시간 latency이 알려져 있다면, 이러한 일수를 늘려 더 유익한 비율을 얻을 수도 있을 것이다. 이러한 설정은 무작위 임상 시험과 유사한 연구 설계를 위해서도 사용할 수 있다(예를 들어, 임상시험에 참여한 피험자가 최소한 N일 동안은 관찰되었다고 할 때).

Time At Risk

Define the time-at-risk window start, relative to target/comparator cohort entry:

days from

Define the time-at-risk window end:

days from

The minimum number of days at risk?

Figure 12.13: 위험 노출 기간 TAR 설정.



코호트 연구를 설계할 때 지켜야 할 황금률 golden rule은 연구 집단을 정의할 때 코호트 시작 일 이후의 정보를 절대로 사용하지 않아야 한다는 것이며, 지키지 않을 경우 비뚤림이 발생할 수 있다. 예를 들어, 모든 피험자에게 적어도 1년의 위험 노출 기간이 있어야 한다고 연구 집단을 정의할 경우, 연구집단에 포함된 환자는 치료를 잘 견디는 피험자로 분석을 제한했다는 뜻이 된다 (역자 주: 부작용이나 여러 가지 이유로 치료를 1년간 유지하지 못하는 환자는 모두 연구 집단에서 탈락한다). 따라서, 이러한 설정은 세심한 주의를 기울여 사용해야 한다.

성향 점수 보정

극단적인 성향 점수를 갖는 피험자를 제거하여, 연구 대상을 잘라낼 trimming 수 있다. 상위 또는 하위 비율을 제거하도록 선택하거나, 선호도 점수 preference score가 지정된 범위를 벗어나는 피험자를 제거할 수 있다. 코호트 트리밍은 관측치를 제거하여 통계적 검정력을 감소시키기 때문에 일반적으로 권장되지는 않는다. IPTW를 사용할 때처럼 경우에 따라서는 트리밍을 하는 것이 바람직할 수 있다.

트리밍에 추가하여, 또는 트리밍 대신에 성향 점수를 이용해 계층화 stratification하거나 짹짓기 matching하도록 선택할 수 있다. 계층화할 때, 계층의 수 number of strata를 지정하고, 표적군, 대조군, 또는 전체 연구 집단을 기준으로 계층을 선택할지 여부를 지정해야 한다. 성향 점수 짹짓기 시, 표적군의 각 피험자와 일치시키기 위한 대조군에서의 최대 피험자 수 짹짓기 비율을 지정하여야 한다. 일반적인 값은 일대일 매칭 one-on-one matching의 경우 1이고, 일대다 매칭 variable-ratio matching의 경우 다수 (예를 들어 100) 이다. 또한, 매칭을 허용하는 성향 점수 사이의 최대 허용 차이를 뜻하는 캘리퍼 caliper를 지정해야 한다: 캘리퍼는 다음과 같이 서로 다른 캘리퍼 척도 caliper scales로 정의할 수 있다:

- **성향 점수 척도 propensity score scale:** 성향 점수 자체
- **표준화 척도 standardized scale:** 성향 점수 분포의 표준편차

- 표준화 로짓 척도 **standardized logit scale**: 성향 점수를 보다 정규분포로 만들기 위해 로그 변환한 성향 점수 분포의 표준편차

잘 모르겠다면, 기본값을 사용하거나, 이 주제에 대한 Austin (2011) 의 연구를 참고하기를 권장한다.

대규모 성향 점수 모델 large-scale propensity model을 최적화하는 것은 많은 컴퓨팅 자원을 요구할 수 있어서, 계산 시 샘플링한 자료를 이용하고자 할 수 있다. 기본 설정상, 표적 및 대조 코호트의 최대 크기는 250,000으로 설정되어 있다. 대부분의 연구에서 코호트의 전체 피험자 수가 이 한도에 도달하지 못할 것이다. 이보다 많은 데이터를 이용한다고 해서 더 나은 모델로 이어질 가능성은 희박하다. 비록 샘플링한 데이터를 이용해 성향 점수 모델을 적합하더라도 전체 집단에 대한 성향 점수는 여전히 계산된다는 점에 유의하자.

각 공변량이 치료배정과 상관성이 있는지 검사하겠습니까? Test each covariate for correlation with the target assignment? 을 '예 yes'로 설정하면, 만약 어떤 공변량이 치료 배정과 비정상적으로 높은 (양 또는 음의) 상관관계가 있으면 오류를 발생시키고 프로세스가 중단된다. 이것은 대규모 성향 점수 모델 계산이 완전히 끝날 때까지 기다리는 것을 방지할 수 있다. 매우 높은 단변량 상관관계를 발견하면 공변량을 검토하여 치료 배정과 상관관계가 높은 이유와 이를 제거해야 하는지를 결정할 수 있다.

모델 적합시에 정규화를 사용하시겠습니까? Use regularization when fitting the model? 매우 많은 공변량 (일반적으로 만 개 이상)이 성향 점수 모델 계산 시 사용된다. 이러한 대규모 모델을 적합하기 위해서는 정규화 regularization가 필요하다. 만약 수동으로 몇 개의 공변량만 사용된다면, 정규화를 사용하지 않아도 모델을 적합할 수 있다.

그림 12.14는 이 연구에 대한 우리의 선택을 보여준다. 최대 짹짓기 비율을 100으로 설정하여서 일-대-다 비율 짹짓기를 선택하였다.

결과 모델 설정

먼저, 표적 코호트와 대조 코호트 간 결과의 상대 위험도 relative risk를 추정하기 위해 사용할 통계 모델을 명시할 필요가 있다. 12.1절에서 간략히 논의했던 것처럼, 콕스 Cox, 포아송 Poisson 및 로지스틱 회귀분석 중에서 선택할 수 있다. 예제에서는 콕스 비례 위험 모형 Cox proportional hazards model을 사용하는데, 이 모델은 중도절단을 고려하여 첫 번째 사건까지의 시간 time to first event 을 고려한다. 다음으로, **총화에 조건부 회귀분석을 사용할지 whether the regression should be conditioned on the strata**를 명시할 필요가 있다. 쉽게 말하자면, 조건부 conditioning는 각 층 strata 내에서 추정치를 계산한 다음, 여러 층의 추정치를 결합한 것으로 생각하면 된다. One-to-one PS matching에서는 이러한 과정이 불필요할 것이며, 통계적 검정력 감소를 유발할 것이다. conditioning는 총화 stratification 또는 일-대-다 비율 짹짓기 시행시 필요하다.(역자 주: 다시 한

Propensity Score Adjustment

How do you want to trim your cohorts based on the propensity score distribution?

None ▾

Do you want to perform matching or stratification?

Match on propensity score ▾

What is the maximum number of persons in the comparator arm to be matched to each person in the target arm within the defined caliper? (0 = means no maximum - all comparators will be assigned to a target person):

100 ▾

What is the caliper for matching:

0.2

What is the caliper scale:

Standardized Logit ▾

What is the maximum number of people to include in the propensity score model when fitting? Setting this number to 0 means no down-sampling will be applied:

250000 ▾

Test each covariate for correlation with the target assignment? If any covariate has an unusually high correlation (either positive or negative), this will throw an error.

Yes ▾

Use regularization when fitting the propensity model?

Yes ▾

Control Settings ▾ Prior ▾

Figure 12.14: 성향 점수 보정 설정.

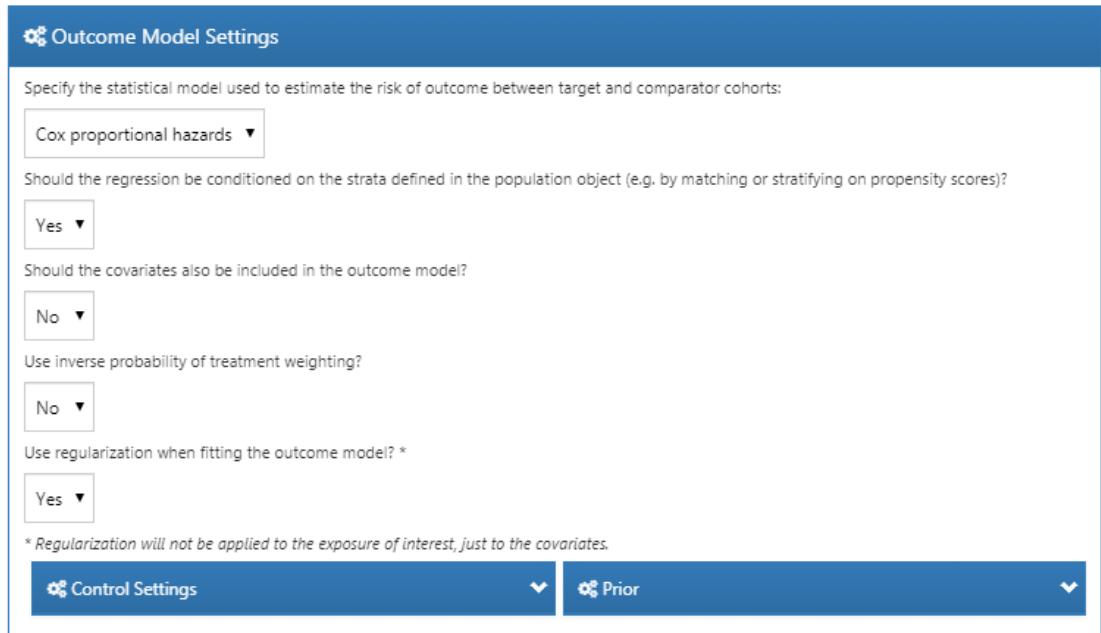


Figure 12.15: 결과 모델 설정

번 말하자면, 일대일 PS 매칭 시에는 conditioned on the strata 조건을 사용하지 말고, 일대다 매칭 또는 PS stratification에는 conditioned on the strata을 사용하기 바란다.)

분석을 보정하기 위해 결과 모델 outcome model에 공변량을 추가할 수도 있다. 이것을 성향 점수 모델 사용에 추가하거나 성향 점수 짹짓기 대신에 수행할 수 있다. 하지만 보통 성향 점수 모델을 적합하기에는 충분한 수의 데이터가 있지만, 결과 모델을 적합하기에는 결과가 발생한 피험자가 적어 데이터가 모자라는 경우가 많다. 그래서 공변량을 결과 모델에 추가하지 말고, 결과 모델을 가급적 간단하게 유지하기를 권장한다.

성향 점수를 이용해 층화하거나 매칭하는 대신에 역 확률 치료 가중치 inverse probability of treatment weighting, IPTW를 사용할 수도 있다.

만약 모든 공변량을 결과 모델에 추가한다면, 공변량이 매우 많기 때문에 결과 모델 적합 시 정규화를 이용하는 것이 합리적일 것이다. 편향이 없는 추정을 위해서 치료 변수 자체에는 정규화가 적용되지 않음에 유의하자.

그림 12.15는 이 연구에 대한 선택을 보여준다. 일-대-다 비율 짹짓기를 사용하기 때문에, 층화에 조건부 회귀분석 condition the regression on the strata (다른 말로 matched sets)를 해야 함에 주목하자.

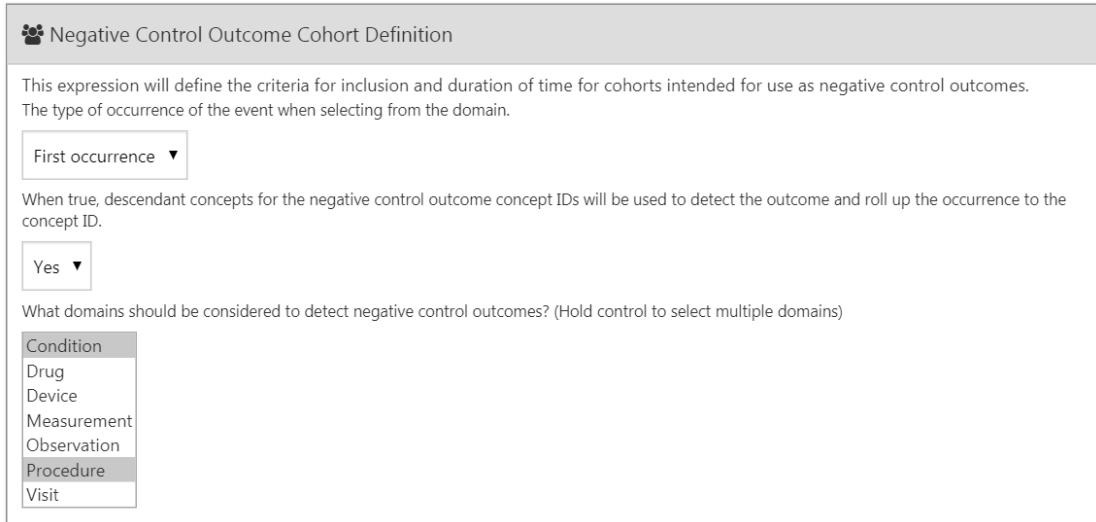


Figure 12.16: 음성 대조 결과 코호트 정의 설정.

12.7.3 평가 설정

18장에서 기술한 바와 같이, 음성 대조군 및 양성 대조군은 연구자가 세팅한 분석방법론에 대한 운영 특성을 평가하고 경험적 보정을 수행하기 위하여 추가하여야 한다.

음성 대조군 결과 코호트 정의

12.7.1절에서 우리는 음성 대조군을 지정하는 concept set을 선택했다. concept을 지정하는 것뿐 아니라, 분석을 위해서는 concept을 기반으로 코호트를 생성하는 프로세스가 필요하다. ATLAS는 세 가지 선택 사항을 가진 표준 프로세스를 제공한다. 첫 번째 선택은 모든 발생을 사용할지 또는 concept의 첫 번째 발생 만을 사용할지 여부이다. 두 번째 선택은 하위 개념의 발생을 포함할지 여부를 결정한다. 하위 개념의 발생을 포함하면, 예를 들어 하위 개념 “내성 발톱 ingrown nail of foot”의 발생은 상위 개념 “내성 손톱 ingrown nail”的 발생으로 간주한다. 세 번째 선택사항은 개념을 찾을 때 고려할 도메인을 지정한다.

양성 대조군 합성

음성 대조군 외에도 양성 대조군도 포함할 수 있는데, 양성 대조군은 알려진 효과 크기 effect size와 함께 인과적 영향 causal effect이 존재하는 것으로 보이는 노출-결과 쌍 exposure-outcome pair을 뜻한다. 여러 가지 이유로 실제 양성 대조군 설정은 문제가 있기 때문에, 대신 18장에서 설명한 대로 음성 대조군을 기반으로 합성된 양성 대조군을 사용한다. **양성 대조군 합성** 여부를 선택할 수 있다. 만약 “예”를 선택하면, 반드시 **모형 유형 model type**을 선택해야 하는데, 현재는 “Poisson”과 “survival”을 지원하고 있다. 이 예제에서는 생존 (콕스) 분석을 시행하기 때문에, 양성 대조

The screenshot shows the 'Positive Control Synthesis' configuration interface. It includes the following settings:

- Should we perform positive control synthesis? (to calibrate confidence intervals)**: Yes
- Model Type:** Survival
- Using OHDSI covariates for model.** (Click to view details)
- Define the time-at-risk window start, relative to target/comparator cohort entry:** 1 days from cohort start date
- Define the time-at-risk window end:** 0 days from cohort end date
- The minimum required continuous observation time (in days) prior to exposure:** 365
- Should only the first exposure per subject be included?**: Yes
- Should only the first outcome per person be considered when modeling the outcome?**: Yes
- Remove people with prior outcomes?**: Yes
- Advanced Settings start here**
- Additional Settings**

Figure 12.17: 음성 대조 결과 코호트 정의 설정.

군 합성 시에도 “survival”을 선택하도록 하자. 양성 대조군 합성 시 가급적 분석 추정 설정에서 사용된 값을 비슷하게 사용하였다 (minimum required continuous observation prior to exposure, should only the first exposure be included, should only the first outcome be included, remove people with prior outcomes). 그림 12.15를 통해 양성 대조군 합성을 위한 설정을 참조할 수 있다.

12.7.4 연구 패키지 실행

이제 연구를 완전히 정의했으므로, 실행 가능한 R 패키지로 추출할 수 있다. 이 패키지에는 CDM 데이터가 있는 사이트에서 연구를 실행하는데 필요한 모든 것이 들어있다. 여기에는 분석을 실행하기 위한 R 코드뿐만 아니라 표적, 대조 및 결과 코호트, 음성 대조군을 정의하기 위한 개념 군과 코호트 생성 프로세스가 포함된다. 패키지를 생성하기 전에 연구를 저장한 다음, Utilities 탭을 클릭하면, 수행될 일련의 분석을 검토할 수 있다. 앞서 언급했듯이, 개개의 비교와 분석 설정의 조합은

각각의 분석 결과를 생성할 것이다. 이번 예시에서는 성향 점수 매칭을 사용하여 두 가지 분석을 지정하였다: 급성 심근경색 위험에 대한 ACEi 대 THZ 비교, 혈관 부종에 대한 ACEi 대 THZ 비교.

“Download”를 클릭하여 zip 파일을 다운로드하기 위해, 패키지의 이름을 입력해야 한다. zip 파일에는 R 패키지의 일반적인 필수 풀더 구조와 함께, R 패키지가 포함되어 있다. (Wickham, 2015) 이 패키지를 사용하려면 R Studio를 사용하는 것이 좋다. R Studio를 로컬로 실행하는 경우 파일의 압축을 푼 다음, .Rproj 파일을 더블 클릭하여 R Studio에서 연다. R Studio 서버에서 R Studio를 실행하는 경우,  Upload 버튼을 클릭하여 파일을 업로드하고 압축을 해제한 다음, .Rproj 파일을 클릭하여 프로젝트를 연다.

R Studio에서 프로젝트를 열면 README 파일을 열고 파일의 지침을 따라 할 수 있다. 모든 파일 경로를 시스템의 기존 경로로 변경하는 것을 잊지 말자.

연구를 진행할 때 나타날 수 있는 흔한 오류 메시지는 “공변량과 치료가 높은 상관관계를 보임 High correlation between covariate(s) and treatment detected”이다. 이는 성향 모델을 적용했을 때 일부 공변량이 노출과 높은 상관관계가 있음을 나타낸다. 오류 메시지에 언급된 공변량을 검토하고 적절한 경우 해당 공변량을 공변량 집합에서 제외하면 된다. (12.1.2절 참조)

12.8 R을 사용한 연구 구현하기

ATLAS를 사용하여 연구를 실행하는 R 코드를 작성하는 대신 R 코드를 직접 작성할 수도 있다. 이는 ATLAS를 이용하는 것보다, 훨씬 큰 유연성을 제공할 수 있다. 예를 들어 사용자 정의 공변량 또는 선형 결과 모델을 사용하려면 사용자 정의 R 코드를 작성하고 이를 OHDSI R 패키지가 제공하는 기능과 결합해야 한다.

예제 연구에서, 우리는 연구를 수행하기 위해 CohortMethod 패키지를 사용할 것이다. CohortMethod는 CDM 데이터베이스에서 필요한 데이터를 추출하고 성향 점수 모델에 대규모의 공변량 집합을 사용할 수 있다. 다음 예시에서는 혈관 부종만을 결과로 사용할 것이다. 12.8.6절에서는 이것이 어떻게 급성 심근경색과 음성 대조군 결과를 포함하도록 확장될 수 있는지 기술한다.

12.8.1 코호트 생성

먼저 표적, 대조 및 결과 코호트를 생성 Instantiation해야 한다. 코호트 실체화 방법은 10장에서 자세히 기술되어 있다. 부록에서 표적 (부록 B.2), 대조 (부록 B.5) 및 결과 (부록 B.4) 코호트의 정의 전체를 제공한다. ACEi, THZ 및 혈관 부종 코호트가 scratch.my_cohorts라고 명명된 테이블에서 코호트 정의 ID 1, 2, 3을 가지고 함께 실체화되었다고 가정한다.

12.8.2 데이터 추출

먼저 R에게 서버에 연결하는 방법을 알려줘야 한다. CohortMethod `createConnectionDetails`라는 함수를 제공하는 DatabaseConnector 패키지를 이용한다. 다양한 데이터베이스 관리 시스템에 필요한 설정에 대해 알아보기 위하여 `?createConnectionDetails`를 입력해보자. 예를 들어, 아래 코드를 사용하여 PostgreSQL 데이터베이스에 연결할 수 있다.

```
library(CohortMethod)

connDetails <- createConnectionDetails(dbms = "postgresql",
                                         server = "localhost/ohdsi",
                                         user = "joe",
                                         password = "supersecret")

cdmDbSchema <- "my_cdm_data"
cohortDbSchema <- "scratch"
cohortTable <- "mycohorts"
cdmVersion <- "5"
```

마지막 네 줄은 `cdmDbSchema`, `cohortDbSchema` 및 `cohortTable` 변수와 CDM 버전을 정의한다. 이를 이용하여 이후에 CDM 데이터가 존재하는 위치, 연구용 코호트가 생성된 위치, 그리고 사용된 CDM 버전 정보를 R에 전달한다. Microsoft SQL Server의 경우 데이터베이스 스키마는 데이터베이스와 스키마 schema를 모두 지정해야 한다 (예를 들어 `cdmDbSchema <- "my_cdm_data.dbo"`).

이제 CohortMethod를 이용해 코호트를 추출하고, 공변량을 구성하며, 분석에 필요한 모든 데이터를 추출할 수 있다.

```
# target and comparator ingredient concepts:
aceI <- c(1335471, 1340128, 1341927, 1363749, 1308216, 1310756, 1373225,
         1331235, 1334456, 1342439)
thz <- c(1395058, 974166, 978555, 907013)

# Define which types of covariates must be constructed:
cs <- createDefaultCovariateSettings(excludedCovariateConceptIds = c(aceI,
                                                                     thz),
                                         addDescendantsToExclude = TRUE)

#Load data:
```

```

cmData <- getDbCohortMethodData(connectionDetails = connectionDetails,
                                  cdmDatabaseSchema = cdmDatabaseSchema,
                                  oracleTempSchema = NULL,
                                  targetId = 1,
                                  comparatorId = 2,
                                  outcomeIds = 3,
                                  studyStartDate = "",
                                  studyEndDate = "",
                                  exposureDatabaseSchema = cohortDbSchema,
                                  exposureTable = cohortTable,
                                  outcomeDatabaseSchema = cohortDbSchema,
                                  outcomeTable = cohortTable,
                                  cdmVersion = cdmVersion,
                                  firstExposureOnly = FALSE,
                                  removeDuplicateSubjects = FALSE,
                                  restrictToCommonPeriod = FALSE,
                                  washoutPeriod = 0,
                                  covariateSettings = cs)

cmData

```

```

## CohortMethodData object
##
## Treatment concept ID: 1
## Comparator concept ID: 2
## Outcome concept ID(s): 3

```

많은 파라미터가 있지만, CohortMethod 매뉴얼에 모두 설명되어 있다. `createDefaultCovariateSettings` 함수는 FeatureExtraction 패키지에 설명되어 있다. 간단히 말해, 코호트를 포함하는 테이블에 함수를 지정하고 해당 테이블의 cohort definition ID가 표적, 대조 및 결과 코호트를 식별하도록 지정한다. index data 당일 혹은 이전에 발견된 모든 진단명, 약물 노출, 시술 기록에 대한 공변량을 포함하여 공변량의 기본 모음을 구성하도록 지시한다. 12.1절에서 언급했듯이, 공변량 집합에서 표적 및 대조 치료를 배제하여야 하며, 이 예제에서는 두 가지의 약물군에 해당하는 성분명 ingredient을 나열하여 이를 달성한다. 또한, FeatureExtraction에 모든 하위 개념을 배제하도록 지시하여 나열된 성분을 포함하는 모든 약물 노출을 공변량에서 제외한다.

코호트, 결과 및 공변량에 대한 모든 데이터는 서버에서 추출되어 `cohortMethodData` object에

저장된다. 이러한 object는 ff 패키지를 사용하여 8.4.2절에서 언급한 것처럼 데이터가 크더라도, R이 메모리를 모두 소모하지 않도록 보장한다.

일반 함수 `summary()`를 사용하여 추출한 데이터에 대한 추가 정보를 볼 수 있다:

```
summary(cmData)
```

```
## CohortMethodData object summary
##
## Treatment concept ID: 1
## Comparator concept ID: 2
## Outcome concept ID(s): 3
##
## Treated persons: 67166
## Comparator persons: 35333
##
## Outcome counts:
##           Event count Person count
## 3                 980        891
##
## Covariates:
## Number of covariates: 58349
## Number of non-zero covariate values: 24484665
```

`cohortMethodData` 파일을 만들면 상당한 시간이 걸릴 수 있으며 향후 세션을 위해 저장하는 것이 좋다. `cohortMethodData`가 `ff`를 사용하므로 R의 일반 저장 기능을 사용할 수 없다. 대신에, `saveCohortMethodData()` 함수를 사용하도록 한다.

```
saveCohortMethodData(cmData, "AceiVsThzForAngioedema")
```

`loadCohortMethodData()` 함수를 사용하여 향후 세션에서 데이터를 로드할 수 있다.

새로운 사용자 정의하기

일반적으로 **새로운 사용자 new user**는 해당 약물 (표적군 또는 대조군 내에서)의 최초 사용으로 정의되며, 최초 사용을 보장하는 확률을 높이기 위해 최초 사용 이전의 최소 기간을 뜻하는 휴약기

간 washout period를 사용할 수 있다. CohortMethod 패키지를 사용할 때 다음 세 가지 방법으로 새로운 사용자를 정의할 수 있다.

1. 코호트 정의 시 지정
2. `getDbCohortMethodData` 함수를 사용하여 코호트를 로딩할 때, `firstExposureOnly`, `removeDuplicateSubjects`, `restrictToCommonPeriod`, `washoutPeriod` 전달 인자 argument를 사용하여 지정
3. 연구 집단 정의 시 `createStudyPopulation` 함수를 이용하여 지정

첫 번째 방법의 장점은 입력 코호트 `input cohort`가 이미 `CohortMethod` 패키지 밖에서 완전히 정의되어 있고, 외부 코호트 특성화 도구가 이 분석에 사용된 것과 동일한 코호트에서 사용될 수 있다는 것이다. 두 번째, 세 번째 방법의 장점은 CDM에서 DRUG_ERAS 테이블을 직접 사용할 수 있는 등, 새로운 사용자를 정의하는 데 생기는 문제를 줄여준다는 것이다. 최초 사용에 대한 데이터만 가져올 것이기 때문에 두 번째 방법이 세 번째 방법보다 더 효율적이다. 세 번째 방법이 컴퓨터 메모리 측면에서 덜 효율적이긴 하지만, 원래 코호트를 연구 대상군과 비교할 수 있다.

12.8.3 연구군 정의

일반적으로, 표적, 대조 코호트와 결과 코호트는 서로 독립적으로 정의된다. 효과 크기 추정치를 생성하려면 노출 전에 결과가 있는 피험자는 제거하고 정의한 위험 노출 기간 risk window에 발생하는 결과 만을 고려하는 등의 방법을 추가해야 한다. 이를 위하여 `createStudyPopulation` 함수를 사용할 수 있다.

```
studyPop <- createStudyPopulation(cohortMethodData = cmData,
                                    outcomeId = 3,
                                    firstExposureOnly = FALSE,
                                    restrictToCommonPeriod = FALSE,
                                    washoutPeriod = 0,
                                    removeDuplicateSubjects = "remove all",
                                    removeSubjectsWithPriorOutcome = TRUE,
                                    minDaysAtRisk = 1,
                                    riskWindowStart = 1,
                                    startAnchor = "cohort start",
                                    riskWindowEnd = 0,
                                    endAnchor = "cohort end")
```

코호트 정의에서 이미 이러한 기준을 적용했기 때문에 `firstExposureOnly`와 `removeDuplicateSubjects`

를 FALSE로, `washoutPeriod`를 0으로 설정하였다. 사용할 결과 ID를 지정하고, 위험노출 기간 risk window 시작일 전에 결과가 발생한 사람은 제거할 것이다. 위험노출 기간은 코호트 시작일 다음 날부터 시작하는 것으로 정의하고 (`riskWindowStart = 1` 및 `startAnchor = "cohort start"`), 코호트 노출이 끝날 때 종료되도록 설정했다 (`riskWindowEnd = 0` and `endAnchor = "cohort end"`). 이것은 코호트 정의에서 치료 노출 종료로써 정의되었었다. 위험노출 기간은 관찰 종료 또는 연구 종료일에 자동으로 결단됨에 유의하자. 또한 위험노출 기간이 0일인 피험자도 제거했다. 연구 대상자에 남아 있는 사람의 수를 보려면, `getAttritionTable` 함수를 사용하면 된다.

```
getAttritionTable(studyPop)
```

	##	description	targetPersons	comparatorPersons	...
## 1		Original cohorts	67212	35379	...
## 2	Removed subs in both cohorts		67166	35333	...
## 3		No prior outcome	67061	35238	...
## 4		Have at least 1 days at risk	66780	35086	...

12.8.4 성향 점수

`getDbcohortMethodData()` 함수로 생성된 공변량을 사용하여 성향 점수 모델을 적합할 수 있으며, 피험자별 성향 점수를 계산할 수 있다.

```
ps <- createPs(cohortMethodData = cmData, population = studyPop)
```

`createPs` 함수는 대규모 정규화 로지스틱 회귀분석 large-scale regularized logistic regression을 적합화하기 위해 Cyclops 패키지를 사용한다. 성향 점수 모델을 적합화하기 위해, Cyclops는 prior의 분산을 지정하는 하이퍼-파라미터 값을 알아야 한다. 기본값으로 Cyclops는 최적의 하이퍼-파라미터를 추정하기 위해 교차 검증 cross-validation을 사용할 것이다. 다만, 이 작업은 오랜 시간이 걸릴 수 있음을 알아 두어야 한다. `createPs` 함수의 `prior` 및 `control`의 매개변수를 사용하여 병렬 처리를 사용하여 교차 유효성 검사 속도를 높이는 등 Cyclops의 동작을 지정할 수 있다.

예제에서는 성향점수 기반의 일-대-다 비율 짹짓기를 수행하였다:

```
matchedPop <- matchOnPs(population = ps, caliper = 0.2,
                           caliperScale = "standardized logit", maxRatio = 100)
```

위와 같은 설정 대신에, `trimByPs`, `trimByPsToEquipoise` 또는 `stratifyByPs` 함수에서 성향 점수를 사용할 수도 있다.

12.8.5 결과 모델

결과 모델은 결과와 어떠한 변수가 관련이 있는지 설명하는 모델이다. 엄격한 가정하에서 치료 변수에 대한 계수 coefficient는 인과적 영향 causal effect으로 해석될 수 있다. 이 경우, 짹짓기 된 군에 대해 총화 조건부 콕스 비례 위험 모델 Cox proportional hazards model conditioned (stratified) on the matched set을 이용하였다.

```
outcomeModel <- fitOutcomeModel(population = matchedPop,
                                    modelType = "cox",
                                    stratified = TRUE)

outcomeModel

## Model type: cox
## Stratified: TRUE
## Use covariates: FALSE
## Use inverse probability of treatment weighting: FALSE
## Status: OK
##
##           Estimate lower .95 upper .95   logRr seLogRr
## treatment    4.3203    2.4531    8.0771 1.4633   0.304
```

12.8.6 다중 분석 실행하기

음성 대조군을 포함하여 다수의 결과에 대해 하나 이상의 분석을 수행하기 원할 수 있다. Cohort-Method는 이러한 연구를 효율적으로 수행하는 기능을 제공한다. 이것은 다중 분석 실행에 대한 패키지 설명(package vignette on running multiple analyses)에 자세히 설명되어 있다. 간단히 말해서 먼저 필요한 코호트가 모두 생성되어 있다면, 분석하고자 하는 모든 표적-대조-결과 조합을 미리 지정하여 한꺼번에 실행 할 수 있다.

```
# Outcomes of interest:
ois <- c(3, 4) # Angioedema, AMI

# Negative controls:
```

```

ncs <- c(434165, 436409, 199192, 4088290, 4092879, 44783954, 75911, 137951, 77965,
       376707, 4103640, 73241, 133655, 73560, 434327, 4213540, 140842, 81378,
       432303, 4201390, 46269889, 134438, 78619, 201606, 76786, 4115402,
       45757370, 433111, 433527, 4170770, 4092896, 259995, 40481632, 4166231,
       433577, 4231770, 440329, 4012570, 4012934, 441788, 4201717, 374375,
       4344500, 139099, 444132, 196168, 432593, 434203, 438329, 195873, 4083487,
       4103703, 4209423, 377572, 40480893, 136368, 140648, 438130, 4091513,
       4202045, 373478, 46286594, 439790, 81634, 380706, 141932, 36713918,
       443172, 81151, 72748, 378427, 437264, 194083, 140641, 440193, 4115367)

tcos <- createTargetComparatorOutcomes(targetId = 1,
                                         comparatorId = 2,
                                         outcomeIds = c(ois, ncs))

tcosList <- list(tcos)

```

다음으로, 하나의 결과를 분석하는 이번 예제에서 전에 설명한 다양한 함수를 호출하기 위해 어떤 조절인자 argument를 사용해야 하는가를 지정한다.

```

aceI <- c(1335471, 1340128, 1341927, 1363749, 1308216, 1310756, 1373225,
         1331235, 1334456, 1342439)

thz <- c(1395058, 974166, 978555, 907013)

cs <- createDefaultCovariateSettings(excludedCovariateConceptIds = c(aceI,
                                                                     thz),
                                         addDescendantsToExclude = TRUE)

cmdArgs <- createGetDbCohortMethodDataArgs(
  studyStartDate = "",
  studyEndDate = "",
  firstExposureOnly = FALSE,
  removeDuplicateSubjects = FALSE,
  restrictToCommonPeriod = FALSE,
  washoutPeriod = 0,
  covariateSettings = cs)

```

```

spArgs <- createCreateStudyPopulationArgs(
  firstExposureOnly = FALSE,
  restrictToCommonPeriod = FALSE,
  washoutPeriod = 0,
  removeDuplicateSubjects = "remove all",
  removeSubjectsWithPriorOutcome = TRUE,
  minDaysAtRisk = 1,
  startAnchor = "cohort start",
  addExposureDaysToStart = FALSE,
  endAnchor = "cohort end",
  addExposureDaysToEnd = TRUE)

psArgs <- createCreatePsArgs()

matchArgs <- createMatchOnPsArgs(
  caliper = 0.2,
  caliperScale = "standardized logit",
  maxRatio = 100)

fomArgs <- createFitOutcomeModelArgs(
  modelType = "cox",
  stratified = TRUE)

```

그럼 다음 이를 하나의 분석 설정 object로 결합하는데, 이것은 고유 분석 ID(unique analysis ID)와 몇 가지 설명을 제공한다. 하나 이상의 분석 설정 object를 하나의 list로 결합할 수 있다.

```

cmAnalysis <- createCmAnalysis(
  analysisId = 1,
  description = "Propensity score matching",
  getDbCohortMethodDataArgs = cmdArgs,
  createStudyPopArgs = spArgs,
  createPs = TRUE,
  createPsArgs = psArgs,
  matchOnPs = TRUE,
  matchOnPsArgs = matchArgs
  fitOutcomeModel = TRUE,

```

```
fitOutcomeModelArgs = fomArgs)

cmAnalysisList <- list(cmAnalysis)
```

이제 모든 비교 및 분석 설정을 포함하여 연구를 실행할 수 있다.

```
result <- runCmAnalyses(connectionDetails = connectionDetails,
                         cdmDatabaseSchema = cdmDatabaseSchema,
                         exposureDatabaseSchema = cohortDbSchema,
                         exposureTable = cohortTable,
                         outcomeDatabaseSchema = cohortDbSchema,
                         outcomeTable = cohortTable,
                         cdmVersion = cdmVersion,
                         outputFolder = outputFolder,
                         cmAnalysisList = cmAnalysisList,
                         targetComparatorOutcomesList = tcosList)
```

`result` object에는 작성된 모든 인공물 artifact에 대한 참조가 들어 있다. 예를 들어, 급성 심근경색의 결과 모델을 추출할 수 있다.

```
omFile <- result$outcomeModelFile[result$targetId == 1 &
                                      result$comparatorId == 2 &
                                      result$outcomeId == 4 &
                                      result$analysisId == 1]

outcomeModel <- readRDS(file.path(outputFolder, omFile))

outcomeModel

## Model type: cox
## Stratified: TRUE
## Use covariates: FALSE
## Use inverse probability of treatment weighting: FALSE
## Status: OK
##
##           Estimate lower .95 upper .95    logRr seLogRr
## treatment   1.1338     0.5921    2.1765  0.1256    0.332
```

또한 하나의 명령으로 모든 결과에 대한 효과 크기 추정치를 검색할 수 있다:

```
summ <- summarizeAnalyses(result, outputFolder = outputFolder)
head(summ)
```

	analysisId	targetId	comparatorId	outcomeId	rr	...
## 1	1	1	2	72748	0.9734698	...
## 2	1	1	2	73241	0.7067981	...
## 3	1	1	2	73560	1.0623951	...
## 4	1	1	2	75911	0.9952184	...
## 5	1	1	2	76786	1.0861746	...
## 6	1	1	2	77965	1.1439772	...

12.9 연구 결과물

결과물로 나오는 추정치는 몇 가지 가정이 충족된 경우에만 유효하다. 유효성을 검증하기 위하여 다양한 진단 기준을 사용할 것이다. 이것은 ATLAS로 생성된 R 패키지를 이용한 결과에 자동으로 포함되어 있으며, 특정 R 함수를 활용하여 즉석에서 생성할 수도 있다.

12.9.1 성향 점수 및 모델

성향 점수 기반의 짹짓기 이후 면지 표적 코호트와 대조 코호트가 어느 정도 비슷한지를 평가할 필요가 있다. 이를 위해 성향 점수 모델에 대한 Area Under the Receiver Operator Curve(AUC) 통계값을 계산할 수 있다. AUC 1은 기저 공변량에 근거해 치료배정이 완전히 예측 가능하다는 것을 나타내므로, 두 군은 비교할 수 없다. `computePsAuc` 함수를 사용하여 AUC를 계산할 수 있는데, 우리의 예제에서는 0.79이다. `plotPs` 함수를 사용하여 그림 12.18과 같이 선호 점수 분포 preference score distribution를 생성할 수도 있다. 많은 피험자에 대해 그들이 받을 치료가 예측 가능했다는 것을 알 수지만 또한 많은 수의 중첩이 있다. 이는 조정을 통해 비교 가능한 군을 선택할 수 있음을 나타낸다.

일반적으로 성향 점수 모델 자체를 검사하는 것이 좋으며, 특히 모델이 매우 예측적일 경우에는 더욱 그렇다. 그렇게 하면 어떤 변수가 가장 예측적인지를 알 수 있다. 표 12.7은 성향 모델에서 상위 예측 변수를 보여준다. 변수가 너무 예측적일 경우, CohortMethod 패키지는 이미 완벽하게 예측된 모델을 적합하려고 시도하기보단 유용한 정보를 주는 에러를 발생시킬 것이다.

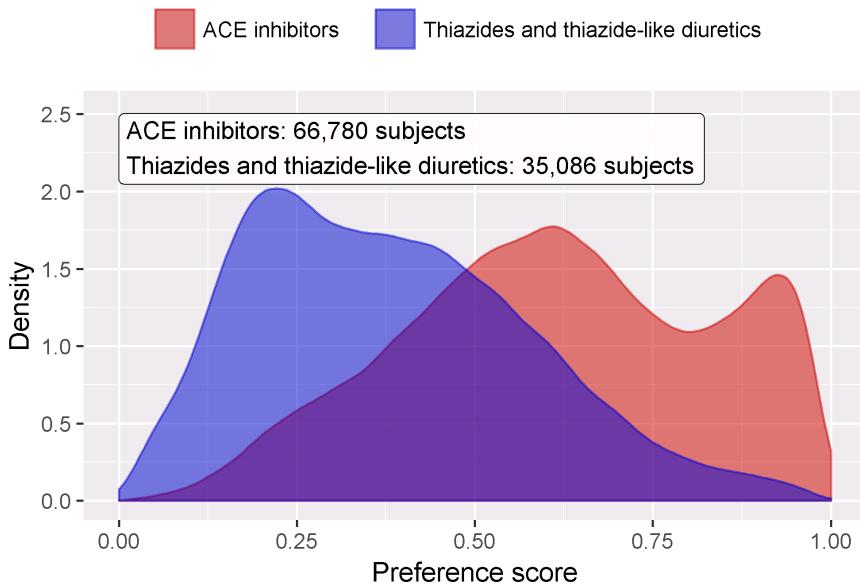


Figure 12.18: 성향 점수 분포.

Table 12.7: ACEi와 THZ 비교 성향점수모델에서 상위 10개의 예측인자. 베타 beta (기울기) 가 양수인 것은 해당 공변량이 있을 경우 표적 치료 target treatment 군에 배정될 확률이 높다는 것을 의미한다. “(Intercept)” 는 로지스틱 회귀 모델의 y 절편값을 의미한다.

Beta	공변량
-1.42	condition_era group during day -30 through 0 days relative to index: Edema
-1.11	drug_era group during day 0 through 0 days relative to index: Potassium Chloride
0.68	age group: 05-09
0.64	measurement during day -365 through 0 days relative to index: Renin
0.63	condition_era group during day -30 through 0 days relative to index: Urticaria
0.57	condition_era group during day -30 through 0 days relative to index: Proteinuria
0.55	drug_era group during day -365 through 0 days relative to index: INSULINS AND ANALOGUES
-0.54	race = Black or African American
0.52	(Intercept)
0.50	gender = MALE

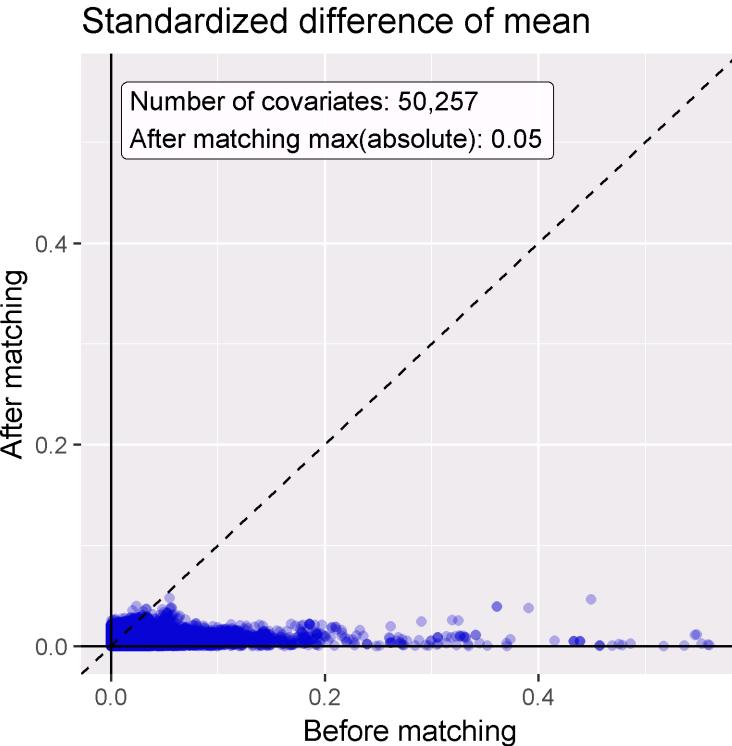


Figure 12.19: 공변량 균형, 성향 점수 매칭 전과 후의 절대 표준화 평균 차이를 보여준다. 각 점은 공변량을 나타낸다.



만약 어떤 변수가 매우 예측적 ‘highly predictive’ 이라면, 두 가지 결론을 내릴 수 있다. 하나는 변수가 노출과 매우 밀접한 관계가 있기 때문에 모델 적합 전에 제외해야 한다든가, 또는 표적군과 대조군이 실제로 비교가 불가능하기 때문에 분석을 멈추어야 한다는 것이다.

12.9.2 공변량 균형

성향점수를 사용하는 목적은 두 군을 비교할 수 있게 만드는 (또는 적어도 비교할 수 있는 군을 선택하는) 것이다. 기저 공변량이 조정 후 실제로 균형을 이루고 있는지 등을 확인하여 이 목적이 달성되었는지 입증해야 한다. `computeCovariateBalance` 및 `plotCovariateBalanceScatterPlot` 함수를 사용하여 그림 12.19를 생성할 수 있다. 한 가지 주요한 원칙은 성향 점수 조정 후 공변량이 0.1보다 큰 표준 차이 \geq absolute standardized difference of means을 가져서는 안 된다는 것이다. 여기서는 성향점수 짹짓기 이전에 상당한 불균형이 있었음에도 불구하고, 짹짓기 이후에는 이 기준을 충족한다는 것을 알 수 있다.

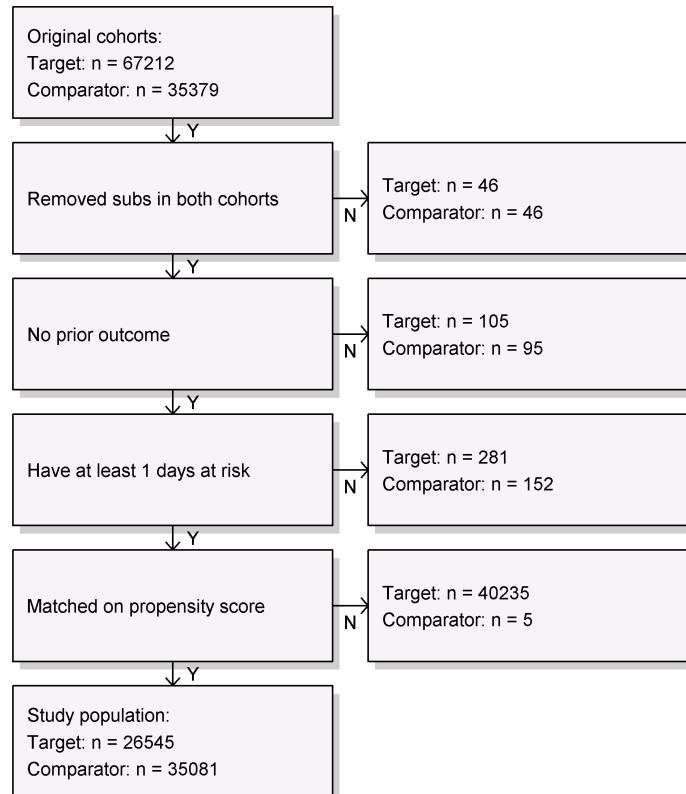


Figure 12.20: Attrition diagram. 맨 위에 표시된 개수는 우리의 목표와 대조군 코호트 정의를 충족하는 개수이다. 하단의 개수는 결과 모델에 입력되는 개수로서, 이 경우는 콕스 회귀 분석이다.

12.9.3 추적 기간과 검정력

결과 모델을 적합하기 전에, 특정 효과 크기를 감지할 수 있는 충분한 검정력이 있는지 파악해보고 싶을 수 있다. 연구 대상 집단이 완전히 정의되면, 다양한 포함/제외 기준 (예를 들어 이전 결과 없음)과 매칭 및/또는 트리밍으로 인한 손실을 고려하여 이러한 검정력 계산을 수행하는 것이 좋다. 그림 12.20과 같이 `drawAttritionDiagram` 함수를 사용하여 연구에서 피험자의 소모를 볼 수 있다.

후향적 연구에서 표본 크기가 고정되어 있고 (데이터가 이미 수집된 상태이므로), 실제 효과 크기를 알 수 없으므로, 예상 효과 크기에 대한 검정력을 계산하는 것은 의미가 없다. 대신 CohortMethod 패키지는 `computeMdrr` 함수를 제공하여 minimum detectable relative risk(MDRR)를 계산한다. 이 사례에서 MDRR은 1.69이다.

추적 기간 follow-up 데이터를 더 잘 이해하기 위해서는 추적 기간의 분포도 검사할 수 있다.

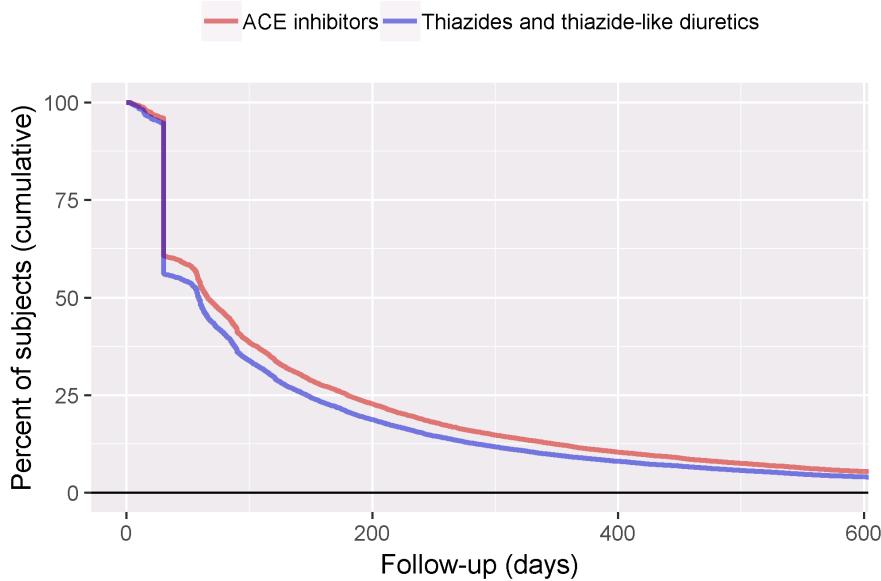


Figure 12.21: 표적 및 대조군 코호트의 추적 시간 분포.

추적 기간을 위험 노출 기간으로 정의했으므로 결과 발생으로 인해서는 중도 절단되지 않는다. `getFollowUpDistribution` 그림 12.21과 같은 간단한 개괄을 제공한다. 이는 두 코호트의 follow-up time이 비슷하다는 것을 의미한다.

12.9.4 Kaplan-Meier

마지막으로 한 가지 검사는 두 코호트에서 시간 경과에 따른 생존율을 보여주는 Kaplan-Meier plot을 검토하는 것이다. `plotKaplanMeier` 함수를 사용하여 그림 12.22를 만들 수 있는데, 여기서 예를 들면 위험 비례 가정이 성립하는지 확인할 수 있다. Kaplan-Meier plot은 성향점수별로 충화 또는 가중치를 자동으로 조정한다. 이 경우, 일-대-다 비율 짹짓기를 사용했으므로, 대조군의 생존 곡선이 조정되어 표적 군이 대조 약물에 노출되었을 경우 생존 곡선이 어떤 모습인지 모방하여 그려진다.

12.9.5 효과 크기 추정치

혈관 부종에 대한 위험 비는 4.32 (95% 신뢰 구간: 2.45 - 8.08)이며, 이는 ACEi가 THZ와 비교하여 혈관 부종의 위험을 증가시키는 것을 의미한다. 마찬가지로, 심근경색에 대한 위험 비는 1.13 (95% 신뢰 구간: 0.59 - 2.18)이며, 심근경색에 대한 영향은 거의 또는 전혀 없음을 알 수 있다. 앞에서 검토한 것처럼 연구가 잘 수행됐는지를 검사하는 진단 방법은 의심의 여지가 없다. 그러나 궁극적으로 이러한 근거의 질과 신뢰 여부는 14장에서 설명한 대로 연구 진단이 커버하지 못하는 많은 요인에 달려 있다.

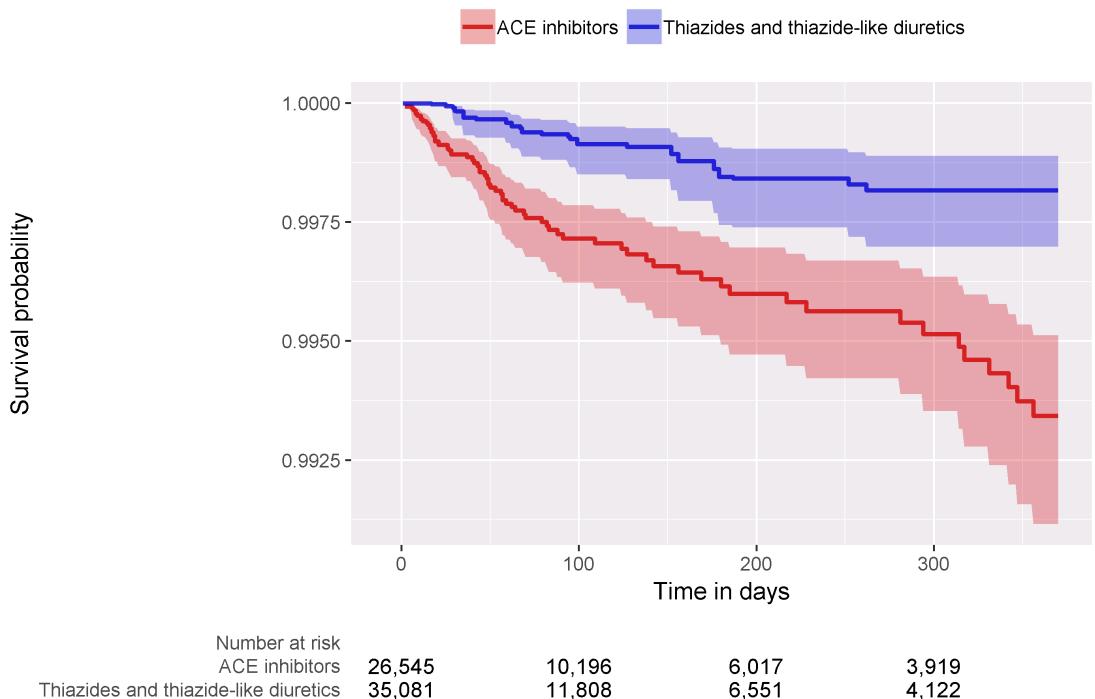


Figure 12.22: Kaplan-Meier plot.

12.10 요약



- 인구 수준 추정은 관찰형 데이터의 인과 관계를 추론하는 것을 목적으로 한다.
- 반사실 **counterfactual**, 즉 피험자가 만일 표적 치료제에 노출되지 않았다거나, 또는 다른 대체 약물에 노출되었다면 어떤 일이 벌어졌을까라고 하는 것은 관찰할 수 없다.
- 각 연구 설계는 서로 다른 방식으로 반사실을 구성하는 것을 목적으로 한다.
- OHDSI 연구방법론 라이브러리 Methods Library에 구현된 다양한 연구설계는 적절한 반사실을 만들기 위한 가정이 충족되었는지 여부를 평가하는 진단 방법을 제공한다.

12.11 예제

전제조건

이 예제를 위해서는 R, R-studio, Java가 8.4.5절에서 설명한 바와 같이 설치되어 있어야 한다. 또한 다음과 같이 SqlRender, DatabaseConnector, Eunomia, CohortMethod 패키지를 모두 설치해야 된다.

```
install.packages(c("SqlRender", "DatabaseConnector", "devtools"))
devtools::install_github("ohdsi/Eunomia", ref = "v1.0.0")
devtools::install_github("ohdsi/CohortMethod")
```

Eunomia 패키지는 당신의 로컬 R 세션에서 작동할 수 있도록 CDM 형태의 가상 데이터를 제공한다. 데이터베이스 접속은 다음과 같이 설정하면 된다.

```
connectionDetails <- Eunomia::getEunomiaConnectionDetails()
```

CDM 데이터 스키마는 “main”이다. 이 연습문제는 또한 몇 가지 코호트를 이용한다. Eunomia 패키지의 `createCohorts` 함수를 이용하여 COHORT 테이블에 코호트를 생성할 수 있다.

```
Eunomia::createCohorts(connectionDetails)
```

문제 정의

디클로페낙Diclofenac 새 사용자와 비교하였을 때 celecoxib 새 사용자의 위장 출혈 위험은 얼마인가?

celecoxib 새 사용자 코호트는 COHORT_DEFINITION_ID = 1 값을 가진다. 디클로페낙 새 사용자 코호트는 COHORT_DEFINITION_ID = 2 값을 가진다. 위장 출혈 코호트는 COHORT_DEFINITION_ID = 3 값을 가진다. Celecoxib와 디클로페낙의 성분 개념 ID는 각각 1118084과 1124300이다. 위험 노출 기간은 치료가 시작된 날부터 시작하며, 관찰이 종료될 때 멈춘다 (intent-to-treat 분석이라고 부른다).

Exercise 12.1. CohortMethod R 패키지를 사용하여, 공변량의 기본 모음을 사용하고 CDM에서 CohortMethodData 추출해 보라. CohortMethodData의 요약본을 생성해 보라.

Exercise 12.2. createStudyPopulation 기능을 사용하여 연구 집단을 생성하는데, 180일의 휴약 기간을 가지며, 사전 결과를 가진 사람을 배제하고 두 코호트에 공통으로 나타나는 사람을 제거해야 한다. 사람의 수가 적어지는가?

Exercise 12.3. 아무 조정을 사용하지 않고 Cox 비례 위험 모델을 만들어라. 이렇게 진행하면 무엇이 잘못되는가?

Exercise 12.4. 성향 모델을 만들어라. 그 두 집단은 비교되는가?

Exercise 12.5. 5개의 계층을 사용하여 PS 계층화를 수행하라. 공변량 균형은 달성되었는가?

Exercise 12.6. PS strata를 사용하여 Cox 비례 위험 모델을 구축하라. 조정되지 않은 모델과 결과가 다른 이유는 무엇인가?

제안된 답변은 부록 E.8에서 찾을 수 있다.

Chapter 13

환자 수준 예측

Chapter leads: Peter Rijnbeek & Jenna Reps

임상 의사 결정 clinical decision making이란 임상 의사가 알 수 있는 환자의 병력에 대한 정보와 현재 임상지침에 따라 진단 또는 치료 경로를 추론해야 하는 복잡한 일이다. 임상 예측 모델은 이러한 의사 결정 과정을 지원하기 위해 개발되었으며 광범위한 전문 분야에서 임상 실무에 사용된다. 이러한 모델은 인구 통계학적 정보, 질병력 및 치료력과 같은 환자 특성을 조합하여 이를 기반으로 진단 또는 예후 결과를 예측한다.

임상 예측 모델을 설명하는 출판물의 수가 지난 10년 동안 많이 증가했다. 현재 사용되는 대부분의 모델은 소규모 데이터 집합을 사용하여 추정되며, 소규모 환자 특성만 고려한다. 이처럼 소 표본이고 그래서 낮아지는 통계적 검정력으로 인해 데이터 분석가는 엄격한 가정하게 모델링을 수행하게 된다. 제한적인 환자 특성을 가진 데이터 집합의 선택은 현재 알고 있는 전문가의 지식에만 의존해서 강하게 설명된다. 이는 환자가 풍부한 디지털 트레일 digital trail을 생성하는 현대 의학의 현실과 크게 대조되며, 이는 모든 의료 전문가가 완전히 동화될 힘을 뛰어넘는다. 현재, 건강 관리는 전자 의무 기록 Electronic Health Records(EHR)에 저장된 엄청난 양의 환자 개인별 정보를 생성하고 있다. 여기에는 진단, 약물치료, 실험실 검사 결과와 같은 정형화된 데이터와 임상적 기술 clinical narratives에 포함된 비정형화된 데이터가 포함되어 있다. 대량의 환자 데이터를 완전한 EHR로부터 얻고 이것을 활용하여도 예측 정확도를 얼마나 얻을 수 있는지는 알 수 없다.

대규모 데이터 모음 분석을 위한 머신 러닝의 발전으로 이러한 유형의 데이터에 환자-수준 예측을 적용하는 데 관심이 높아졌다. 그러나 환자-수준 예측을 위한 많은 수의 출판물은 모델 개발 지침을 따르지 않아 광범위한 외적 타당도를 수행에 실패하거나 또는 독립적인 연구자가 그 모델을 재현

하고 외적 타당도를 검증하기 위해 필요 가능성은 제한하는 모델링을 위한 세부사항을 제공하지 않는다. 이것은 모델의 예측 성능을 공정하게 평가하기 어렵게 하고 임상 실무에서 모델이 적절하게 사용될 가능성을 줄인다. 표준화를 개선하기 위해 예측 모델을 개발하고 보고하는 모범 사례에 대한 지침을 자세히 설명하는 여러 논문이 작성되었다. 예를 들어, 개별 예측 또는 진단(TRIPOD)¹ 선언문에 다변량 예측 모델의 투명한 보고 Transparent Reporting는 예측 모델 개발 및 타당도 보고에 대한 명확한 권장 사항을 제공하고 투명성과 관련된 일부 우려를 해결한다.

OHDSI CDM을 통해 전례 없는 규모의 데이터를 균일하고 투명하게 분석할 수 있게 되어, 대규모의 환자별 예측 모델링이 현실이 되었다. CDM으로 표준화된 데이터베이스 네트워크가 증가하면서 전 세계의 다양한 의료 환경에서 모델의 외적 타당도 검증이 가능하게 되었다. 우리는 이런 환경이 치료의 질적 개선을 가장 필요로 하는 많은 수의 환자 공동체를 돌볼 수 있는 즉각적인 기회를 제공한다고 믿는다. 그러한 모델은 진정한 개인 맞춤형 의료를 제공할 수 있기 때문에, 환자의 예후를 크게 개선할 수 있게 될 것으로 믿는다.

이 장에서는 환자-수준 예측을 위한 OHDSI의 표준화된 프레임워크 (Reps et al., 2018)를 설명하고 개발 및 타당도 검증을 위해 확립된 모범 사례를 구현하는 PatientLevelPrediction R 패키지에 관해 설명한다. 우리는 환자-수준 예측의 개발과 평가에 필요한 이론을 제공하는 것으로 시작하여 구현된 기계 학습 알고리즘에 대한 큰 그림 수준의 개요를 제공할 것이다. 그런 다음 예측 문제에 대한 예제에 대하여 논의하고 ATLAS 또는 사용자 정의 R 코드를 사용하여 예측 문제를 정의하고 실행하는 단계별 지침을 제공할 것이다. 마지막으로 연구 결과를 널리 알리기 위한 Shiny 앱 사용법에 대해서 논의한다.

13.1 예측 문제

그림 13.1은 우리가 다루는 예측 문제를 보여준다. 연구 대상군 중 어떤 환자가 위험 노출 시간 동안에 어떤 결과를 경험할 것인지 특정 시점 ($t = 0$)에서 예측하는 것을 목표로 한다. 예측은 해당 시점 ($t = 0$) 이전의 관찰 기간 observation window에서 관찰된 환자 정보만 사용하여 수행된다.

표 13.1에서 볼 수 있듯이 예측 문제를 밝히려면 대상 코호트 target cohort의 $t = 0$, 결과 코호트에 의해 예측하고자 하는 결과, 그리고 위험 노출 시간을 정의해야 한다. 표준 예측 질문을 다음과 같이 정의한다:

[대상 코호트, T]에서, 누가 [위험에 노출된 시간, t] 내에 [결과 코호트, O]가 발생하는가?

또한 개발하고자 하는 모델에 대해 디자인을 선택하고 내적 및 외적 타당도 검증을 수행할 관찰

¹<https://www.equator-network.org/reporting-guidelines/tripod-statement/>

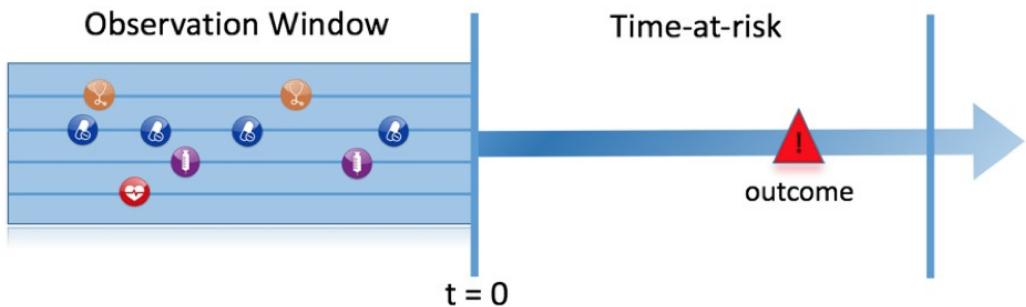


Figure 13.1: 예측 문제

데이터 모음을 결정해야 한다.

Table 13.1: Main design choices in a prediction design.

Choice	Description
Target cohort	How do we define the cohort of persons for whom we wish to predict?
Outcome cohort	How do we define the outcome we want to predict?
Time-at-risk	In which time window relative to $t=0$ do we want to make the prediction?
Model	What algorithms do we want to use, and which potential predictor variables do we include?

이 개념적 프레임워크는 다음과 같은 모든 유형의 예측 문제에 적용된다, 예를 들면:

- 질병 발병 및 진행
- **구조:** [질병 A]로 새로 진단된 환자 중, [진단 시점 t] 내에 *[또 다른 질병이나 합병증, B*] 이 생길 사람은 누구인가?
- **예제:** 새로 진단된 심방세동 환자 중 향후 3년 이내에 허혈성 뇌졸중이 발생할 사람은 누구인가?
- 치료 선택
- **구조:** [치료 1] 또는 [치료 2]로 치료한 [대상 질병, D]에 걸린 환자 중 [치료 1]로 치료받은 환자는 누구인가?
- **예제:** 와파린 또는 리바록사반을 복용한 심방세동 환자 중 어떤 환자가 와파린을 복용했는가?
(예를 들어 성향 모델의 경우)

- 치료 반응
- 구조: [치료 1]을 처음 사용하는 사람 중, 누가 [시간대 t]에서 [어떤 효과, E]를 경험했는가?
- 예제: 메트포민으로 치료받기 시작한 당뇨병 환자 중 어떤 환자가 3년 동안 메트포민을 유지하는가?
- 치료 안전
- 구조: [치료 1]을 처음 사용하는 사람 중 누가 [시간대 t]에서 [이상 반응 E]를 경험하게 되는가?
- 예제: 와파린을 처음 사용하는 사람 중 누가 1년 안에 위장관 출혈이 발생하는가?
- 치료 준수
- 구조: [치료 1]을 처음 사용하는 사람 중 누가 [시간대, t]에서 [준수 지표 수치]를 달성하는가?
- 예제: 메트포민으로 치료를 시작한 당뇨병 환자 중 어떤 환자가 1년 중 80% 이상의 복용 준응도를 보이는가?

13.2 데이터 추출

예측 모델을 만들 때 상태에 따라 분류된 예제 기반으로 공변량과 결과 상태 간의 관계를 유추하기 위하여 기계학습과 같은 지도 학습이라는 프로세스를 사용한다. 따라서, 대상 코호트에 있는 사람의 CDM에서 공변량을 추출하는 방법이 필요하며 그들의 결과 레이블을 얻을 필요가 있다.

공변량 (“예측변수”, “특징” 또는 “독립 변수”라고도 함)은 환자의 특성을 묘사한다. 공변량은 연령, 성별, 특정 질병 존재, 그리고 환자 기록에 있는 노출 코드 그리고 그 외 여러 가지가 될 수 있다. 공변량은 Feature Extraction 패키지를 사용하여 구성되었고, 11장에 자세히 설명돼 있다. 예측을 위해 우리는 오직 대상 코호트에 들어오는 날짜 기준으로 환자의 이전 또는 그때의 데이터만 사용할 수 있다. 이 날짜를 인덱스 날짜라고 한다.

또한, 위험 노출 기간 time-at-risk(TAR) 동안 모든 환자의 결과 상태 (“라벨” 또는 “분류”라고도 함)를 생성할 필요가 있다. 만약 결과가 위험에 노출된 시간 안에 발생하거나 그 결과 상태는 “양성”으로 정의된다.

13.2.1 데이터 추출 예제

표 13.2는 두 개의 코호트가 있는 COHORT에 대한 예를 보여준다. 코호트 정의 ID 1인 코호트는 대상 코호트 (예를 들어 “최근 심방세동 진단을 받은 사람”)이고 코호트 정의 ID 2는 결과 코호트 (예를 들어 “뇌졸중”)이다.

Table 13.2: Example COHORT table. For simplicity the COHORT_END_DATE has been omitted.

COHORT_DEFINITION_ID	SUBJECT_ID	COHORT_START_DATE
1	1	2000-06-01
1	2	2001-06-01
2	2	2001-07-01

표 13.3은 CONDITION_OCCURRENCE에 대한 예제이다. 개념(concept) ID 320128은 “본태성 고혈압”을 나타낸다.

Table 13.3: Example CONDITION_OCCURRENCE table. For simplicity only three columns are shown.

PERSON_ID	CONDITION_CONCEPT_ID	CONDITION_START_DATE
1	320128	2000-10-01
2	320128	2001-05-01

이 예제 데이터를 기반으로, 위험에 노출된 시간이 인덱스 날짜 (대상 코호트 시작 날짜)의 다음 연도라고 가정하고 다음과 같이 공변량 및 결과 상태를 구성할 수 있다. Person ID가 1인 환자 (인덱스 날짜 이후에 발생한 질병)의 경우 “이전 해의 본태성 고혈압”을 나타내는 공변량은 값 0 (현재 아님)으로 하고, person ID가 2인 환자는 값 1 (현재 진행)을 갖는다. 유사하게, 결과 상태는 person ID가 1인 환자 (이 사람은 결과 코호트에 못 들어감)의 경우는 값 0을 갖고 person ID가 2인 환자 (인덱스 날짜 다음 1년 이내에 결과가 발생하였음)의 경우에는 값 1을 갖는다.

13.2.2 결측

관찰 의료 데이터는 데이터 누락 여부를 거의 반영하지 않는다. 이전의 예제에서, 우리는 person ID가 1인 환자가 인덱스 날짜 이전에 본태성 고혈압이 없었음을 관찰했다. 이는 그 당시 본태성 고혈압이 없었거나 기록되지 않았기 때문일 수 있다. 머신 러닝 알고리즘은 두 시나리오를 구분할 수 없고 사용 가능한 데이터 안에서 예측값을 대략 평가한다는 것을 인지해야 한다.

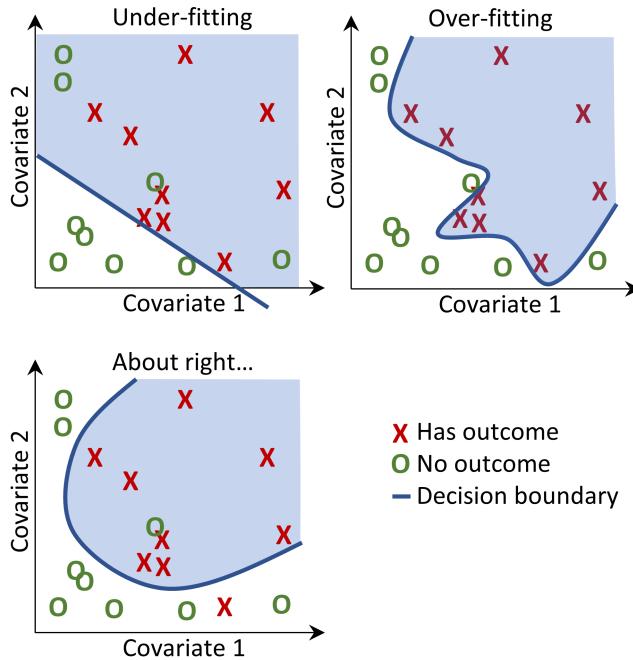


Figure 13.2: 결정 경계.

13.3 모델 적합

예측 모델을 적합할 때 상태에 따라 분류된 예제로부터 공변량과 관찰된 결과 상태 간의 관계를 알려고 노력한다. 만약 수축기 혈압과 이완기 혈압의 두 가지 공변량이 있다고 가정하면 그림 13.2 와 같이 2차원 공간에서 그림으로 각 환자를 나타낼 수 있다. 이 그림에서 데이터를 나타내는 점의 모양은 환자의 결과 상태 (예를 들어, 뇌졸중)를 나타낸다.

지도 학습 모델은 두 결과 분류 classes를 최적으로 분리하는 결정 경계 decision boundaries를 찾아내려고 노력할 것이다. 다른 지도 학습 기법은 다른 분류 결정 경계로 이어지고 분류 결정 경계의 복잡성에 영향을 줄 수 있는 하이퍼-파라미터 hyper-parameters가 종종 있다.

그림 13.2에서 세 가지 다른 결정 경계를 볼 수 있다. 그 경계는 새로운 데이터 포인트의 결과 상태를 추론하기 위해 사용되기도 한다. 새 데이터 포인트가 음영이 있는 영역에 포함되면 모델은 “결과가 있음”을 예측하고, 그렇지 않으면 “결과가 없음”으로 예측한다. 이상적으로는 결정 경계가 두 분류를 완벽하게 구분해야 한다. 그러나 너무 복잡한 모델은 데이터에 “과적합 overfit”할 위험이 있다. 이는 보이지 않는 데이터에 대한 모델의 일반화에 부정적인 영향을 줄 수 있다. 예를 들어, 데이터에 레이블이 없거나 잘못 지정된 데이터 포인트를 갖는 잡음 noise이 포함된 경우 해당 잡음에 예측 모델을 적합하고 싶지 않을 것이다. 그러므로 우리는 학습 데이터 training data를 갖고 완벽하게

판별하지는 않지만 “실제의” 복잡성을 반영하는 결정 경계를 정의하는 것을 선호 할 수 있다. 정규화 regularization와 같은 기술은 복잡성을 최소화하면서 모델 성능을 최대화하는 것을 목표로 한다.

각각의 지도 학습 알고리즘마다 의사 결정 경계를 학습하는 방법이 다르므로 어떤 알고리즘이 데 이터에 가장 적합한지 간단하지 않다. No Free Lunch 정리에 따르면 모든 예측 문제에서 하나의 알고리즘이 언제나 다른 알고리즘보다 성능이 우수하지는 않다는 것을 알 수 있다. 따라서 환자 수준 예측 모델을 개발할 때 다양한 하이퍼-파라미터 설정으로 여러 개의 지도 학습 알고리즘을 사용하는 것이 좋다.

다음의 알고리즘은 PatientLevelPrediction 패키지에서 사용할 수 있다:

13.3.1 정규화된 로지스틱 회귀

Least Absolute Shrinkage and Selection Operator(LASSO) 로지스틱 회귀는 변수의 선형결합을 알 수 있는 일반화 선형 모델 generalized linear models에 속하고 로지스틱 함수는 결국 0과 1 사이 값으로 배치구조를 나타낸다. LASSO 정규화는 모델 학습 시 모델 복잡도에 따른 비용을 목적 함수 objective function에 추가한다. 이 비용은 계수의 선형 결합의 절댓값의 합이다. 모델은 이 비용을 최소화하면서 특징 선택 feature selection을 자동으로 수행한다. 우리는 대규모 정규화 로지스틱 회귀 분석을 수행하기 위하여 Cyclops(Cyclic coordinate descent for logistic, Poisson and survival analysis) 패키지를 사용한다.

Table 13.4: Hyper-parameters for the regularized logistic regression.

Parameter	Description	Typical values
Starting variance	The starting variance of the prior distribution.	0.1

교차 검증에서 표본 외 out-of-sample 우도 likelihood를 최대화하여 분산이 최적화되므로 시작 분산은 결과 모델의 성능에 거의 영향을 미치지 않는다. 그러나 시작 분산이 최적값에서 너무 차이가 나면 모델 적합 시간이 길어질 수 있다.

13.3.2 Gradient Boosting Machines

Gradient boosting machines은 부스팅 양상을 기법 boosting ensemble technique이며 프레임워크 안에서 다중 의사 결정 나무를 연결한다. 부스팅은 의사 결정 나무 tree를 반복적으로 추가하는 것이지만, 다음에 생성될 의사 결정 나무를 학습할 때 비용 함수 cost function에서 이전 의사 결정 나무에 의해 잘못 분류된 데이터 포인트에 더 많은 가중치를 추가한다. CRAN에서 제공하는 xgboost R

패키지로 수행된 gradient boosting framework를 효율적으로 구현하는 Extreme Gradient Boosting 을 사용한다.

Table 13.5: Hyper-parameters for gradient boosting machines.

Parameter	Description	Typical values
earlyStopRound	Stopping after rounds without improvement	25
learningRate	The boosting learn rate	0.005,0.01,0.1
maxDepth	Max levels in a tree	4,6,17
minRows	Min data points in a node	2
ntrees	Number of trees	100,1000

13.3.3 랜덤 포레스트

랜덤 포레스트 Random forest는 다중 의사 결정 나무를 연결하는 배깅 앙상블 기법 bagging ensemble technique^o이다. 배깅의 기본 개념은 유사도가 낮은 분류기를 사용하여 유사도가 높은 분류기로 결합하여 과적합 가능성을 줄이는 것이다. 랜덤 포레스트 Random forest는 다중 의사 결정 나무를 학습하여 저장하는 것으로 하지만 각 나무에서 변수의 하위 집합만 사용하며 변수 하위 집합은 의사 결정 나무마다 다르다. 우리 패키지는 Python의 Sklearn 내 Random forest를 사용한다.

Table 13.6: Hyper-parameters for random forests.

Parameter	Description	Typical values
maxDepth	Max levels in a tree	4,10,17
mtries	Number of features in each tree	-1 = square root of total features,5,20
ntrees	Number of trees	500

13.3.4 K-최근접 이웃

K-최근접 이웃 K-nearest neighbors(KNN)은 몇 개의 거리 척도 distance metric를 사용하여 레이블이 지정되지 않은 새로운 데이터 포인트에 가장 가까운 K개의 레이블이 있는 데이터 포인트를

찾는 알고리즘이다. 새로운 데이터 포인트의 예측은 K-최근접의 레이블이 된 데이터 포인트의 가장 보편적인 분류이다. 모델에 새 데이터에 대한 예측을 수행하기 위해 레이블이 지정된 데이터가 필요하므로 KNN의 공유 제한이 있으며 데이터 사이트 간에 이 데이터를 공유할 수 없는 경우가 종종 있다. 우리는 대규모 KNN classifier인 OHDSI에서 개발된 BigKnn 패키지를 포함했다.

Table 13.7: Hyper-parameters for K-nearest neighbors.

Parameter	Description	Typical values
k	Number of neighbors	1000

13.3.5 나이브 베이즈

나이브 베이즈 Naive Bayes 알고리즘은 클래스 변수의 값이 주어지는 모든 특징 사이의 조건부 독립성의 나이브 추정을 가진 베이즈 Bayes 이론을 적용한다. 클래스의 사전 배포와 데이터가 클래스에 속할 가능성에 기초하여, 사후 배포가 얻어진다. 나이브 베이즈는 하이퍼-파라미터를 갖지 않는다.

13.3.6 AdaBoost

AdaBoost는 부스팅 양상을 기법이다. 부스팅은 분류기 classifier를 반복적으로 추가하여 수행되지만, 다음 분류기가 학습될 때 비용 함수에서 이전 분류기에 의해 잘못 분류된 데이터 포인트에 더 많은 가중치를 준다. 우리는 Python에 있는 sklearn AdaboostClassifier 구현을 사용한다.

Table 13.8: Hyper-parameters for AdaBoost.

Parameter	Description	Typical values
nEstimators	The maximum number of estimators at which boosting is terminated	4

Parameter	Description	Typical values
learningRate	<p>Learning rate</p> <p>shrinks the contribution of each classifier by learning_rate.</p> <p>There is a trade-off between learningRate and nEstimators</p>	1

13.3.7 의사 결정 트리

의사 결정 나무는 탐욕 접근 greedy approach 방식을 사용하여 선택한 개별 테스트를 사용하여 가변 공간을 분할하는 분류기이다. 이것은 클래스를 분리하는 데 가장 많은 정보를 얻는 파티션을 찾는 것을 목표로 한다. 의사 결정 나무는 많은 수의 파티션 tree depth을 사용하게 되면 쉽게 과적합 되기 때문에 종종 일부 정규화 (예를 들어, 모델의 복잡성을 제한하는 하이퍼-파라미터의 정리 또는 지정) 가 필요하다. 우리는 Python에 있는 sklearn DecisionTreeClassifier 구현을 사용한다.

Table 13.9: Hyper-parameters for decision trees.

Parameter	Description	Typical values
classWeight	“Balance” or “None”	None
maxDepth	The maximum depth of the tree	10

Parameter	Description	Typical values
minImpuritySplitThreshold	The minimum threshold for early stopping in tree growth.	10^{-7}
minSamplesLeaf	A node will split if its impurity is above the threshold, otherwise it is a leaf	10
minSamplesSplit	The minimum number of samples per leaf	2

13.3.8 Multilayer Perceptron

Multilayer perceptrons은 비선형 함수를 사용하여 입력에 가중치를 부여하는 여러 계층의 노드를 포함하는 신경망이다. 첫 번째 레이어는 입력 레이어 input layer이고 마지막 레이어는 출력 레이어 output layer이며 그리고 그 사이에는 히든 레이어 hidden layers가 있다. 신경망은 일반적으로 역전파 back-propagation을 사용하여 학습된다. 즉, 학습 입력 training input이 네트워크를 통해 앞으로 전달되어 출력을 생성하고, 출력과 결과 상태 사이의 오류가 계산되며, 이 오류는 네트워크를 통해 뒤로 전달되어 선형 함수 가중치를 업데이트한다.

Table 13.10: Hyper-parameters for Multilayer Perceptrons.

Parameter	Description	Typical values
alpha	The l2 regularization	0.00001
size	The number of hidden nodes	4

13.3.9 딥 러닝

Deep net, convolutional neural networks 또는 recurrent neural networks와 같은 딥 러닝 deep learning은 Multilayer perceptrons와 유사하지만, 예측에 유용한 잠재 표현을 학습하는 것을 목표로 하는 숨겨진 레이어를 여러 개 가지고 있다. PatientLevelPrediction 패키지의 별도의 vignette 이러한 모델과 하이퍼-파라미터에 대해 자세히 기술되어 있다.

13.3.10 다른 알고리즘

다른 알고리즘도 환자-수준 예측 프레임워크에 추가될 수 있다. 이것은 이 장의 범위를 벗어난다. 자세한 내용은 PatientLevelPrediction 패키지의 “Adding Custom Patient-Level Prediction Algorithms” vignette에 있다.

13.4 예측 모델 평가

13.4.1 평가 유형

예측값과 관측값의 일치 여부를 평가함으로써 예측 모델을 평가할 수 있는데, 그렇게 하려면 결과 상태가 있는 데이터가 필요하다.



평가를 위해서는 모델을 개발하는 데 사용된 것과 다른 데이터 모음을 사용해야 한다. 그렇지 않으면 과적합 (13.3절 참조) 되거나 새로운 환자에게 잘 맞지 않는 모델을 만들 위험성이 있다.

평가는 다음 두 가지로 나뉜다:

- **내적 타당도:** 같은 데이터베이스에서 추출된 다른 데이터 모음을 사용하여 모델을 개발하고 평가.
- **외적 타당도:** 한 데이터베이스에서 모델을 개발하고 다른 데이터베이스에서 평가.

내적 타당도를 수행하는 데는 두 가지 방법이 있다:

- **홀드 아웃 모음 holdout set 접근법**은 레이블이 지정된 데이터를 두 개의 독립된 훈련 모음과 테스트 모음으로 나누는 것이다. 훈련 모음은 모델 학습에 사용되고, 테스트 모음은 모델 평가에 사용된다. 환자군을 무작위로 훈련 모음과 테스트 모음으로 나누거나 다음과 같이 선택할 수 있다:

- 날짜를 기반으로 데이터를 분할한다 (시점 타당도). 예를 들어, 특정 날짜 이전의 데이터로 학습시키고, 그 특정 날짜 이후 데이터로 평가하는 것이다. 이것은 모델이 서로 다른 기간에서도 일반화가 가능한지 여부를 알 수 있게 해준다.
- 지리적 위치를 기반으로 데이터를 분할한다 (공간 타당도).
- **교차 검증**은 데이터가 제한적일 때 유용하다. 데이터를 n 개의 동일 크기의 모음으로 분할한다. 여기서 n 은 미리 정해져야 한다 (예를 들어, $n = 10$). 이러한 각 모음에 대해 한 모음의 데이터를 제외한 모든 데이터를 이용해 모델을 학습하며, 감추어 둔 한 모음 (홀드 아웃 모음)은 예측 (평가)에 사용된다. 이를 n 번 반복하여 모든 데이터가 한 번씩 모델을 구축하는 알고리즘을 평가하는데 사용된다. PLP 프레임 워크에서는 최적의 하이퍼-파라미터를 선택하는 데 교차 검증을 이용한다.

외적 타당도는 다른 데이터베이스, 즉 개발에 사용된 데이터베이스가 아닌 다른 데이터베이스로부터 데이터를 얻어 모델의 성능을 평가하는 것을 목표로 한다. 훈련에 사용한 그 데이터베이스뿐만 아니라 다른 데이터베이스에도 우리가 개발한 모델을 적용하기를 원하기 때문에 모델의 타 기관 적용 방법론은 중요하다. 다른 데이터베이스란 다른 환자 집단, 다른 의료 시스템 및 다른 데이터 획득 프로세스를 대변한다. 대규모 데이터베이스 모음에 대한 예측 모델의 외적 타당도 검증은 임상 실무에서 예측 모델을 수용하고 구현하기 위해 결정적이라고 생각한다.

13.4.2 성능 지표

임계값 지표

예측 모델은 위험에 노출된 시간 동안 관심 결과가 발생할 위험이 있는 각 환자에 대해 0과 1 사이의 값을 할당한다. 0값은 0% 위험을 의미하고 0.5값은 50% 위험을 의미하고 1값은 100% 위험을 의미한다. 위험에 처한 시간 동안 환자가 관심 결과를 갖는지 여부를 결정하는 임계치(기준치)를 정하면 정확도, 민감도, 특이도, 양성 예측도와 같은 일반적인 측정 지표를 계산해 낼 수 있다. 예를 들어, 표 13.11에서 임계값을 0.5로 설정하면 환자 1, 3, 7 및 10은 예상 위험이 임계값 0.5보다 크거나 같으므로 관심 결과가 발생할 것으로 예측된다. 다른 모든 환자는 0.5 미만의 예측 위험을 가지고 있으므로 관심 결과는 없을 것으로 예측된다.

Patient ID	Predicted risk	Predicted	Has outcome	Type
		class at 0.5 threshold	during time-at-risk	

Table 13.11: Example of using a threshold on the predicted probability.

Patient ID	Predicted risk	Predicted	Has outcome	Type
		class at 0.5 threshold	during time-at-risk	
1	0.8	1	1	TP
2	0.1	0	0	TN
3	0.7	1	0	FP
4	0	0	0	TN
5	0.05	0	0	TN
6	0.1	0	0	TN
7	0.9	1	1	TP
8	0.2	0	1	FN
9	0.3	0	0	TN
10	0.5	1	0	FP

환자에게 관심 결과가 예측되고 (위험 노출 시간 동안) 관심 결과가 발생했다면 이를 진양성(TP)이라고 한다. 환자에게 관심 결과가 있을 것으로 예상되지만 그 결과가 없는 경우라면 이를 거짓 양성(FP)이라고 한다. 환자에게 관심 결과가 없을 것으로 예상되고 실제 결과가 없는 경우 이를 진음성(TN)이라고 한다. 마지막으로, 환자에게 관심 결과가 없을 것으로 예상되지만 결과가 있는 경우 이를 거짓 음성(FN)이라고 한다.

다음과 같이 임계값-기반 지표를 계산할 수 있다:

- 정확도(accuracy): $(TP + TN) / (TP + TN + FP + FN)$
- 민감도(sensitivity): $TP / (TP + FN)$
- 특이도(specificity): $TN / (TN + FP)$
- 양성예측도(positive predictive value): $TP / (TP + FP)$

임계값을 낮추면 이러한 값이 감소하거나 증가 할 수 있다. 분류기의 임계값을 낮추면 반영되는

결과 수가 증가하여 분모가 증가할 수 있다. 이전에 임계값을 너무 높게 설정한 경우, 새로운 결과는 모두 진양성이 될 수 있으며, 이는 양성 예측도를 증가시킨다. 이전 임계값이 적절하거나 또는 너무 낮다면 임계값을 더 낮추게 되면 거짓 양성이 발생하여 양성 예측도가 감소한다. 민감도의 분모는 분류기의 임계값 ($TP + FN$ 은 상수)에 의존하지 않는다. 이는 분류기의 임계값을 낮추면 진양성 결과 수를 증가 시켜 민감도를 높일 수 있음을 의미한다. 또한, 임계값을 낮추면 민감도가 바뀌지 않고 양성 예측도가 변할 수 있다.

판별력

판별력 discrimination은 위험에 노출된 시간 동안 관심 결과를 경험할 환자에게 더 높은 위험을 할당하는 능력이다. Receiver Operating Characteristics(ROC) 곡선은 가능한 모든 임계값에 대하여 x축은 1-특이도를 그리고 y축에는 민감도를 그린 것이다. ROC 곡선은 이 장의 뒷부분에 있는 그림 13.17에 나와 있다. area under the receiver operating characteristic curve(AUC)가 0.5이면 위험에 무작위로 할당되는 것을 의미하고 값이 1이면 완벽하게 판별한다는 의미이다. 대부분 출판된 예측 모델의 AUC는 0.6-0.8 사이의 값을 갖고 있다.

AUC는 위험에 노출된 시간 동안 관심 결과를 경험한 환자와 그렇지 않은 환자 간에 예측된 위험 분포가 얼마나 다른지 결정하는 방법을 제공한다. AUC가 높으면 위험 분포가 대부분 분리되지만, 겹치는 부분이 많을 때는 그림 13.3과 같이 AUC가 0.5에 가까워진다.

결과가 드물게 발생할 경우, AUC가 높은 값을 갖는 모델이라도 주어진 임계값을 초과하는 모든 양성에 대해 음성이 많을 수 있으므로 (즉, 양성 예측도가 낮을 수 있기 때문에) 실용적이지 않을 수 있다. 결과의 심각성과 일부 중재에 대한 비용 (건강 위험/금전적)에 따라, 거짓 양성비가 커질 수 있다. 결과가 드물게 발생하는 것이라면 area under the precision-recall curve(AUPRC)라고 알려진 다른 측정법이 권장된다. AUPRC는 x축은 민감도 (재현율 recall이라고도 함)와 y축은 양성예측도 (정밀도 precision라고도 함)을 나타내는 곡선하 면적이다.

적합도

적합도 calibration은 모델이 정확한 위험을 할당하는 능력이다. 예를 들어, 모델이 100명의 환자에게 10%의 위험이 발생 가능성이 있다고 할 경우 10명의 환자는 위험 노출 시간 중에 결과를 경험해야 한다. 모델이 100명의 환자에게 80%의 위험 발생 가능성이 있다고 하면 80명의 환자는 위험 노출 시간 동안 결과를 경험해야 한다. 적합도는 일반적으로 예측된 위험에 따라 환자를 십분 위(열 등분)로 분할하고 각 그룹에서 평균 예측 위험과 실제 결과를 경험한 환자의 비율을 이용하여 계산한다. 그런 다음 10개의 점 (y축에 예측 위험과 x축에 관찰된 위험)을 그린 후 그 점들이 $x = y$ 인 선에 가까이 있는지, 즉 모델이 잘 보정되었는지를 확인한다. 적합도 그래프는 이 장의 뒷부분에 있는 그림 13.18에 나와 있다. 또한 점을 사용하여 절편 (0에 가까워 야 함)과 기울기 (1에 가까워야 함)

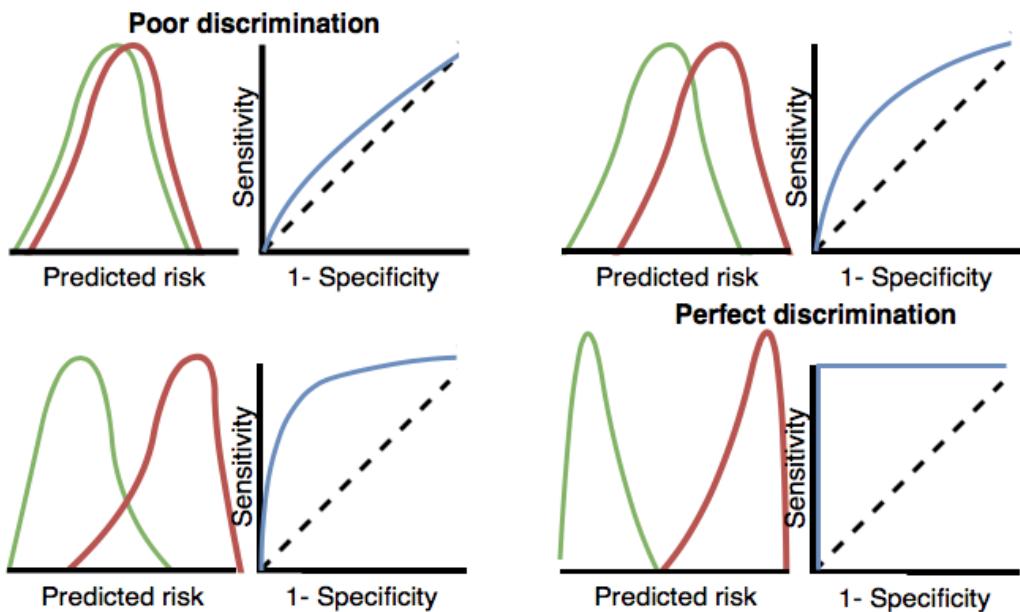


Figure 13.3: ROC 플롯이 어떻게 판별력과 연관되어 있는지를 설명한다. 두 등급의 예측 위험 분포가 유사한 경우, ROC는 대각선에 가까우며 AUC는 0.5에 가깝다 (좌측 상단).

를 계산하여 선형 모델을 만든다. 기울기가 1보다 크면 모델이 실제 위험보다 높은 위험을 할당하고 있고 기울기가 1보다 작으면 모델이 실제 위험보다 낮은 위험을 나타내는 것이다. 예측된 위험과 관찰된 위험 사이의 비선형 관계를 더욱 잘 포착하기 위해서, PLP R 패키지에 Smooth Calibration Curves를 구현했다.

13.5 환자-수준 예측 연구 설계

이 장에서는 예측 연구를 설계하는 방법을 보여준다. 첫 번째 단계는 예측 문제를 명확하게 정의하는 것이다. 흥미롭게도, 많은 출판된 논문에서 예측 문제가 잘 정의되어 있지 않다. 예를 들어 기준 날짜 (대상 코호트의 시작)가 어떻게 정의되어 있는지 명확하지 않다. 잘못 정의된 예측 문제는 임상 실무에서의 구현은 물론 다른 사람에 의한 외적 타당도 검정이 불가능하다. 환자-수준 예측 프레임워크에서 표 13.1에 정의된 주요 선택 사항을 명시적으로 정의하게 함으로써 예측 문제에 대한 적절한 명세서를 만들도록 한다. 여기서는 “치료 안전성”을 보는 유형의 예측 문제를 예로 들어 이 프로세스를 살펴보자.

13.5.1 문제 정의

혈관 부종 Angioedema은 ACE 억제제의 잘 알려진 부작용이며, ACE 억제제에 대한 레이블에 보고된 혈관 부종의 발생률은 0.1 % - 0.7% 범위이다. (Byrd et al., 2006) 혈관 부종은 드물지만, 생명을 위협하여 호흡 정지 및 사망으로 이어질 수 있기 때문에 이러한 부작용에 대한 환자 모니터링은 중요하다. (Norman et al., 2013) 또한, 혈관 부종이 조기에 인식되지 않으면, 식별할 원인의 범위가 넓어지고 값비싼 정밀검사를 해야 할 수 있다. (Norman et al., 2013; Thompson and Frable, 1993) 아프리카계 미국인 환자 사이에서 발생한 높은 위험 외에 ACE 억제제 관련 혈관 부종의 발생에 대해 알려진 소인은 없다. (Byrd et al., 2006) 대부분의 반응은 초기 요법의 첫 주 또는 한 달 안에, 그리고 종종 초기 복용 후 몇 시간 내에 발생한다 (Cicardi et al., 2004). 그러나 치료가 시작된 후 몇 년이 걸릴 수도 있다. (O'Mara and O'Mara, 1996) 위험에 처한 사람을 구체적으로 식별하는 진단 테스트는 없다. 우리가 위험에 처한 사람을 식별 할 수 있다면, 예를 들어, 의사는 ACE 억제제 처방을 중단하고 다른 고혈압 약물을 처방할 수도 있다.

관찰 의료 데이터에 환자-수준 예측 프레임워크를 적용하여 다음의 환자-수준 예측 질문에 응용해보자:

처음에 ACE 억제제로 치료를 시작한 환자 중, 그다음 해에 혈액 부종을 경험한 환자는 누구인가?

13.5.2 연구 모집단 정의

예측 모델을 개발하기 위한 최종 연구 모집단은 종종 대상 코호트의 하위 집단인데, 왜냐하면 관심 결과에 의존하는 기준을 적용하거나 또는 대상 코호트의 부분 모집단 sub-population의 민감도 분석을 수행하고자 하기 때문이다. 이것을 위하여 우리는 다음의 질문을 설명해야만 한다:

- 대상 코호트의 기준 날짜 이전의 최소 필요 관찰 기간은 얼마인가? 이 선택은 학습 데이터에서 사용 가능한 환자 시간에 따라 달라질 수 있지만, 장차 모델을 적용하려는 데이터 소스에서 사용 가능할 것으로 예상되는 시간에 따라 달라질 수도 있다. 최소 관측 시간이 길어질수록 특징 추출에 사용할 수 있는 기저력 시간 baseline history time이 길어지지만, 대신 분석에 사용할 수 있는 환자 수는 줄어든다. 또한, 단기 또는 장기의 과거력 관찰 시간을 선택해야 하는 임상적 이유가 있을 수 있다. 예제에서는 기준 날짜로부터 365일 이전까지의 기록을 기저 관찰 기간 (휴약기 washout period)으로 사용한다.
- 환자가 대상 코호트에 여러 번 포함될 수 있는가? 대상 코호트 정의에서, 사람은 서로 다른 시간 간격 동안, 예를 들어 서로 다른 에피소드의 질병이 있거나 의료 제품에 대한 또 다른 노출 기간이 있는 경우 그 코호트에 여러 번 포함될 수 있다. 코호트 정의는 환자가 한 번만

들어갈 수 있도록 제한을 적용할 필요는 없지만, 특정 환자-수준 문제와 관련하여 코호트를 첫 번째 관찰로 제한 할 수 있다. 이 예제에서는 기준이 ACE 역제제의 첫 번째 사용을 기반으로 했기 때문에 대상 코호트에 한 번만 들어갈 수 있다.

- 이전에 관심 결과를 이미 경험 한 사람이 코호트에 들어가도록 허용하는가? 대상 코호트에 포함되기 전에 관심 결과를 이미 경험한 사람이 대상 코호트에 다시 들어가도록 허용하는가? 특정 환자-수준 예측 문제에 따라, 결과가 처음 발생하게 되는 것을 예측하고자 할 수 있으며, 이 경우 이전에 결과를 경험한 환자는 그 결과가 처음 발생한 것이 아니므로 대상 코호트에서 제외돼야 한다. 다른 상황에서, 유행하는 에피소드를 예측하고자 하는 경우가 있을 수 있는데, 이로 인해 사전에 관심 결과를 가진 환자가 분석에 포함될 수 있고 사전 결과 자체가 미래 결과를 예측하기 위한 변수가 될 수 있다. 예측을 위한 예제로, 우리는 이전에 혈관 부종을 앓는 사람을 포함하지 않도록 선택할 것이다.
- 대상 코호트 시작 날짜를 기준으로 결과 발생을 예측할 기간을 어떻게 정의하는가? 이 질문에 답하기 위해 두 가지 결정을 내려야 한다. 첫째, 결과가 발생할 수 있는 위험 기간은 대상 코호트가 시작된 그 날짜 (기준 날짜)에 시작되는가 혹은 그 후에 시작되는가? 결과 발생 예측 기간을 나중으로 하자는 주장은, 결과 발생이 실제로는 대상 코호트가 시작되기 전에 이미 발생했지만, 기록이 늦게 되는 경우를 방지하기 원하기 때문이거나, 혹은 결과를 막기 위한 개입이 일어날 수 있는 여유시간을 남겨 두고 싶을 수 있기 때문이다. 둘째, 대상 코호트 시작 또는 종료 날짜를 기준으로 며칠을 오프셋으로 정하여 결과 발생 가능 기간의 끝에 더하여 정의해야 한다. 예제에서는 대상 코호트의 시작일 하루 뒤부터 365일까지를 위험 발생 가능 기간으로 정하여 예측할 것이다.
- 결과 발생 최소 위험 시간 *minimum amount of time-at-risk*이 필요한가? 관심 결과가 생기지는 않았지만, 정의한 위험 발생 기간이 끝나기 전에 데이터베이스에 남지 않는 (자료가 없어서 관찰이 종료된) 환자를 포함할 것인지 결정해야 한다. 그러한 환자는 우리가 그들을 더 이상 관찰하지 않는 동안에 결과가 생길 수도 있다. 예제에서는 위험 노출 최소 시간이 필요한 것으로 했는데, 그 이유는 더 이상 관찰하지 않는 동안에 결과가 생길 수도 있기 때문이다. 또한 이 제약 조건이 결과를 경험한 사람에게도 적용되게 할지 결정해야 한다. 그렇지 않으면 전체 위험 발생 시간과 관계없이 결과가 발생한 모든 사람을 연구에 포함하게 될 것이다. 예를 들어, 결과가 사망인 경우 전체 위험 발생 기간이 완료되기 전에 사망한 사람이 중도 절단될 수 있다.

13.5.3 모델 개발 설정

예측 모델을 개발하기 위해 학습하고자 하는 알고리즘을 결정해야 한다. 특정 예측 문제에 대해서 최상의 알고리즘을 선택하는 방법은 경험적 질문이라고 본다. 즉, 데이터를 이용해서 그 자체가 여러 개의 접근법을 시도하게 하고 그중에서 최상의 것을 선택하도록 하는 것을 선호한다. PLP 프레임워크에서는 13.3절에 설명된 대로 많은 알고리즘을 구현했을 뿐더러, 연구자가 다른 알고리즘을 추가 할 수도 있다. 이 예제에서는 작업을 단순하게 하기 위해 Gradient Boosting Machines 알고리즘을 선택한다.

또한, 모델을 학습하기 위하여 사용할 공변량을 결정해야 한다. 이 예제에서는 성별, 연령, 모든 질병, 의약품과 의약품 그룹 및 방문 횟수를 추가한다. 우리는 기준 날짜 이전부터 1년 전까지의 이러한 여러 임상적 사건을 사용할 것이다.

13.5.4 모델 평가

마지막으로 모델 평가 방법을 정의해야 한다. 편의상 여기서는 내적 타당도를 선택했다. 학습과 테스트를 위한 데이터 모음에서 이것을 나누는 방법과 환자를 이 두 데이터 모음으로 할당하는 방법을 결정해야 한다. 여기에서는 일반적으로 하는 75%-25% 분할을 사용한다. 매우 큰 데이터 모음의 경우 학습데이터 모음에 더 많은 데이터를 포함해서 사용할 수 있다.

13.5.5 연구 요약

우리 연구를 위하여 완벽하게 정의한 내용을 표 13.12에 나타내었다.

Table 13.12: Main design choices for our study.

Choice	Value
Target cohort	Patients who have just started on an ACE inhibitor for the first time. Patients are excluded if they have less than 365 days of prior observation time or have prior angioedema.
Outcome cohort	Angioedema.
Time-at-risk	1 day until 365 days from cohort start. We will require at least 364 days at risk.

Choice	Value
Model	Gradient Boosting Machine with hyper-parameters ntree: 5000, max depth: 4 or 7 or 10 and learning rate: 0.001 or 0.01 or 0.1 or 0.9. Covariates will include gender, age, conditions, drugs, drug groups, and visit count. Data split: 75% train - 25% test, randomly assigned by person.

13.6 ATLAS에서의 연구 구현하기

예측 연구를 설계하기 위한 인터페이스는 ATLAS 메뉴  Prediction 버튼을 누르면 열 수 있다. 새로운 예측 연구를 만든다. 새로운 예측 연구를 생성하라. 만든 연구에 알기 쉬운 이름을 부여하라. 연구 디자인은  버튼을 클릭하면 언제든지 저장이 가능하다.

예측 디자인 기능에는 4개의 세션이 있다: 예측 문제 설정, 분석 설정, 실행 설정, 학습 설정. 다음은 각 세션에 대한 설명이다.

13.6.1 예측 문제 설정

여기서는 대상 코호트와 결과 코호트를 선택할 수 있다. 대상 코호트와 결과 코호트의 모든 조합에 대해 예측 모델이 개발될 것이다. 예를 들어, 만약 두 개의 대상 모집단과 두 개의 대상 결과를 지정한다면 네 개의 예측 문제가 지정된다.

대상 코호트를 선택하기 위해서는 사전에 ATLAS에서 정의하여야 한다. 예시 코호트가 10장에 있다. 부록은 이 예시에 사용된 대상 (부록 B.1)과 결과 (부록 B.4) 코호트의 전체 정의를 제공한다. 대상 모집단을 코호트에 추가하기 위해서는 “대상 코호트 추가 [Add Target Cohort]” 버튼을 클릭 해라. 결과 코호트를 추가하는 것은 “결과 코호트 추가 [Add Outcome Cohort]” 버튼을 클릭하면 마찬가지로 작동된다. 완료되면, Dialog는 그림 13.4처럼 보일 것이다.

13.6.2 분석 설정

분석 설정에서 지도 학습 알고리즘, 공변량 및 모집단 설정을 할 수 있다.

모델 설정

모델 개발을 위해 하나 혹은 더 많은 지도 학습 알고리즘을 선택할 수 있다. 지도 학습 알고리즘을 추가하기 위해서는 “모델 설정 추가 [Add Model Settings]” 버튼을 눌러라. 현재 ATLAS에서 지

The screenshot shows the 'Prediction Problem Settings' interface in ATLAS. It is divided into two main sections: 'Target Cohorts' and 'Outcome Cohorts'.

Target Cohorts:

- Header: 'Target Cohorts' with a 'Remove' button and a 'Name' column.
- Table: A single entry named 'New users of ACE inhibitors as first-line monotherapy for hypertension' with a red 'X' icon.
- Buttons: 'Show 10 entries', 'Filter:', 'Previous 1 Next'.
- Text: 'Showing 1 to 1 of 1 entries'.

Outcome Cohorts:

- Header: 'Outcome Cohorts' with a 'Remove' button and a 'Name' column.
- Table: A single entry named 'Angioedema outcome' with a red 'X' icon.
- Buttons: 'Show 10 entries', 'Filter:', 'Previous 1 Next'.
- Text: 'Showing 1 to 1 of 1 entries'.

Figure 13.4: 예측 문제 설정.

원되는 모든 모델이 포함된 드롭다운이 나타난다. 드롭다운 메뉴에 있는 이름을 클릭하여 원하는 연구가 포함된 지도 학습 모델을 선택 할 수 있다. 그리고 나면 지정된 모델의 창을 볼 수 있고, 하이퍼-파라미터값을 선택 할 수 있다. 여러 값이 제공되는 경우 가능한 모든 값의 조합으로 교차 검증을 사용하여 최적의 조합을 선택하기 위해 그리드서치 Grid Search를 시행한다.

여기의 예에서는 점진적 부스팅 머신 Gradient Boosting Machine(GBM)을 선택하고 그림 13.5에 지정된 것처럼 하이퍼-파라미터를 설정한다.

공변량 설정

우리는 이미 CDM 포맷의 관찰 데이터에서 추출할 수 있는 표준 공변량 covariate을 정의했다. 공변량 설정 창에서 포함할 표준 공변량을 선택할 수 있다. 다양한 유형의 공변량을 정의 할 수 있으며 각 모형은 각각 지정된 공변량 설정과 함께 별도로 생성될 것이다.

연구에서 공변량 설정을 추가하려면, “공변량 설정 추가 [Add Covariate Settings]”을 클릭하라. 그러면 공변량 설정 창이 열린다.

공변량 설정 창의 첫 번째 부분은 제외/포함 옵션이다. 공변량은 일반적으로 모든 개념에 맞게 구성된다. 그러나, 어떤 개념이 대상 코호트 정의와 연결된 경우와 같이 특정한 개념을 추가/제외하기 원할 수도 있다. 특정 개념만 포함하려면 ATLAS에서 개념 모음을 설정한 다음에 “환자-수준 예측 모형의 기저 공변량에 어떤 개념을 포함하시겠습니까? (모든 것을 포함하려면 비워 두십시오) What concepts do you want to include in baseline covariates in the patient-level

Gradient Boosting Machine Model Settings
Use the options below to edit the model settings

The boosting learn rate (default = 0.01,0.1):

Boosting learn rate	Action
0.001	Remove
0.01	Remove
0.1	Remove
0.9	Remove
	Add Reset to default

Maximum number of interactions - a large value will lead to slow model training (default = 4,6,17):

Maximum number of interactions	Action
4	Remove
7	Remove
10	Remove
	Add Reset to default

The minimum number of rows required at each end node of the tree (default = 20):

Minimum number of rows	Action
20	Remove
	Add Using default

The number of trees to build (default = 10,100):

Trees to build	Action
5000	Remove
	Add Reset to default

The number of computer threads to use (how many cores do you have?) (default = 20):

20	Using default
----	-------------------------------

Figure 13.5: 점진적 부스팅 머신 설정.

What concepts do you want to include in baseline covariates in the propensity score model? (Leave blank if you want to include everything)

✖

Should descendant concepts be added to the list of included concepts?

No ▾



What concepts do you want to exclude in baseline covariates in the propensity score model? (Leave blank if you want to include everything)

✖

Should descendant concepts be added to the list of included concepts?

No ▾



A comma delimited list of covariate IDs that should be restricted to:

Figure 13.6: 공변량 포함 및 제외 설정.

prediction model?” 아래 를 클릭하여 개념 모음을 선택한다. “**포함한 개념 목록에 그 하위 개념을 추가합니까? Should descendant concepts be added to the list of included concepts?**”라는 질문에 “예”라고 답함으로써 모든 하위 개념을 자동으로 개념 모음에 추가할 수 있다. 같은 절차는 **환자 수준 예측 모형의 기저 공변량에 어떤 개념을 제외하시겠습니까? (모든 것을 포함하려면 비워 두십시오) What concepts do you want to exclude in baseline covariates in the patient-level prediction model** 질문에서 동일한 과정을 반복할 수 있고 선택된 개념에 해당하는 공변량을 제거할 수 있다. 마지막 옵션인 “**쉼표로 구분되는 공변량 ID의 리스트의 범위를 정한다. A comma delimited list of covariate IDs that should be restricted to**”를 사용하면 공변량의 ID 모음 (개념 ID가 아닌) 을 쉼표로 구분하여 추가할 수 있다. 이 옵션은 고급 사용자에게만 필요하다. 완료되면 포함 및 제외 옵션이 그림 13.6과 같아야 한다.

다음에서는 시간과 연계하지 않는 변수를 선택할 수 있다.

- 성 [Sex]: 남자 또는 여자 성별을 나타내는 이항 변수
- 나이 [Age]: 연령에 해당하는 연속 변수
- 나이 그룹 [Age group]: 5년 단위의 이항 변수 (0-4, 5-9, 10-14, ..., 95+)
- 인종 [Race]: 각 인종의 이항 변수, 1은 환자의 인종 기록이 있음을 의미하며, 그렇지 않으면 0이다.
- 민족 [Ethnicity]: 민족에 대한 이항 변수, 1은 환자의 민족 기록이 있음을 의미하며, 그렇지 않으면 0이다.
- 기준 연도 [Index year]: 각 코호트 연도 시작 날짜에 대한 이항 변수, 1은 환자 코호트 시작 날짜 연도를 의미하고, 그렇지 않으면 0이다. **미래에 이 모형을 적용하기 원할 수 있기 때문에, 때때로는 색인 연도를 포함하는 것은 합리적이지 않다.**
- 기준인 월 [Index month]: 각 코호트 달 시작 날짜에 대한 이항 변수, 1은 환자 코호트 월 시작

Select Covariates

	Gender	Age	Age Groups	Race	Ethnicity	Index Year	Index Month	Prior Observation Time	Post Observation Time	Time In Cohort	Index Year & Month
Demographics	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 13.7: 공변량 선택.

Time bound covariates

Set the time windows for the time bound covariates in days relative to the cohort index

	Any Time Prior	Long Term	Medium Term	Short Term	End Days
Time Windows	All Time	-365	-180	-30	0

Figure 13.8: Time bound 공변량.

날짜를 의미하고, 그렇지 않으면 0이다.

- 사전 관찰 시간 [Prior observation time]: [예측에 권장되지 않음] 코호트 시작일 이전에 환자가 데이터베이스에 있었던 기간(일)에 해당하는 연속 변수
- 사후 관찰 시간 [Post observation time]: [예측에 권장되지 않음] 코호트 시작일 이후에 환자가 데이터베이스에 있었던 기간(일)에 해당하는 연속 변수
- 코호트 내 시간 [Time in cohort]: 환자가 코호트에 있었던 기간에 해당하는 연속 변수 (코호트 종료일에서 코호트 시작일을 뺀)
- 기준 연도 및 월 [Index year and month]: [예측에 권장되지 않음] 각 코호트 시작 연월 날짜에 대한 이항변수, 1은 환자 코호트 시작 연월 날짜이고, 그렇지 않으면 0이다.

완료된다면, 그림 13.7과 같아야 한다.

표준 공변량은 공변량에 대해 세 개의 유동적인 시간 간격을 가능하게 한다:

- 종료일: 코호트 시작 날짜를 기준으로 종료 시기와의 간격 [기본값 0]
- 장기간 [기본 값: 코호트 시작일 이전 365일부터 코호트 시작일 바로 하루 전까지]
- 중기간 [기본 값: 코호트 시작일 이전 180일부터 코호트 시작일 바로 하루 전까지]
- 단기간 [기본 값: 코호트 시작일 이전 30일부터 코호트 시작일 바로 하루 전까지]

완료가 된다면, 그림 13.8과 같을 것이다.

다음 옵션은 ERA 테이블을 이용해 추출한 공변량이다:

- Condition: 각 상태(질병) 개념의 연속된 기간(era)과 선택한 시간 간격을 이용해서 공변량 구축. 연구자가 정한 사전(기저) 관찰 기간 내에 era의 시작과 끝이 포함되거나, 혹은 era의

Set the time bound era covariates

Domain	Any Time Prior	Long Term (-365 days)	Medium Term (-180 days)	Short Term (-30 days)	Overlapping	Era Start		
						Long Term (-365 days)	Medium Term (-180 days)	Short Term (-30 days)
Condition	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Condition Group	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Drug	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Drug Group	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 13.9: Time bound era 공변량.

시작과 끝이 지정한 사전 관찰 기간을 내포하는 경우 공변량 값은 1이고, 그렇지 않으면 0이다.

연구자가 정한 사전 관찰기간은 코호트 시작일보다 이전이어야 한다. (역자 주: 질병 개념의 연속된 기간(era)은 CDM 데이터베이스를 만들 때 모두 계산해서 era 테이블에 저장해 둔다.)

- Condition group: 각 질병 개념 및 그 하위(자식) 개념을 포함하는 연속된 기간 (era)과 선택한 시간 간격을 이용해서 공변량 구축. 환자가 질병 era 테이블에서 코호트 시작일 이전에 지정된 시간 간격 동안, era가 있는 개념 ID 또는 하위 개념 ID(**any descendant concept ID**)를 갖는 경우, 공변량 값은 1이고, 그렇지 않으면 0이다. (역자 주: 위 condition에서 하위(자식) 개념도 포함한 것)
- Drug: 각 약물 개념의 연속된 기간(era)과 선택한 시간 간격을 이용해서 공변량 구축. 환자가 약물 era 테이블에서 코호트 시작일 이전의 지정된 시간 간격 동안 era가 있는 개념 ID를 가지고 있는 경우 공변량 값은 1이며, 그렇지 않으면 0이다. (역자 주: 위 condition 변수와 같은 구성인데 condition era 대신 drug era를 쓴 것)
- Drug group: 각 약물 개념 및 그 하위(자식) 개념을 포함하는 연속된 기간(era)과 선택한 시간 간격을 이용해서 공변량을 구축. 환자가 약물 era 테이블에서 코호트 시작일 이전의 지정된 시간 간격 기간 동안 개념 ID 또는 하위 개념 ID(**any descendant concept ID**)를 가진 경우 공변량 값은 1이며, 그렇지 않으면 0이다.

겹치는 시간 간격 설정은 약물 또는 질병 발생대(era)가 코호트 시작 날짜 이전에 시작하고 코호트 시작 날짜 이후에 끝나야 하므로 코호트 시작 날짜와 겹친다. Era start 옵션은 선택한 시간 간격 동안 시작되는 질병 또는 약물 era를 찾는 것으로 제한된다. (역자 주: 위 drug에서 하위(자식) 개념도 포함한 것)

완료된다면, 그림 13.9과 같이 보일 것이다.

다음은 도메인별 개념 ID에 대해 다양한 시간 간격을 옵션으로 가진 공변량을 선택한다.

- Condition: 각 질병 개념 ID와 선택한 시간 간격으로 공변량을 구축. 코호트 시작일 이전을 기준으로 선택한 시간 간격 동안 질병 개념 ID가 나타나면 공변량 값은 1이며, 그렇지 않으면

0이다. (역자 주: 예를 들어 연구 대상군에 포함된 환자는 100명이고, 그들 환자 데이터에서 코호트 시작일 이전 사전 관찰 기간 동안 총 1,000종류의 고유한 진단명이 나타났다고 하면, 총 1,000개의 진단명 변수가 생성된다. 이하 아래 모든 도메인별 변수도 마찬가지. 결과적으로 입력변수가 아주 많은 초기 토폴로지가 구성되고, 이후 변수 선택과정에서 학습에 기여하지 않는 변수는 제거된다)

- Condition Primary Inpatient: 입원 시 주 진단명 별로 생성되는 이항 공변량.
- Drug: 각 약물 개념 ID 및 시간 간격으로 공변량을 계산하고 약물 노출 테이블의 코호트 시작 일 이전에 지정된 시간 간격 동안 기록된 개념 ID를 가지고 있는 경우 공변량 값은 1, 그렇지 않으면 0이다.
- 시술 Procedure: 각 시술 개념 ID와 선택한 시간 간격으로 공변량을 구축. 환자의 수술 발생 테이블의 코호트 시작일 이전에 지정된 시간 간격 동안 기록된 개념 ID를 가지고 있는 경우 공변량 값은 1, 그렇지 않으면 0이다.
- 검사 Measurement: 각 검사 개념 ID와 선택한 시간 간격으로 공변량을 구축. 환자가 측정 테이블의 코호트 시작일 이전에 지정된 시간 간격 동안 기록된 개념 ID를 가지고 있는 경우 공변량 값은 1, 그렇지 않으면 0이다.
- 검사 값 Measurement Value: 각 검사 개념 ID, 값 및 선택한 시간 간격으로 공변량을 구축. 환자가 측정 테이블의 코호트 시작일 이전에 지정된 시간 간격 동안 기록된 개념 ID를 가지고 있는 경우 값은 측정값, 그렇지 않으면 0이다.
- 검사 값 범위 그룹 Measurement range group: 검사 값이 정상 범위 이하인지 이내인지 또는 그 이상인지를 나타내는 이항 공변량이다.
- 관찰 Observation: 각 관찰 개념 ID와 선택한 시간 간격으로 공변량을 구축. 환자가 관찰 테이블의 코호트 시작일 이전에 지정된 시간 간격 동안 기록된 concept ID를 가지고 있는 경우 공변량 값은 1이며, 그렇지 않으면 0이다.
- 기기 Device: 각 기기 개념 ID와 선택한 시간 간격으로 공변량을 구축. 환자가 기기 테이블의 코호트 시작일 이전에 지정된 시간 간격 동안 기록된 개념 ID를 가지고 있는 경우 공변량 값은 1이며, 그렇지 않으면 0이다.
- 방문 수 Visit count: 각 방문과 선택한 시간 간격으로 공변량을 구축. 시간 간격 동안 기록된 방문 수를 공변량 값으로 계산한다.
- 방문 개념 수 Visit Concept Count: 각 방문 개념 ID와 선택한 시간 간격으로 공변량을 구축. 방문 유형 및 시간 간격 동안 기록된 도메인 당 기록의 수를 공변량 값으로 계산한다.

중복배제 distinct count 옵션은 도메인 및 지정한 시간 간격별로 고유한 개념 IDs의 수를 계산한다.

완료된다면, 그림 13.10과 같이 보일 것이다.

마지막 옵션은 흔하게 사용하는 위험 점수를 공변량으로 포함할 것인지 여부이다. 위험점수 설정을

Set the time bound covariates

Domain	Any Time Prior	Long Term (-365 days)	Medium Term (-180 days)	Short Term (-30 days)	Distinct Count		
					Long Term (-365 days)	Medium Term (-180 days)	Short Term (-30 days)
Condition	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Condition - Primary Inpatient	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Drug	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Procedure	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Measurement	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Measurement - Value	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Measurement - Range Group	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Observation	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Device	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Visit - Count		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
Visit - Concept Count		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

Figure 13.10: Time bound 공변량.

Set the index score covariates

Index Score Type	
CHADS ₂	<input type="checkbox"/>
CHA ₂ DS ₂ VASc	<input checked="" type="checkbox"/>
DCSI	<input checked="" type="checkbox"/>
Charlson	<input checked="" type="checkbox"/>

Figure 13.11: 위험 점수 공변량 설정.

다 마치면 그림 13.11과 같이 된다.

모집단 설정

모집단 설정에서 추가적인 선정 기준과 위험 노출 기간 time-at-risk(TAR)을 연구 대상 군에 적용할 수 있다. 연구에서 모집단 설정을 추가하기 위해서는 “모집단 설정 추가 [Add Population Settings]” 버튼을 클릭하면 모집단 설정 창이 열린다.

옵션의 첫 번째 설정에서 위험 노출 기간을 사용자가 지정할 수 있다. 위험 노출 기간이란 관심 결과가 발생하는지 알아보는 시간 간격이다. 만약 환자의 위험 노출 기간 내에 관심 결과가 있다면 그것을 “결과가 있음 [Has outcome]”이라고 분류하고, 그렇지 않으면 “결과가 없음 [No outcome]”으로 분류할 것이다. “대상 코호트 시작 날짜를 기준으로 위험 노출 기간 시작 정의 Define the time-at-risk window start, relative to target cohort entry:”는 대상 코호트 시작과 끝 일자에 비례하여 위험 노출 기간을 정의한다. 마찬가지로, “위험 노출 기간 종료 정의 Define the time-at-risk window end:”는 위험 노출 기간의 끝을 결정한다.

“최소 기저(선행) 관찰 기간 Minimum lookback period applied to target cohort”는 코호트 시작일 이전에 환자의 기저상태를 관찰하기 위해 필요한 최소의 선행 관찰 기간을 정의한다. 365일이 기본값이다. 이 숫자를 더 크게 하면 환자를 보다 완벽하게 파악할 수 있지만 (더 오래 관찰하므로), 최소 기저 관찰 기간이 짧은 환자는 연구 대상에서 제거될 것이다.

“위험 노출 기간이 없는 대상자를 제거해야 하는가? Should subjects without time at risk be removed?” 을 예로 설정하면, “최소 위험 노출 기간 Minimum time at risk” 값도 필요하다. 이를 통해 추적 기간 동안 중도 절단된 대상 (즉, 위험 노출 기간 동안 데이터베이스에서 떠난 경우) 을 제거 할 수 있다. 예를 들어, 위험 노출 기간이 코호트 시작 다음 날부터 365일까지라면 전체 위험 노출 기간은 364 (365-1)일이다. 만약 전체 관찰 기간에 환자가 포함되길 원한다면 최소 위험 기간을 364일이라고 설정한다. 사람들이 처음 100일 동안 위험 노출 기간이 있는 것이 좋다면, 최소 위험 노출 기간을 100일로 선택한다. 위험 노출 기간의 시작이 코호트 시작으로부터 1일 후이기 때문에 코호트 시작일로부터 적어도 101일 동안 데이터베이스에 남아있는 환자가 포함될 것이다. 만약 “위험 노출 기간이 없는 대상자를 제거해야 하는가? Should subjects without time at risk be removed?”에서 아니오라고 선택하면, 모든 환자, 즉, 위험 노출 기간 동안 데이터베이스에서 이탈한 환자들까지도 연구에 포함될 것이다.

“최소 위험 기간 이내에 결과가 관찰된 사람을 포함하겠습니까? Include people with outcomes who are not observed for the whole at risk period?” 옵션은 이전 옵션과 관련된다. 예라고 설정하면 위험 노출 기간 Time-at-Risk(TAR) 중에 결과가 발생했지만, 최소 위험 기간을 채우지 못해서 탈락할 사람도 연구에 포함된다. (역자 주: 예를 들어 TAR를 365일이라고 설정했는데, 한 환자가 60일째 결과가 발생하고 그 이후 기록이 없으면 그 환자의 TAR은 60일이 되어 연구 대상에서 탈락하게 된다. 이 가능은 그것을 방지하여 설정한 TAR 이내에 결과가 발생해서 TAR이 짧은 환자도 당연히 연구에 포함될 수 있도록 하는 기능이다.)

“대상자마다 첫 번째 노출만 포함되어야 하는가?” 옵션은 한 환자가 코호트 시작일이 다른 여러 개의 관찰 기간을 가질 경우에 유용하다. 예를 선택하면 환자당 가장 처음의 대상 코호트만 유지될 것이다. 그렇지 않으면 같은 환자가 한 데이터 모음에 여러 번 포함되게 된다.

“코호트 시작일 전에 해당 관심 결과가 있던 환자를 제거하시겠습니까? Remove patients who have observed the outcome prior to cohort entry?”에서 예를 선택하면 위험 노출 기간 시작일 이전에 해당 결과가 발생한 환자를 제거할 수 있다. 즉 그 관심 결과를 단 한 번도 경험하지 않은 환자를 위한 모형이 된다. 아니오를 선택한다면 환자는 코호트 시작 이전에도 해당 결과를 경험했을 수 있다. 종종, 이전에 결과를 발생한 경우에는 또다시 그런 결과가 발생할 것이라고 쉽게 예측할 수 있다.

완료된다면, 모집단 설정 화면은 그림 13.12와 같다.

Define the time-at-risk window start, relative to target cohort entry:
 days from

Define the time-at-risk window end:
 days from

Minimum lookback period applied to target cohort:

Should subjects without time at risk be removed?
 Yes Minimum time at risk: days

Include people with outcomes who are not observed for the whole at risk period?
 Yes

Should only the first exposure per subject be included?
 Yes

Remove patients who have observed the outcome prior to cohort entry?
 No

Figure 13.12: 모집단 설정.

이제 분석 설정을 끝냈으므로, 전체적인 설정 화면은 그림 13.13과 같다.

13.6.3 실행 설정

여기에는 세 가지의 옵션이 있다:

- “**표본추출 실행 Perform sampling**”: 표본추출을 실행할지 말지 선택할 수 있다 (기본값 = “아니오 NO”). 만약 “예 yes”라고 설정하면, 또 다른 옵션이 나타날 것이다: “**몇 명의 환자를 부분집합으로 사용할 것인가?**”. 여기서 표본 크기는 결정될 수 있다. 표본추출은 대규모 모집단 (예로 1,000만 환자)을 대상으로 한 모형에서 환자의 표본을 가지고 모형을 테스트하여 구축함으로써 그 모델이 예측력이 있는지 아는데 효율적 수단이 될 것이다. 예를 들어, 그 추출한 표본에서 AUC가 0.5에 가까우면, 그 모형은 쓸모가 없으므로 버려야 할 것이다.
- “**최소 공변량 발생**: 만약 어떤 공변량이 지정한 값보다 적은 수의 대상에서만 발생하면 제거된다: **Minimum covariate occurrence: If a covariate occurs in a fraction of the target population less than this value, it will be removed:**” 여기서 최소 공변량 발생을 선택할 수 있다 (기본값 = 0.001). 전체 모집단을 대표하지 않은 드문 사건을 제거하려면 공변량 발생의 최소 임계값이 필요하다.
- “**공변량 정규화 Normalize covariate**”: 여기서 공변량을 표준화할 것인지 선택할 수 있다 (기본값 = yes). 공변량의 표준화는 LASSO 모형을 적용하기 위해서는 늘 필요하다.

예를 들어, 그림 13.14처럼 선택한다.

Analysis Settings

Model Settings

Show 10 entries Filter:

Remove	Model	Options
	GradientBoostingMachineSettings	{"nTrees": [5000], "nThread": 20, "maxDepth": [4, 7, 10], "minRows": [20], "learnRate": [0.001, 0.01, 0.1, 0.9], "seed": null}

Showing 1 to 1 of 1 entries Previous 1 Next

Covariate Settings

Column visibility Copy CSV Show 10 entries Filter:

Remove	Options
	DemographicsGender, DemographicsAgeGroup, DemographicsRace, DemographicsEthnicity, DemographicsIndexMonth, ConditionGroupEraLongTerm (+12 more covariate settings)

Showing 1 to 1 of 1 entries Previous 1 Next

Population Settings

Column visibility Copy CSV Show 10 entries Filter:

Remove	Risk Window Start	Risk Window End	Washout Period	Include All Outcomes	Remove Subjects With Prior Outcome	Minimum Time At Risk
	1d from cohort start date	365d from cohort start date	365d	true	false	364d

Showing 1 to 1 of 1 entries Previous 1 Next

Figure 13.13: 분석 설정.

Execution Settings

Perform sampling:

Yes ▾

How many patients to use for a subset? 500000 patients

Minimum covariate occurrence: If a covariate occurs in a fraction of the target population less than this value, it will be removed:

0.001

Normalize covariates:

Yes ▾

Figure 13.14: 실행 설정.

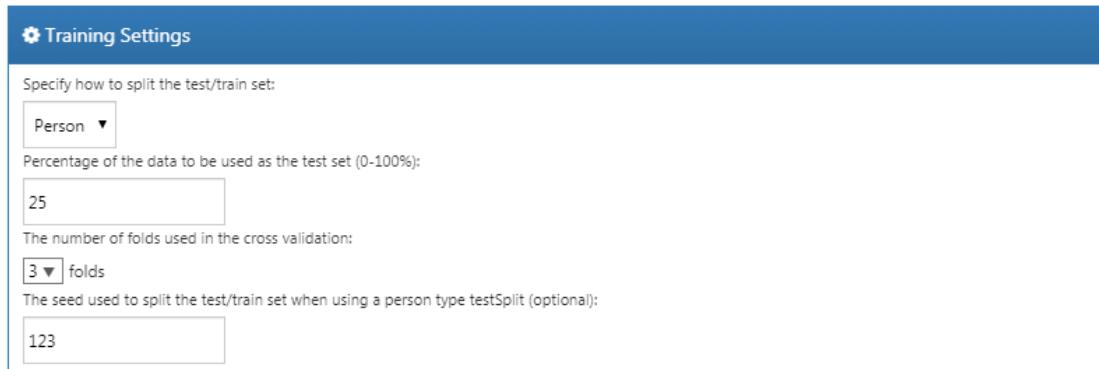


Figure 13.15: 학습 설정.

13.6.4 학습 설정

여기에는 네 가지 옵션이 있다:

- “검증/학습 모음을 분할하는 방법 지정 Specify how to split the test/train set.”: 학습/검증 데이터를 사람 (관심 결과에 따라 분류됨) 별로 구분할지, 시간 (모형을 학습하기 위해서는 이전 데이터를, 모형을 평가하기 위해서는 최근 데이터 사용) 별로 구분할지를 선택한다.
- “테스트 모음으로 사용될 데이터의 백분율 Percentage of the data to be used as the test set (0-100%)”: 테스트 데이터로 사용될 데이터의 백분율을 선택한다 (기본값 = 25%).
- “교차 검정에 사용된 폴드 수 The number of folds used in the cross validation”: 최적 하이퍼-파라미터 선택에 사용되는 교차 검증을 위한 폴드 수를 선택한다 (기본값 = 3).
- “사람을 기준으로 검증/학습 모음을 분할하는 경우, 사용할 시드 (선택적) The seed used to split the test/train set when using a person type testSplit”: 사용자 유형 testSplit에서 학습/검증 모음을 분할하는 데 사용되는 seed로 임의의 초기값을 선택한다.

이 예에서는 그림 13.15와 같이 선택했다.

13.6.5 연구 가져오기 및 내보내기

연구를 내보내려면 “유ти리티 Utilities” 아래의 “내보내기 Export” 탭을 클릭한다. ATLAS는 연구를 실행할 때 필요한 연구 이름, 코호트 정의, 선택된 모형, 공변량, 설정과 같은 모든 데이터를 직접 복사하여 붙여넣을 수 있는 JSON 형식의 파일을 생성할 것이다. 연구를 가져오려면 “유ти리티 Utilities” 아래의 “가져오기 Import” 탭을 클릭한다. 환자 수준 예측 연구 JSON 파일의 내용을 이 창에 붙여넣은 다음 다른 탭 버튼 아래에 있는 가져오기 버튼을 클릭한다. 이 작업은 해당 연구에 대한 이전 설정을 모두 덮어쓰므로 일반적으로 비어있는 새 연구 디자인에 수행해야 한다는 점을 유의해야 한다.

13.6.6 연구 패키지 다운로드

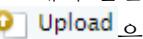
“유ти리티 Utilities” 탭 아래의 “리뷰 & 다운로드 Review & Download” 클릭한다. “연구 패키지 다운로드 Download Study Package” 부분에서 R 패키지 이름으로 허용되지 않은 모든 문자는 ATLAS가 파일 이름에서 자동으로 제거한다는 점에 유의하여 R 패키지 이름을 작성해야 한다.  **Download**를 클릭하여 R 패키지를 로컬 폴더로 다운로드할 수 있다.

13.6.7 연구 실행

R 패키지를 실행하려면 8.4.5절에 설명된 대로 R, RStudio, Java가 설치되어 있어야 한다. 또한, 다음과 같이 R에 설치할 수 있는 PatientLevelPrediction 패키지가 필요하다:

```
install.packages("drat")
drat::addRepo("OHDSI")
install.packages("PatientLevelPrediction")
```

기계학습 알고리즘의 몇몇은 추가적인 소프트웨어 설치를 요구한다. PatientLevelPrediction 패키지를 설치하는 방법에 대한 자세한 내용은 “Patient-Level Prediction Installation Guide” vignette를 참조하면 된다.

작성된 연구 R 패키지를 사용하려면 R Studio를 이용하는 것을 추천한다. 로컬에서 R Studio를 사용 중인 경우 ATLAS에서 생성한 파일의 압축을 풀고 .Rproj를 두 번 누르면 RStudio에서 열린다. RStudio 서버에서 RStudio를 실행 중인 경우 파일을 업로드하고 압축을 푼 다음  **Upload**을 클릭하여 연구프로젝트를 열면 된다.

일단 R Studio에서 연구프로젝트를 열면 README 파일을 열 수 있고 그 설명을 따르면 된다. 모든 파일 경로를 시스템의 기존 경로로 변경해야 한다.

13.7 R에서의 연구 실행

ATLAS 사용하여 연구 디자인을 실행하는 방법은 R에서 코드를 직접 작성하는 것이다. PatientLevelPrediction 패키지에서 제공하는 함수를 사용할 수 있다. 패키지는 OMOP CDM으로 변환된 데이터로부터 데이터 추출, 모델 구축 및 모델 평가를 가능하게 한다.

13.7.1 코호트 예시화

우선 대상 코호트와 결과 코호트를 만들어야 한다. 코호트 만드는 것은 10장에 설명되어 있다. 부록 예선 대상 코호트 (부록 B.1)와 결과 코호트 (부록 B.4)에 대한 전체 정의를 제공한다. 이 예제에서

우리는 ACE 억제제 코호트를 ID 1, 혈관부종 코호트를 ID 2로 가정한다.

13.7.2 데이터 추출

우선 R에서 서버를 연결해야 한다. `PatientLevelPrediction`은 다양한 데이터베이스 관리 시스템(DBMS)에 필요한 구체적인 설정을 위한 `createConnectionDetails`. Type `?createConnectionDetails` 라고 불리는 기능을 제공하는 `DatabaseConnector` 패키지를 사용한다. 예를 들어 이 코드를 사용하여 PostgreSQL 데이터베이스에 연결할 수 있다.

```
library(PatientLevelPrediction)
connDetails <- createConnectionDetails(dbms = "postgresql",
                                         server = "localhost/ohdsi",
                                         user = "joe",
                                         password = "supersecret")

cdmDbSchema <- "my_cdm_data"
cohortsDbSchema <- "scratch"
cohortsDbTable <- "my_cohorts"
cdmVersion <- "5"
```

마지막 4줄은 `cdmDbSchema`, `cohortsDbSchema` 및 `cohortsDbTable` 변수와 CDM 버전을 정의한다. 나중에 이것을 사용하여 CDM 형식의 데이터 위치, 관심 있는 코호트 위치, 사용되는 CDM 버전을 R에 입력한다. Microsoft SQL의 경우 데이터베이스 스키마는 데이터와 스키마 모두 지정해야 한다. 예를 들어 `cdmDbSchema <- "my_cdm_data.dbo"`이다.

먼저 코호트 항목 수를 세어 코호트 생성이 되었는지 확인하는 것이 좋다:

```
sql <- paste("SELECT cohort_definition_id, COUNT(*) AS count",
             "FROM @cohortsDbSchema.cohortsDbTable",
             "GROUP BY cohort_definition_id")
conn <- connect(connDetails)
renderTranslateQuerySql(connection = conn,
                        sql = sql,
                        cohortsDbSchema = cohortsDbSchema,
                        cohortsDbTable = cohortsDbTable)
```

```
## cohort_definition_id count
```

```
## 1          1 527616
## 2          2 3201
```

이제 PatientLevelPrediction을 통해 분석에 필요한 모든 데이터를 추출할 수 있다. 공변량은 FeatureExtraction 패키지를 사용하여 추출된다. FeatureExtraction 패키지에 대한 자세한 내용은 해당 vignettes에서 볼 수 있다. 예제 연구에서는 다음과 같은 설정을 사용하기로 하였다:

```
covariateSettings <- createCovariateSettings(
  useDemographicsGender = TRUE,
  useDemographicsAge = TRUE,
  useConditionGroupEraLongTerm = TRUE,
  useConditionGroupEraAnyTimePrior = TRUE,
  useDrugGroupEraLongTerm = TRUE,
  useDrugGroupEraAnyTimePrior = TRUE,
  useVisitConceptCountLongTerm = TRUE,
  longTermStartDays = -365,
  endDays = -1)
```

데이터를 추출의 마지막 단계는 getPlpData 함수를 실행하고 코호트가 저장되는 데이터베이스 스키마, 코호트와 결과를 위한 코호트 정의 ID, 해당 사람이 데이터에 포함되도록 관찰되어야 하는 코호트 색인 일자 이전의 최소 일자인 최소 휴약기 washout period와 같은 연결 세부사항을 입력하는 것이고 마지막으로 이전에 생성된 공변량 설정을 입력하는 것이다.

```
plpData <- getPlpData(connectionDetails = connDetails,
  cdmDatabaseSchema = cdmDbSchema,
  cohortDatabaseSchema = cohortsDbSchema,
  cohortTable = cohortsDbSchema,
  cohortId = 1,
  covariateSettings = covariateSettings,
  outcomeDatabaseSchema = cohortsDbSchema,
  outcomeTable = cohortsDbSchema,
  outcomeIds = 2,
  sampleSize = 10000
)
```

(번역 생략된 부분) There are many additional parameters for the getPlpData function which are all documented in the PatientLevelPrediction manual. The resulting plpData object uses

the package `ff` to store information in a way that ensures R does not run out of memory, even when the data are large. (번역 생략된 부분) Creating the `plpData` object can take considerable computing time, and it is probably a good idea to save it for future sessions. Because `plpData` uses `ff`, we cannot use R’s regular `save` function. Instead, we’ll have to use the `savePlpData` function:

```
savePlpData(plpData, "angio_in_ace_data")
```

(번역 생략된 부분) We can use the `loadPlpData()` function to load the data in a future session.

13.7.3 추가 포함 기준

최종 연구 모집단은 이전에 정의된 2개의 코호트에서 추가적인 제약조건을 적용하여 얻어진다. 예를 들어 최소 위험에 노출된 시간을 적용할 수 있으며 (`requireTimeAtRisk`, `minTimeAtRisk`), 이것이 결과를 가진 환자에게도 적용되는지 여부를 지정할 수 있다 (`includeAllOutcomes`). 또한 여기서 대상 코호트 시작에 관련된 위험 기간 risk window의 시작과 끝을 지정한다. 예를 들어 위험 코호트가 시작된 후 30일부터 위험 기간으로 시작하고 1년 후 종료하려면 `riskWindowStart = 30`, `riskWindowEnd = 365`로 설정할 수 있다. 때에 따라 위험 기간은 코호트 종료일에 시작해야 한다. `addExposureToStart = True`로 설정하면 코호트 (노출) 시간을 시작일에 추가할 수 있다.

아래의 예에서는 연구를 위해 정의한 모든 설정을 시행할 것이다:

13.7.4 모델 개발

알고리즘의 설정 기능에서 사용자는 각 하이퍼-파라미터에 대한 적합한 값의 목록을 지정할 수 있다. 하이퍼-파라미터에서 가능한 모든 조합은 학습 모음에 교차 검증을 사용하는 이른바 그리드서치에 포함된다. 만일 사용자가 어떤 값도 지정하지 않으면 기본값이 사용된다.

예를 들어 점진적 부스팅 머신에 다음 설정을 사용하는 경우: `ntrees = c(100,200)`, `maxDepth = 4` 그리드서치는 점진적 부스팅 머신 알고리즘을 다른 하이퍼-파라미터의 기본 설정을 더한 `ntrees = 100`과 `maxDepth = 4`, 다른 하이퍼-파라미터의 기본 설정을 더한 `ntrees = 200`과 `maxDepth = 4`에 적용할 것이다. 최고의 교차 검증 실행을 이끄는 하이퍼-파라미터는 마지막 모델로 선택될 것이다. 이 문제를 위해 여러 하이퍼-파라미터값을 가지고 점진적 부스팅 머신을 만들기로 하였다:

```
gbmModel <- setGradientBoostingMachine(ntrees = 5000,
                                         maxDepth = c(4,7,10),
                                         learnRate = c(0.001,0.01,0.1,0.9))
```

`runPlp` 함수는 모델을 훈련하고 평가하기 위해 모집단, `plpData` 및 모델 설정을 사용한다. 데이터를 75% ~ 25%로 분할하기 위해 `testSplit`(사람/시간)과 `testFraction` 파라미터를 사용하고 환자 수준 예측 파이프라인을 실행할 수 있다:

```
gbmResults <- runPlp(population = population,
                      plpData = plpData,
                      modelSettings = gbmModel,
                      testSplit = 'person',
                      testFraction = 0.25,
                      nfold = 2,
                      splitSeed = 1234)
```

패키지 안에 R `xgboost` 패키지를 사용하여 데이터의 75%를 사용하는 점진적 부스팅 머신 모델을 맞추고 나머지 25%에 대해 모델을 평가한다. 결과 데이터 구조는 모델과 성능에 대한 정보가 포함되어 있다.

`runPlp` 함수에는 기본적으로 TRUE로 설정된 `plpData`, `plpResults`, `plpplots`, `evaluation` 등을 저장할 수 있는 몇 가지 파라미터가 있다.

다음을 사용하여 모델을 저장할 수 있다:

```
savePlpModel(gbmResults$model, dirPath = "model")
```

(번역 생략된 부분) We can load the model using:

```
plpModel <- loadPlpModel("model")
```

(번역 생략된 부분) You can also save the full results structure using:

```
savePlpResult(gbmResults, location = "gbmResults")
```

(번역 생략된 부분) To load the full results structure use:

```
gbmResults <- loadPlpResult("gbmResults")
```

13.7.5 내부 검증

연구를 실행하면 runPlp함수는 학습/테스트 모음에서 학습된 모델과 학습/테스트 모음에서 모델의 평가를 해준다. viewPLP(runPLP = gbmResults)를 실행하여 양방향의 결과를 볼 수 있다. 이것은 대화식 그림을 포함하여 프레임워크에 생성한 모든 측정값을 볼 수 있는 Shiny 앱을 열 것이다 (Shiny 애플리케이션 섹션의 그림 13.16 참조).

모든 평가 그림을 폴더에 생성하고 저장하려면 다음 코드를 실행하면 된다:

```
plotPlp(gbmResults, "plots")
```

도표는 13.4.2장에서 더욱 자세하게 기술된다.

13.7.6 외부 검증

항상 외적 타당도를 수행하는 것을 권장한다. 즉 가능한 많은 새로운 데이터에 최종모델을 적용하고 성능을 평가해야 한다. 여기서 이미 두 번째 데이터베이스에서 데이터 추출이 수행되어 newData 폴더에 저장되었다고 가정한다. 이전에 장착된 모델을 model 폴더로부터 로딩한다.

```
# load the trained model
plpModel <- loadPlpModel("model")
```

```

#load the new plpData and create the population
plpData <- loadPlpData("newData")

population <- createStudyPopulation(plpData =
                                      outcomeId = 2,
                                      washoutPeriod = 364,
                                      firstExposureOnly = FALSE,
                                      removeSubjectsWithPriorOutcome = TRUE,
                                      priorOutcomeLookback = 9999,
                                      riskWindowStart = 1,
                                      riskWindowEnd = 365,
                                      addExposureDaysToStart = FALSE,
                                      addExposureDaysToEnd = FALSE,
                                      minTimeAtRisk = 364,
                                      requireTimeAtRisk = TRUE,
                                      includeAllOutcomes = TRUE
)

# apply the trained model on the new data
validationResults <- applyModel(population, plpData, plpModel)

```

또한 필요한 데이터를 추출하는 외부 검증을 더욱 쉽게 하기 위해 externalValidatePLP 함수를 제공한다. `result <- runPlp(...)`을 실행했다고 가정했을 때 모델에 필요한 데이터를 추출하여 새 데이터에 대해 평가할 수 있다. 검증 코드트가 ID 1과 2가 있는 테이블 `mainschema.dob.cohort` 과 CDM 데이터가 스키마 `cdmschema.dob` 라고 가정한다:

```

valResult <- externalValidatePlp(
  plpResult = result,
  connectionDetails = connectionDetails,
  validationSchemaTarget = 'mainschema.dob',
  validationSchemaOutcome = 'mainschema.dob',
  validationSchemaCdm = 'cdmschema dbo',
  databaseNames = 'new database',
  validationTableTarget = 'cohort',
  validationTableOutcome = 'cohort',
  validationIdTarget = 1,
)

```

```
    validationIdOutcome = 2  
)
```

모델을 검증할 데이터베이스가 여러 개 있는 경우 다음을 실행 할 수 있다:

```
valResults <- externalValidatePlp(  
  plpResult = result,  
  connectionDetails = connectionDetails,  
  validationSchemaTarget = list('mainschema.dob',  
    'difschema.dob',  
    'anotherschema.dob'),  
  validationSchemaOutcome = list('mainschema.dob',  
    'difschema.dob',  
    'anotherschema.dob'),  
  validationSchemaCdm = list('cdms1schema.dbo',  
    'cdm2schema.dbo',  
    'cdm3schema.dbo'),  
  databaseNames = list('new database 1',  
    'new database 2',  
    'new database 3'),  
  validationTableTarget = list('cohort1',  
    'cohort2',  
    'cohort3'),  
  validationTableOutcome = list('cohort1',  
    'cohort2',  
    'cohort3'),  
  validationIdTarget = list(1,3,5),  
  validationIdOutcome = list(2,4,6)  
)
```

13.8 결과 보급

13.8.1 모델 성능

viewPlp 함수를 사용하면 예측 모델의 성능을 탐색하기에 가장 쉽다. 이 함수는 결과 객체를 입력값으로 한다. R에서 모델을 개발하는 경우에는 runPlp의 결과를 입력값으로 사용할 수 있다. ATLAS

로 만든 연구 패키지를 이용한다면 모델 중 하나를 로딩해야 한다 (이 예제에서는 Analysis_1을 로딩할 것이다):

```
plpResult <- loadPlpResult(file.path(outputFolder,
                                         'Analysis_1',
                                         'plpResult'))
```

여기서 “Analysis_1”은 앞에서 설명한 분석에 해당한다.

이후에 다음을 실행하여 Shiny 앱을 시작할 수 있다:

```
viewPlp(plpResult)
```

Shiny 앱은 그림 13.16에서 볼 수 있듯이 훈련 모음과 평가 모음에 있는 성능 지표의 요약본을 보여준다. 결과를 보면 훈련 모음에서 AUC가 0.78이고 평가 모음에서는 0.74까지 떨어지는 것을 보여준다. 평가 모음 AUC가 좀 더 정확한 측정값이다. 전반적으로 이 모델은 ACE 억제제를 처음 사용하는 사용자에서 결과를 예측할 수 있을 것처럼 보이지만 훈련 모음이 평가 모음보다 성능이 더 좋기 때문에 다소 과적합 되었다. ROC 도표는 그림 13.17에 제시되어 있다.

그림 13.18에 있는 모델 적합 도표는 점들이 대각선 주위에 있을 때 일반적으로 관찰된 위험이 예측된 위험과 일치함을 보여준다. 그러나 그림 13.19에 있는 인구통계학적 그래프는 하늘색 선 (예측위험)이 40세 미만의 적색선 (관측된 위험)과 다르기 때문에 모델이 젊은 환자에 대해 잘 보정되지 않았음을 보여준다. 이것은 대상 모집단 target population에서 40대 미만을 제거해야 할 필요가 있다는 것을 보여 준다 (젊은 환자에서는 관찰된 위험이 거의 0이므로).

마지막으로, 손실 도표 attrition plot는 선정/제외 기준에 기초해서 라벨링 된 데이터에서 손실된 환자 수를 나타낸다 (그림 13.20 참조). 아래 표는 대상 모집단에서 많은 연구대상자가 위험 노출 기간 조건 (1년)을 만족하지 못해서 손실됐다는 것을 보여준다. 흥미롭게도, 결과가 발생한 환자는 상대적으로 적게 손실되었다.

13.8.2 모델간 비교

ATLAS가 생성한 연구 패키지는 다른 예측 문제에 대해 다른 예측 모델을 생성하고 평가하는데 사용할 수 있다. 이를 위해서 특별히, 연구 패키지가 생성한 결과에 대해 여러 모델을 볼 수 있도록 Shiny 앱을 개발하였다. 앱을 시작하기 위해서는 `viewMultiplePlp(outputFolder)`라고 실행하는데 `outputFolder`는 `execute` 명령을 실행할 때 지정된 분석 결과를 저장하는 경로이다 (우리 예제에서는 “Analysis_1”이라는 하위 폴더가 포함되어야 한다).

The screenshot shows a Shiny application interface. At the top, there's a header bar with a logo, the URL "http://127.0.0.1:4888", an "Open in Browser" button, and a "Publish" dropdown. Below the header, the title "PatientLevelPrediction Explorer" is followed by tabs for "Internal Validation" and "External Validation". A horizontal navigation bar contains links for "Evaluation Summary", "Characterization", "ROC", "Calibration", "Demographics", "Preference", "Box Plot", and "Settings". The "Evaluation Summary" tab is currently selected and highlighted with a blue border. The main content area is titled "Evaluation Summary" and contains a table. The table has a header row with columns for "Metric", "test", and "train". There are 11 data rows, each corresponding to a metric: AUC, AUC_lb95ci, AUC_ub95ci, AUPRC, BrierScaled, BrierScore, CalibrationIntercept.Intercept, CalibrationSlope.Gradient, outcomeCount, populationSize, and Incidence. The "test" column values are: 0.72130, 0.70057, 0.74203, 0.10971, 0.03755, 0.03355, -0.00089, 1.02041, 601.00000, 16685.00000, and 3.60204. The "train" column values are: 0.75348, 0.74215, 0.76482, 0.13571, 0.04902, 0.03304, -0.00813, 1.22457, 1802.00000, 50054.00000, and 3.60011. Below the table, a message says "Showing 1 to 11 of 11 entries". At the bottom right, there are buttons for "Previous", "1" (selected), and "Next".

Metric	test	train
1 AUC	0.72130	0.75348
2 AUC_lb95ci	0.70057	0.74215
3 AUC_ub95ci	0.74203	0.76482
4 AUPRC	0.10971	0.13571
5 BrierScaled	0.03755	0.04902
6 BrierScore	0.03355	0.03304
7 CalibrationIntercept.Intercept	-0.00089	-0.00813
8 CalibrationSlope.Gradient	1.02041	1.22457
9 outcomeCount	601.00000	1802.00000
10 populationSize	16685.00000	50054.00000
11 Incidence	3.60204	3.60011

Figure 13.16: Shiny 앱에서의 요약 평가 통계.

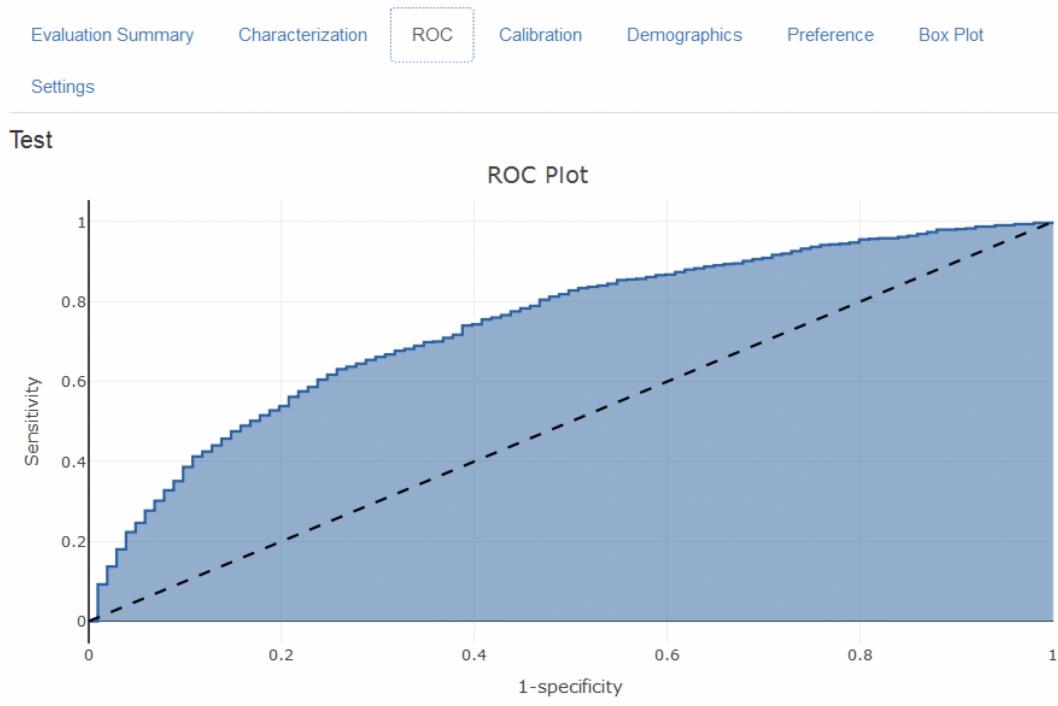


Figure 13.17: ROC 도표.

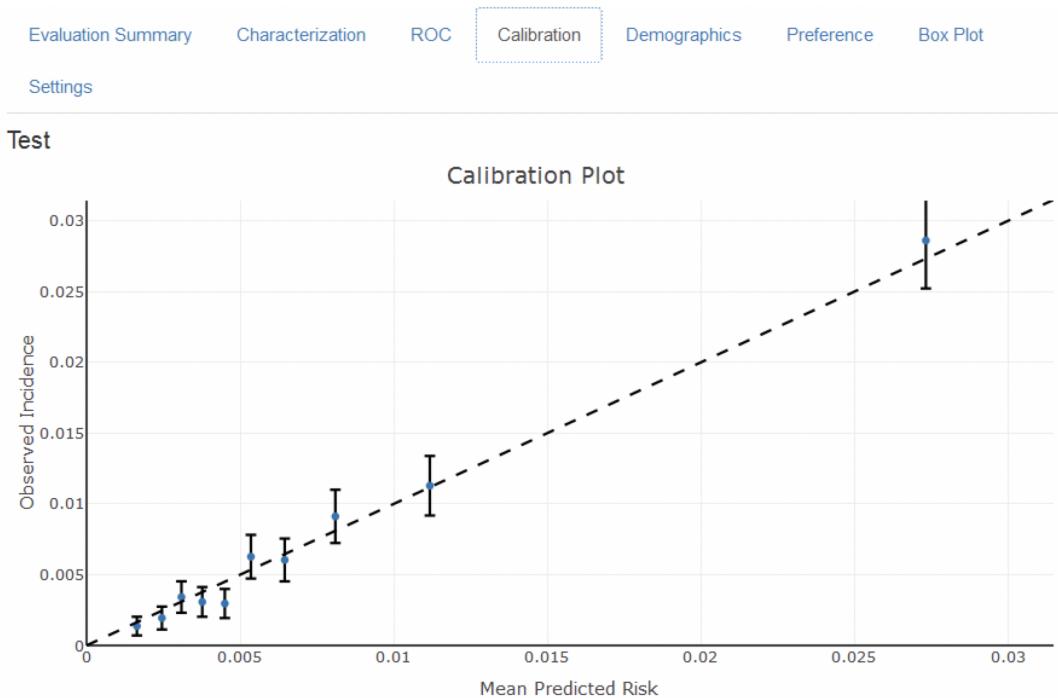


Figure 13.18: 모델 보정

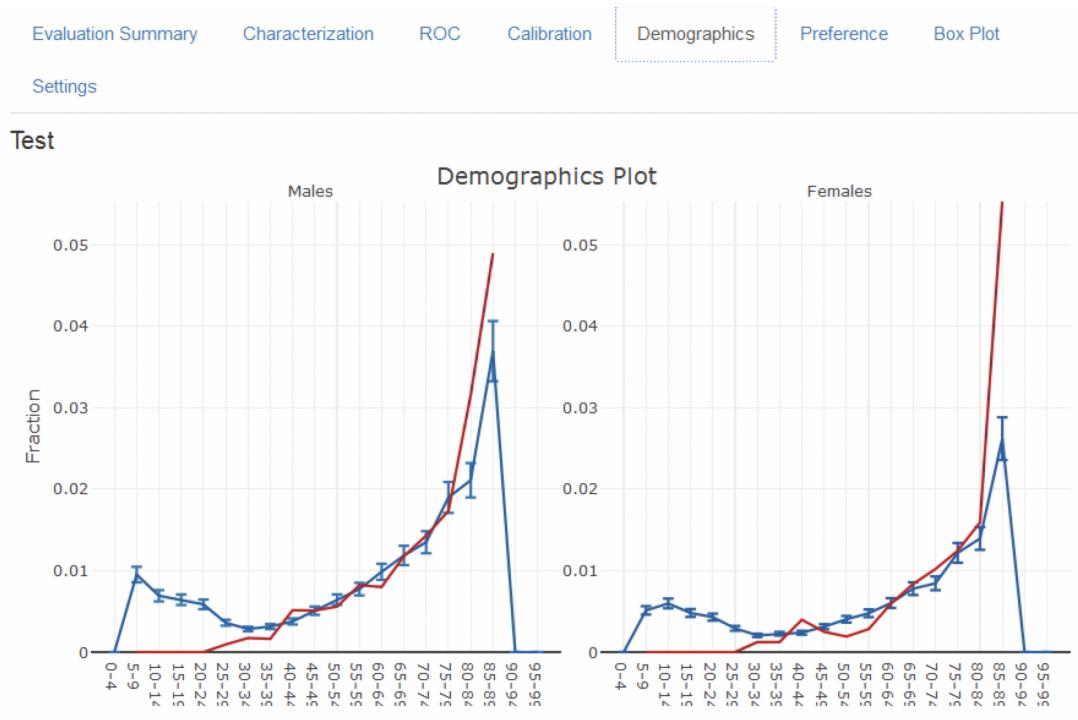


Figure 13.19: 모델의 인구 통계학적 보정

The table displays information about four cohorts used in the analysis. It includes columns for description, targetCount, uniquePeople, and outcomes.

	description	targetCount	uniquePeople	outcomes
1	Original cohorts	500000	500000	13746
2	First exposure only	500000	500000	13746
3	At least 365 days of observation prior	500000	500000	13746
4	Have time at risk	351028	351028	12726

Showing 1 to 4 of 4 entries

Previous 1 Next

Figure 13.20: 분석한 예측 문제에서의 손실 도표

The screenshot shows a shiny web application interface. On the left, there are four dropdown menus under 'Filters': 'Development Database' (All), 'Validation Database' (All), 'Target Cohort' (New users of ACE inhibitors as first-line monotherapy for hypertension selected), and 'Outcome Cohort' (All). At the top right, there are tabs for 'Results', 'Model Settings', 'Population Settings', and 'Covariate Settings'. Below these are buttons for 'Show 10 entries' and 'Search'. The main area contains a table with columns: Analysis, Dev, Val, T, O, Model, TAR start, TAR end, AUC, AUPRC, T Size, O Count, and Incidence (%). Four rows of data are shown:

Analysis	Dev	Val	T	O	Model	TAR start	TAR end	AUC	AUPRC	T Size	O Count	Incidence (%)
Analysis_1	Optum claims	Optum claims	New users of ACE inhibitors as first-line monotherapy for hypertension	Acute myocardial infarction events	Lasso Logistic Regression	1	365	0.74486	0.03094	87757	650	0.74068
Analysis_3	Optum claims	Optum claims	New users of ACE inhibitors as first-line monotherapy for hypertension	Angioedema events	Lasso Logistic Regression	1	365	0.60523	0.00254	87615	148	0.16892
Analysis_5	Optum claims	Optum claims	New users of ACE inhibitors as first-line monotherapy for hypertension	Acute myocardial infarction events	Random forest	1	365	0.71687	0.03102	87757	650	0.74068
Analysis_7	Optum claims	Optum claims	New users of ACE inhibitors as first-line monotherapy for hypertension	Angioedema events	Random forest	1	365	0.64163	0.02447	87615	148	0.16892

At the bottom left, it says 'Showing 1 to 4 of 4 entries'. At the bottom right, there are buttons for 'Previous', '1', and 'Next'.

Figure 13.21: 학습된 각 모델에 대해서 핵심 성능 지표를 보여주는 shiny 요약 페이지

모델 요약 보기 및 설정

대화형 Shiny 앱은 그림 13.21과 같이 요약 페이지를 보여준다.

요약 페이지 테이블에는 아래 내용이 있다:

- 모델에 관한 기본 정보 (예를 들어, 데이터베이스 정보, 분류 유형, 위험 노출 기간 설정, 대상 모집단, 결과 명)
- 대상 모집단 수와 결과 발생률
- 판별 지표 discrimination metrics: AUC, AUPRC

표 왼쪽에는 필터 옵션이 있는데 여기서는 해당 코호트의 개발/검증 데이터베이스, 모델 유형, 관심 있는 위험 노출 기간 설정이 있다. 예를 들어, 대상 모집단 “고혈압의 일차 단일 요법으로 ACE 억제제의 신규 사용자 [New users of ACE inhibitors as first line mono-therapy for hypertension]”에 해당하는 모델을 선택하려면 대상 코호트/[Target Cohort] 옵션을 선택하면 된다.

해당 행을 클릭하여 모델을 탐색하면, 선택된 행이 강조 표시된다. 행을 선택하면 모델 설정 Model Settings 탭을 클릭하여 모델을 개발할 때 사용되는 모델 설정을 탐색할 수 있다:

비슷하게, 다른 탭에서 모델을 생성하는데 사용된 모집단과 공변량 설정을 탐색할 수 있다.

모델 성능 보기

일단 모델 행을 선택하면 모델 성능도 볼 수 있다. 임계값 성능 요약을 보기 위하여 를 클릭하면 그림 13.23처럼 나타난다.

이 요약 보기에는 표준화한 형식의 예측 질문, 임계값 선택기 threshold selector 및 대시보드에 양성예측도 PPV, 음성예측도 NPV, 민감도 및 특이도 (13.4.2절 참조)와 같은 주요 임계값 기반 지표를 포함하고 있다. 그림 13.23에서 임계값 0.00482에서 민감도는 83.4% (다음 1년간 결과가

Model Settings: help

Show 10 entries

	Setting	Value
1	Model	lr_lasso
2	variance	0.01
3	seed	50975614

Showing 1 to 3 of 3 entries

Figure 13.22: 모델을 개발에 사용된 설정 보기.

Prediction Question

Within New users of ACE inhibitors as first-line monotherapy for hypertension predict who will develop Acute myocardial infarction events during 1 day/s after cohort start and 365 day/s after cohort start

Input

Threshold value slider: **0.0048175**

Dashboard

THRESH...	INCIDENCE	PPV
0.00482	0.741%	1.2%
SPECIFICITY	SENSITIVITY	NPV
49.2%	83.4%	99.7%

Cutoff Performance

Ground Truth Negative	Ground Truth Positive
Predicted Positive	44215
Predicted Negative	542
	42892
	108

Figure 13.23: 설정된 임계값에 따른 성능 측정값 요약표

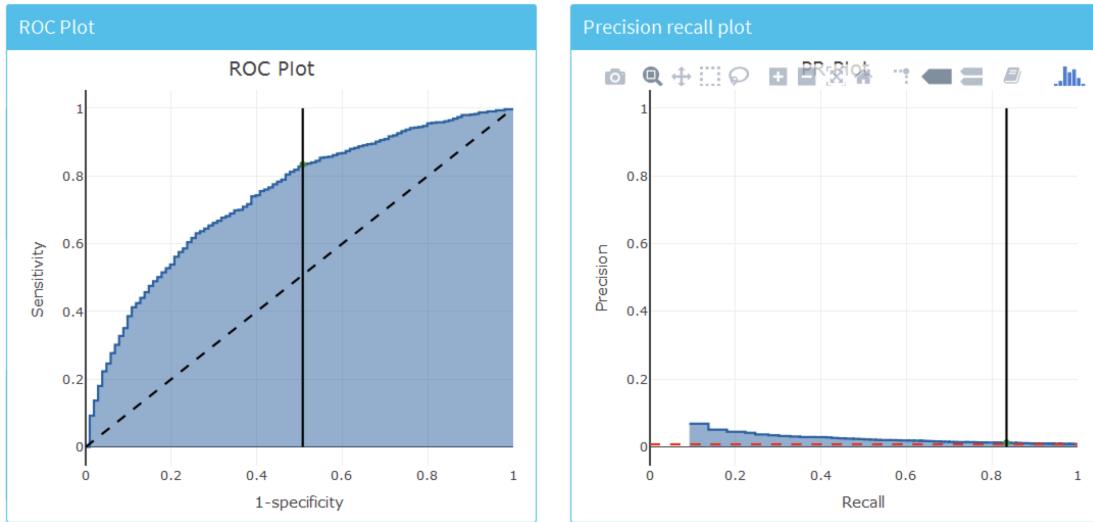


Figure 13.24: 모델의 전체적인 판별도를 평가하기 위한 ROC와 정밀도-검출률 도표.

발생한 83.4% 환자는 0.00482 이상의 위험을 가지고 있다) 이고 PPV는 1.2% (0.00482보다 크거나 같은 위험을 가진 환자의 1.2%는 다음 1년간 그 결과가 발생한다)이다. 연간 결과 발생률이 0.741% 이므로 위험이 0.00482 이상인 환자를 식별하는 것은 모집단의 평균 위험의 거의 두 배 (1.2%)가 되는 환자그룹을 찾을 수 있다는 것이다. 슬라이더를 사용하여 임계값을 조정할 수 있다.

모델의 전체적인 예측력을 보려면 “Discrimination” 템을 클릭하면 ROC 도표, 정밀도-검출률 precision-recall 도표, 분포 도표를 볼 수 있다. 그림의 수직선은 선택한 임계값 포인트에 해당한다. 그림 13.24는 ROC와 정밀도-검출률 도표를 보여준다. ROC 도표는 모델이 1년 이내에 결과가 생길 사람과 그렇지 않은 사람을 구별할 수 있음을 보여준다. 그러나 결과 발생률이 낮다는 것은 거짓 양성률이 높다는 것을 의미하기 때문에 정밀도-재현율 도표를 보면 성능이 그다지 인상적이지는 않아 보인다.

그림 13.25는 예측과 선호 점수 분포를 보여준다.

마지막으로 “Calibration” 템을 클릭하여 모델의 보정 calibration을 조사할 수 있다. 그림 13.26은 보정 도표 calibration plot와 인구통계학적 보정을 보여준다.

평균 예측 위험은 1년 이내 결과를 경험한 관측된 비율과 일치하는 것으로 나타나므로 모델은 잘 보정되어 있다. 흥미롭게도 인구통계학적 보정을 보면 젊은 환자에게서 기대 선이 관찰 선보다 더 높게 나왔다는 것을 보여주고 있어서, 모델이 젊은 연령 집단에 있어서 위험을 실제보다 더 높게 예측하는 것을 알 수 있다. 이것은 젊은 환자나 고령 환자를 위한 모델을 분리하여 별도로 개발해야 할 수도 있다는 것을 의미한다.



Figure 13.25: 결과가 생긴 군과 생기지 않은 군에서 예측 위험 분포. 이 도표간에 중복이 심할 수록 판별력은 나빠진다.

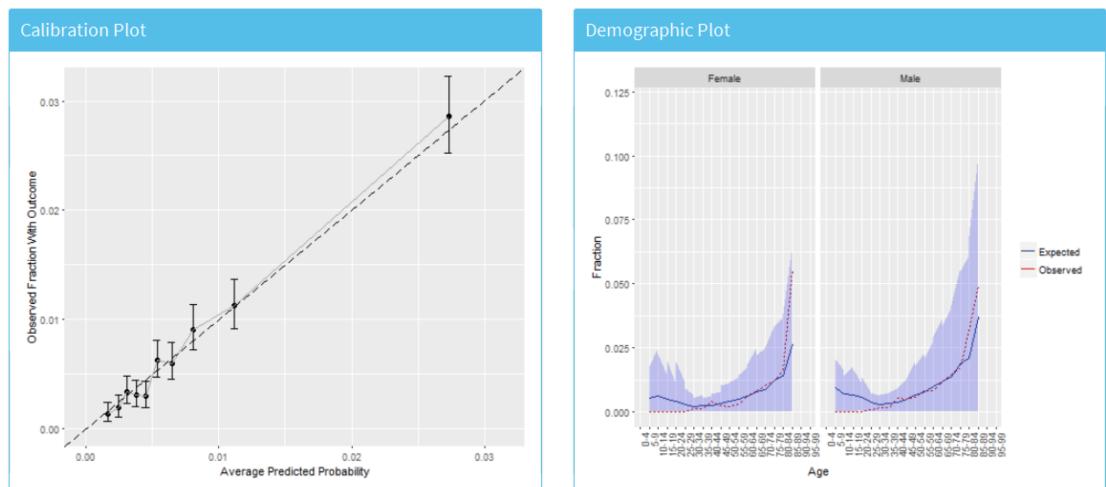


Figure 13.26: 위험 충화 보정 및 인구통계학적 보정

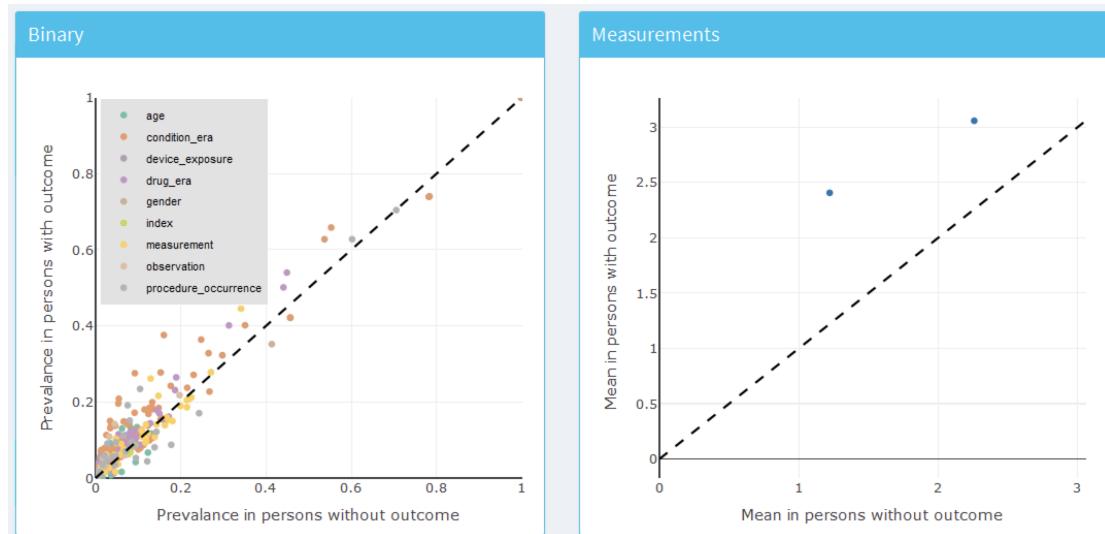


Figure 13.27: 모델 요약 도표. 각 점은 모델에 포함된 변수에 해당한다.

모델 보기

최종 모델을 확인하려면 왼쪽 메뉴에서 Model 옵션을 선택하면 된다. 옵션을 선택하면 그림 13.27과 13.28과 같이 모델의 각 변수에 대한 그래프와 공변량에 대한 요약표를 볼 수 있다. 변수 그래프는 범주형 변수와 연속형 변수로 구분된다. x축은 결과가 없는 환자의 유병률/평균이고 y 축은 결과가 있는 환자의 유병률/평균이다. 그래프를 보면 결과가 있는 환자는 대각선 아래보다 대각선 위에 더 많이 분포하고 있다.

그림 13.28의 표에는 공변량과 공변량으로 사용될 수 있는 모든 변수의 값 (일반 선형 모델을 사용할 경우 계수, 그렇지 않을 경우 변수 중요도), 그리고 결과 평균 (결과가 있는 사람의 평균), 비-결과 평균 (결과가 없는 사람의 평균)이 나타나 있다.



예측 모델은 인과관계를 판단하는 모델이 아니며 예측 변수를 결과의 원인으로 오인해서는 안 된다. 그림 13.28의 변수를 수정한다 해도 결과 발생 위험에 영향을 미친다는 보장은 할 수 없다.

Model Table

[Download Model](#)

Show 10 entries Search:

Covariate Name	Value	Outcome Mean	Non-outcome Mean
1 age group: 00-04	0	0.0004	0.0001
2 age group: 05-09	0	0	0.0003
3 index month: 1	0	0.1307	0.1096
4 observation during day -365 through 0 days relative to index: Domain	0	0.1188	0.0514
5 Charlson index - Romano adaptation	0	2.4783	1.3817
6 Diabetes Comorbidity Severity Index (DCSI)	0.1478	2.4056	1.2207
7 CHADS2VASc	0.9279	3.0573	2.2576
8 visit_occurrence concept count during day -365 through 0 concept_count relative to index	0	19.5263	13.8837
9 age group: 10-14	0	0	0.001
10 index month: 2	0	0.0934	0.0909

Showing 1 to 10 of 67,897 entries

Previous 1 2 3 4 5 ... 6790 Next

Figure 13.28: 모델 세부사항 표.

13.9 추가적 환자-수준 예측 변수

13.9.1 논문 제출용 문서 작성

논문지에 논문을 실을 수 있도록 워드 문서를 자동으로 생성하는 기능이 추가되었다. 그 문서에는 도출된 연구의 많은 세부사항과 결과가 포함되어 있다. 외적 타당도를 수행한 경우 그 결과도 추가 할 수 있다. 선택적으로, 대상 집단의 공변량이 포함된 표를 추가할 수 있다. 다음 기능을 사용하여 논문의 초안을 작성할 수 있다:

```
createPlpJournalDocument(plpResult = <your plp results>,
                         plpValidation = <your validation results>,
                         plpData = <your plp data>,
                         targetName = "<target population>",
                         outcomeName = "<outcome>",
                         table1 = F,
                         connectionDetails = NULL,
                         includeTrain = FALSE,
                         includeTest = TRUE,
                         includePredictionPicture = TRUE,
                         includeAttritionPlot = TRUE,
                         outputLocation = "<your location>")
```

더욱 자세한 내용에 대해서는 기능의 도움 페이지를 참조하라.

13.10 요약



- 환자-수준 예측은 과거의 데이터를 사용하여 미래의 사건을 예측하는 모델을 개발하는 것을 목표로 한다.
- 모델 개발을 위한 최고의 기계학습 알고리즘 선택은 경험적인 문제이다. 즉 당면한 문제 와 데이터에 의해 결정된다.
- 환자 수준 예측 PatientLevelPrediction 패키지는 OMOP-CDM의 데이터를 사용하여 예측 모델을 개발하고 검증하기 위한 사례를 제공한다.
- 모델 및 그 성능 지표 보급은 대화식 대시보드로 수행된다.

- OHDSI의 예측 프레임워크는 임상 허가의 전제 조건인 예측 모델의 대규모 외적 타당도 검증을 가능하게 한다.

13.11 예제

전제조건

이 내용을 연습하기 위하여 8.4.5절에 설명한 대로 R, R-Studio, Java가 설치돼야 한다. 또한 SqlRender, DatabaseConnector, Eunomia, 그리고 PatientLevelPrediction 패키지도 필요하며 다음을 설치하면 된다: SqlRender, DatabaseConnector, Eunomia and PatientLevelPrediction

```
install.packages(c("SqlRender", "DatabaseConnector", "devtools"))
devtools::install_github("ohdsi/Eunomia", ref = "v1.0.0")
devtools::install_github("ohdsi/PatientLevelPrediction")
```

Eunomia 패키지는 자신의 PC에서 R을 실행할 수 있도록 모의 CDM 데이터를 제공한다. 세부 내용은 다음을 통해 확인 할 수 있다:

```
connectionDetails <- Eunomia::getEunomiaConnectionDetails()
```

CDM 데이터베이스 스키마는 “main”이다. 이 예제는 여러 코호트를 사용한다. Eunomia 패키지의 `createCohorts` 함수는 코호트 테이블에서 다음과 같이 작성된다:

```
Eunomia::createCohorts(connectionDetails)
```

문제 정의

NSAID를 처음 사용하기 시작한 환자에서 누가 내년에 위장관 출혈을 일으킬지 예측.

NSAID 신규 사용자 코호트는 COHORT_DEFINITION_ID = 4를 갖고, 위장관 출혈 코호트는 COHORT_DEFINITION_ID = 3을 갖는다.

Exercise 13.1. PatientLevelPrediction R 패키지를 사용하여 예측에 사용할 공변량을 정의하고, CDM에서 PLP 데이터를 추출하고, PLP 데이터를 요약하라.

Exercise 13.2. 최종 대상 모집단을 정의하기 위하여 연구 선택사항을 다시 살펴보고 `createStudyPopulation` 함수를 사용하여 이를 지정하라. 선택한 것이 최종 모집단의 크기에 어떤 영향을 미칠 것인가?

Exercise 13.3. LASSO를 사용하여 예측 모델을 만들고 Shiny 앱을 사용하여 성능을 평가하라. 모델 성능은 어느 정도인가?

답안은 부록 E.9에서 찾을 수 있다.

Part IV

근거의 질

Chapter 14

근거의 질

Chapter leads: Patrick Ryan & Jon Duke

14.1 신뢰성 있는 근거의 속성

본격적인 여정 시작에 앞서, 우리가 바라는 이상적인 종착지가 어디인지 상상해보는 것은 도움이 될 것이다. 데이터를 근거로 만들기 위한 우리의 여정을 지원하기 위해서, 우리는 근거의 신뢰도를 높일 수 있는 바람직한 속성을 강조하고자 한다.

신뢰할 수 있는 근거는 **반복 가능 repeatable** 해야 한다. 즉, 연구자가 주어진 질문에 대해 동일한 데이터를 이용하여 동일한 분석을 수행할 때 동일한 결과가 나올 것이라 기대할 수 있어야 한다. 근거의 반복 가능성에 대한 최소한의 요구 조건은 근거가 특정 데이터를 입력하고 정의된 절차를

Desired attribute	Question	Researcher	Data	Analysis	Result
Repeatable	Identical	Identical	Identical	Identical =	Identical
Reproducible	Identical	Different	Identical	Identical =	Identical
Replicable	Identical	Same or different	Similar	Identical =	Similar
Generalizable	Identical	Same or different	Different	Identical =	Similar
Robust	Identical	Same or different	Same or different	Different =	Similar
Calibrated	Similar (controls)	Identical	Identical	Identical =	Statistically consistent

Figure 14.1: 이상적인 신뢰성을 위한 근거의 속성

수행하여서 나온 결과라는 점과, 사후 의사 결정 과정에서 수동적인 개입에서 벗어나야 한다는 점이다. 조금 더 이상적으로는, 신뢰할 수 있는 근거는 **재현 가능 reproducible**해야 하는데, 다른 연구자가 주어진 데이터와 분석 방법을 가지고 동일한 업무를 수행하였을 때 첫 연구자의 수행 결과와 동일한 결과를 낼 수 있어야 한다는 것이다. 재현 가능성은 위해서 연구 절차는 일반적인 사람이 읽을 수 있고, 컴퓨터가 실행할 수 있으며 충분히 구체화되어 있어 추가적인 연구자의 결정이 연구 결과에 반영되지 않도록 해야 한다. 반복성과 재현성을 충족시킬 수 있는 가장 효과적인 방법은 사전에 정의한 데이터의 입출력을 이용하여 표준화된 분석 방법을 사용하고, 이러한 절차를 버전이 관리되는 데이터베이스에 적용하는 것이다.

또한 우리는 동일한 질문에 대해 비슷한 데이터를 가지고 동일 분석 방법을 적용하여 비슷한 결과를 얻을 수 있는, **복제 가능한 replicable** 것으로 보인다면 우리가 주장하는 근거는 더욱 신뢰할 만하다고 자신할 수 있다. 예를 들어, 한 대규모 보험사의 청구 데이터베이스에 대한 분석에서 생성된 근거는 다른 보험사의 청구 데이터베이스를 이용하여 복제가 가능할 경우 그 근거가 강화될 수 있다. 인구 수준 효과 추정의 관점에서도 이 속성은 Austin Bradford Hill 경의 인과적 관점과 잘 일치한다. “다른 사람, 다른 장소, 환경 및 시간에서도 반복적으로 관찰되었습니까? …(중략)… 반복적인 상황과 관찰만이 우연으로 설명되는 현상인지 혹은 실제하는 위험인지 답할 수 있다.” (Hill, 1965) 환자 수준 예측의 맥락에서 복제 가능성은 외부 검증 external validation의 시행에 대한 중요성뿐 아니라, 한 데이터베이스에서 훈련된 모델이 다른 데이터베이스에 적용될 때 결과를 구별할 수 있는 판별 정확도 discriminative accuracy와 보정 calibration을 관찰함으로써 모델의 성능을 평가할 수 있는 능력을 강조할 수 있다. 서로 다른 데이터베이스에 대해 동일한 분석을 수행하고, 여전히 유사한 결과를 보이는 상황에서 우리는 그 근거가 **일반화될 수 있다 generalizable**는 확신을 얻는다. OHDSI 연구 네트워크의 핵심 가치는 다른 인구, 지역, 자료 획득 과정 등으로 대표되는 다양성이다. Madigan et al. (2013b) 은 효과 추정치 effect estimates가 데이터의 선택에 따라 민감하게 변할 수 있음을 보여주었다. 각 데이터 원천이 단일 연구의 신뢰도를 하락시킬 수 있는 고유의 한계점과 비뚤림이 있다는 점을 인식한 상태에서, 서로 다른 데이터 세트를 사용하여도 유사한 결과 패턴이 관측된 것은 어마어마하게 강력한 의미가 있다. 이는 데이터 원천 각각이 가지고 있는 비뚤림의 가능성을 상당 부분 감소 시켜, 연구 결과를 설명할 수 있기 때문이다. 네트워크 연구의 인구 수준 효과 추정치가 미국, 유럽, 아시아 그리고 다양한 청구데이터, 전자의무기록 데이터상에서 일관된 결과를 보여줄 때 해당 의학적 중재는 의학적 의사 결정 과정에서 더 큰 영향을 줄 수 있는 더욱 강력한 근거로서 인식되어야 한다.

신뢰할 만한 근거는 분석 내에서 주관적 선택에 지나치게 민감하지 않은 **강건성 robust** 을 가져야 한다. 주어진 연구에 대해서 잠재적으로 합당하다고 생각되는 대안적인 통계 방법이 있다면, 결과에 따라서 다른 분석 방법을 통해 얻은 동일한 결과로 기존 연구 결과에 대해 확신을 더하거나, 혹은 상충하는 결과를 통해 기존 연구에 대한 경각심을 얻을 수 있다. (Madigan et al., 2013a) 인구 수준

효과 추정에서 민감도 분석에는 연구 설계 선정 (코호트 비교 연구, 자기 통제 환자군 self-controlled case series 연구 등)과 분석적 고려사항의 선정 (코호트 비교에서 혼란 변수 조정을 위한 성향점수 매칭, 계층화 또는 가중치 유무)과 같은 고급 연구 설계의 문제를 포함할 수 있다.

마지막으로 가장 중요할 수도 있는 부분은 근거는 **보정되어야 한다 calibrated**는 점이다. 근거 생성 시스템에 대한 성능이 검증되지 않은 상태에서는 해당 시스템이 미지의 연구 질문에 대한 답변을 제공한다고 말하기 불충분하다. 폐쇄형 시스템은 잘 알려진 작동 특성을 가져야 하며, 이는 측정 가능하고 시스템이 생성하는 어떠한 결과에 대해서도 그 상황을 잘 전달할 수 있어야 한다. 통계적 표현은 경험적으로 잘 정립된 특성이 있음을 보여줄 수 있어야 한다. 예를 들어 95% 신뢰구간이란 95%의 확률 범위를 갖는다는 뜻이고, 10%의 예상 확률이란 인구 집단에서 관측된 사건 발생의 비율이 10%이라는 뜻이다. 관찰 연구에서는 항상 연구 설계, 연구 방법, 연구 데이터에 대한 가정을 검정할 방법을 수반해야 한다. 이 검정 방법은 연구 타당성에 일차적인 위협 (선택비뚤림, 교란변수, 측정 오차)에 대해 먼저 집중하여 평가하여야 한다. 음성 대조군 Negative controls은 관찰연구에서 발생할 수 있는 계통 오차를 확인하고 감소시킬 수 있는 강력한 도구인 것으로 보고되었다. (Schuemie et al., 2016, 2018a,b)

14.2 근거의 질에 대한 이해

하지만 우리의 연구 결과가 충분히 신뢰할만한 수준인지 어떻게 알 수 있을까? 누군가가 우리의 연구에서 설정해놓은 특정 환경을 신뢰할까? 규제 당국의 의사결정은 어떨까? 향후 연구의 기반이 될 수 있을까? 새로운 연구가 발표되거나 확산하는 과정에서 독자는 연구의 형태 (무작위 대조시험, 관찰 연구, 혹은 다른 유형의 분석 방법)에 관계없이 이러한 질문을 염두에 두어야 한다.

흔히 관찰 연구 observational study 즉, 실세계 데이터 real world data를 활용한 연구를 진행하면서 마주하게 되는 우려는 바로 데이터 품질에 관한 부분이다. (Botsis et al., 2010; Hersh et al., 2013; Sherman et al., 2016) 일반적으로 관찰 연구에 사용된 데이터는 원래 연구 목적으로 수집된 것이 아니므로 내재적 비뚤림 inherent biases과 같은 불완전하거나 부정확한 데이터의 수집으로 인한 문제를 겪을 수 있다. 이러한 우려로 인해 데이터 품질을 측정하고 특성화하고 이상적으로 데이터 품질을 개선하려는 방법에 대한 연구가 계속해서 증가하고 있다. (Kahn et al., 2012; Liaw et al., 2013; Weiskopf and Weng, 2013) OHDSI 커뮤니티는 이러한 연구를 강력히 지지하며, 커뮤니티 회원은 OMOP CDM 및 OHDSI 네트워크의 데이터 품질을 조사하는 많은 연구를 직접 주도하고 참여하였다. (Huser et al., 2016; Kahn et al., 2015; Callahan et al., 2017; Yoon et al., 2016)

지난 10년간의 결과를 고려해보면, 데이터 품질이라는 것은 결코 완벽해질 수 없다는 것이 명백해졌다. 이 개념은 의료정보학 분야의 개척자인 Clem McDonald 박사의 인용에도 잘 반영되어 있다.

:

사실 데이터 충실도의 감소는 의사의 뇌에서 의료기록으로 데이터가 이동하는 것에서부터 시작된다.

그러므로 우리는 공동체로서 질문해야 할 필요가 있다. –불완전한 데이터가 주어지면, 어떻게 우리는 신뢰할만한 근거를 얻을 수 있을까?

이 문제에 대한 대답은 “근거의 품질”에 대한 다음과 같은 전반적인 과정을 살펴보는 데 있다: 데이터에서부터 근거로의 과정에 대한 검토, 근거 생성 과정의 구성 요소에 대한 확인, 각 구성 요소의 질에 대한 신뢰 구축 방법의 결정, 그리고 이것을 투명하게 전달하는 방법. 근거의 질이란 단순히 관찰 데이터의 품질뿐 아니라 관찰 분석에 사용된 방법, 소프트웨어 및 임상적 정의의 타당성을 고려해야 한다.

뒤이어 나오는 단원에서 우리는 근거의 품질에 해당하는 네 가지 구성요소에 대한 부분을 살펴볼 것이며, 이를 표 14.1에 나타내었다.

Table 14.1: 근거의 품질에 해당하는 네 가지 구성요소

구성요소	측정 대상
데이터 품질	합의된 구조와 방법을 이용하여 타당한 값을 가진 데이터가 온전히 입력되었는가?
임상적 타당성	수행된 분석이 임상적 의도와 어느 정도 일치하고 있는가?
소프트웨어의 타당성	데이터의 변환과 분석 과정이 우리가 의도한대로 진행되었다고 신뢰할 수 있는가?
방법론적 타당성	주어진 데이터의 강점과 약점을 인지하고 있는 상태에서, 적절한 연구 방법론을 사용하고 있

14.3 근거 품질의 전달

근거 품질의 중요한 측면은 데이터에서 근거로의 여정에서 발생하는 불확실성을 표현하는 능력이다. OHDSI의 활동을 통해 이루고자 하는 거시적인 목표는 OHDSI에서 생성된 근거가 –비록 여러 방면으로 불완전하더라도– 강점과 약점에 대하여 일관되게 측정되고, 엄격하고 공개적인 방식으로 전달되어 생성되었다는 신뢰감을 의료 전문가에게 제공해주는 것이다.

14.4 요약



- 우리가 생성한 근거는 반복 가능성 *repeatable*, 재현 가능성 *reproducible*, 복제 가능성 *replicable*, 일반화 가능성 *generalizable*, 강건성 *robust*을 갖추어야 하며 보정된 *calibrated* 결과여야 한다.

- 근거의 품질은 그 근거의 신뢰성 여부를 판단하기 위해 단순히 데이터의 품질만이 아닌 그 이상의 것을 추구한다:
 - * 데이터 품질
 - * 임상적 타당성
 - * 소프트웨어 타당성
 - * 방법론적 타당성
- 근거를 전달하는 과정에서, 근거의 품질에 대한 다양한 위협으로부터 나타나게 되는 불확실성 또한 표현해야 한다.

Chapter 15

데이터의 질

Chapter leads: Martijn Schuemie, Vojtech Huser & Clair Blacketer

관찰 의료 연구에서 사용되는 대부분의 데이터는 연구를 목적으로 수집되지 않는다. 예를 들어, 전자 의무 기록 Electronic Health Records(EHR)은 환자의 진료를 지원하는데 필요한 정보를 수집하기 위해, 청구 데이터는 비용 지불자에게 비용을 청구하기 위한 근거를 제공하기 위해 수집된다. 많은 이가 이러한 데이터를 임상 연구에 사용하는 것이 적합한지 여부에 의문을 가지고 있으며 심지어, van der Lei (1991) 는 “데이터는 수집된 목적으로만 사용되어야 한다 (Data shall be used only for the purpose for which they were collected)”고 주장하였다. 문제는 데이터가 우리가 원하는 연구를 위해 수집되지 않았기 때문에, 충분한 품질을 보장할 수 없다는 것이다. 데이터의 품질이 낮으면 (garbage in), 그 데이터를 사용한 연구 결과의 품질도 낮을 수밖에 없다 (garbage out). 따라서 관찰 의료 연구에 있어서 데이터 품질을 평가하는 것은 중요하며, 다음의 질문에 답하는 것을 목표로 한다:

연구 목적에 적합한 데이터인가 (Are the data of sufficient quality for our research purposes) ?

우리는 데이터 품질 Data Quality(DQ)을 다음과 같이 정의할 수 있다 (Roebuck, 2012):

데이터를 특정 목적에 적합하게 만드는 완전성 Completeness, 유효성 Validity, 일관성 Consistency, 적시성 Timeliness, 정확성 Accuracy의 상태 (The state of completeness, validity, consistency, timeliness and accuracy that makes data appropriate for a specific use).

주목할만한 것은 우리의 데이터가 완벽하지는 않지만, 목적에 충분히 적합할 수 있다는 것이다.

DQ를 직접적으로 관찰할 수는 없지만, 이를 평가하기 위한 방법론이 개발되어 왔다. DQ 평가는 2 가지 유형으로 구분될 수 있다 (Weiskopf and Weng, 2013): 보편적인 DQ를 확인하기 위한 평가, 특정 연구의 맥락에서 DQ를 확인하기 위한 평가.

본 장에서 우리는 먼저, DQ 문제가 발생할 수 있는 원인에 대해 검토하고, 보편적인 DQ와 연구 목적별 DQ 평가 이론에 대해 논의 후, OHDSI 도구를 사용하여 이러한 평가를 어떻게 수행하는지 단계별로 설명하고자 한다.

15.1 데이터 품질 문제에 대한 원인

14장에서 언급한 바와 같이, 의사들이 본인의 생각을 기록할 때 데이터 품질과 관련된 많은 위험요소가 발생한다. Dasu and Johnson (2003)은 데이터의 수명 주기에 따른 단계를 명시하였고, 각 단계를 통합한 DQ 진행을 제시하였다. 그들은 이를 DQ 연속체 DQ continuum라 하였다:

- 데이터 수집 및 통합 Data gathering and integration.** 자료의 수기 입력 시 오류와 비풀림 등의 발생 가능한 문제 (예를 들어 upcoding in claims; 청구를 위하여 진단명과 시술 등을 보다 심각하게 혹은 높은 비용으로 작성하는 것), EHR에서 잘못된 테이블 간의 결합, 결측값을 기본값으로 대체하는 것 등을 포함한다.
- 데이터 저장 및 지식 공유 Data storage and knowledge sharing.** 데이터 모델에 대한 문서화 부족, 메타 데이터의 부족이 잠재적인 문제로 여겨진다.
- 데이터 분석 Data analysis.** 잘못된 데이터 변환, 부정확한 데이터 해석, 그리고 부적절한 방법론 사용 등의 문제가 포함될 수 있다.
- 데이터 공유 Data publishing.** 후속 사용을 위해 데이터를 게시하는 경우 (When publishing data for downstream use).

우리가 사용하는 데이터는 대부분 이미 수집되고 통합되어 있기 때문에, 1단계에서 개선할 수 있는 것은 거의 없다. 이 단계에서 생성된 DQ를 확인할 방법은 다음 절에서 논의될 것이다.

유사하게, 특정 형식으로 데이터를 받기 때문에 2단계에 대한 영향을 줄 수 있는 부분도 미미하다. 하지만 OHDSI에서는 관찰 데이터를 CDM으로 변환하기 때문에 이 변환 프로세스에 대한 주도권을 가지고 있다. 몇몇은 이러한 특정 단계가 DQ를 저하할 것이라 우려를 표한다. 하지만 우리는 이 변환 프로세스를 통제하기 때문에, 이후 15.2.2절에서 논의하는 것과 같이 DQ를 보존하기 위한 엄격한 안전장치를 구축할 수 있다. 여러 연구 (Defalco et al., 2013; Makadia and Ryan, 2014; Matcho et al., 2014; Voss et al., 2015a,b; Hripcsak et al., 2018)에 따르면 이 과정이 제대로 실행된다면 CDM으로 변환했을 때 오류가 거의 발생하지 않는 것으로 나타났다. 실제로, 대규모 공동체에 의해

공유되는 잘 문서화된 데이터 모델은 명백하고 명확한 방법으로 데이터 저장을 용이하게 한다.

3단계 (데이터 분석) 또한 우리의 통제 아래에 있다. OHDSI에서 우리는 이 단계의 품질 문제에 대해 DQ라는 용어 대신 각각 16장, 17장, 그리고 18장에서 다루는 임상적 타당성 *clinical validity*, 소프트웨어의 타당성 *software validity* 그리고 방법론적 타당성 *method validity*이라는 용어를 사용한다.

15.2 보편적인 데이터 품질

우리는 우리의 데이터가 관찰 연구의 보편적인 목적에 적합한지 여부에 대해 의문을 가질 수 있다. Kahn et al. (2016) 은 보편적인 DQ가 3가지 구성요소로 구성되어있다고 정의하였다:

1. **적합성 Conformance:** 데이터값이 지정된 표준과 형식을 준수하는가? 3가지 하위 유형으로 식별된다:
 - **Value:** 기록된 데이터의 요소가 지정된 형식과 일치하는가? 예를 들어 모든 의료 제공자 Provider의 진료와 specialties는 유효한 전문 분야인가?
 - **Relational:** 기록된 데이터가 지정된 관계적 제약 relational constraints 과 일치하는가? 예를 들어 DRUG_EXPOSURE 테이블의 PROVIDER_ID가 PROVIDER 테이블에도 상응하는 기록을 가지고 있는가?
 - **Computation:** 데이터에 대한 계산이 의도한 결과를 산출하는가? 예를 들어 키와 몸무게에서 계산된 BMI와 데이터에 기록된 BMI가 일치하는가?
2. **완전성 Completeness:** 특정 변수가 존재하는지 여부 (예를 들어 진료실에서 측정된 체중이 기록되어 있는가?) 와 모든 변수의 값이 기록되어 있는지 (예를 들어 모든 사람이 성별에 관련된 데이터를 가지고 있는가?) 를 나타낸다.
3. **타당성 Plausibility:** 데이터의 값을 믿을 수 있는가? 3가지 하위 유형으로 정의된다:
 - **Uniqueness:** 예를 들어 각각의 PERSON_ID는 PERSON 테이블에서 한 번만 발생하는가?
 - **Atemporal:** 값, 분포 또는 밀도가 예상되는 값과 일치하는가? 예를 들어 데이터에 의해 계산된 당뇨병 유병률이 실제 알려진 유병률과 일치하는가?
 - **Temporal:** 값의 변화가 예상 범위 내에서 일어나는가? 예를 들어 예방접종 순서는 권고사항과 일치하는가?

각각의 구성요소는 두 가지 방법으로 평가될 수 있다:

- **검증 Verification** 외부 참조에 의존하지 않고 모델과 메타데이터의 데이터 제약, 시스템 추

정, 그리고 기관 내 지식을 집중적으로 확인한다. 검증의 주요 특징은 기관 환경 내의 자원을 사용하여 예상되는 값과 분포를 설명하는 능력이다.

- **검토 Validation** 관련된 외부 기준 benchmarks과 관련된 데이터 값과의 일치에 주력한다. 외부 기준으로 사용 가능한 원천으로는 다기관의 데이터를 결합한 결과가 될 수 있다.

15.2.1 데이터 품질 검사

Kahn은 데이터가 주어진 요구 조건을 준수하는지 확인하기 위해 데이터 품질 확인 data quality check (때로는 데이터 품질 규칙 data quality rule이라고도 함)이라는 용어를 도입하였다 (예를 들어 부정확한 출생 연도 또는 사망 사건의 누락으로 인해 141세라는 환자의 신뢰할 수 없는 연령 자료 삭제). 우리는 자동화된 DQ 도구를 만들어 소프트웨어 내에서 위와 같은 검사를 진행할 수 있다. 이러한 도구 중 하나가 ACHILLES Automated Characterization of Health Information at Large-scale Longitudinal Evidence Systems(ACHILLES)이다. (Huser et al., 2018) ACHILLES는 CDM에 부합하는 데이터베이스의 특성과 시각화를 제공하는 소프트웨어 도구이다. 따라서 데이터 베이스 네트워크에서 DQ를 평가하는데 사용할 수 있다 (Huser et al., 2016). ACHILLES는 독립형 도구로써 사용 가능하며, “데이터 원천 Data Sources” 기능으로 ATLAS 안에도 통합되어 있다.

ACHILLES는 분석마다 분석 ID와 간단한 설명을 지닌 170개 이상의 데이터 특성 분석을 사전 계산한다. 이와 관련된 두 가지 예시는 다음과 같다. “715: DRUG_CONCEPT_ID에 의한 DAYS_SUPPLY의 분포” 그리고 “506: 성별에 따른 사망 연령 분포”. 이러한 분석 결과는 데이터 베이스에 저장되며, 웹 뷰어 web viewer 또는 ATLAS에서 확인 할 수 있다.

공동체에서 DQ 평가를 위해 만든 또 다른 도구로 Data Quality Dashboard(DQD)가 있다. ACHILLES가 특성화 분석을 실행하여 CDM 인스턴스 instance에 대한 전반적인 시각적 이해를 제공한다면, DQD는 테이블별, 필드 별로 주어진 규격에 적합하지 않은 CDM의 기록 수를 제공한다. 전체적으로, 1,500건 이상의 확인이 수행되고, 각 확인은 Kahn의 프레임워크로 구성된다. 각 DQ의 결과는 임계값과 비교되며, FAIL은 임계값을 위반하는 행을 백분율로 계산한 결과로 결정된다. 표 15.1은 체크포인트 예시를 보여준다.

Table 15.1: 데이터 품질 Dashboard에서 데이터 품질 규칙의 예시.

위반 행의				
분율	확인 내용 설명	임계값	상태	
0.34	VISIT_OCCURRENCE의 provider_id가 CDM specification에 규정된 데이터 형식인가를 예 / 아니오로 나타낸 값.	0.05	FAIL	

위반 행의

분율	확인 내용 설명	임계값	상태
0.99	MEASUREMENT 테이블의 measurement_source_value 필드에서 0으로 매핑된 고유 원천 데이터의 수와 백분율.	0.30	FAIL
0.09	DRUG_ERA 테이블의 drug_concept_id 필드가 성분명 등급에 적합하지 않은 값을 가진 레코드 수와 백분율.	0.10	PASS
0.02	DRUG_EXPOSURE 테이블에서 verbatim_end_date 필드에 drug_exposure_start_date 이전에 발생한 값이 있는 레코드 수와 백분율.	0.05	PASS
0.00	PROCEDURE_OCCURRENCE 테이블의 procedure_occurrence_id 필드에 중복되는 값이 있는 레코드 수와 백분율.	0.00	PASS

DQ 확인 도구는 여러 가지 방법으로 구성되며 테이블, 필드, concept 수준의 확인이 예시가 될 수 있다. 테이블 점검은 CDM 내에서 상위 수준에서 수행되는 점검으로 예를 들면 모든 페일 테이블이 존재하는지를 확인하는 것이다. 필드 수준의 확인은 모든 테이블의 모든 필드가 CDM 규격에 적합한지 평가하는 방법으로 수행된다. 이는 모든 기본 키가 실제로 고유한지 확인하는 것과 모든 표준 concept 필드가 수많은 concept_ID 중 적절한 도메인의 concept_ID를 사용하는지 확인하는 것이 포함된다. Concept 수준의 검사는 개별적인 concept_id를 확인하기 위해 조금 더 깊이 들어간다. 이 중 상당수가 Kahn의 프레임워크 중 타당성 Plausibility 항목에 해당하며 성별과 관련된 특정 개념이 부적절한 성별에 할당되지 않도록 보장하는 것이 예시로 해당된다 (예를 들어 여성 환자에서 전립선암).



ACHILLES와 DQD는 CDM 데이터를 대상으로 실행된다. 이렇게 식별된 DQ 문제는 CDM으로의 변환 과정이 원인일 수 있지만, 원천 데이터 상에서 이미 존재하는 DQ 문제를 반영할 수도 있다. 만일 변환 과정의 문제로 확인되는 경우 일반적으로 문제 해결을 연구자의 역량 내에서 진행할 수 있지만, 원천 데이터의 오류로 인한 문제의 유일한 조치는 오류 데이터 자체를 삭제하는 것이다.

15.2.2 ETL 단위 검정

상위 레벨의 데이터 품질 확인뿐만 아니라, 개별 수준의 데이터 품질 확인도 수행되어야 한다. 데이터가 CDM으로 변환되는 추출 변환 적재 Extract-Transform-Load(ETL) 과정은 종종 상당히 복잡하고, 이러한 복잡성으로 인해 실수를 눈치채지 못할 위험이 된다. 더욱이, 시간이 지남에 따라 원천 데이터 모델이 변경되거나, CDM 버전이 업데이트될 수 있으므로, ETL 과정의 수정이 필수적으로 진행되어야 한다. ETL과 같이 복잡한 과정의 변경은 의도하지 않은 결과를 초래할 수 있어, ETL의 모든 측면을 재고하고 검토해야 한다.

ETL의 향후 계획을 명확히 하고 지속적인 작업 진행을 위해 하나의 단위 검정 Unit test을 구성하는 것을 적극적으로 권장한다. 단위 검정이란 하나의 측면을 자동으로 확인하는 작은 코드 조각이다. 6장에서 설명한 Rabbit-in-a-Hat 도구로 이러한 단위검정을 더욱 쉽게 작성할 수 있는 단위 검정 프레임워크를 만들 수 있다. 이 프레임 워크는 원천 데이터베이스와 대상으로 하는 CDM 버전의 ETL을 위해 특별히 작성된 R 함수의 집합이다. 이러한 함수 중 일부는 원천 데이터 스키마를 준수하는 가짜 데이터 항목을 만들기 위한 것이며, 다른 일부는 CDM 형식으로 데이터에 대한 예상값을 정하는 데 사용될 수 있다. 단위 검정에 대한 예시는 다음과 같다:

```
source("Framework.R")
declareTest(101, "Person gender mappings")
add_enrollment(member_id = "M000000102", gender_of_member = "male")
add_enrollment(member_id = "M000000103", gender_of_member = "female")
expect_person(PERSON_ID = 102, GENDER_CONCEPT_ID = 8507)
expect_person(PERSON_ID = 103, GENDER_CONCEPT_ID = 8532)
```

예제에서, Rabbit-in-a-Hat에 의해 생성된 프레임워크는 나머지 코드에서 사용되는 함수를 불러오는 출처가 된다. 이후에 성별 매핑 Person gender mappings에 대한 테스트를 시작할 것이라 선언하였다. 원천 스키마는 ENROLLMENT 테이블을 가지고 있고, 우리는 Rabbit-in-a-Hat에서 생성된 add_enrollment 함수를 사용하여 MEMBER_ID와 GENDER_OF_MEMBER 필드에 대해 서로 다른 값을 지닌 두 개의 항목을 만들었다. 마지막으로, ETL 이후 PERSON 테이블에서 다양한 예상값을 지닌 두 개의 항목이 존재해야 한다는 것을 명시한다.

ENROLLMENT 테이블에는 다른 필드가 많이 존재하지만, 이 테스트의 맥락에서는 다른 필드가 어떤 값을 가지는지를 설명하지 않을 것이다. 하지만 이러한 값을 비워두면 (예를 들어 생년월일), 레코드를 삭제하거나 오류를 발생시키는 ETL의 원인이 될 수 있다. 테스트 코드를 읽기 쉽게 유지하면서 이러한 문제를 해결하기 위해서, add_enrollment 함수는 사용자가 명확하게 지정하지 않은 필드의 값에 기본값 (White Rabbit 스캔 보고서에서 관찰된 가장 일반적인 값)을 할당한다.

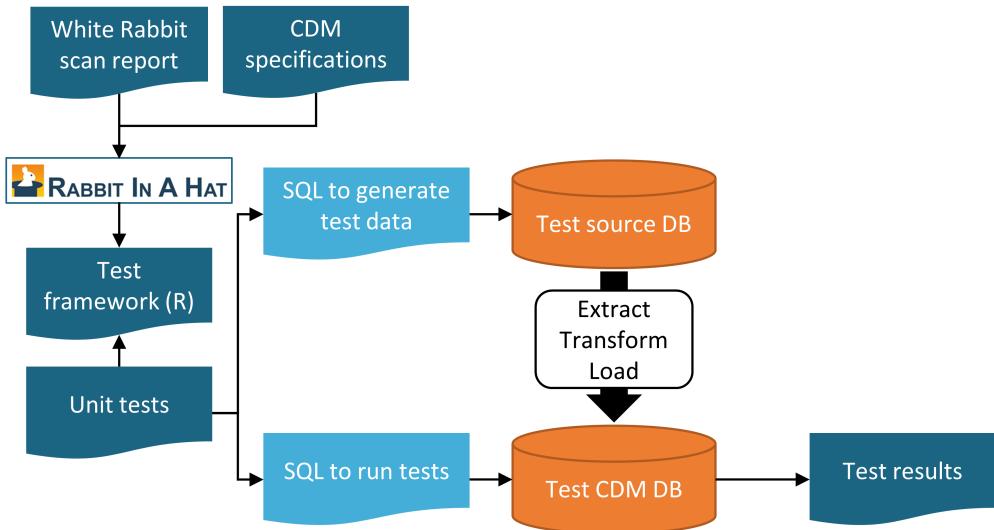


Figure 15.1: Rabbit-in-a-Hat 테스팅 프레임워크를 사용한 ETL 단위 검정 과정.

ETL의 모든 다른 논리에 대해 유사한 단위 검정이 만들어질 수 있으며, 일반적으로 수백 개의 시험을 진행할 수 있다. 테스트를 정의하는 것이 끝나면 프레임워크를 사용하여 두 개의 SQL 구문 모음을 만들 수 있다. 하나는 가짜 원천 데이터를 만드는 것이고, 다른 하나는 ETL된 데이터에 대한 테스트를 진행할 수 있는 구문이다.

```

insertSql <- generateInsertSql(databaseSchema = "source_schema")
testSql <- generateTestSql(databaseSchema = "cdm_test_schema")
  
```

전반적인 과정은 그림 15.1에 묘사된 것과 같다.

SQL을 통한 테스트는 표 15.2와 같은 테이블을 반환한다. 이 표에서는 우리가 앞서 정의한 두 가지 테스트를 통과하는 것을 알 수 있다.

Table 15.2: ETL 단위 검정 결과 예시.

ID	설명	상태
101	성별 매핑(Person gender mappings)	PASS
101	성별 매핑(Person gender mappings)	PASS

이 단위 검정의 강점은 ETL 프로세스가 변경될 때마다 쉽게 재실행할 수 있다는 것이다.

15.3 연구 별 검사

지금까지 보편적인 DQ 검사에 초점을 맞췄다. 이러한 검사는 데이터가 연구에 사용되기 이전에 실행되어야 한다. 이러한 검사는 연구 문제와 무관하게 수행되어야 하므로 이후에 연구 목적의 DQ 평가를 수행할 것을 권한다.

이러한 평가 중 일부는 특별히 연구와 관련된 DQ 규칙 rule의 형태를 취할 수 있다. 예를 들어, 관심 노출에 대한 레코드의 최소 90%가 노출 기간을 명시한다는 새로운 규칙 도입을 원할 수도 있다.

표준으로 시행하는 검사는 연구와 가장 관련된 concept, 예를 들어 코호트 정의에서 정의된 concept 을 ACHILLES에서 검토하는 것이다. 전체기간에서 특정 코드의 사용 빈도가 급격히 변한다면 이것은 DQ 문제가 있다는 것을 알려주는 힌트가 될 수도 있다. 몇몇 예시는 이 장의 뒷부분에서 recommend를 설명하고 있다.

또 다른 평가는 연구를 위해 설정된 코호트 정의를 사용해 생성된 코호트 결과에 대한 유병률과 시간에 따른 유병률의 변화를 검토하고 이것이 외부 임상 지식에 기반한 예상값과 일치하는지 확인하는 것이다. 예를 들어, 신약의 노출은 시장에 소개되기 전에는 없어야 하고, 도입 이후에 시간이 지남에 따라 증가할 가능성이 있다. 유사하게 결과에 대한 유병률은 모집단에서 질환의 유병률에 대해 알려진 것과 일치해야 한다. 만약 연구가 데이터베이스의 네트워크에서 실행한다면, 우리는 데이터베이스 간의 코호트 유병률을 비교할 수 있다. 한 데이터베이스에서 높은 유병률을 보이지만, 다른 데이터베이스에서는 누락된 경우, DQ 문제가 있을 수 있다. 이러한 평가는 16장에서 논의한 바와 같이, 임상적 타당성 *clinical validity*의 개념과 중복된다는 것을 유의해야 한다. 몇몇 데이터베이스에서는 예상하지 못한 유병률 결과가 나올 수가 있는데, 이는 DQ 문제가 아니라 코호트 정의에서 연구 주제와 부합하는 건강 상태를 온전히 잡아내지 못했거나 데이터베이스마다 환자 모집단이 상이하여 발생할 수 있다.

15.3.1 매핑 검사하기

우리가 통제할 수 있는 오류의 원인 중 한 가지는 원천 코드를 표준 concept에 매핑하는 것이다. 용어 매핑은 정교하게 제작되었으며, 매핑 상의 문제가 있다면 공동체 구성원에 의해 발견되어¹에 보고된 후 다음 업데이트에 반영된다. 그런데도 모든 매핑을 직접 확인하는 것은 불가능하고 오류가 계속 존재할 수 있다. 그렇기 때문에, 연구를 수행할 때 연구와 관련 있는 concept의 매핑을 검토해보는 것을 권장한다. 다행히도, CDM에서 표준 용어(Concept)뿐만 아니라 원천 코드도 같이 저장하기 때문에 이러한 작업은 쉽게 할 수 있다. 연구에 사용된 concept에 매핑된 원천 코드뿐만 아니라 그렇지 않은 원천 코드도 검토할 수 있다.

¹<https://github.com/OHDSI/Vocabulary-v5.0/issues>

% per month	Max monthly %	Person count	Description
	26.81	92,019,885	Depressive Disorder
	6.64	15,969,198	Depressive disorder 440383
	6.64	15,686,275	311 (ICD9CM) Depressive disorder, not elsewhere classified
	0.46	188,230	F328 (ICD10CM) Other depressive episodes
	0.38	94,693	F3289 (ICD10CM) Other specified depressive episodes
	3.10	12,010,783	Adjustment disorder with mixed emotional features 433454
	3.07	9,839,712	30928 (ICD9CM) Adjustment disorder with mixed anxiety and depressed mood
	3.03	2,049,618	F4323 (ICD10CM) Adjustment disorder with mixed anxiety and depressed mood
	0.04	121,453	3091 (ICD9CM) Prolonged depressive reaction
	3.17	9,237,192	Dysthymia 433440

Figure 15.2: checkCohortSourceCodes 기능의 output 예시.

원천 코드를 검토하는 방법 중 하나는 MethodEvaluation R 패키지의 `checkCohortSourceCodes` 함수를 사용하는 것이다. 이 함수는 ATLAS에서 생성된 코호트 정의를 입력 `input`으로 사용하고 코호트 정의에서 사용된 각 concept 모음에 대해 concept과 매핑되는 원천 코드를 확인한다. 또한 전체 기간에 대한 코드의 빈도를 계산하여 특정 코드에서 발생하는 시간적인 문제를 확인하는 데 도움이 될 수 있다. 그림 15.2 예시 결과는 “우울증 Depression disorder”이라 불리는 concept 모음의 분석을 보여준다. 관심 분야의 데이터베이스에서 이 concept 모음의 가장 보편적인 concept은 440383 (우울증; Depressive disorder)이다. 데이터베이스 내의 ICD-9 코드의 3.11, ICD-10 코드의 F32.8과 F32.89 이 세 가지 코드가 해당 concept으로 매핑이 된 걸 볼 수 있다. 그림의 왼쪽부터 보면 전체로서의 concept은 시간이 지남에 따라 초반에는 증가하지만, 그 후에 급격히 감소하는 것을 볼 수 있다. 개별 코드를 살펴보면, 이러한 하락은 하락 시점에 ICD-9 코드의 사용이 중단되는 것으로 설명될 수 있다는 것을 알 수 있다. 이것이 ICD-10 코드가 사용되기 시작한 것과 같은 시간임에도 불구하고, 결합된 ICD-10 코드의 빈도가 ICD-9 코드의 빈도보다 훨씬 적다. 이 구체적인 예시는 ICD-10 코드 F32.9 (“주요 우울 장애, 단일 에피소드, 불특정”)도 이 concept으로 매핑돼야 했었기 때문이다. 이 문제는 Vocabulary에서 해결되었다.

앞의 예시는 매핑되지 않은 원천 코드를 발견하는 것을 묘사한 것으로, 일반적으로 누락된 매핑을 식별하는 것이 존재하는 매핑을 확인하는 것 보다 더 어렵다. 이는 어떤 원천 코드가 매핑되어야 하지만 매핑되지 않았는지 알아야 한다. 이를 평가하는 반-자동화된 방법은 MethodEvaluation R 패키지의 `findOrphanSourceCodes` 함수를 사용하는 것이다. 이 함수는 간단한 텍스트 검색을 통해 원천 코드에 대한 Vocabulary를 검색할 수 있게 하고, 이 원천 코드가 특정 concept이나 그 concept의 하위 concept 중 하나와 매핑되는지 여부를 확인한다. 원천 코드의 결과는 현재 CDM 데이터

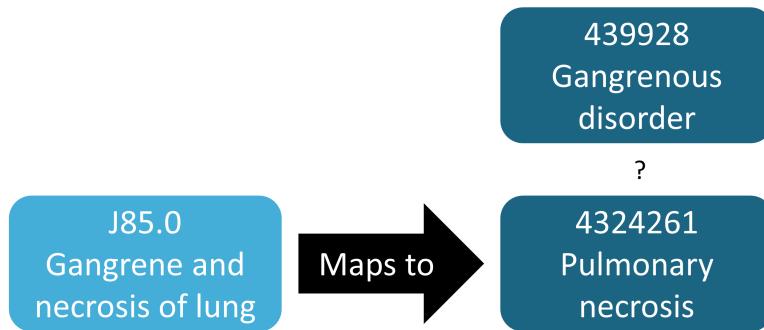


Figure 15.3: orphan 원천 코드 예시.

베이스에 나타나는 코드로만 제한된다. 예를 들어, “괴저 장애 Gangrenous disorder” (439928) 와 모든 하위 concept은 모든 괴저 발생을 찾기 위해 사용되었다. 이것이 실제로 괴저를 나타내는 모든 원천 코드를 포함하는지 여부를 평가하기 위해, 원천 코드를 식별하기 위한 CONCEPT 테이블과 SOURCE_TO_CONCEPT_MAP 테이블의 설명을 검색하는데 몇 가지 용어 (예를 들어 “괴저 gangrene”)가 사용되었다. 자동 검색은 데이터에 나타나는 각 괴저 코드가 “괴저 장애”라는 concept에 직접 또는 간접적으로 매핑되었는지 여부를 평가하기 위해 사용된다. 이러한 평가의 결과는 그림 15.3과 같으며, ICD-10 코드 J85.0 (“폐의 괴저 및 괴사”; Gangrene and necrosis of lung)은 “괴저 장애의 하위 concept이 아닌 concept 4324261 (“폐 괴사”; Pulmonary necrosis)에만 매핑된 것을 알게 되었다.

15.4 ACHILLES 실습

여기서는 CDM 형식의 데이터베이스에 대해 ACHILLES를 실행하는 방법을 보여준다.

먼저, R에서 서버를 연결하는 방법에 관해 설명할 필요가 있다. ACHILLES는 `createConnectionDetails`라는 함수를 제공하는 DatabaseConnector 패키지를 사용한다. 다양한 데이터베이스 관리 시스템 Database management systems(DBMS)에 필요한 특정 설정을 ?`createConnectionDetails`을 입력하여 확인할 수 있다. 예를 들어, 다음 코드를 이용하여 PostgreSQL과 연결할 수 있다:

```

library(Achilles)
connDetails <- createConnectionDetails(dbms = "postgresql",
                                         server = "localhost/ohdsi",
                                         user = "joe",
                                         password = "supersecret")

cdmDbSchema <- "my_cdm_data"
  
```

```
cdmVersion <- "5.3.0"
```

마지막 두 줄은 CDM 버전뿐만 아니라 `cdmDbSchema` 변수를 정의한다. 이를 사용하여 CDM 형식의 데이터가 어디에 있는지, 어떤 버전의 CDM이 사용되었는지 R에 입력한다. Microsoft SQL Server의 경우, `cdmDbSchema <- "my_cdm_data.dbo"`와 같이 데이터베이스와 스키마를 모두 지정해야 한다.

다음으로, ACHILLES를 실행한다:

```
result <- achilles(connectionDetails,
                      cdmDatabaseSchema = cdmDbSchema,
                      resultsDatabaseSchema = cdmDbSchema,
                      sourceName = "My database",
                      cdmVersion = cdmVersion)
```

이 함수는 `resultsDatabaseSchema`에 여러 테이블을 생성하며, 여기에서는 CDM 데이터와 동일한 데이터베이스 스키마로 설정하였다.

ATLAS를 ACHILLES 결과 데이터베이스로 지정하거나 ACHILLES 결과를 JSON 파일로 내보내서 ACHILLES 데이터베이스의 특징을 볼 수 있다:

```
exportToJson(connectionDetails,
              cdmDatabaseSchema = cdmDatabaseSchema,
              resultsDatabaseSchema = cdmDatabaseSchema,
              outputPath = "achillesOut")
```

JSON 파일은 `achillesOut` 하위 폴더에 작성되고, 결과 확인을 위해 AchillesWeb 웹 애플리케이션과 함께 사용할 수 있다. 예를 들어, 그림 15.4 ACHILLES 데이터 밀도 도표를 보여준다. 이 도표는 2005년에 시작된 대량의 데이터를 보여준다. 하지만 1961년경에 몇 개의 레코드가 있는 것으로 나타나며, 이는 데이터에 오류가 있는 것일 수도 있다.

또 다른 예시로는 그림 15.5로, 당뇨병 진단 코드의 유병률에 급격한 변화를 보여주고 있다. 이러한 변화는 특정 국가에서 보험 청구 규정이 변경됨에 따라 진단 수가 증가한 것이지 실제로 유병률이 증가한 것은 아니다.

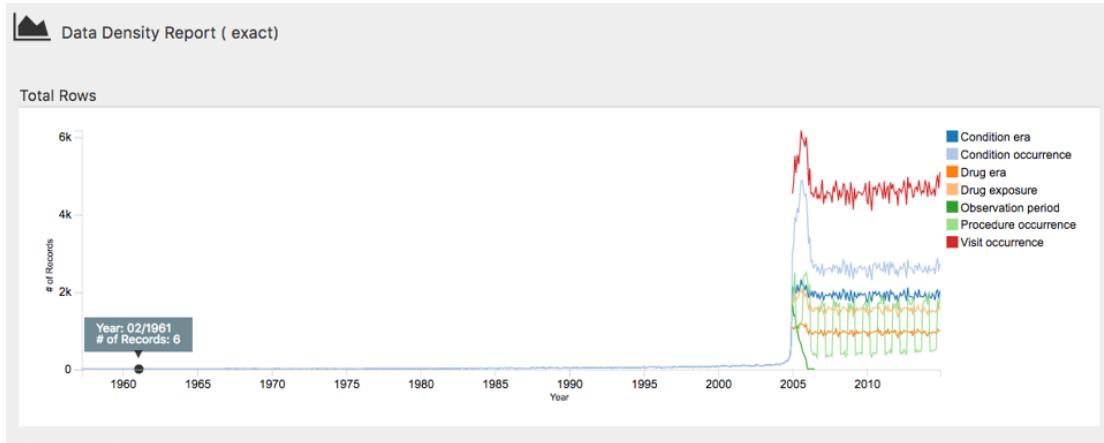


Figure 15.4: ACHILLES 웹 뷰어의 데이터 밀도 그림.

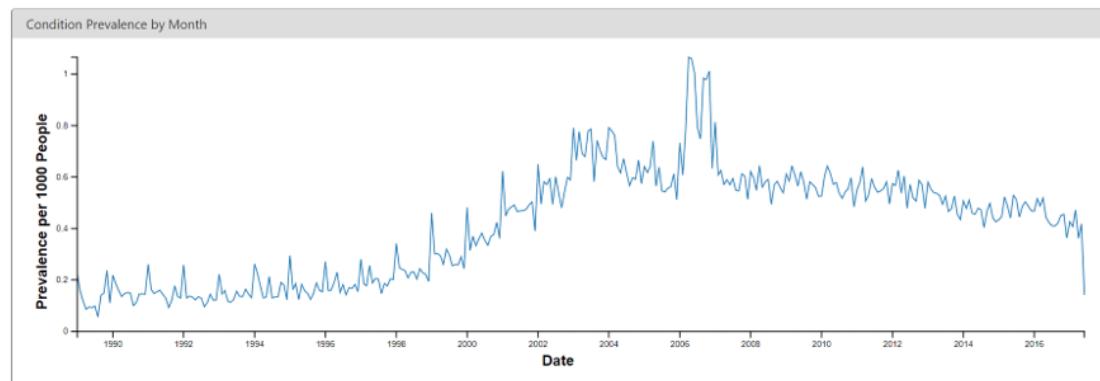


Figure 15.5: ACHILLES 웹 뷰어에서 코딩된 월별 당뇨병 발생률.

15.5 Data Quality Dashboard 실습 (Data Quality Dashboard in Practice)

여기에서는 CDM 형식의 데이터베이스에서 Data Quality Dashboard를 실행하는 방법을 보여준다. 15.4절에서 설명한 CDM connection에 대해 더 많은 checks를 수행한다. 현재 DQD는 CDM v5.3.1만 지원하기 때문에 실행 전 데이터베이스가 올바른 버전인지 확인이 필요하다. ACHILLES 와 마찬가지로 cdmDbSchema를 작성하여 데이터를 찾을 위치를 R에 입력해야 한다.

```
cdmDbSchema <- "my_cdm_data.dbo"
```

다음으로, Dashboard를 실행한다...

```
DataQualityDashboard::executeDqChecks(connectionDetails = connectionDetails,
                                         cdmDatabaseSchema = cdmDbSchema,
                                         resultsDatabaseSchema = cdmDbSchema,
                                         cdmSourceName = "My database",
                                         outputFolder = "My output")
```

위의 함수는 지정된 스키마에서 사용 가능한 모든 데이터 품질 체크포인트를 실행한다. 그런 다음 CDM과 동일한 스키마로 설정한 resultsDatabaseSchema에 테이블을 작성한다. 이 테이블은 CDM 테이블, CDM 필드, 검사명, 설명, Kahn의 카테고리와 하위 카테고리, 위반 행의 수, 임계값 레벨 그리고 검사의 통과 여부 등 각 체크포인트의 실행에 대한 모든 정보가 포함된다. 이 함수는 테이블뿐만 아니라 outputFolder로 JSON 파일을 작성할 위치를 지정한다. JSON 파일을 사용해서 웹 뷰어를 시작해 결과를 확인할 수 있다.

```
viewDqDashboard(jsonPath)
```

jsonPath 변수는 위의 executeDqChecks 함수가 호출될 때, 지정된 outputFolder에 위치한 Dashboard의 결과가 포함된 JSON 파일의 경로여야 한다.

처음 Dashboard를 열면 그림 15.6과 같이 개요 테이블이 표시된다. 여기에는 내용별 Kahn의 카테고리에서 실행된 총 검사의 수, 각 검사의 PASS 수와 백분율 및 전체 통과 비율이 표시된다.

왼쪽 메뉴에서 *Results*를 클릭하면 실행된 각 검사에 대한 상세 결과 페이지로 이동한다 (그림 15.7 참조). 예시의 테이블은 개별적인 CDM 테이블의 완전성을 확인하거나 CDM에서 특정 테이블에 최소 1개 이상의 레코드를 가진 인원수 및 백분율을 확인하기 위한 검사에 대한 것이다. 이 경우

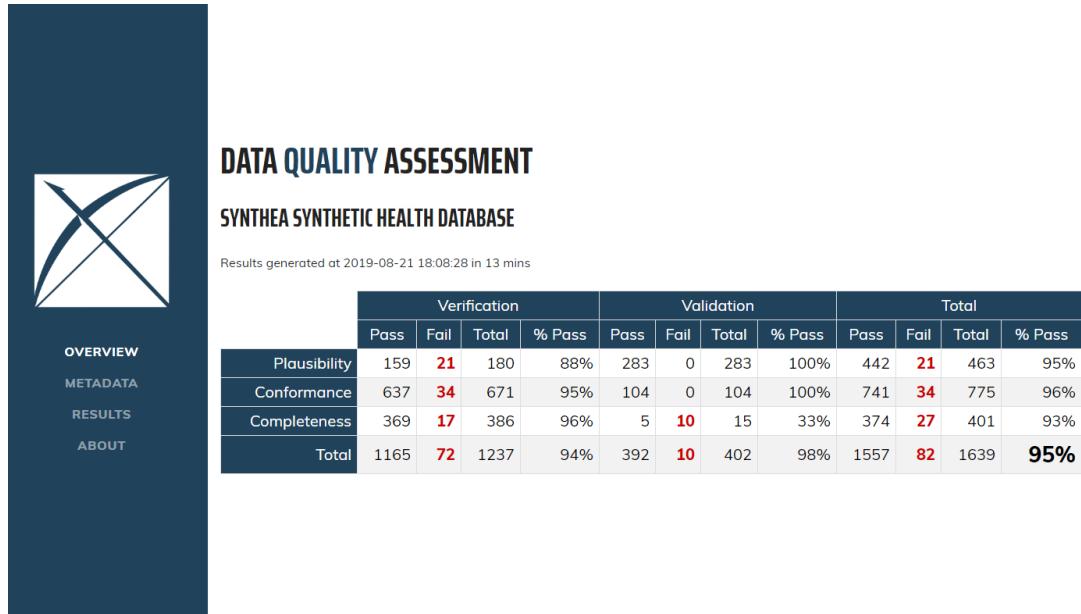


Figure 15.6: 데이터 품질 Dashboard에서의 데이터 품질 점검 개요.

나열된 5개의 테이블이 Dashboard에 Fail로 나타났으며 모두 비어있다. 아이콘을 클릭하면 나열된 결과를 생성하기 위해 데이터에서 실행된 정확한 쿼리를 보여주는 창이 열린다. 이를 통해 Dashboard에서 Fail로 간주한 행을 쉽게 식별할 수 있다.

15.6 연구별 검사 실습

다음으로, 부록 B.4에 제공된 혈관 부종 코호트 정의에 대한 몇 가지 검사를 수행할 것이다. 15.4 절에 설명된 것처럼 연결 세부사항이 설정되어 있고, 코호트 정의 JSON과 코호트 정의에 대한 SQL 이 각각 “cohort.json”과 “cohort.sql” 파일에 저장되어 있다고 가정한다. JSON 파일과 SQL 파일은 ATLAS 코호트 정의 기능의 내보내기 탭에서 얻을 수 있다.

```
library(MethodEvaluation)

json <- readChar("cohort.json", file.info("cohort.json")$size)
sql <- readChar("cohort.sql", file.info("cohort.sql")$size)

checkCohortSourceCodes(connectionDetails,
  cdmDatabaseSchema = cdmDbSchema,
  cohortJson = json,
  cohortSql = sql,
  outputFile = "output.html")
```

RESULTS

SYNTHEA SYNTHETIC HEALTH DATABASE

Results generated at 2019-08-22 14:15:06 in 29 mins

							Column visibility	CSV
Search:								
Status	Context	Category	Subcategory	Level	Description		% Records	
				FIELD				
[+]	FAIL	Verification	Plausibility	Atemporal	FIELD	The number and percent of records with a value in the gap_days field of the DRUG_ERAS table less than 0. (Threshold=0%).	24.07%	
[+]	FAIL	Verification	Completeness	None	FIELD	The number and percent of records with a value of 0 in the standard concept field race_concept_id in the PERSON table. (Threshold=0%).	16.74%	
[+]	FAIL	Verification	Conformance	Relational	FIELD	The number and percent of records that have a value in the ethnicity_concept_id field in the PERSON table that does not exist in the CONCEPT table. (Threshold=0%).	16.15%	
[+]	PASS	Verification	Completeness	None	FIELD	The number and percent of records with a NULL value in the condition_end_date of the CONDITION_OCCURRENCE. (Threshold=100%).	13.24%	
[+]	PASS	Verification	Completeness	None	FIELD	The number and percent of records with a NULL value in the condition_end_datetime of the CONDITION_OCCURRENCE. (Threshold=100%).	13.24%	

Figure 15.7: 데이터 품질 Dashboard에서의 데이터 품질 평가 구체적 관찰.

그림 15.8과 같이 웹 브라우저에서 출력 output 파일을 열 수 있다. 여기서 혈관 부종 코호트 정의에 “Inpatient or ER visit”과 “Angioedema” 두 가지 concept이 있는 것을 확인할 수 있다. 이 예제 데이터베이스에서, 방문은 ETL 중에 표준 concept과 매핑되었지만, Vocabulary에는 없는, “ER”과 “IP”라는 데이터베이스 특정 원천 코드를 통해 발견되었다. 혈관 부종은 하나의 ICD-9 코드와 두 개의 ICD-10 코드를 통해 발견되었다. 개별 코드에 대한 피크 라인을 봤을 때, 두 가지 코딩 시스템 간의 교대 시점을 명확하게 알 수 있지만, 전체적인 concept에서는 불연속성이 없다.

다음으로, 표준 concept 코드에 매핑되지 않은 원천 코드인 orphan 원천 코드를 검색할 수 있다. 표준 concept인 “혈관 부종 Angioedema”을 찾은 다음 “혈관 부종” 또는 그 이름이 일부 포함되어 있거나 동의어가 있는 concept과 코드를 찾는다:

% per month	Max monthly %	Person count	Description
	60.60	24,189,656	Inpatient or ER visit
	39.50	15,003,249	Emergency Room Visit 9203
	39.50	15,003,249	ER (None) No matching concept
	23.90	9,186,407	Inpatient Visit 9201
	23.90	9,186,407	IP (None) No matching concept
	0.27	76,711	Angioedema
	0.27	76,711	Angioedema 432791
	0.26	64,726	9951 (ICD9CM) Angioneurotic edema, not elsewhere classified
	0.20	8,822	T783XXA (ICD10CM) Angioneurotic edema, initial encounter
	0.09	3,163	T783XXD (ICD10CM) Angioneurotic edema, subsequent encounter

Figure 15.8: Angioedema 코호트 정의에서 사용된 원천 코드.

```
"Periodic edema"))
View(orphans)
```

code	설명	vocabularyId	overallCount
T78.3XXS	Angioneurotic edema, sequela	ICD10CM	508
10002425	Angioedemas	MedDRA	0
148774	Angioneurotic Edema of Larynx	CIEL	0
402383003	Idiopathic urticaria and/or angioedema	SNOMED	0
232437009	Angioneurotic edema of larynx	SNOMED	0
10002472	Angioneurotic edema, not elsewhere classified	MedDRA	0

데이터에서 실제로 사용된 유일한 잠재적 orphan 코드는 “혈관 신경성 부종 Angioneurotic edema, 후유증 sequela”이며, 이는 혈관 부종과 매핑되어서는 안 된다. 따라서 이 분석에서는 누락된 코드가 발견되지 않았다.

15.7 요약



- 대부분의 관찰형 의료 데이터는 연구를 위해 수집되지 않는다.
- 데이터 품질은 데이터가 연구 목적에 적합한지를 확인하기 위해 평가되어야 한다.
- 보편적인 연구 목적을 위해, 특정 연구의 맥락에서 비판적으로 데이터 품질을 평가해야 한다.
- 데이터 품질의 일부 측면은 데이터 품질 Dashboard의 예시와 같이 사전 정의된 많은 규칙을 통해 자동적으로 평가될 수 있다.
- 특정 연구와 관련된 코드 매핑을 평가하기 위한 다른 도구가 있다.

15.8 예제

전체조건

예제 실습을 위해 8.4.5절에서 설명한 것과 같이 R, R-studio 및 Java가 설치되어 있다고 가정한다. 또한 SqlRender, DatabaseConnector, ACHILLES 및 Eunomia 패키지가 필요하다. 아래의 코드를 사용하여 설치할 수 있다:

```
install.packages(c("SqlRender", "DatabaseConnector", "devtools"))
devtools::install_github("ohdsi/Achilles")
devtools::install_github("ohdsi/DataQualityDashboard")
devtools::install_github("ohdsi/Eunomia", ref = "v1.0.0")
```

Eunomia 패키지는 로컬 R 세션 내에서 실행되는 CDM에서 시뮬레이션 된 데이터 모음을 제공한다. 자세한 접속정보는 아래를 활용하여 얻을 수 있다:

```
connectionDetails <- Eunomia::getEunomiaConnectionDetails()
```

CDM 데이터베이스 스키마는 “main”이다.

Exercise 15.1. Eunomia 데이터베이스에 대해 ACHILLES를 실행하라.

Exercise 15.2. Eunomia 데이터베이스에 대해 DataQaulityDashbiard를 실행하라.

Exercise 15.3. DQD 검사 목록을 추출하라.

제안된 답변은 부록 E.10에서 찾을 수 있다.

Chapter 16

임상적 타당성

Chapter leads: Joel Swerdel, Seng Chan You, Ray Chen & Patrick Ryan

질량을 에너지로 바꾸는 확률은 마치 몇 마리의 새만 존재하는 나라에서 눈을 감고 총을 쏘아 나는 새를 맞출 확률과 비슷하다. 알베르트 아인슈타인, 1935

OHDSI의 이상은 ‘관찰형 연구를 통하여 세계에 건강과 질병에 대한 포괄적인 이해를 제공’ 하는 것이다. 후향적 연구는 이미 존재하는 데이터를 이용할 수 있다는 편리함이 있지만, 이전 14장에서 기술한 바와 같이 다양한 타당성의 한계를 지니고 있다. 데이터의 질과 통계적 방법론과 별개로 임상적 타당성 clinical validity을 논하기는 쉽지 않지만 이번 장에서는 보건의료 데이터의 특성, 코호트 정의 검증, 근거의 일반화, 이 세 가지에 집중해보려 한다. 인구 수준 추정(PLE) 12장의 예제로 돌아가 보자. 앞장에서 ’ACE inhibitor가 thiazide 또는 thiazide-like diuretics에 비해 혈관부종 위험성이 높은가?’에 대해 대답하려 했었고, ACE inhibitor가 thiazide 또는 thiazide-like diuretics에 비해 혈관부종을 더 많이 야기함을 증명했었다. 이번 장은 다음의 질문에 답하기 위해 쓰였다: “수행한 분석이 어느 정도로 임상적 의도와 일치하는가?”

16.1 보건의료 데이터의 특성

우리가 확인했던 것이 사실 ACE inhibitor 사용과 혈관부종의 관계가 아니라, ACE inhibitor의 **처방**과 혈관부종 간의 관계일 수도 있다. 이전 15장에서 데이터의 질에 대해서 이미 다루었다. CDM으로 변환된 데이터의 질은 원본 데이터의 질을 결코 넘어설 수 없다. 여기서 우리는 대부분의 의료 서비스 사용 데이터의 특성에 대해 다룬다. OHDSI의 많은 데이터베이스는 청구 데이터 또는 전자 의무기록에서 유래된다. 청구데이터와 전자의무기록 모두 연구를 위해서 만들어진 데이터베이스가

아니며, 둘은 서로 다른 데이터 수집 과정을 거친다. 청구 데이터는 보험금 청구 및 환급을 위하여 만들어진 데이터베이스로, 데이터 요소는 제공된 의료 서비스의 청구를 정당화하기 위한 목적하에 만들어진다. EHR의 데이터 요소는 임상 의료 행위 및 운영을 뒷받침하기 위하여 수집되며, 주어진 의료 시스템 기반에서 현재의 의료 서비스 및 추후 필요하리라 예상되는 정보 위주로 수집된다. 이들 모두 환자의 완전한 병력 medical history을 반영하거나 서로 다른 의료 기관 간의 데이터를 통합하지 못한다.

관찰 데이터로부터 믿을만한 근거를 생성하기 위해서 연구자는 환자가 의료 서비스를 찾는 그 순간부터 의료 서비스에 대한 데이터가 만들어져 분석에 사용되기까지의 전 과정에 대해 이해할 필요가 있다. 예를 들어 다양한 원천 관찰형 데이터에서 “약물 노출”은 의사의 처방 기록, 약국 조제 기록, 병원에서의 직접 주입, 또는 환자가 자가보고한 약물 복용 기록까지 여러 가지를 의미할 수 있다. 데이터가 어디에서 유래했는지는 환자가 약을 실제로 복용했는지 안 했는지 뿐 아니라 약물 복용 기간에 대한 신빙성에 영향을 미칠 수 있다. 데이터 수집 과정은 의사 처방 없이 복용할 수 있는 일반의약품 over-the-counter(OTC) 등의 약물 노출을 과소평가할 수 있고, 환자가 처방만 받고 실제로 약국에서 조제 받지 않거나, 복용하지 않은 경우 약물 노출을 과대평가할 수도 있다. 치료 노출과 결과 간의 가능한 비뚤림을 이해하고, 보다 이상적으로는 이러한 오류를 정량화하고 보정하는 것은 가용한 데이터로부터 만들어 낸 근거의 신뢰성을 향상할 수 있다.

16.2 코호트 유효성 검사

Hripcsak and Albers (2017) 은 “표현형 phenotype은 유기체의 유전적 구성에서 파생된 유전형 genotype과 구별되는, 관찰할 수 있고 잠재적으로 변화하는 유기체의 상태를 나타내는 것”이라고 설명했다. 표현형이라는 용어는 전자 의무 기록(EHR) 데이터로부터 추정되는 환자 특성에 적용될 수 있다. 연구자는 정보학이 시작된 이후 구조화된 데이터와 서술적 데이터 모두에서 EHR phenotyping을 수행해 왔다. 목표는 원 EHR 데이터, 청구 데이터 또는 기타 임상 관련 데이터를 기반으로 대상 개념에 대한 결론을 도출하는 것이다. 표현형 알고리즘 (예를 들어, 표현형을 식별하거나 특성화하는 알고리즘)은 공학적인 최근의 연구 또는 다양한 형태의 머신 러닝을 통해 분야의 전문가나 분야 지식을 가지고 있는 엔지니어에 의해 생성될 수 있다.

이 설명은 임상적 타당성 고려 시 보강에 유용한 몇 가지 속성을 강조한다. 1) 그것은 관찰할 수 있는 어떤 것에 대해 말하고 있다는 것을 분명히 한다. (따라서 관찰형 데이터에서 수집이 가능함을 의미한다) 2) 그것은 표현형 정의에 시간의 개념을 포함한다. (사람의 상태가 변할 수 있기 때문에) 3) 원하는 의도인 표현형과 의도에 대한 구현인 표현형 알고리즘을 구별한다.

OHDSI는 “코호트”라는 용어를 채택하여 일정 기간 하나 이상의 포함 기준을 충족하는 사람 집합을 정의했다. “코호트 정의”는 관찰 데이터베이스에 대해 코호트를 구현하는데 필요한 논리를 나타낸

다. 이와 관련하여 코호트 정의 (또는 표현형 알고리즘) 는 관찰 가능한 관심 임상 상태에 속하는 환자의 표현형을 반영하기 위한 코호트 생성에 사용된다.

임상적 특성 분석, 인구 수준 추정, 환자 수준 예측을 포함한 대부분의 관찰형 분석의 연구 프로세스의 일부로서 하나 이상의 코호트를 설정해야 한다. 이러한 분석에 의해 도출된 근거의 타당성을 평가하기 위해, 반드시 각각의 코호트에 대해 다음의 질문을 던져야 한다: ‘코호트 정의 따라 가용한 관찰 데이터에서 식별된 코호트의 피험자가 의도했던 표현형에 실제로 얼마나 부합하는가?’

다시 12장의 예제로 돌아가 보자. ‘ACE 억제제가 thiazide 계열 이뇨제와 비교해서 혈관부종 위험성을 높이는가?’에 대한 질문에 대답하기 위하여 우리는 3가지 코호트를 생성했다. ACE 억제제를 처음 사용한 대상 코호트, Thiazide 계열 이뇨제를 처음 사용한 대조 코호트, 혈관부종이 발생한 결과 코호트. ACE 억제제와 thiazide 이뇨제를 사용한 사람을 모두 식별하였다고 얼마나 자신 있게 말할 수 있는가? 처음 사용한 환자 new user라는 조건에 이전 (관찰되지 않은) 사용이 모두 배제되었다고 믿을 수 있는가? 약물에 노출되었다는 기록이 있는 환자가 실제로 약물에 노출되었고, 기록이 없는 환자는 실제로 약물에 노출되지 않았다고 믿을 수 있는가? ACE 억제제를 복용한다고 했을 때, 코호트 시작 시점과 종료 시점이 실제 약물을 시작하고, 종료한 시점과 일치할까? “혈관부종” 발생의 기록이 있는 환자가 다른 알레르기 피부 질환과 다른, 실제 피하조직의 부종을 경험했을 것인가? 실제 혈관 부종을 겪은 환자 중 얼마나 많은 숫자의 환자가 실제로 의사에게 진찰을 받고 해당 질환을 진단받고 기록되었을까? 우리가 약물에 의한 것으로 의심한 혈관부종이 음식 알레르기나 바이러스 감염 등에 의한 다른 원인과 얼마나 구별 지어질 수 있는가? 질병의 발생 시기가 약물 노출과 부작용 발생 간의 시계열적 연관성을 도출하기에 확신을 가질 수 있을 만큼 잘 포착되었는가? 이러한 유형의 질문에 답하는 것이 임상적 타당성의 핵심이다.

이 장에서는 코호트 정의를 검증하는 방법에 대해 논의한다. 먼저 코호트 정의의 타당성을 측정하는데 사용되는 측정기준을 설명한다. 다음으로, 이러한 측정기준을 추정하는 두 가지 방법을 설명한다. 1) 원천 기록 검증을 통한 임상 판정과 2) PheEvaluator, 진단적 예측 모델링을 이용하는 반자동화 방법이다.

16.2.1 코호트 판단 측정기준

연구에 대한 코호트 정의가 결정되면 정의의 타당성을 평가할 수 있다. 타당성을 평가하는 일반적인 접근방식은 정의된 코호트의 일부 또는 모든 사람을 ‘최적 표준 gold standard’에 비교하고 그 결과를 코호트 정의 내의 최적 표준 분류 및 자격 검정에 따라 계층화하여 2x2의 혼동 행렬 confusion matrix로 표현하는 것이다. 그림 16.1은 혼동 행렬의 요소를 보여준다.

코호트 정의의 참과 거짓 결과는 그 정의를 사람들의 집합에 적용함으로써 결정된다. 정의에 포함된 사람은 특정 건강 상태에 대해 양성으로 간주하며 “양성 true”으로 표시된다. 코호트 정의에 포함되

		Gold Standard	
		True	False
Cohort Definition	True	True Positive	False Negative
	False	False Negative	True Negative

Figure 16.1: 혼동 행렬.

지 않은 사람은 건강 상태에 대해 음성으로 간주하며 “음성 false”으로 표시된다. 코호트 정의에서 고려된 개인의 건강 상태에 대한 절대적 진리는 알기 어렵지만, 기준이 되는 최적 표준을 확립하는 방법은 여러 가지가 있는데, 그중 두 가지는 이번 장 후반부에 기술할 것이다. 사용한 방법과 관계없이, 이러한 사람에 대한 라벨링은 코호트 정의에 설명된 것과 동일하다.

표현형 지정의 이항 표시 오류 외에도, 건강 상태의 시점도 부정확할 수 있다. 예를 들어, 코호트 정의는 어떤 사람이 특정 표현형에 속한다고 올바르게 정의할 수 있지만, 언제부터 해당 건강 조건을 갖게 되었는지 시점을 정하는 데에는 부정확할 수 있다. 이 오류는 생존 분석 결과 (예를 들어 위험비 hazard ratio를 효과 측정으로 effect measure 이용하는 분석에서 비뚤림을 유발할 수 있다.

이 과정의 다음 단계는 코호트 정의와 최적 표준과의 일치성을 평가하는 것이다. 최적 표준과 코호트 정의에 의해 “양성”이라고 표기된 사람을 “진양성 true positive”이라고 부른다. 최적 표준에 의해 “음성”으로, 코호트 정의에 의해 “양성”으로 분류된 사람을 “위양성 false positive”이라고 부른다. 예를 들어, 코호트 정의가 실제 특정 상태가 없는 환자를 잘못하여 특정 상태가 있다고 판단할 수 있다. 최적 표준과 코호트 정의에 의해 모두 “음성”으로 정의된 환자는 “진음성 true negative”이라고 부른다. 최적 표준에 의해 “양성”으로 분류되었으나 코호트 정의에 의해 “음성”으로 분류된 환자는 “위음성 false negative”이라고 부른다. 예를 들어 실제로는 환자가 특정 건강 상태를 지니고 있으나, 코호트 정의에서는 그렇지 않게 분류될 수 있다. 혼동 행렬의 네 칸의 숫자를 세어, 우리는 코호트 정의가 사람을 실제 표현형으로 분류하는 데 얼마나 정확한지 여부를 정량화할 수 있다. 다음은 코호트 정의 성능을 측정하기 위한 표준 성능 평가 기준이다 (역자 주: 민감도, 특이도는 최적 표준을 분모로 하는 성능평가도구이고, 양성예측도와 음성예측도는 측정도구의 평가를 분모로 하는 성능평가도구이다).

1. 코호트 정의 민감도 sensitivity – 실제 표현형에 속하는 피험자 중 얼마나 많은 비율의 피험자가 코호트 정의에 의해 양성으로 분류되는가? 다음의 공식으로 구한다:

$$\text{민감도} = \frac{\text{진양성}}{\text{진양성} + \text{위음성}}$$

2. 코호트 정의 특이도 specificity – 실제 표현형에 속하지 않는 피험자 중 얼마나 많은 비율의 피험자가 코호트 정의에 의해 음성으로 분류되는가? 다음의 공식으로 구한다:

특이도 = 진음성 / (진음성 + 위양성)

3. 코호트 정의 양성 예측도 Positive predictive value(PPV) – 코호트 정의에 의해 양성으로 분류되는 환자 중 표현형을 실제로 가지고 있는 환자가 얼마나 되는가? 다음의 공식으로 구한다:

양성 예측도 = 진양성 / (진양성 + 위양성)

4. 코호트 정의 음성 예측도 Negative predictive value(NPV) – 코호트 정의에 의해 음성으로 분류되는 환자 중 표현형을 실제로 가지고 있지 않은 환자가 얼마나 되는가? 다음의 공식으로 구한다:

음성예측도 = 진음성 / (진음성 + 위음성)

위의 측정기준에서 만점은 100%이다. 관측 데이터의 특성상 만점은 보통 평균과 거리가 멀다. Rubbo et al. (2015)은 심근경색에 대한 코호트 정의를 검증하는 연구를 검토했다. 그들이 조사한 33개의 연구 중, 오직 하나의 데이터 집합에서 하나의 코호트 정의만이 양성 예측도에 대해 만점을 얻었다. 전체적으로 33개 연구 중 31개에서 70% 이상의 양성 예측도가 보고되었다. 그러나 그들은 또한 33개 연구 중 11개만이 민감도를 보고했고 5개만이 특이도를 보고했다는 것을 발견했다. 양성예측도는 민감성, 특이성, 유병률의 함수다. 유병률에 대한 값이 다른 데이터에서는 민감도와 특이도가 일정하게 유지되더라도 양성예측도에 대해서는 다른 값을 생성한다 (역자 주: 같은 검사법이라 할지라도 질병의 유병률이 바뀌면 PPV, NPV가 달라진다). 민감도와 특이도가 없다면 불완전한 코호트 정의로 인한 비뚤림을 수정할 수 없다. 게다가, 건강 상태의 오분류가 수행될 수 있는데, 즉 대상 및 대조 코호트 정의 수행 시 오분류의 정도가 비슷할 수도 있지만, 그 정도가 두 그룹 간에 매우 다를 수도 있다는 점이다. 이전의 코호트 정의 검증 연구는 실제 추정치에 강한 비뚤림을 초래할 수 있음에도 불구하고 대상 및 대조 코호트 간의 오분류의 잠재적 가능성에 대해 시험을 하지 않았다.

코호트 정의에 대한 성능 평가 기준이 마련되면, 이러한 정의를 사용하여 연구 결과를 조정하는 데 사용될 수 있다. 이론적으로, 이러한 측정 오차 추정치를 이용해 연구 결과를 보정하는 방법은 잘 확립되어 있다. 그러나 실제로는 성능 평가 점수를 얻기 어렵기 때문에, 이러한 보정은 거의 고려되지 않고 있다. 최적 표준을 결정하는 방법은 이 절의 나머지 부분에 설명되어 있다.

16.3 원천 기록 검증

코호트 정의를 검증하는 데 사용되는 일반적인 방법은 원천 기록 확인을 통한 임상적 판단이다. 즉 관심 임상 조건이나 특성을 분류할 수 있는 충분한 지식을 갖춘 하나 분야 이상의 전문가가 개인의

기록을 철저히 검사하는 것이다. 차트 검토는 보통 다음의 단계를 따른다:

1. 기관생명윤리위원회 institutional review board(IRB) 또는 환자 개인에게 직접 차트 검토 및 연구에 대한 승인을 받는다.
2. 평가할 코호트 정의를 사용하여 코호트를 생성한다. 수동으로 전체 코호트를 판단할 만큼 시간적, 인적 자원이 충분하지 않은 경우 검토 대상 중 일부를 표본으로 추출한다.
3. 환자 차트를 검토할 수 있는 충분한 임상 전문 지식을 가진 사람 한 명 이상을 섭외한다.
4. 환자가 원하는 임상 조건이나 특성이 있는지에 대해 양성 및 음성을 판단하기 위한 지침을 결정한다.
5. 임상 전문가는 각 환자가 표현형에 속하는지 여부를 분류하기 위해 표본 내의 사람에 대한 모든 가용 데이터를 검토하고 판단한다.
6. 코호트 정의 분류 및 임상 판정 분류에 따라 혼동 행렬로 도표화하고 수집된 데이터에서 가능한 성능을 평가한다.

차트 검토의 결과는 일반적으로 하나의 성능 지표인 양성예측도 평가로 제한된다. 평가 대상이 되는 코호트 정의는 원하는 조건이나 특성을 가진 것으로 생각하는 사람만 생성하기 때문이다. 따라서 코호트의 표본에 있는 각 개인은 임상적 판단에 근거하여 진양성 또는 위양성 중 하나로 분류된다. 전체 모집단의 표현형 (코호트 정의에 의해 식별되지 않은 사람을 포함)에 있는 모든 사람에 대한 지식이 없으면, 위음성의 식별이 불가능하며, 따라서 혼동행렬의 나머지 부분을 채워 나머지 성능 특성을 채울 수 없다. 모집단 전체의 표현형을 식별하는 가능한 방법에는 모집단이 작지 않은 경우 불가능한 전체 데이터에 대한 차트 검토 또는 이미 충분한 임상 정보와 특정 표현형 유무가 결정된 임상 레지스트리 (예로 암 레지스트리)의 활용 등이 포함되나 이는 일반적으로는 실행이 불가능하다 (아래 예시 참조). 또는 코호트 정의에 적합하지 않은 사람을 표본으로 추출하여 예측된 음성 집단의 표본을 생성한 다음, 위 차트 검토의 3~6단계를 반복하여 이러한 환자가 진정으로 관심의 임상 조건이나 특성이 없었는지 여부를 확인할 수 있다. 이렇게 하면 음성예측도를 추정할 수 있으며, 표현형의 유병률에 대한 적절한 추정치를 구할 수 있다면, 민감도와 특이도를 추정할 수 있다.

원천 기록 확인, 차트 검토를 통한 임상적 판단에는 여러 가지 한계가 있다. 앞서 언급했듯이, 차트 검토는 양성예측도와 같은 단일 지표의 평가에도 매우 많은 시간이 소요되고 자원이 많이 소요되는 과정이 될 수 있다. 이러한 한계는 혼동 행렬을 완전히 채우기 위해서 전체 모집단을 평가해야 하므로 실용성을 크게 저해한다. 또한 상기 프로세스의 다중 단계는 연구의 결과를 편향시킬 수 있는 잠재력을 가지고 있다. 예를 들어, EHR에서 기록에 동일하게 접근할 수 없거나, EHR이 없거나, 개별적인 환자 동의가 필요한 경우, 추출된 평가 대상의 표본은 정말로 무작위적이지 않을 수 있으며, 선택비뚤림을 야기할 수 있다. 또한 수동적인 프로세스는 인간의 실수나 잘못된 분류에 취약하므로 완벽하게 정확한 측정 기준을 나타내지 못할 수 있다. 환자의 기록이 불충분하거나, 결정이 주관적이거나, 충분한 전문성이 없는 등의 이유로 종종 임상 전문가들 사이에 의견 불일치가 있을 수

있다. 많은 연구에서, 전문가들 사이의 불일치에 대해 충분히 고려하지 않는 다수결의 원칙을 통해 일방적으로 해결된다.

16.3.1 원천 기록 검증에 대한 예시

차트 검토를 활용한 코호트 정의 유효성을 검사하는 과정의 예시는 Columbia University Irving Medical Center(CUIMC)에 수행하는 연구로부터 제공되는데, 그 연구는 National Cancer Institute(NCI)에서 행하는 타당성 연구의 일환인 다수의 암에 대한 코호트 정의를 검수하는 것이다. 이 과정은 이러한 암 중 하나인 전립선암을 검증하는 데 사용되었는데, 과정은 다음과 같다.

1. 제안서를 제출하고, OHDSI 암 표현형 연구를 위한 IRB의 동의를 얻었다.
2. 전립선암에 대한 코호트 정의를 개발하였다: 어휘를 탐구하기 위해 ATHENA와 ATLAS를 사용해서, 전립선 2차 신경세포 (concept ID 4314337) 또는 Non-Hodgkin's의 전립선 림프종 (concept ID 4048666) 을 제외한 전립선 악성 종양 (concept ID 4163261) 을 가지고 있는 모든 환자를 포함한 코호트 정의를 생성했다.
3. ATLAS를 사용하여 코호트를 생성하고, 수동 검토를 위한 100명의 임의 환자를 지정하여 개개인의 PERSON_ID를 환자 MRN을 매핑 테이블을 사용하여 매핑하였다. PPV의 성능 측정 기준에 맞는 원하는 통계적 정밀도를 달성하기 위하여 100명의 환자가 선정되었다.
4. 위의 임의로 선정한 각각의 환자가 참 혹은 거짓 양성 반응을 보이는지 밝히기 위하여 입원 환자와 외래 환자를 모두 포함한 다양한 EHR의 기록을 수동으로 검토하였다.
5. 한 의사가 수동 검토와 임상 판정을 실행하였다. (비록 이상적으로는 미래에 합의와 계층 간의 신뢰성을 평가하기 위해 더욱 엄격한 검증 연구가 더욱더 많은 검토자에 의해 실행되어야 할 것이다)
6. 참고 기준의 결정은 임상 문서, 병리학 보고서, 실험실, 의약품, 사용 가능한 전자 환자기록 전체 문서가 기반이 되었다.
7. 환자들은 1) 전립선암 2) 비 전립선암 3) 인식 불가로 분류되었다.
8. PPV의 보수적인 추정치는 다음과 같이 계산되었다: 전립선암 / (비 전립선암 + 인식 불가)
9. 그리고, CUIMC 전체 인구를 아우르는 참조 기준을 식별하기 위한 최적 표준과 같은 종양 레지스트리를 사용하여, 코호트 정의에 정확히 식별되지 않은 종양 레지스트리 안의 환자 수를 세었다. 이 방법은 우리가 이 값을 진양성 혹은 위음성으로 사용하여 민감도를 측정할 수 있게 했다.
10. 측정된 민감도, PPV, 유병률을 사용하여, 코호트 정의에 대한 특이도를 측정할 수 있었다. 위에서 언급한 바와 같이, 이 과정은 각각의 코호트 정의가 수동 차트 검토를 통하여 개별 적으로 평가되어야 하고, 또한 모든 성능 지표를 확인하기 위해 CUIMC 종양 레지스트리와 연관되어야 하므로 상당한 시간과 노동력이 필요하다. 최대한 신속히 종양 레지스트리의 접근

권한을 얻는 검토를 이행하지만, IRB 승인 과정 만으로 몇 주가 소요되고, 수동 차트 검토의 과정 자체로도 몇 주가 더 소요된다.

Rubbo et al. (2015) 는 심근경색(MI) 코호트 정의에 대한 검증 노력의 검토는 연구에 사용된 코호트 정의뿐만이 아니라 검증 방법과 보고된 결론에서도 상당한 이질성이 존재한다는 것을 확인하였다. 그들은 급성 심근경색에 대한 최적 표준의 코호트 정의가 존재하지 않다고 결론 지었다. 그들은 이 과정이 상당한 비용과 시간이 소모되었다고 언급하였다. 이러한 제약 때문에, 대부분의 연구는 그들의 검증에 대한 작은 표본 크기를 가졌고 이는 곧 성능 특성에 대한 추정치의 큰 변화를 가져왔다. 그들은 또한 33개의 연구 중에서, 모든 연구가 양성 예측도에 대해 보고하는 가운데 오직 11개의 연구가 민감도에 대해 보고하였고, 오직 5개의 연구만이 특이도에 대해 보고하였다. 위에서 언급한 바와 같이 민감도와 특이도의 예측 없이, 분류비뚤림에 대한 통계적 교정은 시행될 수 없다.

16.4 PheEvaluator

OHDSI 공동체는 진단 예측 모델을 사용하여 최적 표준을 구축하는 다른 접근방식을 개발하였다. (Swerdel et al., 2019) 일반적인 생각은 건강 결과의 확인을 의료진이 원천 기록 검증을 수행하는 방식과 비슷하게 모방하지만, 규모에 맞게 자동화된 방법으로 시행된다. 이 도구는 PheEvaluator라고 불리는 오픈 소스 R 패키지로 개발되었다.¹ PheEvaluator은 환자 수준 예측(PLP) 기능을 사용한다.

과정은 다음과 같다:

1. 극도로 특이도가 높은(“xSpec”) 코호트를 생성한다: 진단 예측 모델을 학습할 때 관심 결과가 생길 가능성이 아주 높은 사람으로서 noisy positive label로 사용될 집단을 결정한다.
2. 극도로 민감도가 높은(“xSens”) 코호트를 생성한다: 결과가 생길 가능성이 있는 모든 이를 포함하는 집단을 결정한다. 이 코호트는 그 집단의 상반되는 집단을 식별하는데 사용될 것이다. 즉, 이는 진단 예측 모델을 학습할 때 noisy negative labels로 사용될 결과를 갖지 않을 것이라고 확신하는 사람 집단이다.
3. xSpec과 xSens 코호트를 사용하여 예측 모델을 학습시킨다: 13장에서 설명된 바와 같이, 다양한 환자 특징을 예측자로 사용하여 모델을 적합시키고, 어떤 개인이 xSpec 코호트에 포함되는지 (결과가 있다고 생각되는 사람들) 혹은 xSens 코호트에 상반되는 집단 (결과가 없다고 생각되는 사람들)에 포함되어 있는지에 대한 여부를 예측하는 것을 목표로 한다.
4. 학습된 모델을 코호트 정의 성능을 평가하는데 사용될 홀드 아웃 집단 결과의 확률을 예측하는데 적용한다: 모델의 예측 집단은 개인이 표현형에 포함되는 예측된 확률을 추정하기 위해서 개인의 데이터로 적용될 수 있다. 우리는 이 예측을 확률적 최적 표준 probabilistic

¹<https://github.com/OHDSI/PheEvaluator>

gold standard로 사용한다.

5. 코호트 정의의 성능 특성을 평가하라: 우리는 예측된 확률을 코호트 정의의 이진 분류로 비교 한다 (혼동 행렬의 시험 조건). 시험 조건과 참 조건의 예측을 사용하여, 우리는 혼동 행렬을 완전히 채울 수 있고 성능 특성, 다시 말하면 민감도, 특이도, 예측값의 전체 집단을 예측할 수 있다. 이 접근 방식을 사용하는 데 있어 주요 제약은 데이터베이스 내 건강 결과를 갖는 것이 데이터에 의해 제약된 사람의 확률을 예측하는 것이다. 데이터베이스 혹은 의료기록과 같은 중요한 정보에 따라 유효하지 않을 수도 있다.

진단 예측 모델링에서, 병이 있는 사람과 없는 사람을 식별하는 모델을 생성한다. 환자 수준 예측 장에서 언급된 바와 같이, 예측 모델은 표적(대상) 코호트와 결과 코호트를 사용하여 개발되었다. 표적 코호트는 건강 결과를 소유한 개인과 그렇지 않은 개인 모두를 포함한다; 결과 코호트는 표적 코호트 안의 개인 중에서 건강 결과가 생긴 개인을 식별한다. PheEvaluator 과정에서 우리는 극도의 명확한 코호트 정의, 즉 “xSpec” 코호트를 예측 모델을 위한 결과 코호트를 결정하는 데 사용한다. xSpec 코호트는 관심 질병에 걸릴 확률이 매우 높을 가능성을 가진 집단을 찾기 위한 정의를 사용한다. xSpec 코호트는 관심 건강 결과에 대해 다중 발병 기록을 가진 개인의 집단으로 정의될 수 있다. 심방세동을 예로 들자면 심방세동 진단 코드를 10번 이상 가진 사람으로 선정할 수 있다. 심근 경색에 대해서는 5번의 심근경색이 생겼고 그 중 최소 2번은 입원 중에 생긴 것으로 정의할 수 있다. 예측 모델에 사용할 표적 코호트는 관심 건강 결과를 얻을 가능성이 낮은 집단의 그룹과 xSpec 코호트 안의 집단을 합쳐서 구축한다. 관심 건강 결과를 얻을 가능성이 낮은 집단을 결정하기 위해서, 전체 데이터베이스에서 표본을 추출하여 관심 결과를 나타내는 표현형에 속함을 암시하는 증거를 가진 개인을 제외하는데, 보통 xSpec 코호트를 정의하는 데 사용된 개념 concept을 포함하고 있는 기록을 가진 개인을 제외함으로써 얻을 수 있다. 이 방법에는 제약이 있다. 이 xSpec 코호트 내 사람들이 병을 가진 이와 다른 특성을 가질 가능성이 있기 때문이다. 이것은 또한 이 사람들이 첫 진단 후 보통의 환자와 비교해 관찰 시간이 더욱 길었을 수 있다. 우리는 확률적 최적 표준을 생성하는데 사용되는 예측 모델을 생성하는데 LASSO 로지스틱 회귀를 사용한다. (Suchard et al., 2013) 이 알고리즘은 간명한 모델을 생성하고 일반적으로 데이터 집합 전체에서 존재할 수 있는 많은 공변량을 제거한다. 현재 PheEvaluator 소프트웨어 버전에서, 결과 상태 (예/아니오)는 개인에 대한 모든 데이터에 근거하여 평가되고 (모든 관측 시간), 코호트 시작 날짜의 정확도를 평가하지 않는다.

16.4.1 PheEvaluator에 의한 검증 예시

우리는 PheEvaluator를 한 코호트 정의의 완전한 성능 특성을 평가하는데 사용할 수 있는데, 이는 그 코호트 정의가 심근 경색을 가졌던 이력이 있는 개인을 확진할 필요가 있는 연구에 사용될 수 있게 한다.

다음은 PheEvaluator를 사용하여 심근경색 코호트 정의를 검토하는 과정이다:

첫 번째 단계 : xSpec 코호트 정의

높은 확률 (높은 특이도)로 심근경색을 가지고 있을 환자를 정의하자 (xSpec). 심근경색을 뜻하는 개념 또는 그것의 하위 개념이 입월 5일 이내에 발생하거나, 1년 이내 4번 이상 condition occurrence 테이블에 기록된 환자가 필요하다. 그림 16.2가 ATLAS에서 해당 코호트의 정의가 어떻게 표현되는지 보여준다.

두 번째 단계 : xSens 코호트 정의

그리고 극도로 민감도가 높은 코호트 (xSens)를 정의하자. 이 코호트는 한 번이라도 심근경색이라는 컨셉에 해당하는 condition occurrence 기록을 가지고 있는 환자로 정의할 수 있을 것이다. 그림 16.3은 심근경색의 xSens 코호트를 ATLAS에서 어떻게 만드는지 보여준다.

세 번째 단계 : 예측 모델에 적합

‘createPhenoModel’ 함수는 평가용 코호트 환자가 우리가 관심 있는 임상 결과를 실제로 가지고 있을 확률을 평가하기 위한 진단 예측 모델을 개발한다. 이 기능을 사용하기 위해 1단계와 2단계에서 개발한 xSpec 및 xSens 코호트를 사용한다. xSpec 코호트는 함수에서 xSpecCohort 파라미터로 입력된다. xSens 코호트는 xSens 코호트에 있는 코호트가 모델링 프로세스에 사용된 대상 코호트에서 제외되어야 함을 나타내기 위해 ‘exclCohort’ 파라미터로 입력된다. 이 배제 방법을 사용하여 실제로 임상 결과가 있을 가능성성이 낮은 사람을 결정할 수 있다. 우리는 이 그룹을 “불확실 음성 noisy negative” 그룹, 즉 해당 임상결과가 있는 사람을 소수 포함하여 대부분은 해당 임상결과가 없을 가능성이 큰 사람이 포함된 그룹으로 생각할 수 있다. 함수에서 xSens 코호트를 ‘prevCohort’ 파라미터로 사용할 수도 있다. 이 파라미터는 해당 임상 결과에 대한 모집단에서의 대략적인 유병률을 결정하기 위해서 사용된다. 일반적으로 데이터베이스에서 무작위로 추출한 표본에서의 유병률은 모수의 유병률과 비례해야 한다. 설명한 방법을 사용하면, 더 이상 표본을 임의로 추출하여 해당 결과가 있는 사람 대비 결과가 있는 환자의 비율을 이용한 재교정 re-calibration이 필요하지 않다.

xSpec 코호트를 정의하는 데 사용된 모든 컨셉은 모델링 과정에서 제외해야 한다. 이를 위해 excludedConcepts 파라미터를 xSpec 정의에 사용된 컨셉 목록으로 설정한다. 예를 들어 심근경색의 경우 심근 경색에 대한 컨셉과 모든 하위 컨셉을 사용하여 ATLAS에 컨셉 모음을 만들었다. 예제에서는 excludedConcepts 파라미터를 심근 경색의 concept ID인 4329847로 설정하고 addDescendantsToExclude 파라미터를 TRUE로 설정해서, 모든 하위 컨셉이 제외되도록 하였다.

모델링에 포함될 환자의 임상 특성을 지정할 수 있는 몇 가지 파라미터가 있다. lowerAgeLimit 및 upAgeLimit 을 이용하여 모델링에 포함될 환자의 연령의 하한과 상한을 설정할 수 있다. 계획한

Cohort #10934

MI xSpec Cohort

Definition Concept Sets Generation Reporting Export

[460] MI xSpec Model

Cohort Entry Events

Events having any of the following criteria:

+ Add Initial Event ▾

a condition occurrence of [460] Myocardial Infarction ▾ + Add attribute... ▾ Delete Criteria

with continuous observation of at least 365 days before and 0 days after event index date

Limit initial events to: earliest event per person.

Restrict initial events to:

having all of the following criteria: + Add criteria to group... ▾

with at least 1 using all occurrences of:

a condition occurrence of [460] Myocardial Infarction ▾ + Add attribute... ▾ Delete Criteria

✖ with a Visit occurrence of: ✖ Inpatient Visit Add Import

where event starts between 0 days Before and 5 days After ▾ index start date add additional constraint

restrict to the same visit occurrence

allow events from outside observation period

and with at least 4 using all occurrences of:

a condition occurrence of [460] Myocardial Infarction ▾ + Add attribute... ▾ Delete Criteria

where event starts between 1 days After and 365 days After ▾ index start date add additional constraint

restrict to the same visit occurrence

allow events from outside observation period

Limit initial events to: earliest event per person.

Remove initial event restriction

Figure 16.2: 극도로 특이도가 높은(xSpec) 심근경색 코호트 정의

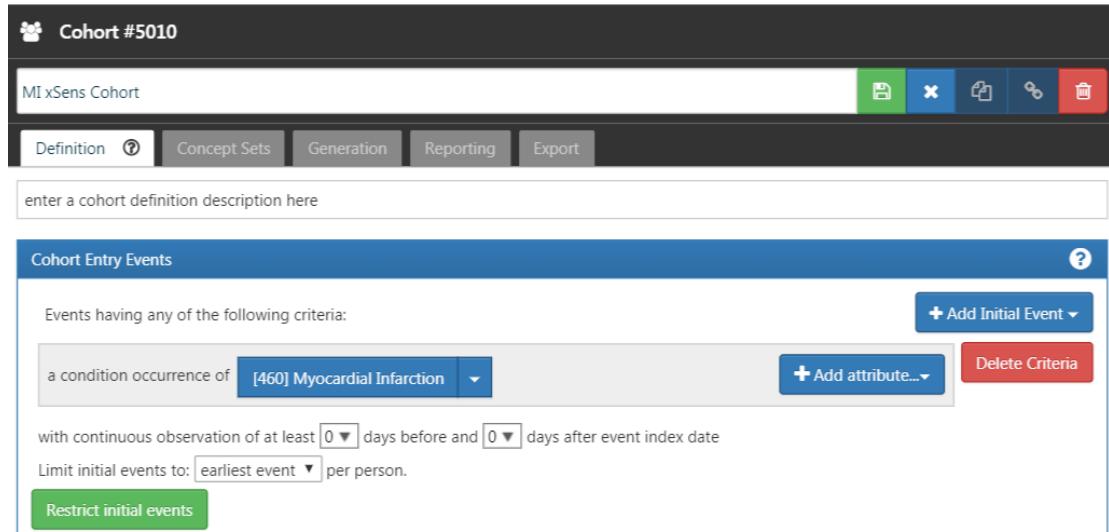


Figure 16.3: 극도로 민감도가 높은 (xSens) 심근경색 코호트 정의

연구가 특정 연령군에 대한 연구일 경우 이를 이용할 수 있다. 예를 들어, 연구에 사용될 코호트 정의가 소아의 제1형 당뇨병에 대한 것이라면, 진단 예측 모델을 개발하는 데 사용되는 연령을 5~17 세로 제한할 수 있다. 이를 통해 테스트할 코호트 정의에서 선택한 환자와 더 밀접하게 관련 있는 모델을 생성할 수 있다. 또한 gender 파라미터를 남성 또는 여성의 concept ID로 설정하여 모델에 포함할 성별을 지정할 수 있다. 기본적으로 이 파라미터는 남성과 여성 모두를 포함하도록 설정되어 있다. 이 기능은 전립선암과 같은 특정 성별에만 발생하는 질병에 유용 할 수 있다. 환자별 의료기관 최초 방문 일자를 기준으로 startDate 및 endDate 파라미터를 각각 날짜 범위의 하한 및 상한으로 설정하여 대상 환자에 대한 제한을 설정할 수 있다. 마지막으로 mainPopnCohort 파라미터를 사용하여 대상 및 결과 코호트에 있는 모든 사람을 선택할 대규모 집단 코호트를 지정할 수 있다. 대부분의 경우 이 값은 0으로 설정되어 대상 및 결과 코호트에 대한 사람을 선택하는 데 제한이 없음을 나타낸다. 그러나 때에 따라 건강 결과의 유병률이 매우 낮거나 0.01% 이하인 경우, 이 파라미터가 더 나은 모델을 구축하는 데 유용한 경우가 있을 수 있다. 예를 들면

```
setwd("c:/temp")
library(PheEvaluator)
connectionDetails <- createConnectionDetails(
  dbms = "postgresql",
  server = "localhost/ohdsi",
  user = "joe",
  password = "supersecret")
phenoTest <- createPhenoModel(
```

```

connectionDetails = connectionDetails,
xSpecCohort = 10934,
cdmDatabaseSchema = "my_cdm_data",
cohortDatabaseSchema = "my_results",
cohortDatabaseTable = "cohort",
outDatabaseSchema = "scratch.dbo", #should have write access
trainOutFile = "5XMI_train",
exclCohort = 1770120, #the xSens cohort
prevCohort = 1770119, #the cohort for prevalence determination
modelAnalysisId = "20181206V1",
excludedConcepts = c(312327, 314666),
addDescendantsToExclude = TRUE,
cdmShortName = "myCDM",
mainPopnCohort = 0, #use the entire person population
lowerAgeLimit = 18,
upperAgeLimit = 90,
gender = c(8507, 8532),
startDate = "20100101",
endDate = "20171231")

```

이 예제에서, 코호트 테이블("cohort")과 그것이 존재하는 데이터베이스("my_results")를 지정하였고 (cohortDatabaseSchema, cohortDatabaseTable - "my_results.cohort"), CONDITION_OCCURRENCE, DRUG_EXPOSURE 등을 찾을 수 있는 CDM 데이터베이스의 위치를 설정했다 (cdmDatabaseSchema - "my_cdm_data"). CDM에서 2010년 1월 1일부터 2017년 12 월 31일 사이에 첫 번째 의료기관 방문이 있는 환자만이 모델에 포함되었다. xSpec 코호트 생성 시 '312327', '314666' 및 그들의 하위 컨셉 등이 사용되지 않도록 하였다. 첫 번째 방문 시 환자의 나이는 18세에서 90 사이여야 한다. 이러한 파라미터를 이용하여 만들어진 예측 모델 결과는 "c:/temp/lr_results_5XMI_train_myCDM_ePPV0.75_20181206V1.rds"에 저장된다.

네 번째 단계 : 평가용 코호트 생성

createEvalCohort 기능은 관심 건강 결과의 예측된 가능성을 가진 각각의 사람들의 대규모 코호트를 생성하기 위하여 환자 수준 예측 패키지 기능 applyModel를 사용한다. 이 기능은 xSpec 코호트를 지정해야 한다 (파라미터 xSpecCohort를 xSpec 코호트 ID로 지정함으로써). 우리는 앞 과정에서 진행했었던 바와 같이 평가 코호트에 포함된 사람들의 특징도 지정해야 할 수도 있다. 이것은 나이 제한의 상한과 하한선을 지정하는 것도 포함할 수 있다 (나이를 각각 lowerAgeLimit와

`upperAgeLimit` 전달 인자로 설정함으로써), 성별 (파라미터 `gender`를 concept ID의 남성 그리고/혹은 여성으로 지정함으로써), 시작과 종료 날짜 (날짜를 각각 `startDate`와 `endDate` 전달인자로 설정함으로써), 그리고 사용할 인구의 코호트 ID를 `mainPopnCohort`로 지정하여 사람을 선별할 대규모 인구를 설계한다.

예를 들면:

```
setwd("c:/temp")
connectionDetails <- createConnectionDetails(
  dbms = "postgresql",
  server = "localhost/ohdsi",
  user = "joe",
  password = "supersecret")
evalCohort <- createEvalCohort(
  connectionDetails = connectionDetails,
  xSpecCohort = 10934,
  cdmDatabaseSchema = "my_cdm_data",
  cohortDatabaseSchema = "my_results",
  cohortDatabaseTable = "cohort",
  outDatabaseSchema = "scratch.dbo",
  testOutFile = "5XMI_eval",
  trainOutFile = "5XMI_train",
  modelAnalysisId = "20181206V1",
  evalAnalysisId = "20181206V1",
  cdmShortName = "myCDM",
  mainPopnCohort = 0,
  lowerAgeLimit = 18,
  upperAgeLimit = 90,
  gender = c(8507, 8532),
  startDate = "20100101",
  endDate = "20171231")
```

이 예시에서, 파라미터는 평가 코호트 파일을 생성하기 위하여 (“c:/temp/lr_results_5XMI_eval_myCDM_ePPV0.”) 이 모델 파일을 사용해야 한다고 명시한다 (“c:/temp/lr_results_5XMI_train_myCDM_ePPV0.75_20181206V1.rda”) 이 과정에서 생성된 모델과 평가 코호트 파일은 다음 과정에서 제공될 코호트 정의의 평가에 사용될 것이다.

다섯 번째 단계 : 코호트 정의 생성 및 검증

다음 과정은 평가받을 수 있게 코호트 정의를 생성하고 테스트하는 것이다. 이상적인 성능 특성은 관심 연구 문제를 설명하기 위하여 코호트의 의도된 사용에 따라 달라질 수 있다. 몇몇 특정한 문제에는, 굉장히 민감한 알고리즘이 필요할 수 있다; 다른 것은 더욱 구체적인 알고리즘이 필요할 수 있다. PheEvaluator를 사용한 코호트 정의의 성능 특성을 결정하는 과정은 그림 16.4가 보여준다.

그림 16.4의 A 부분에서, 테스트가 진행될 코호트 정의에서 사람들을 조사하고 코호트 정의에 포함되었던 사람 (Person ID가 016, 019, 022, 023, 025), 포함되지 않았던 사람 (Person Id가 017, 018, 020, 021, 024)의 평가 코호트 (앞 과정에서 생성되었었다)로부터 사람들을 찾았다. 이전에 이미 만든 예측 모델 ($p(O)$)을 사용하여 각각 포함된/포함되지 않은 사람들의 건강 결과 발생 가능성을 예측했었다.

다음과 같이 진양성, 진음성, 위양성, 위음성 결과를 추정하였다. (그림 16.4의 B 부분 참조):

1. 사용자가 정의한 코호트가 평가용 코호트의 한 사람을 포함한다면, 즉 사용자 정의 코호트가 어떤 사람을 '양성'으로 간주하는 확률, 즉 양성에 대한 기대값이 진양성에 추가되며, 1에서 해당 확률을 뺀, 음성에 대한 확률이 기대값으로 진음성에 추가된다. 예를 들어 Person ID 016 인 사람이 해당 임상 결과 outcome를 가질 (양성일) 확률이 99%라면, 기대값으로 0.99가 진 양성에 더해지고, 기대값으로 0.01(1.00-0.99)이 위양성에 더해진다. 이러한 과정이 사용자가 정의한 코호트의 모든 사람 (예를 들어 Person ID가 19, 22, 23, 25인 사람)에 대해 적용된다.
2. 마찬가지로, 사용자 정의 코호트가 평가 코호트에 존재하는 사람을 포함하지 않는 경우, 즉 사용자 정의 코호트가 음성으로 예측한 사람의 경우, 1에서 표현형에 대한 예측 확률에서 뺀 값이 음성에 대한 기대값으로 진음성에 추가되고, 표현형에 대한 예측 확률이 기대값으로 진 양성에 추가된다. 예를 들어 Person ID가 17인 사람이 결과를 가질 확률이 1%라고 예측했다면 (결과가 없을 확률이 99%라고 예측했다면), $1.00 - 0.01 = 0.99$ 가 진음성에 추가되고 0.01이 위음성에 추가된다. 이런 과정이 사용자 정의 코호트에 포함되지 않은 평가용 코호트의 사람 모두 (예를 들어 Person ID 018, 020, 021, 024인 사람)에게 반복된다.

평가용 코호트의 전체 집합에 대해 이러한 값을 추가한 후, 혼동 행렬의 4개 칸을 모두 추정값으로 채울 수 있다. 그리고 민감도, 특이도, 양성예측도와 같이 PA 성능의 추정치 point estimate를 만들 수 있다 (16.4 C). 추정치만을 예측할 수 있고, 추정치의 분산을 측정할 수는 없다는 것에 주의하자. 이 예제에서는 민감도, 특이도, 양성 예측도, 음성 예측도는 각각 0.99, 0.63, 0.42, 0.99로 나타났다.

`testPhenotype` 함수를 이용하여 사용자 정의 코호트의 성능을 측정할 수 있다. 이 함수는 우리가 모델과 평가용 코호트를 만들었던 이전 두 단계의 결과값을 사용한다. `createPhenoModel` 함수의 결과 상 나오는 RDS file에 대한 `modelFileName` 을 입력한다. 이 예제에서는 “c:/temp/lr_results_5XMI_train_myC

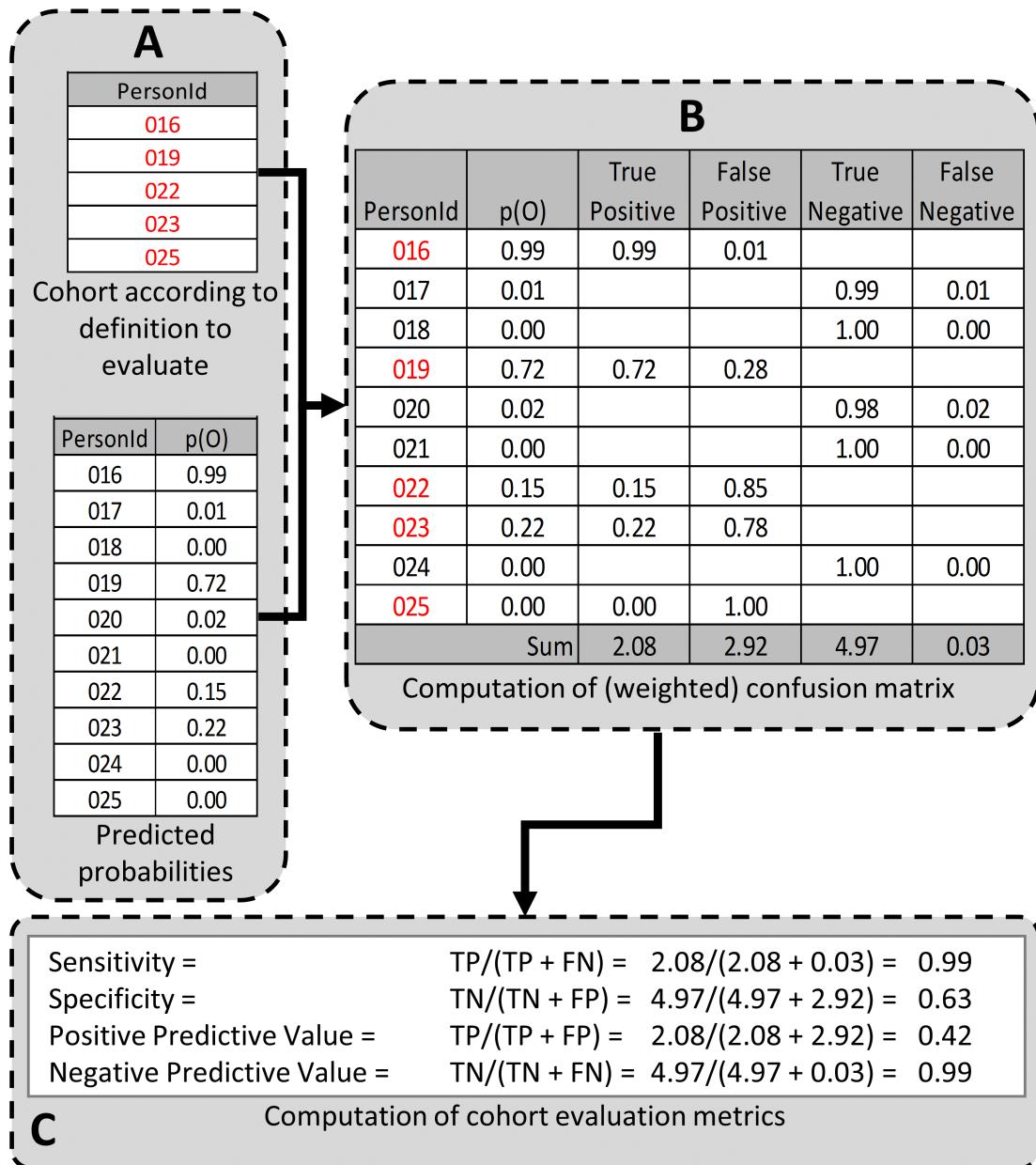


Figure 16.4: Determining the Performance Characteristics of a cohort definition using PheEvaluator.
 $p(O)$ = Probability of outcome; TP = True Positive; FN = False Negative; TN = True Negative;
FP = False Positive.

을 입력했다. `createEvalCohort` 함수의 결과로 출력되는 RDS 파일의 이름을 `resultsFileName` 인자로 정할 수 있다. 이 예제에서는 “c:/temp/lr_results_5XMI_eval_myCDM_ePPV0.75_20181206V1.rds”를 사용했다. 사용자 정의 코호트의 성능을 테스트하기 위해, `cohortPheno`의 코호트 ID를 해당 코호트의 ID로 입력한다. `phenText` 인자에는 ‘MI Occurrence, Hospital In-Patient Setting’과 같이 해당 코호트 정의에 대한 설명을 입력한다. `testText`에는 “5 X MI”와 같이 xSpec 정의에 대한 설명을 입력한다. 이 단계의 출력물은 코호트 정의의 성능의 추정치를 data frame 형태이다. `cutPoints` 인자에는 성능 추정치에 대한 참고치 값을 list 형태로 입력한다. 코호트 정의 성능은 `@ref(fig:phevaluatorDiagram` 그림에서 본 것과 같이 “기대값 expected value”으로 계산된다. 예측값 기반으로 성능을 추정하기 위하여, 우리는 `cutPoints` 인자에 “EV(expected value)”를 포함한 목록 list을 입력한다. 특정한 예측값을 기준으로 성능을 평가하고 싶을 수도 있을 것이다. 예를 들어, 예측 확률 0.5 이상에서 모든 성능이 결과에 대해 양성으로 간주하고 예측 확률 0.5 미만에서 모두 음성으로 간주하는 것을 확인하려면 “0.5”를 `cutPoints` 인자 목록에 추가한다. 예를 들어 다음과 같다.

```
setwd("c:/temp")
connectionDetails <- createConnectionDetails(
  dbms = "postgresql",
  server = "localhost/ohdsi",
  user = "joe",
  password = "supersecret")
phenoResult <- testPhenotype(
  connectionDetails = connectionDetails,
  cutPoints = c(0.1, 0.2, 0.3, 0.4, 0.5, "EV", 0.6, 0.7, 0.8, 0.9),
  resultsFileName =
  "c:/temp/lr_results_5XMI_eval_myCDM_ePPV0.75_20181206V1.rds",
  modelFileName =
  "c:/temp/lr_results_5XMI_train_myCDM_ePPV0.75_20181206V1.rds",
  cohortPheno = 1769702,
  phenText = "All MI by Phenotype 1 X In-patient, 1st Position",
  order = 1,
  testText = "MI xSpec Model - 5 X MI",
  cohortDatabaseSchema = "my_results",
  cohortTable = "cohort",
  cdmShortName = "myCDM")
```

이 예제에서 넓은 범위의 예측 임계값 prediction threshold이 예측값 expected value(“EV”)와

함께 cutPoints로 사용되었다. 주어진 설정에서, 이 단계에서 각각의 예측 임계값에 대한 민감도와 특이도와 같은 성능을 측정하여 제공한다. 평가는 이전 단계에서 만든 평가용 코호트과의 비교를 통해 이루어진다. 더욱 자세한 정보는 결과값으로 나온 데이터 프레임을 CSV로 저장하여 확인할 수 있다.

이러한 과정을 통해서 5개의 데이터 모음에서 4가지의 심근경색 코호트의 성능을 평가한 자료를 표 16.1에서 볼 수 있다. Cutrona 등이 평가한 것과 비슷한 코호트 (“ ≥ 1 X HOI, In-Patient”)의 경우 우리가 확인한 양성 예측도는 67% (59%-74%)이다.

Table 16.1: pheEvaluator을 이용하여 여러 데이터 모음에서 심근경색에 대한 4가지 코호트 정의에 대한 성능 평가. Sens – Sensitivity ; PPV – Positive Predictive Value ; Spec – Specificity; NPV – Negative Predictive Value; Dx Code – Diagnosis code for the cohort.

Phenotype Algorithm	Database	Sens	PPV	Spec	NPV
≥ 1 X HOI	CCAE	0.761	0.598	0.997	0.999
	Optum1862	0.723	0.530	0.995	0.998
	OptumGE66	0.643	0.534	0.973	0.982
	MDCD	0.676	0.468	0.990	0.996
	MDCR	0.665	0.553	0.977	0.985
≥ 2 X HOI	CCAE	0.585	0.769	0.999	0.998
	Optum1862	0.495	0.693	0.998	0.996
	OptumGE66	0.382	0.644	0.990	0.971
	MDCD	0.454	0.628	0.996	0.993
	MDCR	0.418	0.674	0.991	0.975
≥ 1 X HOI, In-Patient	CCAE	0.674	0.737	0.999	0.998
	Optum1862	0.623	0.693	0.998	0.997
	OptumGE66	0.521	0.655	0.987	0.977
	MDCD	0.573	0.593	0.995	0.994
	MDCR	0.544	0.649	0.987	0.980
1 X HOI, In-Patient, 1st Position	CCAE	0.633	0.788	0.999	0.998
	Optum1862	0.581	0.754	0.999	0.997
	OptumGE66	0.445	0.711	0.991	0.974
	MDCD	0.499	0.666	0.997	0.993
	MDCR	0.445	0.711	0.991	0.974

16.5 근거의 일반화

코호트는 특정 관찰 데이터베이스의 맥락 안에서 잘 정의되고 완전히 평가될 수 있지만, 임상 타당성은 그 결과가 관심 대상 모집단에 일반화될 수 있다고 간주하는 정도에 의해 제한된다. 동일한 주제에 대한 복수의 관찰형 연구는 다른 결과를 산출할 수 있는데, 이는 설계와 분석 방법뿐만 아니라 데이터 출처의 선택에도 의해 발생할 수 있다. Madigan et al. (2013b)은 데이터베이스의 선택이 관찰 연구의 결과에 영향을 미친다는 것을 보여주었다. 그들은 10개의 관찰형 데이터베이스에 걸쳐 53개의 약물-결과 쌍과 2개의 연구 설계 (코호트 연구 설계 및 자기 대조군 환자 연구 설계)에 대한 결과에서 이질성을 체계적으로 조사했다. 연구 설계를 일정하게 유지했음에도 불구하고, 실제 추정치의 상당한 이질성이 관찰되었다.

OHDSI 네트워크의 관찰형 데이터베이스는 그들이 반영하는 인구 집단 (예를 들어 소아 대 노인, 사보험 가입자 대 공공보험 가입자), 데이터가 수집되는 치료 환경 (예를 들어 입원 환자 대 외래 환자, 1차 의료기관 대 2차, 특수 의료기관), 데이터 수집 프로세스 (예를 들어 행정 청구, 전자 건강 기록, 임상 레지스트리) 및 치료의 기반이 되는 국가 및 지역 보건 시스템 등에서 상당한 차이가 있다. 이러한 차이는 질병과 의료 개입의 영향을 연구할 때 관찰되는 결과의 이질성으로 나타날 수 있으며 네트워크 연구를 위한 각 데이터베이스의 품질에 대한 신뢰도에 영향을 미칠 수도 있다. OHDSI 네트워크 내의 모든 데이터베이스는 CDM으로 표준화되지만, 표준화가 모집단 전체에 존재하는 진정한 고유의 이질성을 없애기 위함이 아니라, 네트워크 전체의 이질성을 인식하고 더 잘 이해하기 위한 일관된 프레임워크 제공을 강화한다는 점이 중요하다. OHDSI 연구 네트워크는 전 세계의 다양한 데이터베이스에 동일한 분석 프로세스를 적용할 수 있는 환경을 제공하기 때문에 연구자는 다른 방법론적 측면을 일정하게 유지하면서 여러 데이터베이스에 걸쳐 결과를 해석할 수 있다. OHDSI의 커뮤니티의 오픈 사이언스에 대한 협력적 접근은 임상 영역의 전문가와 방법론적 분석 전문가가 함께 OHDSI 네트워크의 데이터에 걸쳐 임상적 타당성을 이해하는 집단 지성에 도달하기 위한 한 가지 방법이며, 이는 이 데이터를 이용하여 만들어진 근거의 신뢰를 구축하기 위한 근본이 되어야 한다.

16.6 요약



- 임상적 타당성은 원천 데이터의 특성을 이해하고, 분석 내의 코호트 정의에 대한 성능을 평가하고, 대상 모집단에 관한 연구의 일반성을 평가함으로써 확립할 수 있다.
- 코호트 정의 및 사용한 관찰 데이터에 근거하여 식별된 개인이 의도한 표현형에 진정으로 속하는지 여부를 판단하여 코호트 정의의 성능을 평가할 수 있다.
- 코호트 정의 타당성 검사를 위해서는 측정 오류를 완전히 요약하고 조정할 수 있도록

민감도, 특이도, 양성예측도를 포함한 여러 성능 지표를 추정해야 한다.

- 원천 기록 확인을 통한 임상적 판단 및 PheEvaluator는 코호트 정의 타당성 검사에 대한 두 가지 대안적 접근 방식이다.
- OHDSI 네트워크 연구는 데이터베이스의 이질성을 평가하고 연구 결과의 일반성을 확장하여 실세계 근거의 임상적 타당성을 향상하는 메커니즘을 제공한다.

Chapter 17

소프트웨어의 타당성

Chapter lead: Martijn Schuemie

이 장의 주요 질문은 다음과 같다.

소프트웨어가 우리의 예상대로 작동하는가?

소프트웨어의 타당성은 근거 품질의 필수적인 구성 요소이며, 분석 소프트웨어가 예상하는 기능을 수행하는 경우에만 신뢰 있는 근거를 생성할 수 있다. 17.1.1절에 설명한 대로 모든 연구를 소프트웨어 개발 활동으로 보고 CDM 데이터부터 추정치, 그림과 표에 이르는 결과를 생성하는 전체 분석을 실행하는 자동화된 스크립트를 만드는 것이 필수적이며, 이 스크립트와 스크립트 내에 사용된 모든 소프트웨어에 대해서 반드시 유효성 검증을 해야 한다. 8.1절에서 언급하였듯이, 전체 분석 과정을 사용자가 직접 자신의 코드로 작성하거나, 또는 오딧세이 연구방법론 라이브러리 OHDSI Methods Library에서 적절한 기능을 가져다 사용할 수도 있다. 연구방법론 라이브러리를 사용하면 이미 타당성을 보장하기 위해 여러 노력이 가해진 방법을 사용함으로써 전체 분석과정의 타당성을 정립하는 데 있어 부담을 덜 수 있다는 장점이 있다.

이 단원에서는 먼저 타당한 분석 코드를 작성하기 위한 몇 가지 모범 사례를 살펴볼 것이고, 그 이후 소프트웨어 개발 과정과 검사를 통해 연구방법론 라이브러리의 타당성을 검사하는 방법에 관해 설명할 것이다.

17.1 분석 코드의 타당성

17.1.1 재현성 충족을 위한 자동화

전통적으로 관찰연구는 종종 일련의 과정이라기보다는 여행에 비유되기도 한다. 데이터베이스 전문가는 데이터베이스에서 데이터 모음을 추출하여 이를 데이터 분석가에게 넘겨주고, 데이터 분석가는 스프레드시트 편집기나 다른 편집 도구를 사용하여 데이터 모음을 열고 분석을 수행한다. 마지막으로 분석 결과가 생성되지만, 어떻게 이 결과가 도출되었는지는 거의 보존되지 않는다. 여행의 목적지에는 도달했지만, 그곳에 도달하기 위해 취한 정확한 단계를 추적할 수는 없다. 이 방법은 재현할 수 없고 not reproducible, 투명성이 부족하기 때문에 lack transparency 온전히 받아들여지지 않는다.

따라서 (이런 한계를 극복하기 위해서) 근거를 생성하는 모든 분석은 완전히 자동화되어야 한다. 분석의 자동화라는 의미는 단일 스크립트에 CDM 형식의 데이터베이스에서부터 표와 그림을 포함하는 전체 분석 결과를 생성해 내는 전 과정을 담아내어, 단 하나의 명령으로 모든 전체 과정을 재실행 할 수 있도록 구현하는 것이다. 분석은 단순히 수를 세는 것부터 수백만 건의 연구 문제에 대해 경험적으로 보정된 추정치를 생성하는 것까지 다양한 복잡성을 띠지만, 동일한 원칙이 적용된다. 이 스크립트는 다른 스크립트를 호출하여 하위 분석 절차를 진행하도록 할 수 있다.

분석 스크립트는 모든 컴퓨터 언어를 통해 구현할 수 있으나, 오딧세이에서는 R 언어를 선호한다. DatabaseConnector라는 R 패키지 덕분에 R 환경에서 CDM 형식의 데이터에 직접 연결하여 사용할 수 있으며, 오딧세이 연구방법론 라이브러리 OHDSI Methods Library 내의 다른 R 패키지를 통해 고급 분석을 사용할 수 있다.

17.1.2 프로그래밍 모범 사례

관찰연구 분석 방법은 최종 결과를 생성하기 위해 많은 단계를 거치거나 매우 복잡해질 수 있다. 이러한 복잡성으로 인해 분석 코드를 유지 관리하기 더욱 어려워지고, 오류가 발생할 가능성이 높아질 뿐 아니라 오류를 인식하기조차 어려울 수 있다. 다행히도 많은 컴퓨터 프로그래머는 수년 동안 복잡한 코드를 작성하고 다루는 데 있어서 이 코드를 읽고, 재사용하고, 적용하고, 검증하는 과정이 수월하도록 관리하는 몇 가지 모범 사례를 개발해 두었다. (Martin 2008) 이 우수 사례는 상당한 분량을 차지하므로, 여기서는 4가지 중요한 원칙을 강조하도록 하겠다.

- **축약화 Abstraction:** 모든 것을 수행하는 하나의 큰 스크립트 (소위 “스파게티 코드”라고 하며 코드 라인간 종속성이 있다. 예를 들면, 10행에서 설정된 값이 1000행에서 사용되는 경우를 들 수 있다) 를 작성하는 대신 코드를 “함수”라고 하는 작은 단위로 구성할 수 있다. 함수는 명확한 목표를 가져야 하며 (예를 들면 “무작위 샘플 추출”) 일단 생성하고 나면, 다

른 스크립트에서도 직관적으로 사용할 수 있다. 이처럼 우리는 함수를 통해서 이해하기 쉬운 개념으로 코드를 추상화하고 축약할 수 있다.

- **캡슐화 Encapsulation:** 축약 작업이 진행되기 위해서는 함수 간의 의존성을 명확하게 정의하고 최소화해야 한다. 예로 들었던 무작위 샘플 추출 기능에서는 몇 가지 인수 arguments (예를 들어, 데이터 모음과 추출 집단의 크기)와 하나의 출력값 (예를 들어, 추출 집단)이 있어야 한다. 이 함수가 수행하는 기능에 있어서 어떠한 것도 영향을 줄 수 없어야 하며, 함수 외부에서 설정된 소위 “전역 변수 global variables”를 함수 내에서 사용하지 말아야 한다.
- **명확한 명명 Clear naming:** 변수 혹은 함수의 이름은 명확해야 하며, 자연어처럼 읽을 수 있도록 하라. 예를 들어, `x <- spl(y, 100)` 보다는 우리가 읽을 수 있도록 `sampledPatients <- takeSample(patients, sampleSize = 100)`처럼 작성하라. 축약어를 사용하고자 하는 충동을 억눌러라. 현대 언어는 변수, 함수의 이름으로 사용하는 데 있어 충분히 다양하게 사용할 수 있다.
- **재사용 Reuse:** 명확하고 잘 캡슐화된 기능을 작성하였을 때 얻는 장점 중 하나는 계속해서 재사용할 수 있다는 점이다. 이렇게 하면 시간이 절약될 뿐 아니라 코드가 줄어 복잡성이 줄고, 오류가 발생할 가능성성이 줄어 든다.

17.1.3 코드 검증

소프트웨어 코드의 타당성을 검증하기 위한 여러 가지 방법이 있지만, 관찰연구에서 사용하는 코드와 관련하여 두 가지 방법을 소개하고자 한다.

- **코드 삼자 검토 Code review:** 한 사람이 코드를 작성하고, 다른 사람이 코드를 검토한다.
- **이중 코딩 Double coding:** 두 사람이 독립적으로 분석 코드를 작성하고, 이후에 스크립트 실행 결과를 비교한다.

코드 삼자 검토는 일반적으로 작업량이 적지만, 검토자가 일부 오류를 놓칠 수 있다는 단점이 있다. 이중 코딩은 다소 노동 집약적이지만 오류를 놓칠 가능성이 적다. 이중 코딩의 다른 단점은 두 개별적인 코드의 구현이 대부분, 아니 언제나 다른 결과를 나타낸다는 점이다. 이는 임의적인 사소한 선택으로 인해 발생한다. (예를 들어 “노출 종료까지”라는 말은 노출 종료일을 포함하는가 포함하지 않는가?) 결과적으로, 독립적인 두 프로그래머는 독립적으로 이중 코딩을 수행해야 함에도, 분석을 상호 조정하기 위해 협력하여야 할 필요가 있다.

단위 검정 unit testing과 같은 다른 소프트웨어 검증 방법은 관찰 연구 특성상 데이터의 입력 (CDM 내의 데이터) 과 출력 (연구 결과) 사이에 높은 복잡도의 관계를 가진 일회성 과정이므로 다소 유용하지 못하기 때문에 관련성이 적다고 할 수 있다. 이러한 다른 검증방법은 연구방법론 라이브러리 내에서는 적용 되어있다는 점을 주의하라.

17.1.4 연구 방법론 라이브러리의 활용

오딧세이 연구방법론 라이브러리 OHDSI Methods Library 는 수많은 기능을 제공하기 때문에, 대부분의 관찰 연구를 몇 줄의 코드만으로도 구현할 수 있다. 따라서 연구방법론 라이브러리를 사용하면 개인 연구 내에서 연구자가 타당성을 입증해야 하는 부담이 연구방법론 라이브러리로 옮겨가게 된다. 연구방법론 라이브러리의 타당성은 자체의 소프트웨어 개발 과정과 광범위한 시험을 통해 보장된다.

17.2 연구 방법론 라이브러리 소프트웨어의 개발 과정

오딧세이 연구방법론 라이브러리는 오딧세이 커뮤니티에서 개발하였으며, 라이브러리에 변경이 제안된 사항은 GitHub의 issue tracker (예를 들어 CohortMethod issue tracker¹⁾)와 오딧세이 포럼², 이 두가지 장소에서 논의된다. 두 장소 모두 공개되어 있다. 커뮤니티의 모든 구성원은 소프트웨어 코드를 라이브러리에 제공할 수 있지만, 기준에 배포된 소프트웨어 버전에 대한 변경사항은 오딧세이 인구 수준 추정(PLE) 그룹 리더십 (현재 Marc Sucahrd 박사, Martigin Schuemie 박사)과 환자 수준 예측(PLP) 그룹 리더십 (현재 Peter Rijinbeek 박사, Jenna Reps 박사)만이 최종 결정할 수 있다.

사용자는 연구방법론 라이브러리의 GitHub 저장소(master branch)에서 직접 설치할 수 있고, “drat”이라는 시스템을 이용하여서도 최신 버전을 설치할 수 있다. R의 Comprehensive R Archive Network(CRAN)을 통해서 다양한 연구방법론 라이브러리 패키지를 사용할 수 있으며, 이용할 수 있는 패키지의 수는 점점 증가할 것으로 예상된다.

오딧세이 연구방법론 라이브러리의 정확성, 신뢰성 및 일관성을 최대화하기 위해서 합리적인 소프트웨어 개발법 및 시험 방법을 사용한다. 연구방법론 라이브러리의 모든 원천 코드는 Apache License V2로 배포됨에 따라, R, C++, SQL, Java 등 어떤 언어로 작성이 되었어도, 오딧세이 커뮤니티의 모든 회원과 대중이 동료 평가 peer review를 할 수 있다. 따라서, 연구방법론 라이브러리 내부에 구현된 모든 기능은 정확성, 신뢰성 그리고 일관성의 향상을 위해서 지속적인 비판과 이로 인한 개선이 이루어져야 한다.

17.2.1 원천 코드 관리

연구방법론 라이브러리의 모든 원천 코드는 github을 통해 접근할 수 있는 원천 코드 버전 관리 시스템인 git을 통해 관리되며, 오딧세이 연구방법론 라이브러리 저장소의 접근을 관리하고 있다. 전 세계 누구나 원천 코드를 볼 수 있으며, 오딧세이 커뮤니티의 멤버 누구나 pull request라고 부르는

¹<https://github.com/OHDSI/CohortMethod/issues>

²<http://forums.ohdsi.org/>

코드 변경 요청을 제출할 수 있다. 오딧세이 인구 수준 추정(PLE) 그룹과 환자 수준 예측(PLP) 그룹 리더십은 이 코드 변경 요청을 승인할 수 있고, master branch을 변경하고 새로운 버전을 배포할 수 있다. 지속적인 코드 변경사항 로그는 GitHub 저장소에 유지되며, 코드와 문서의 모든 변동 사항을 반영한다. 이러한 변경사항 로그가 대중으로부터의 검토를 가능하게 한다.

새로운 버전은 필요 시 두 오딧세이 그룹의 리더십의 판단 하에 배포 된다. 프로그램 패키지의 DESCRIPTION 파일에 정의된 대로 패키지 버전 번호가 배포 버전의 번호보다 큰 master branch로 변경사항을 push 하여 새 배포가 시작된다. 이는 자동으로 패키지를 테스트하고, 모든 검사를 통과하면 버전관리 시스템에서 새 버전에 자동으로 태그가 지정되고 패키지가 오딧세이 drat 저장소에 자동으로 업로드된다. 새 버전은 3가지 표기 원칙에 따라 번호가 부여된다

- **세부 버전 Micro version** (4.3.2에서 4.3.3으로 변경하는 경우) 버그를 수정한 경우에 한함. 새로운 기능 추가는 없으며, 상위, 하위호환성을 보장됨.
- **부 버전 Minor version** (4.3.3에서 4.4.0으로 변경하는 경우) 기능적으로 추가가 되었을 때. 하위 호환성이 보장됨.
- **주 버전 Major version** (4.4.0에서 5.0.0으로 변경하는 경우) 주요 개선사항이 생겼을 때. 호환성을 보장하지 않음.

17.2.2 문서화

연구방법론 라이브러리의 모든 패키지는 R 내부 문서화 프레임워크를 통해 문서화 된다. 각 패키지에는 패키지에서 사용 가능한 기능을 설명하는 정의서를 가지고 있다. 기능 정의서 및 기능 구현에 관한 사항을 정리하기 위해서 기능 문서와 원천 코드를 단일 파일로 결합하는 roxygen2 소프트웨어를 사용한다. 패키지 설명서는 R의 명령 입력을 통해 패키지 저장소에 PDF 형태로 제공된다. 또한 많은 패키지는 패키지의 활용법을 담은 도움글 vignettes을 가지고 있다. 모든 문서는 연구방법론 라이브러리의 웹사이트에서 확인할 수 있다.³

모든 연구방법론 라이브러리의 원천 코드는 실제 사용자가 사용할 수 있으며, GitHub의 issue 시스템 및 오딧세이 포럼을 사용하여 커뮤니티의 피드백을 받을 수 있다.

17.2.3 현재 및 과거 버전으로의 접근

연구방법론 라이브러리 패키지의 현재 및 과거 버전은 아래 두 위치에서 접근할 수 있다. 먼저 GitHub 버전관리 시스템은 각 패키지의 전체 개발 과정을 가지고 있으며, 각 단계의 패키지의 상태를 재구성하고 검색할 수 있다. 각각의 출시 버전이 GitHub에 태그 되어있다. 두 번째는 오딧세이 GitHub의 drat 저장소에 R 소스 패키지가 저장되어 있다.

³<https://ohdsi.github.io/MethodsLibrary/>

17.2.4 유지 보수, 지원 및 중단

오딧세이는 각 최신 버전의 연구방법론 라이브러리 내 버그를 보고하고 수정하고 패치하는 것을 적극적으로 지원하고 있다. GitHub 이슈 시스템과 오딧세이 포럼을 활용하여 관련 문제를 제기 및 보고할 수 있다. 각 패키지는 패키지 설명서와 추가적으로 도움글, 온라인 비디오 튜토리얼 영상이나 자료가 제공된다.

17.2.5 검증된 인력

오딧세이 커뮤니티의 회원은 통계학의 여러 분야에 해당되는 사람들로 구성되어 있고, 학계, 비영리단체, 산업계 등의 다양한 기반을 가진 전 세계 사람들로 구성되어 있다.

오딧세이의 인구 수준 추정(PLE) 그룹과 환자 수준 예측(PLP) 그룹의 리더는 공인된 교육기관의 박사 학위를 보유하고 있으며, 동료평가 저널에 다양하게 논문을 게재해오고 있다.

17.2.6 물리적, 논리적 보안체계

오딧세이 연구방법론 라이브러리는 GitHub⁴ 시스템에서 호스팅되고 있다. GitHub의 보안에 관한 부분은 다음 사이트에서 확인할 수 있다. <https://github.com/security> 오딧세이 커뮤니티의 모든 구성원은 연구방법론 라이브러리를 변경 요청할 수 있으며, 이 때 사용자아이디와 비밀번호가 요구된다. 변경사항을 승인은 인구 수준 추정(PLE) 그룹과 환자 수준 예측(PLP) 그룹의 리더를 통해서 가능하다. 사용자 계정은 표준 보안정책 및 기능 요구사항에 따라 접근이 제한된다.

17.2.7 복구 체계

오딧세이 연구방법론 라이브러리는 GitHub 시스템에 호스팅되어 있다. GitHub의 사고 복구 체계는 다음 사이트에서 확인할 수 있다: <https://github.com/security>

17.3 연구방법론 라이브러리 기능 검사

우리는 연구 방법론 라이브러리를 패키지의 단순 기능 검사 (단위 검정)와 시뮬레이션을 이용한 고난도 기능검사의 두 가지로 나누어 테스트를 수행하고 있다.

17.3.1 단위 검증

잘 알려진 데이터 및 결과에 대해서는 원천 코드를 자동으로 테스트 할 수 있는 자동 유효성 검사가 오딧세이에 의해서 운영되고 개선되고 있다. 각 유효성 검사는 일부 입력데이터를 지정하고, 검사

⁴<https://github.com/>

대상의 패키지 중 하나의 기능을 실행하고 출력이 정상적인지의 여부를 평가한다 (예를 들어 소수의 환자군을 가진 임시 데이터를 가지고 성향점수 매칭을 시행함). 더욱 복잡한 기능의 경우 R에서 사용할 수 있는 다른 기능을 조합하여 예상 결과를 생성해볼 수 있다 (예를 들어 대용량 회귀 분석 엔진인 Cyclops는 간단한 문제에 대한 결과를 여러 회귀 방법을 통해서 비교하여 테스트함). 오딧세이에서는 실행 가능한 코드라면 100% 테스트 할 수 있도록 하는 것을 목표로 하고 있다.

이 검사 기능은 패키지가 수정되었을 때 자동으로 수행되게 되어 있다 (정확히는 변화된 패키지가 GitHub 저장소에 push 되었을 때). 검사 도중 에러 발생 시 자동으로 그룹 리더들에게 이메일이 발송되고, 새로운 패키지 버전 이전에 문제를 반드시 해결하도록 하고 있다. 이 검사에 대한 코드와 예상 결과는 검토 가능할 뿐 아니라 적절한 다른 환경에서도 적용할 수 있으며, 관리자뿐만 아니라 일반 사용자도 설치 과정의 일부로 실행하여 연구방법론 라이브러리의 정확성, 신뢰성 및 일관성에 대한 객관적 증거를 제공할 수 있다.

17.3.2 모의시행

더 복잡한 기능은 입력했을 때 어떤 결과를 나타내는지 뚜렷하지 않은 경우도 있다. 이러한 경우 모의 시행 simulation을 하기도 하는데, 특정한 통계 모델에서 나온 값을 입력하고 알려진 모델의 결과값을 생성하는지 여부를 확인한다. 예를 들어 SelfControlledCaseSeries 패키지에서 모의 시행은 분석 방법이 임시 데이터를 이용해서 시간의 흐름을 적절히 파악해서 모델을 만들었는지 검증하는데 사용한다.

17.4 요약



- 관찰 연구는 재현성과 투명성을 보장하기 위해서 CDM 데이터에서부터 결과에 이르기 까지 전체 분석을 실행하는自動화된 스크립트를 구현해야 한다.
- 연구에 사용하는 분석 코드는 축약화, 캡슐화, 명확한 명명법, 코드 재사용이라는 좋은 프로그래밍 방법을 준수해야 한다.
- 코드 삼자 검토 또는 이중 코딩을 사용하여 사용자의 코드를 검증할 수 있다.
- 연구방법론 라이브러리는 관찰 연구에 사용할 수 있는 검증된 기능을 제공한다.
- 연구 방법론 라이브러리는 검증된 소프트웨어 및 검사법 개발을 목표로 하는 개발 과정을 통해 검증된다.

Chapter 18

방법론적 타당성

Chapter lead: Martijn Schuemie

이번 장에서 우리는 다음의 질문에 대해 답하려 한다.

우리의 방법론은 연구 질문에 대해 답하기에 유효한가?

“방법”에는 연구 설계뿐만 아니라 데이터와 설계의 시행도 포함된다. 따라서, 방법론적 타당성은 다소 포괄적이다. 좋은 자료의 질, 임상적 타당성 및 소프트웨어 타당성 없이 좋은 방법의 유효성을 논하는 것은 때때로 불가능하다. 근거의 질적 측면은 이미 방법론적 타당성을 고려하기 전에 별도로 다루어져야 한다.

방법론적 타당성을 확립할 때 핵심 행동은 분석에서 중요한 가정이 충족되었는지의 여부를 평가하는 것이다. 예를 들어, 우리는 성향 점수 짹짓기가 두 집단을 비교할 수 있게 한다고 가정하지만, 이것이 사실인지 평가할 필요가 있다. 가능한 경우, 이러한 가정을 검증하기 위해 경험적 검정을 해야 한다. 예를 들어, 우리는 두 집단이 실제로 짹짓기 후에 넓은 범위의 특성에서 비교 가능하다는 것을 보여주기 위해 진단을 생성할 수 있다. OHDSI에서 우리의 분석이 수행될 때마다 생성되고 평가되어야 하는 많은 표준화된 진단을 개발하였다.

이 장에서는 인구 수준 추정에 사용되는 방법의 타당성에 초점을 맞출 것이다. 먼저, 일부 연구 설계 별 진단을 간략히 강조한 다음, 모든 모집단 수준의 추정 연구가 아닌 대부분의 진단에 적용 가능한 진단에 대해 논의할 것이다. 다음은 OHDSI 도구를 사용하여 이러한 진단을 수행하는 방법에 대한 단계별 설명이다. 우리는 OHDSI Methods Benchmark와 OHDSI 연구방법론 라이브러리 Methods Library에 대한 적용에 대해 검토하면서 이 장의 주제를 마무리한다.

18.1 설계별 진단

각 연구 설계에 대해 해당 설계마다 특정한 진단이 있다. 이러한 진단 중 많은 부분이 OHDSI 연구방법론 라이브러리의 R 패키지로 구현이 되어있고, 쉽게 사용할 수 있다. 예를 들어, 12.9절에는 OHDSI Methods Library 패키지에서 생성된 광범위한 진단이 나열되어 있다:

- 코호트의 초기 비교성을 평가하기 위한 **성향 점수 분포 Propensity score distribution**.
- 모형에서 제외해야 하는 잠재적 변수를 식별하기 위한 **성향 모델 Propensity model**.
- 성향 점수 propensity score 보정을 통해 코호트를 비교할 수 있는지 평가 (기저 공변량을 통해 측정)를 위한 **공변량 균형 Covariate balance**.
- 다양한 분석 단계에서 몇 명의 피험자가 제외되었는지를 관찰하여, 초기 관심 집단에 대한 결과의 일반화 가능성을 알리기 위한 **손실표 Attrition**.
- 질문에 대답하기에 충분한 데이터가 있는지 평가하기 위한 **검정력 Power**.
- 일반적인 관찰 시작 시간을 평가하고 콕스 Cox 모델의 비례 가정이 충족되는지 여부를 위한 **Kaplan Meier curve**.

다른 연구 설계는 이러한 설계의 다른 가정을 검정하기 위해 다른 진단이 필요하다. 예를 들어, 자가-대조군 환자 시리즈(SCCS) 설계의 경우 관찰의 종료가 결과와는 무관하다는 필수 가정을 확인할 수 있다. 이 가정은 심근 경색과 같이 심각하고 잠재적으로 치명적인 사건의 경우에서 종종 위반된다. 우리는 관찰 기간 종료의 시점이 중도 절단과 중도 절단 되지 않은 시점을 보여주는 히스토그램인 그림 18.1에 표시된 도표를 생성하여 가정이 유지되는지의 여부를 평가할 수 있다. 데이터에서 관찰 기간이 데이터 수집의 종료 날짜 (전체 데이터베이스에서 관찰이 중단된 날짜, 예를 들어, 추출 날짜 또는 연구 종료 날짜)를 중도 절단 되지 않은 것으로 간주하고, 다른 모든 관찰을 중도 절단한 것으로 간주한다. 그림 18.1에서 우리는 두 분포 사이의 사소한 차이만을 보고도, 우리의 가정이 유지된다는 것을 시사한다.

18.2 모든 추정을 위한 진단법

연구 설계 특이적인 진단 후, 모든 인과 효과 추정 방법에 걸쳐 적용 가능한 몇 가지 진단법이 있다. 이 중 대부분이 대조 가설의 사용과 해답이 이미 알려진 연구 질문에 의존한다. 대조군 가설을 활용하면 우리의 설계가 사실과 일치하는 결과를 산출하는지 평가할 수 있다. 대조군은 음성 대조군과 양성 대조군으로 나눌 수 있다.

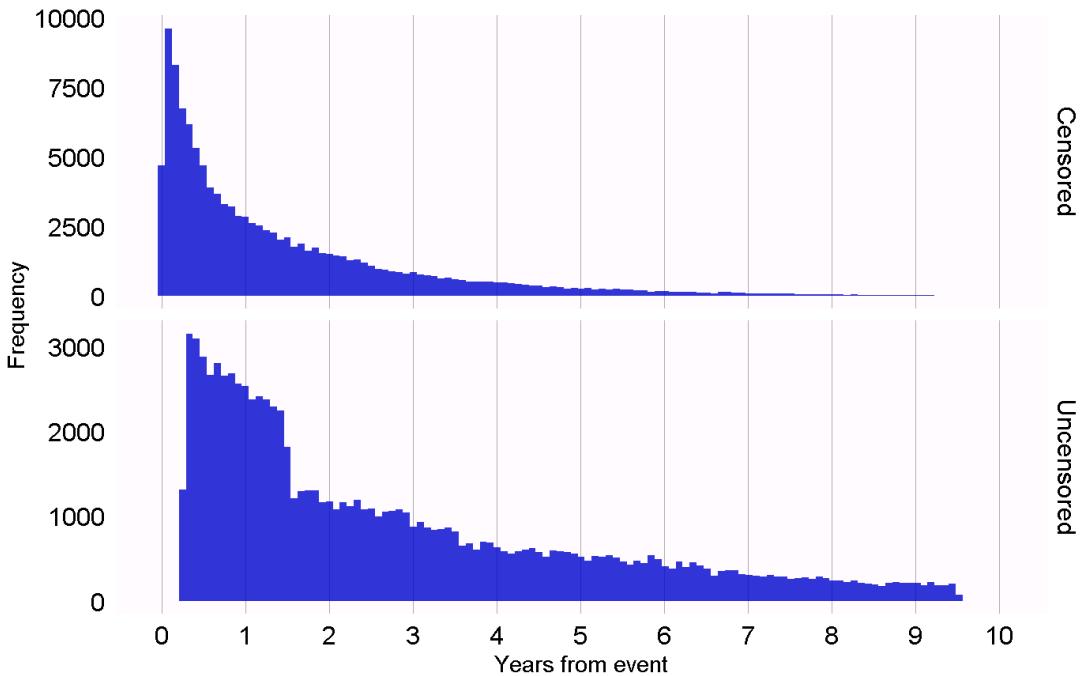


Figure 18.1: 중도 절단된 사람과 중도절단 되지 않은 사람에 대한 관찰 종료 시간.

18.2.1 음성 대조군

음성 대조군은 인과적 효과가 없을 것으로 생각되는 노출-결과 쌍이며, 교란 confounding, (Lipsitch et al., 2010) 선택비뚤림 selection bias, 측정 오류 measurement error를 감지하기 위한 수단으로 권장해온 음성 대조군 혹은 “falsification endpoints” (Prasad and Jena, 2013) 을 포함한다. (Arnold et al., 2016) 예를 들어, 소아기 질병과 후기 다발성 경화증 multiple sclerosis의 관계를 조사하는 한 연구에서 (Zaadstra et al., 2008), 저자는 다발성 경화증을 유발하지 않는 것으로 생각되는 세 가지 음성 대조군을 포함했다: 팔 골절, 뇌진탕, 편도선 절제술. 연구자의 분석 결과 이 세 가지 대조군 중 두 가지가 다발성 경화증과 통계적으로 유의한 연관성이 있는 것으로 나타났는데, 그것은 그 연구에 비뚤림이 있을 수 있음을 의미한다.

여러 음성 대조군 중에서 관심 가설과 비교 가능한 음성 대조군을 선택해야 한다. 즉, 일반적으로 관심 가설과 동일한 결과 (“결과 대조군 outcome controls”이라고 한다) 또는 동일한 노출(“exposure controls”)을 갖는 노출-결과 쌍을 선택해야 한다. 음성 대조군은 다음 기준을 충족해야 한다:

- 노출이 결과의 원인이 되어선 안 된다. 인과 관계를 생각하는 한 가지 방법은 반사실적 counterfactual인 사고이다: 환자가 노출된 경우와 비교하여 환자가 노출되지 않은 경우에서 결과가 초래 (혹은 예방) 될 수 있는가? 때로는 명확한데, 예를 들어 ACEi는 혈관 부종을 유발하는 것으로 알려져 있다. 다른 경우에는 훨씬 덜 명확하다. 예를 들어 고혈압을 유발할 수 있는

약물은, 고혈압의 결과인 심혈관 질환을 간접적으로 유발할 수 있다.

- 노출은 또한 결과를 예방하거나 치료해서도 안 된다. 이것은 단지 또 다른 인과관계로서, 실제 효과 크기가 1이라고 믿어야 할 경우에는 존재해서는 안 되는 인과 관계이다.
- 음성 대조군은 연구자가 가지고 있는 데이터에 존재해야 하며, 이상적으로는 충분한 숫자를 가지고 있어야 한다. 음성 대조군의 유병률에 따라 우선순위를 정해서 달성한다.
- 음성 대조군은 이상적으로 독립적이어야 한다. 예를 들어, 서로의 조상 (예를 들어, “내성 손톱”과 “내성 발톱”)이거나, 형제자매 (예를 들어, “fracture of left femur”나 “fracture of right femur”)인 음성 대조군은 피해야 한다 (개념간에 조상-자녀, 형제 관계에 있으면 안 된다).
- 음성 대조군은 이상적으로 어느 정도의 비뚤림의 가능성성이 있어야 한다. 예를 들어, 누군가의 사회 보장 번호의 마지막 숫자는 기본적으로 임의의 숫자이며 교란을 의미하진 않는다. 따라서 음성 대조군으로 사용해서는 안된다.

음성 대조군이 노출-결과 쌍과 동일한 교란 요인 구조를 가져야 한다는 주장도 있다. (Lipsitch et al., 2010) 그러나, 우리는 이 교란 구조는 알 수 없다고 믿는다. 현실에서 발견되는 변수 사이의 관계는 때때로 사람들이 상상하는 것보다 훨씬 더 복잡하다. 또한, 교란 요인의 구조가 알려진 경우에도, 정확히 동일한 구조로 되어 있지만, 직접적인 인과 효과가 없는 음성 대조군이 존재할 가능성은 작다. 이런 이유로 OHDSI에서는 많은 수의 음성 대조군을 동시에 사용하는데, 그러한 다양한 음성 대조군 모음은 연구자가 설계한 연구의 비뚤림을 포함하여 다양한 유형의 비뚤림을 가지고 있을 것이기 때문이다.

노출과 결과 사이에 인과 관계가 없다는 것은 거의 출판 되어 있지 않다. 그래서 우리는 연관성에 대한 근거가 부족하다는 것이 연관성의 결여를 의미한다고 종종 가정한다. 노출과 결과 모두 광범위하게 연구되었으므로 연관성이 탐지될 수 있었다면 이 가정은 유지될 가능성이 더 높다. 예를 들어, 완전히 새로운 약물에 대한 근거의 부족은 연관성의 부족이 아니라 아직은 관련 지식이 부족함을 의미한다. 이러한 원칙을 염두에 두고 음성 대조군을 선택하기 위한 반자동적인 절차를 개발하였다. (Voss et al., 2016) 간략하게 설명하면, 문헌, 제품 라벨 및 자발적 보고에 대한 정보를 자동으로 추출하고 합성하여 음성 대조군 후보 목록으로 생성한다. 이 목록은 자동화된 추출이 정확한지 검증하고, 또한 생물학적 타당성과 같은 추가 기준도 고려해야 하므로 사람의 직접 검토를 거쳐야 한다.

18.2.2 양성 대조군

실제 상대 위험이 1보다 작거나 클 때, 연구자의 연구 방법론이 어떻게 행동하는지 이해하려면 귀무가설이 참이 아닌 것으로 간주하는 양성 대조군을 사용해야 한다. 불행하게도, 관측 연구에 대한 실제 양성 대조군은 세 가지 이유로 문제가 되는 경향이 있다. 첫 번째, 대부분의 연구 상황에서, 예를 들어 두 치료의 효과를 비교할 때, 특정 상황과 관련된 양성 대조군이 부족하다는 점이다. 두

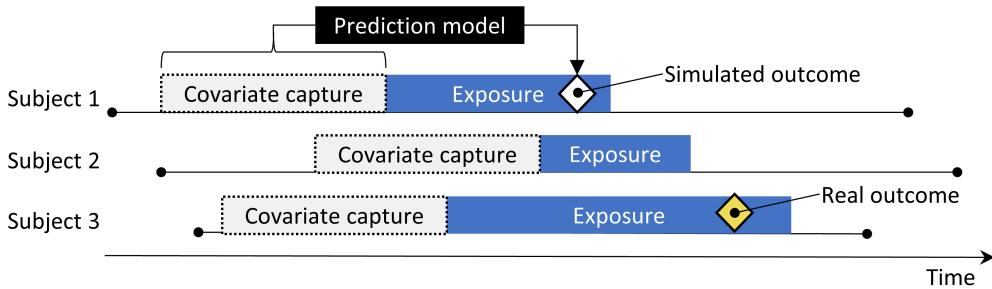


Figure 18.2: 음성 대조군을 이용해서 양성 대조군 합성하기.

번째, 양성 대조군을 찾을 수 있더라도 효과 크기의 규모를 아주 정확히 알 수는 없으며, 때때로 효과 크기를 측정하는 모집단에 의존해야 한다는 점이다. 세 번째, 치료가 특정 결과를 유발하는 것으로 널리 알려진 경우, 예를 들면 원치 않는 결과의 위험을 완화하기 위한 조치를 하는 등, 그것을 쳐방하는 의사의 행동에 변화가 생긴다는 점 때문에 양성 대조군을 평가 수단으로 사용할 수 없게 만든다. (Noren et al., 2014)

그래서 오딧세이에서는 합성 양성 대조군을 이용하는데, (Schuemie et al., 2018a) 음성 대조군을 수정하여 위험 노출 기간 동안 결과가 발생한 것처럼 모의 결과를 주입하여 만든다. 예를 들어, ACEi에 노출되는 동안, 음성 대조군의 결과 “내성 손톱”이 n번 발견되었다고 가정한다. 노출 중에 n개의 모의 발생을 추가하면 위험이 두 배가 된다. 이것이 음성 대조군이었기 때문에, 반 사실적으로 비교했을 때의 상대 위험은 1이지만, 모의 결과를 주입한 후에는 2가 된다.

중요한 문제 중 하나는 교란을 유지하는 것이다. 음성 대조군은 강력한 교란을 가질 수 있는데, 추가 결과를 무작위로 주입하게 된다면 이러한 새로운 결과는 교란을 가지고 있지 않으므로, 우리가 양성 대조군에서 교란을 대하는 능력에 긍정적 평가를 할 수 있게 된다. 교란을 보존하기 위해, 환자 특성에 기인한 기저 공변량이 기존의 결과 outcome에 보이는 것과 마찬가지의 유사한 연관성이 새로운 결과에도 나타나기를 원한다. 이를 달성하기 위해 각 결과에 대해 위험 노출 전에 발생한 기저 공변량을 사용하여 노출 중 결과에 대한 생존율을 예측하는 모델을 학습시킨다. 공변량은 인구 통계학적 뿐만 아니라, 기록된 모든 진단, 약물 노출, 측정 및 의료 처치가 포함된다. 정규화된 하이퍼-파라미터를 선택하기 위해 10회 교차 검증된 L1-regularized Poisson regression (Suchard et al., 2013) 예측 모형을 적합시킨다. 그런 다음, 예측된 비율을 시뮬레이션한 결과에 적용하여, 실제 효과 크기를 원하는 규모로 늘린다. 결과적으로 양성 대조군은 실제 결과와 모의한 결과 모두를 포함한다.

그림 18.2가 과정을 보여준다. 이 절차는 몇 가지 중요한 비뚤림의 요인을 모의하지만, 모든 것을 다 포착하진 않는다. 예를 들어, 일부 측정 오류는 포함되지 않는다. 합성 양성 대조군은 일정한 양성 예측도와 민감도를 암시하지만, 실제로는 그렇지 않을 수 있다.

각각의 대조군에 대해 하나의 실제 “효과 크기”를 언급하지만, 상이한 방법은 상이한 치료 효과 통계량을 추정한다. 인과 관계가 존재하지 않는다고 판단되는 음성 대조군의 경우, 상대 위험 relative risk, 위험 비 hazard ratio, 교차비 odds ratio, 발생률 비 incidence rate ratio, 조건부 conditional, 한계 marginal뿐만 아니라 average treatment effect in the treated(ATT)와 overall average treatment effect(ATE)까지도 포함한 모든 통계량은 1로 동일하다. 양성 대조군을 생성하는 과정은 한계 효과가 달성되는 지점까지 이 비율이 일정하게 유지되는 환자에 대해 조건부 모형을 사용하여 시간에 따라, 그리고 환자 간에 일정한 발생률 비율로 결과를 합성한다. 따라서 실제 효과 크기는 치료에서 한계 발생률 비율로 유지된다. 합성 중에 사용된 결과 모델이 정확하다는 가정하에 조건부 효과 크기와 ATE도 동일하다. 모든 결과가 드물기 때문에 교차비 odds ratios는 상대 위험 relative risk과 거의 동일하다.

18.2.3 경험적 평가

사용하고자 하는 방법론에 대하여 음성 및 양성 대조군의 추정치를 기반으로 다음과 같은 다양한 측정 기준을 계산하여 작동 특성을 이해할 수 있다:

- **Area Under the receiver operator Curve(AUC):** 양성 대조군과 음성 대조군 구별력.
- **Coverage:** 실제 효과 크기가 95% 신뢰 구간 내에 포함된 정도.
- **Mean precision:** 정밀도는 $1/(standard\ error)^2$ 로 계산되며 정밀도가 높을수록 신뢰구간이 좁아진다. 기하 평균을 사용하여 정밀도의 치우친 분포를 설명한다.
- **Mean squared error(MSE):** 효과 크기의 점추정치 로그와 실제 효과 크기 로그 사이의 평균 제곱 오차.
- **Type 1 error:** 음성 대조군의 경우, 귀무가설이 기각된 정도 (at $\alpha = 0.05$). 이는 위 양성도 및 $1 - specificity$ 에 해당한다.
- **Type 2 error:** 양성 대조군의 경우, 귀무가설이 기각되지 않는 정도 (at $\alpha = 0.05$). 이는 위 음성도 및 $1 - sensitivity$ 에 해당한다.
- **Non-estimable:** 대조군 방법에서 추정치를 산출할 수 없었던 대조군은 몇 개인가? 추정할 수 없는 이유는 여러 가지가 있을 수 있는데, 예를 들어 성향 점수 매칭 후 남은 환자나 결과를 가진 환자가 없는 경우이다.

활용 사례에 따라 이러한 작동 특성이 우리의 목표에 적합한지 평가할 수 있다. 예를 들어, 징후 탐지를 수행하고자 하는 경우, type 1과 type 2 오류를 고려하거나 α 의 임계 값을 수정하려는 경우 AUC를 대신 검사할 수 있다.

18.2.4 P-Value 보정

때로는 type 1 error (at $\alpha = 0.05$) 가 5%보다 크다. 다시 말해서, 실제로 귀무가설이 참임에도 불구하고 귀무가설을 기각할 가능성이 5% 이상인 경우가 많다. 그 이유는 p-value는 임의 오차만 반영하기 때문이다. 즉, 제한된 표본 크기로 인한 오차가 발생한다. 예를 들어, 교란으로 인해 발생한 체계적인 오차의 경우 반영되지 않는다. OHDSI는 p-value를 보정하여 type 1 error를 명목값 (nominal)으로 복원하는 과정을 개발하였다. (Schuemie et al., 2014) 우리는 음성 대조군에 대한 실제 효과 추정치로부터 경험적 귀무분포를 유도한다. 음성 대조군의 추정치는 귀무가설이 참일 때 기대할 수 있는 것에 대한 지표를 제공하며 경험적 귀무 분포를 추정하는 데 사용한다.

공식적으로, 우리는 각 추정치의 표집 오차를 고려하여 추정치에 가우스 확률 분포를 적합시킨다. $\hat{\theta}_i$ 는 i 번째 음성 대조군 약물-결과 쌍으로 추정된 로그 효과 추정치 (상대 위험, 오즈 혹은 발생률 비)를 나타내고, $\hat{\tau}_i$ 는 이에 해당하는 추정된 표준 오차 $i = 1, \dots, n$ 를 나타낸다. θ_i 는 실제 로그 효과 크기를 나타내고 (음성 대조군의 경우 0으로 가정), β_i 가 실제 (하지만 알 수 없는) 짹지어진 i 와 연관된 비뚤림을 나타낸다. 즉, 실제 효과 크기의 로그와 연구에서 i 로 반환될 것으로 추정하는 값의 차이는 무한으로 커진다. 일반적인 p-value 계산처럼, 우리는 $\hat{\theta}_i$ 가 평균이 $\theta_i + \beta_i$ 이고, 표준 편차가 $\hat{\tau}_i^2$ 인 정규 분포를 따른다고 가정한다. 전통적인 p-value 계산에서, β_i 는 항상 0과 같다고 가정하지만, β_i 는 평균이 μ 이고 분산이 σ^2 인 정규분포에서 구해진다. 이는 귀무 (비뚤림) 분포를 나타낸다. 우리는 최대 우도를 통해 μ 와 σ^2 를 추정한다. 요약해서 우리는 다음과 같이 가정한다:

$$\beta_i \sim N(\mu, \sigma^2) \text{ and } \hat{\theta}_i \sim N(\theta_i + \beta_i, \hat{\tau}_i^2)$$

$N(a, b)$ 는 평균 a 와 분산 b 를 갖는 가우시안 분포를 나타내고, 다음과 같이 우도를 최대화하여 μ 와 σ^2 을 추정하여:

$$L(\mu, \sigma | \theta, \tau) \propto \prod_{i=1}^n \int p(\hat{\theta}_i | \beta_i, \theta_i, \hat{\tau}_i) p(\beta_i | \mu, \sigma) d\beta_i$$

최대우도 추정량 $\hat{\mu}$ 과 $\hat{\sigma}$ 를 얻는다. 우리는 경험적 귀무 분포를 이용하여 보정된 p-value를 계산한다. 여기서, $\hat{\theta}_{n+1}$ 은 새로운 약물-결과 쌍으로부터 추정된 효과의 로그를 나타내고, $\hat{\tau}_{n+1}$ 은 이에 상응하는 표준 오차이다. 앞서 언급한 가정으로부터, 동일한 귀무 분포로부터 β_{n+1} 를 얻는다고 가정할 경우 다음과 같다:

$$\hat{\theta}_{n+1} \sim N(\hat{\mu}, \hat{\sigma} + \hat{\tau}_{n+1})$$

$\hat{\theta}_{n+1} < \hat{\mu}$ 보다 작은 경우, 새로운 쌍에 대해 보정된 단측 p-value는:

$$\phi \left(\frac{\theta_{n+1} - \hat{\mu}}{\sqrt{\hat{\sigma}^2 + \hat{\tau}_{n+1}^2}} \right)$$

여기서 $\phi(\cdot)$ 은 표준 정규 분포의 누적 분포 함수를 나타낸다. $\hat{\theta}_{n+1} < \hat{\mu}$ 보다 클 때, 보정된 단측 p-value는:

$$1 - \phi \left(\frac{\theta_{n+1} - \hat{\mu}}{\sqrt{\hat{\sigma}^2 + \hat{\tau}_{n+1}^2}} \right)$$

18.2.5 신뢰 구간 보정

마찬가지로, 우리는 일반적으로 95% 신뢰 구간의 적용 범위가 95% 미만임을 관찰한다: 95% 이내의 시간 동안 실제 효과 크기가 95% 신뢰구간 내에 존재. 신뢰 구간의 보정 (Schuemie et al., 2018a)을 위해 우리는 양성 대조군을 사용하여 p-value 보정을 위한 프레임 워크를 확장한다. 일반적으로 보정된 신뢰 구간은 명목 신뢰 구간보다 넓지만, 표준적인 절차에서는 설명되지 않은 문제 (측정되지 않은 교란, 선택비뚤림, 그리고 측정 오차와 같은)가 반영된다.

공식적으로, 우리는 쌍 i 와 관련된 비뚤림인 beta_i 가 다시 가우시안 분포에서 나온다고 가정하지만, 이번에는 실제 효과 크기인 theta_i 와 선형으로 관련된 평균 및 표준편차를 사용한다:

$$\beta_i \sim N(\mu(\theta_i), \sigma^2(\theta_i))$$

일때,

$$\mu(\theta_i) = a + b \times \theta_i \text{ and } \sigma(\theta_i)^2 = c + d \times |\theta_i|$$

우리는 관찰되지 않은 β_i 를 통합하는 주변 우도를 최대화하여 a, b, c, d 를 추정한다:

$$l(a, b, c, d | \theta, \hat{\theta}, \hat{\tau}) \propto \prod_{i=1}^n \int p(\hat{\theta}_i | \beta_i, \theta_i, \hat{\tau}_i) p(\beta_i | a, b, c, d, \theta_i) d\beta_i,$$

최대우도 추정량인 $(\hat{a}, \hat{b}, \hat{c}, \hat{d})$ 를 얻는다.

우리는 체계적 오차 모형을 사용하는 보정된 신뢰구간을 계산한다. 여기서 $\hat{\theta}_{n+1}$ 는 다시 추정 효과의 로그를 다시 표시하고, $\hat{\tau}_{n+1}$ 는 해당하는 추정의 표준 오차를 나타낸다. 위 가정에서 β_{n+1} 이 동일한 체계 오차 모형에서 발생한다고 가정하면 다음과 같다:

$$\hat{\theta}_{n+1} \sim N(\theta_{n+1} + \hat{a} + \hat{b} \times \theta_{n+1}, \hat{c} + \hat{d} \times |\theta_{n+1}|) + \hat{\tau}_{n+1}^2).$$

θ_{n+1} 에 대한 방정식의 해로 보정된 95% 신뢰구간의 하한을 찾는다:

$$\Phi\left(\frac{\theta_{n+1} + \hat{a} + \hat{b} \times \theta_{n+1} - \hat{\theta}_{n+1}}{\sqrt{(\hat{c} + \hat{d} \times |\theta_{n+1}|) + \hat{\tau}_{n+1}^2}}\right) = 0.025,$$

여기서 $\Phi(\cdot)$ 는 표준 정규 분포의 누적 분포 함수를 나타낸다. 상한도 유사하게 확률 0.975에서 찾는다. 확률 0.5를 이용하여 보정된 점 추정치를 정의한다.

EmpiricalCalibration 패키지에서 p-값 보정과 신뢰 구간 보정이 모두 구현된다.

18.2.6 기간 관 분석 반복 Replication Across Sites

방법 검증의 또 다른 형태는 다른 인구와 다른 의료 시스템 혹은 데이터 수집 과정이 다른 데이터 베이스에서 연구를 반복 실행하는 것이다. 기존 연구에서 서로 다른 데이터베이스에서 동일한 연구 설계를 실행하면 효과 크기 추정치가 크게 다른 것으로 나타났으며, (Madigan et al., 2013b) 이는 모집단마다 효과 크기가 다르거나 설계가 다른 데이터베이스에서 발견된 다양한 비뚤림을 적절하게 해결하지 못하고 있음을 시사한다. 실제로, 우리는 신뢰구간의 경험적 보정을 통한 데이터베이스의 잔차 비뚤림을 고려하면 연구 간의 이질성이 크게 줄어듦을 보고 있다. (Schuemie et al., 2018a)

데이터베이스간의 이질성을 표현하는 하나의 방법은 I^2 점수이며, 우연히 발생한 것이 아닌 이질성으로 인해 전체 연구에서 전체 분산의 백분율을 나타낸다. (Higgins et al., 2003) I^2 값에 대해 단순한 분류는 모든 상황에서 적합하지 않긴 하지만, 25%, 50%, 그리고 75%에서 I^2 점수에 대해 낮음, 중간, 높음의 표현을 임시로 할당할 수 있다. 대규모 성향 점수 보정을 사용한 새로운 사용자 코호트 new-user cohort 설계를 사용하여 많은 우울증 치료 효과를 추정하는 연구에서, (Schuemie et al., 2018b) 추정치의 58%만이 I^2 의 25% 미만인 것으로 관찰되었다. 이는 경험적 보정 후에 83%로 증가했다.



데이터베이스 간의 이질성을 관찰하면 추정치의 유효성에 의문이 생긴다. 불행하게도, 그 반대는 사실이 아니다. 이질성을 관찰하지 않는 것이 비뚤림없는 추정을 보장하지 않는다. 모든

데이터베이스가 유사한 비뚤림을 공유하거나 추정치가 일관되게 잘못되었을 가능성은 거의 없다.

18.2.7 민감도 분석

연구를 설계할 때, 때때로 설계 선택이 불확실하다. 예를 들어, 총화 성향 점수 짹짓기를 사용해야 하는지? 총화를 한다면, 어느 정도 총화를 해야 하는가? 적절한 위험 노출 기간은 무엇인가? 이러한 불확실성에 직면할 때, 하나의 해결법은 다양한 옵션을 평가하고, 선택한 디자인에 대한 결과의 민감도를 관찰하는 것이다. 다양한 옵션에서 추정치가 일관되고 유지되면 불확실성에 대해 연구가 굳건하다고 말할 수 있다.

민감도 분석에 대한 이러한 정의는 “연구의 결론이 다양한 규모의 숨겨진 비뚤림에 의해 어떻게 변경될 수 있는지 평가하는 것”이라고 민감도 분석을 정의한 Rosenbaum (2005) 과 같은 다른 연구자가 사용하는 정의와 혼동되어서는 안 된다.

18.3 실무에서의 연구 방법론 검증

여기서, 우리는 thiazides and thiazide-like diuretics(THZ)보다 ACE 억제제(ACEi)와 비교하여 혈관 부종 및 급성 심근 경색의 위험에 대한 ACEi의 효과를 조사하는 12장의 예를 바탕으로 한다. 이 장에서는 우리가 이미 사용했던 설계, 즉 CohortMethod에 관한 많은 진단을 살펴보았다. 여기에는 다른 디자인을 사용한 경우에도 적용할 수 있는 추가 진단법이 적용된다. 12.7절에 설명한 대로, ATLAS를 사용하여 연구를 구현하는 경우 ATLAS에서 생성한 연구의 R 패키지에 포함된 Shiny 앱에서 이러한 진단을 사용할 수 있다. 연구를 12.8절에 설명한 것처럼 R을 이용하여 구현한다면, 다음 절에서 설명한 대로 다양한 패키지에서 사용할 수 있는 R 함수를 사용해야 한다.

18.3.1 음성 대조군 선택하기

연구자는 인과 관계에 영향이 없는 것으로 생각되는 음성 대조군, 노출-결과 쌍을 선택해야 한다. 예제 연구와 같은 비교 효과 추정을 위해, 목적이나 비교 노출로 인해 야기되지 않는다고 여겨지는 음성 대조군을 선택한다. 대조군에서 다양하게 혼합된 비뚤림을 확보하고 경험적 보정을 가능하도록 표현된 충분한 음성 대조군을 원한다. 실용적으로 50-100개의 음성 대조군을 목표로 한다. 이러한 대조군을 완전히 직접 만들어 쓸 수도 있지만, 다행히 ATLAS는 문헌, 제품 라벨 및 자발적 보고서의 데이터를 사용하여 음성 대조군을 선택할 수 있는 기능을 제공한다.

음성 대조군 후보 목록을 생성하려면, 먼저 관심 있는 모든 노출을 포함하는 개념 모음을 생성해야 한다. 이 경우 그림 18.3과 같이 ACEi 및 THZ 클래스의 모든 성분을 선택한다.

Concept Set Expression		Included Concepts 14	Included Source Codes	Explore Evidence	Export	Compare	
Show 25 entries				Search: <input type="text"/>			
Showing 1 to 14 of 14 entries				Previous 1	Next		
Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	Exclude	Descendants	Mapped
1342439	38454	trandolapril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1334456	35296	Ramipril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1331235	35208	quinapril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1373225	54552	Perindopril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1310756	30131	moexipril	Drug	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 18.3: 대상군 및 비교군의 노출을 정의하는 개념을 포함하는 개념 모음.

다음으로, ‘Explore Evidence’ 탭으로 가서 **▶ Generate** 버튼을 클릭한다. 근거 개요를 작성하는 데는 몇 분이 걸리고, 그 후 **View Evidence** 버튼을 클릭할 수 있다. 그러면 그림 18.4에 표시된 결과와 같은 목록이 열린다.

이 목록은 진단 개념과 우리가 정의한 노출 중 하나를 연결하는 근거의 개요를 보여준다. 예를 들어, 다양한 전략을 사용하여 PubMed에서 찾은 결과와 노출을 연결하는 출판물 수, 우리의 관심거리가 되는 노출의 제품 라벨에서 발생 가능한 부작용 진단 리스트의 수, 그리고 자발적인 보고서 수를 볼 수 있다. 기본적으로 이 목록은 음성 대조군의 후보를 먼저 표시하도록 정렬된다. 그런 다음 관측 형 데이터베이스에서 수집된 진단의 유병률을 나타내는 “Sort Order”로 정렬된다. Sort Order가 높을수록 유병률이 높다. 이 데이터베이스의 유병률은 우리가 연구를 수행하려는 데이터베이스의 유병률과 일치하지 않을 수 있지만, 좋은 근사치일 수 있다.

다음 단계는 일반적으로 후보의 유병률이 가장 높은 진단부터 시작하여, 충분히 낍득할 수 있을 때까지 대조군의 목록을 직접 검토하는 것이다. 이 작업을 수행하는 일반적인 방법 중 하나는 목록을 CSV (쉼표로 구분된 값) 파일로 내보내어 임상의가 18.2.1절에서 언급된 기준을 고려하여 이를 검토하는 것이다.

예제 연구에서는 부록 C.1에 나열된 76개의 음성 대조군을 선택한다.

18.3.2 대조군 포함하기

음성 대조군의 집합을 정의했다면, 그것을 연구에 포함해야 한다. 우리는 먼저 음성 대조군의 개념을 결과 코호트로 바꾸기 위한 몇 가지 논리를 정의해야 한다. 12.7.3절은 ATLAS가 사용자가 선택해야 하는 몇 가지 선택을 기반으로 이러한 코호트를 생성하는 방법을 설명한다. 우리는 종종

Evidence for all conditions for ACEi and THZ combined

Save New Concept Set From Selection Below View database record counts (RC) and descendant record counts (DRC) for: SYNPUF 5% ▾

Column visibility Copy CSV Show 15 entries Filter: []

Showing 1 to 15 of 13,787 entries

	Name	Suggested Negative Control	Sort Order	Publication Count (Descendant Concept Match)	Publication Count (Exact Concept Match)	Publication Count (Parent Concept Match)	Product Label Count (Descendant Concept Match)	Product Label (Exact Concept Match)	Product Label (Parent Concept Match)
▼ Suggested Negative Control	Rift valley fever	Y	13,781	0	0	0	0	0	0
No (12777)	Obstruction due to foreign body accidentally left in operative wound	Y	13,780	0	0	0	0	0	0
Yes (1010)	AND/OR body cavity during a procedure								
▼ Found in Publications	Infection by Shigella	Y	13,766	0	0	0	0	0	0
No (12398)									
Yes (Parent) (1160)									
Yes (Exact) (229)									
▼ Found on Product Label									
No (12667)									
Yes (Parent) (878)									
Yes (Exact) (242)									
▼ Found in Product Label Or Publications									
Yes (10576)									
No (3211)									
▼ Signal in FAERS									
No (10951)									
Yes (Parent) (1949)									

Figure 18.4: 문헌, 제품 라벨, 자발적 보고서의 개요에서 발견된 증거를 가진 후보 대조군 결과.

음성 대조군의 혹은 그 자손의 발생에 기초하여 코호트를 만들기로 선택한다. 본 연구가 R에서 구현된다면, Structured Query Language(SQL)를 사용하여 음성 대조군 코호트를 구성할 수 있다. 9장에서는 SQL과 R을 사용하여 코호트를 만드는 방법을 설명한다. 독자가 적절한 SQL과 R을 작성하는 연습을 남겨둔다.

OHDSI 도구는 음성 대조군에서 파생된 양성 대조군을 자동으로 생성하고 포함하는 기능 역시 제공한다. 이 기능은 12.7.3절에서 설명된 ATLAS Evaluation Settings에서 찾을 수 있으며, MethodEvaluation 패키지의 `syntheticPositiveControls`함수에서 구현된다. 여기서 생존 모델을 사용하여 실제 효과 크기가 1.5, 2, 그리고 4인 각 음성 대조군에 대해 세 가지 양성 대조군을 생성한다:

```
library(MethodEvaluation)

# Create a data frame with all negative control exposure-
# outcome pairs, using only the target exposure (ACEi = 1).
eoPairs <- data.frame(exposureId = 1,
                      outcomeId = ncs)

pcs <- synthesizePositiveControls(
  connectionDetails = connectionDetails,
  cdmDatabaseSchema = cdmDbSchema,
```

```

exposureDatabaseSchema = cohortDbSchema,
exposureTable = cohortTable,
outcomeDatabaseSchema = cohortDbSchema,
outcomeTable = cohortTable,
outputDatabaseSchema = cohortDbSchema,
outputTable = cohortTable,
createOutputTable = FALSE,
modelType = "survival",
firstExposureOnly = TRUE,
firstOutcomeOnly = TRUE,
removePeopleWithPriorOutcomes = TRUE,
washoutPeriod = 365,
riskWindowStart = 1,
riskWindowEnd = 0,
endAnchor = "cohort end",
exposureOutcomePairs = eoPairs,
effectSizes = c(1.5, 2, 4),
cdmVersion = cdmVersion,
workFolder = file.path(outputFolder, "pcSynthesis"))

```

우리의 추정 연구 설계에 사용된 위험 노출 기간 time-at-risk(TAR)을 모방해야 한다. `synthesizePositiveControls` 함수는 노출과 음성 대조군 결과에 대한 정보를 추출하고, 쌍마다 모델링 결과를 적합하고, 결과를 합성한다. 양성 대조군의 결과 코호트는 `cohortDbSchema` 와 `cohortTable`로 지정된 코호트 테이블에 추가된다. 결과인 pcs data frame은 합성된 양성 대조군에 대한 정보가 포함된다.

다음으로, 우리는 음성 및 양성 대조군에 대한 영향을 추정하기 위해 관심 효과를 추정하는 데 사용된 동일한 연구를 실행해야 한다. ATLAS의 비교 대화상자에서 양성 대조군 세트를 설정하면, ATLAS가 이러한 대조군에 대한 추정치를 계산하도록 한다. 마찬가지로 Evaluation Settings에서 양성 대조군을 생성하도록 지정하면, 분석에 이러한 대조군이 포함된다. R에서, 음성 및 양성 대조군은 서로 다른 결과로 취급되어야 한다. OHDSI Methods Library의 모든 추정 패키지는 효율적인 방식으로 많은 효과를 쉽게 추정할 수 있다.

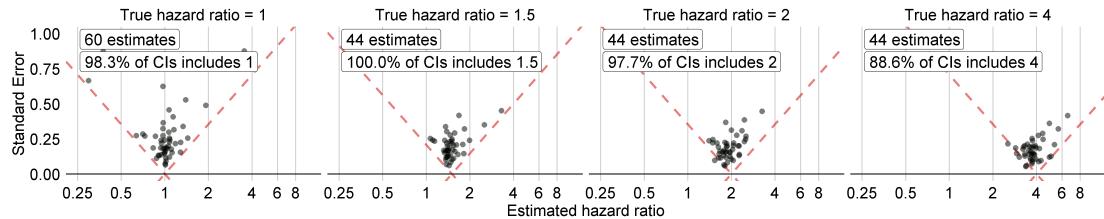


Figure 18.5: 음성 (실제 위험 비율 = 1) 및 양성 대조 (실제 위험 비율 > 1)의 추정치. 각각의 점은 대조군을 뜻한다. 점선 아래의 추정치는 실제 효과 크기를 포함하지 않는 신뢰 구간을 가진다..

18.3.3 경험적 성능

그림 18.5는 예제 효과 연구에 포함된 음성 및 양성 대조군에 대한 추정 효과 크기를 실제 효과 크기로 계층화 한 것이다. 이 도표는 ATLAS에서 생성한 R 패키지와 함께 제공되는 Shiny 앱에 포함되어 있으며, MethodEvaluation 패키지의 `plotControls` 함수를 사용하여 생성할 수 있다. 추정치를 생성하거나 양성 대조군을 합성하기에 충분한 데이터가 없기 때문에 대조군의 수는 때때로 정의된 것보다 적을 수 있다.

이러한 추정치를 기반으로 MethodEvaluation 패키지의 `computeMetrics` 함수를 사용하여 표 18.1에 표시된 측정 기준을 계산할 수 있다.

Table 18.1: Method performance metrics derived from the negative and positive control estimates.

Metric	Value
AUC	0.96
Coverage	0.97
Mean Precision	19.33
MSE	2.08
Type 1 error	0.00
Type 2 error	0.18
Non-estimable	0.08

적용 범위와 type 1 error는 각각 명목 값인 95%와 5%에 매우 가깝고, AUC는 매우 높다. 항상 그런 것은 아니다.

그림 18.5에서 실제 위험 비율이 1일 때 모든 신뢰 구간에 1이 포함되는 것은 아니지만 표 18.1의 type 1 error는 0%이다. Cyclops 패키지의 신뢰 구간은 우도 profiling을 사용하여 추정되므로 기존

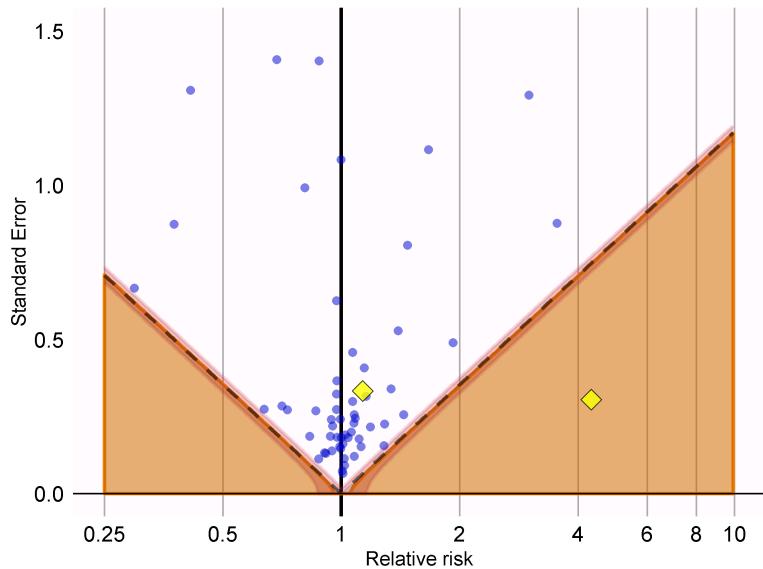


Figure 18.6: P-value 눈금: 점선 아래의 추정치는 conventional $p < 0.05$ 를 의미한다. 음영 처리된 영역의 추정치는 calibrated $p < 0.05$ 를 의미한다. 음영 처리된 영역의 가장자리 둘레에 있는 좁은 밴드는 95% credible interval을 의미한다. 점들은 음성 대조를 가리킨다. 다이아몬드 모양은 관심을 둔 연구 결과를 의미한다.

방법보다 정확하지만, 비대칭적인 신뢰 구간이 발생할 수 있기 때문에 예외적인 상황이다. 대신 p-value는 대칭 신뢰 구간을 가정하여 계산되며 이는 type 1 error를 계산하는 데 사용된 것이다.

18.3.4 P-Value 보정

p-value를 보정하기 위해 음성 대조군에 대한 추정값을 사용할 수 있다. 이는 Shniy 앱에서 자동으로 수행되며, R에서 수동으로도 수행할 수 있다. 12.8.6절에 설명된 대로 요약된 객체 summ을 생성했다고 가정하면 경험적 보정 효과 도표를 그릴 수 있다:

```
# Estimates for negative controls (ncs) and outcomes of interest (ois):
ncEstimates <- summ[summ$outcomeId %in% ncs, ]
oiEstimates <- summ[summ$outcomeId %in% ois, ]

library(EmpiricalCalibration)
plotCalibrationEffect(logRrNegatives = ncEstimates$logRr,
                      seLogRrNegatives = ncEstimates$seLogRr,
                      logRrPositives = oiEstimates$logRr,
                      seLogRrPositives = oiEstimates$seLogRr,
                      showCis = TRUE)
```

그림 18.6에서 우리는 음영 영역이 점선으로 표시된 영역과 거의 정확하게 겹치는 것을 볼 수 있는데, 이는 음성 대조군에 대해 어떠한 비뚤림도 관찰되지 않았음을 나타낸다. 관심 있는 결과 중 하나 (급성 심근경색)는 점선과 음영 위에 있으며, 보정되거나 되지 않은 두 p-value에 따라 귀무 가설을 기각할 수 없음을 나타낸다. 다른 결과 (혈관 부종)는 음성 대조군에서 분명히 두드러지며 보정되거나 되지 않은 두 p-value 모두 0.05보다 작은 영역에 잘 포함된다.

보정된 p-value를 계산할 수 있다:

```
null <- fitNull(logRr = ncEstimates$logRr,
                 seLogRr = ncEstimates$seLogRr)
calibrateP(null,
            logRr= oiEstimates$logRr,
            seLogRr = oiEstimates$seLogRr)
```

```
## [1] 1.604351e-06 7.159506e-01
```

그리고 이것을 보정되지 않은 p-value와 비교하면:

```
oiEstimates$p
```

```
## [1] [1] 1.483652e-06 7.052822e-01
```

예상한 대로, 비뚤림이 거의 없거나 전혀 관찰되지 않았기 때문에 보정되지 않은 p-value는 보정된 값과 매우 유사하다.

18.3.5 신뢰 구간 보정

마찬가지로, 음성 및 양성 대조군에 대한 추정값을 사용하여 신뢰 구간을 보정할 수 있다. Shiny 앱은 보정 신뢰 구간을 자동으로 보고한다. R에서는 “Empirical calibration of confidence intervals” vignette에 자세히 설명된 대로, EmpiricalCalibration 패키지의 `fitSystematicErrorModel` 및 `calibrateConfidenceInterval` 함수를 사용하여 구간을 보정할 수 있다.

보정하기 전에, 혈관 부종 및 급성 심근 경색의 추정 위험 비 (95% 신뢰구간)는 4.32 (2.45-8.08) 및 1.13 (0.59-2.18)이다. 보정된 위험 비는 4.75 (2.52-9.04) 및 1.15 (0.58-2.30)이다.

18.3.6 데이터베이스 간 이질성

한 데이터베이스에서 분석을 수행한 것처럼, IBM MarketScan Medicaid (MDCD) 데이터베이스에서, 우리는 다른 공통 데이터 모델(CDM) 데이터베이스에서도 동일한 분석 코드를 실행할 수 있다.

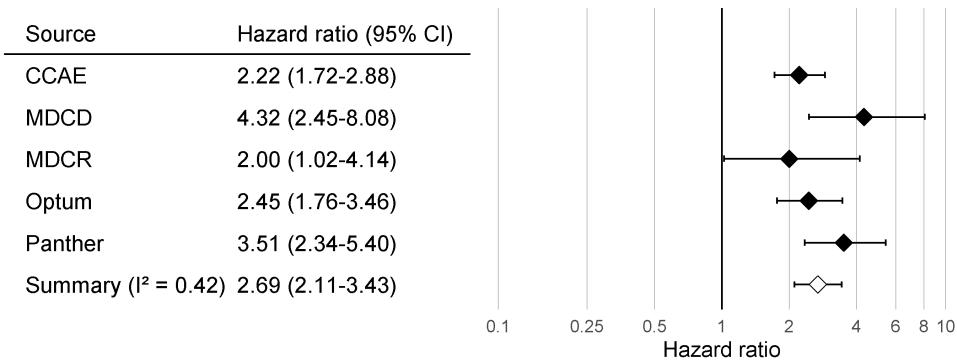


Figure 18.7: 혈관부종의 위험을 위해 ACE 억제제를 thiazide 및 thiazides 유사 이뇨제와 비교할 때 5개 데이터베이스의 효과 크기 추정 및 95% 신뢰 구간(CI).

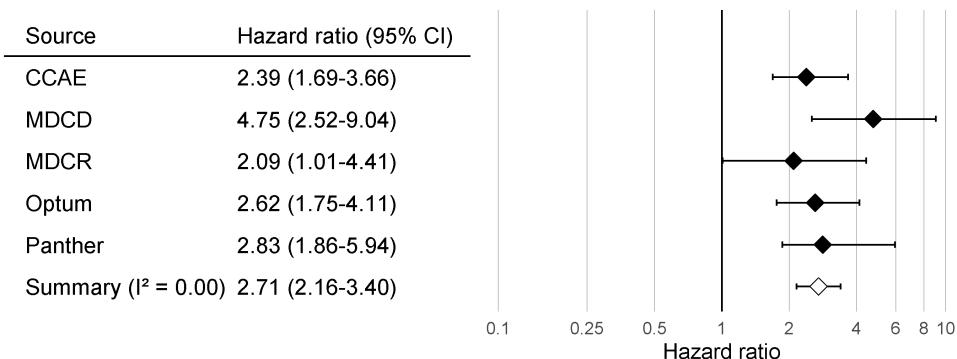


Figure 18.8: ACE 억제제를 thiazide 및 thiazide 유사 이뇨제와 비교할 때 5가지 데이터베이스에서 보정된 효과 크기 추정치 및 95% 신뢰 구간 및 혈관부종의 위험 비율에 대한 메타 분석 추정치.

그림 18.7은 혈관 부종의 결과에 대한 총 5개의 데이터베이스에 대한 forest 도표와 메타 분석 추정치 (임의 효과를 가정) (DerSimonian and Laird, 1986)를 보여준다. 이 그림은 EvidenceSynthesis 패키지의 `plotMetaAnalysisForest` 함수를 사용하여 생성되었다.

모든 신뢰 구간이 1 이상이고 효과가 있다는 사실에 일치하지만 I^2 는 데이터베이스 간 이질성이 있음을 표시한다. 하지만 그림 18.8과 같이 보정된 신뢰구간을 사용하여 I^2 를 계산하면 이질성이 음성 및 양성 대조군을 이용하여 각 데이터베이스에서 측정한 비뚤림에 의해 설명될 수 있음을 알 수 있다. 경험적 보정은 이 비뚤림을 올바르게 고려한 것으로 보인다.

18.3.7 민감도 분석

분석에서 선택된 설계 중 하나는 성향 점수에 일-대-다 비율 짹짓기 variable-ratio를 사용하는 것이다. 그러나 우리는 성향 점수에 계층화를 사용할 수도 있다. 우리는 이 선택에 대해 불확실하기 때문에, 둘 다 사용하기로 할 수 있다. 표 18.2는 일-대-다 비율 짹짓기 및 계층화 (10개의 동일한 크기의 충화)를 사용할 때 보정 및 보정되지 않은 급성 심근 경색 및 혈관 부종의 효과 크기 추정치를

보여준다.

Table 18.2: Uncalibrated and calibrated hazard ratios (95% confidence interval) for the two analysis variants.

Outcome	Adjustment	Uncalibrated	Calibrated
Angioedema	Matching	4.32 (2.45 - 8.08)	4.75 (2.52 - 9.04)
Angioedema	Stratification	4.57 (3.00 - 7.19)	4.52 (2.85 - 7.19)
Acute myocardial infarction	Matching	1.13 (0.59 - 2.18)	1.15 (0.58 - 2.30)
Acute myocardial infarction	Stratification	1.43 (1.02 - 2.06)	1.45 (1.03 - 2.06)

짝짓기와 층화 분석으로부터 얻은 추정치가 강하게 일치하는 것과 함께, 계층화에 대한 신뢰구간이 짝짓기 된 신뢰 구간 내에 완전히 포함되는 것을 볼 수 있다. 이는 연구 설계 선택에 대한 우리의 불확실성이 추정치의 타당성에 영향을 미치지 않음을 시사한다. 계층화는 우리에게 더 큰 검정력 (더 좁은 신뢰 구간)을 제공하는 것으로 보이는데, 왜냐하면 짝짓기는 많은 데이터의 손실을 초래하지만, 계층화는 그렇지 않기 때문에 이것은 놀라운 일은 아니다. 층화분석을 사용한 대가로 비뚤림이 증가할 수 있는데, 비록 보정된 신뢰구간에 비뚤림이 증가했다는 증거는 보이지 않지만 계층 내 남아있는 교란으로 인해서 그럴 수 있다.



연구 진단을 통해 연구를 완전히 실행하기 전에 연구에 따른 선택을 평가할 수 있다. 모든 연구 진단을 생성하고 검토하기 전에 프로토콜을 완성하지 않는 것이 좋다. P-hacking (원하는 결과를 얻도록 디자인을 조정)을 배제하기 위해서 연구목표에 해당하는 효과 크기 추정값을 보지 않은 채 수행해야 한다.

18.4 OHDSI 방법론 벤치마크

비록 권고안이 적용된 맥락 안에서 방법론적인 성능을 경험적으로 평가하는 것이라 하더라도, 연구에서 쓰인 데이터베이스와 관심 노출-결과 쌍과 비슷한 방식 (예를 들어 같은 노출 혹은 같은 결과를 사용하는 것)의 음성과 양성 대조군을 사용하여 방법론적인 성능을 일반적으로 평가하는 것도 가치 있다. 이것이 OHDSI Methods Evaluation⁶⁰ 개발된 이유이다. 성능 평가 기준은 만성 혹은 급성

결과 또는 단기 노출을 포함한 광범위한 대조군 질의를 사용하여 성능을 평가한다. 이 성능 평가 기준의 결과는 분석 방법의 전반적인 유용성을 입증하는 데 도움이 될 수 있으며 상황별로 경험적 평가를 아직 사용할 수 없는 경우 분석법의 성능에 대한 사전 확신을 형성하는데 사용될 수 있다. 사용 평가 기준은 8개의 범주로 분류할 수 있는 200개의 신중하게 선택된 음성 대조군으로 구성되며 각 범주의 대조군은 동일한 노출 또는 동일한 결과를 공유한다. 이러한 200개의 음성 대조군으로부터, 18.2.2절에 기술된 바와 같이 600개의 합성 양성 대조군이 유도된다. 분석 방법을 평가하려면 모든 대조군에 대해 효과 크기 추정치를 생성하는데 사용해야 하며, 그 후 18.2.3절에 설명된 측정기준을 계산할 수 있다. 성능 평가 기준은 공개적으로 사용이 가능하며 MethodEvaluation 패키지의 Running the OHDSI Methods Benchmark vignette 비네팅 실행에 설명된 대로 배포할 수 있다.

각 분석 방법별로 다양한 옵션을 적용하여 OHDSI 연구방법론 라이브러리의 모든 분석법에 대해 벤치마크를 실행했다. 예를 들어, CohortMethod는 성향 점수 짹짓기, 충화, 그리고 가중화를 사용하여 평가했다. 이 실험은 4개의 대규모 관찰형 의료 데이터베이스에서 실행되었다. 온라인상의 Shiny 앱¹에서 볼 수 있는 결과는 여러 방법이 높은 AUC (음성대조군에서 양성 대조군을 구분하는 능력)를 보여주지만, 그럼 18.9와 같이, 대부분의 설정에서 대부분의 방법이 높은 1종 오류와 (실제로 효과 없는데 효과가 있다고 판단) 95% 신뢰 구간에 포함되는 비율이 낮은 것을 보여준다.

이것은 경험적 평가와 보정의 필요성을 강조한다: 이미 출판된 거의 모든 관찰연구처럼 경험적 평가가 수행되지 않는다면, 우리는 그럼 18.9의 결과와 같은 사실을 미리 알고 있어야 하며, 그 연구 결과의 실제 효과 크기가 95% 신뢰 구간에 포함되지 않을 가능성이 높다고 결론지어야 한다!

연구방법론 라이브러리의 설계에 평가까지 경험적 교정은 종종 제2종 오류 (실제로 효과가 있는데 없다고 판단)를 증가시키고 정밀도를 낮추는 대가를 지불하지만, 제1종 오류와 포함 범위를 명목 값으로 복원하는 것을 보여준다.

18.5 요약



- 방법의 타당성은 연구 방법의 기본 가정이 충족되는지에 따라 다르다.
- 가능하면 이러한 가정은 실험 진단을 사용하여 경험적으로 검사를 해야 한다.
- 답이 알려진 질문 대조군 가설을 사용하여 특정 연구 설계가 실제와 일치하는 답을 생성하는지 평가해야 한다.
- 종종 p-value와 신뢰 구간은 대조군 가설을 사용하여 측정한 명목 특성을 나타내지 않는다.

¹<http://data.ohdsi.org/MethodEvalViewer/>

- 이러한 특성은 경험적 보정을 사용하여 명목값으로 복원될 수 있다.
- 연구 진단은 연구자가 관심 있는 효과에 대해 눈가림 된 채 분석을 수행해야만, 분석 설계별 옵션을 안내하고 프로토콜을 개선하는 데 사용될 수 있다.



Figure 18.9: 연구방법론 라이브러리에서 방법에 대한 95% 신뢰 구간 적용 범위. 각 점은 특정 분석 선택 세트의 성능을 나타낸다. 점선은 명목 성능을 나타낸다 (95% 적용 범위). SCCS = Self-Controlled Case Series, GI = Gastrointestinal, IBD = inflammatory bowel disease.

Part V

OHDSI 연구

Chapter 19

연구단계

Chapter leads: Sara Dempster & Martijn Schuemie

이 장에서는 OHDSI 도구를 사용한 관찰 연구의 설계 및 구현에 대한 일반적인 단계별 가이드를 제공할 것이다. 연구 과정의 각 단계를 세분화하고, 단계를 일반적으로 설명하고, 어떤 경우에는 이 책의 앞장에서 설명한 다음과 같은 주요 연구 유형을 설명할 것이다: 1) 임상적 특성 분석 characterization 2) 인구 수준 추정 population level estimation(PLE) 3) 환자 수준 예측 patient level prediction(PLP). 이를 위해, 앞장에서 설명한 많은 요소를 초보자가 접근할 수 있는 방식으로 합성할 것이다. 동시에, 이 장은 필요에 따라 다른 장에서보다 심층적인 자료를 추구할 수 있는 옵션을 통해 실제적인 높은 수준의 설명을 원하는 독자에게 적합할 수 있다. 마지막으로, 몇 가지 주요 예를 통해 설명할 것이다.

또한, OHDSI 커뮤니티에서 권장하는 관찰 연구 지침과 모범 사례를 요약할 것이다. 여기서 논의할 일부 원칙은 일반적이고 다른 많은 관찰 연구 지침의 모범 사례 권장 사항과 일치되지만, 다른 권장 프로세스는 OHDSI 프레임워크에 더 적합하다. 따라서 OHDSI 도구 스택에서 OHDSI 관련 접근 방식을 사용할 수 있는 부분을 강조할 것이다.

독자가 OHDSI 도구, R 및 SQL 인프라를 활용할 수 있다는 가정하에, 이 장 전체에서 이러한 인프라 설정에 대해 논의는 하지 않는다(설정 지침은 8장과 9장 참조). 또한 독자가 OMOP CDM 데이터베이스로 구축한 자체 사이트의 데이터를 활용하여 연구를 수행하는 데 관심이 있는 것으로 가정한다(OMOP ETL의 경우 6장 참조). 그러나 연구 패키지가 아래에 논의된 대로 준비되면, 원칙적으로 다른 사이트에서 배포 및 실행될 수 있음을 강조한다. 조직 및 기술 세부 사항을 포함하여 OHDSI 네트워크 연구 실행에 대한 추가 고려 사항은 20장에서 자세히 설명한다.

19.1 일반 모범 사례 지침

19.1.1 관찰 연구 정의

관찰 연구는 환자를 단순히 관찰하고 특정 환자 치료에 개입하려는 시도가 없는 연구로 정의된다. 때로는 관찰 데이터가 레지스트리 연구에서와 같이 특정 목적으로 수집되는 경우도 있지만, 대부분의 경우 이러한 데이터는 현재 특정 연구 질문 이외의 다른 목적으로 수집된다. 후자 데이터 유형의 일반적인 예로는 전자 의무 기록 Electronic Health Records(EHR) 또는 행정 청구 데이터 administrative claims data가 있다. 관찰 연구는 종종 데이터의 2차 사용이라고 불린다. 관찰 연구 수행을 위한 기본 지침 원칙은 연구 질문을 명시적으로 설명하고 연구 수행 전에 접근 방식을 완전히 지정하는 것이다. 이와 관련하여, 관찰 연구는 임상시험에서 일반적으로 치료 중재의 효능 및/또는 안전성에 관한 특정 질문에 답하기 위한 주된 목적으로 환자를 제시간에 모집하고 추적하는 것을 제외하고는 임상 시험과 다르지 않아야 한다. 관찰 연구에 사용된 분석 방법은 임상 시험에 사용된 분석 방법과 여러 면에서 다르다. 가장 주목할만한 점은, PLE 관찰 연구에서는 무작위 배정이 없어, 연구 목적이 인과적 추론을 도출하는 경우라면 혼란 변수를 통제하는 접근법이 필요하다 (OHDSI 가 지원하는 연구 설계 및 여러 특성에 근거해 연구 집단의 균형을 맞춤으로써 관찰된 혼란 변수를 제거하는 PLE 방법에 대한 자세한 설명은 12장과 18장을 참조).

19.1.2 연구 설계의 사전 지정

관찰 연구 설계 및 매개 변수의 사전 지정 pre-specification은 때로 p-hacking이라고 불리는 원하는 결과를 달성하기 위해 잠재 의식적으로 또는 의식적으로 차츰 발전하는 추가적인 비뚤림 bias 을 도입하지 않도록 하는 데 중요하다. 사전에 연구 세부 사항을 완전히 지정하지 않으려는 유혹은 전자 의무 기록 및 청구와 같은 데이터가 연구자에게 무한한 가능성에 대한 구불구불한 조사선 meandering line of inquiry을 초래하기 때문에 데이터의 1차 사용보다 2차 사용에서 더 크다. 핵심은 기존 데이터를 쉽게 이용할 수 있음에도 불구하고 과학적 탐구의 엄격한 구조를 여전히 적용하는 것이다. 사전 지정 원칙은 인구 수준 추정 또는 환자 수준 예측에서 엄밀하거나 재현 가능한 결과를 보장하기 위해 특히 중요한데, 이러한 결과는 궁극적으로 임상 실무 또는 규제 결정에 영향을 줄 수 있기 때문이다. 특성화 연구 characterization study가 순수하게 탐색적 목적으로 수행되는 경우에도, 잘 지정된 계획 수립은 여전히 바람직하다. 그렇지 않으면 차츰 발전하는 연구 설계 및 분석 프로세스가 문서화, 설명 및 재생산에 어려움을 준다.

19.1.3 프로토콜

관찰 연구 계획은 연구 수행 전에 만들어진 프로토콜 형태로 문서화되어야 한다. 프로토콜은 최소한 주요 연구 질문, 접근 방식 및 질문에 답하는데 사용될 측정 항목을 설명한다. 연구 집단은

다른 연구자가 연구 집단을 완전히 재생산할 수 있도록 상세 수준으로 기술되어야 한다. 또한, 모든 방법 또는 통계 절차 및 측정 항목, 표 및 그래프와 같은 예상 연구 결과의 형태를 설명해야 한다. 종종, 프로토콜은 연구의 타당성 또는 통계적 검정력 statistical power을 평가하려고 설계된 일련의 사전 분석을 설명할 것이다. 더욱이, 프로토콜에는 민감도 분석이라고 하는 주요 연구 질문에 대한 변수 variation 설명이 포함될 수 있다. 민감도 분석은 연구 설계 선택이 전체 연구 결과에 미치는 잠재적 영향을 평가하도록 설계되었으며, 가능할 때마다 미리 설명해야 한다. 프로토콜이 완료된 후, 때론 프로토콜 수정이 필요할 수 있는 예기치 않은 문제가 발생한다. 프로토콜 수정이 필요한 경우, 프로토콜 자체의 변경 및 그 이유를 문서화하는 것이 중요하다. 특히 PLE 또는 PLP의 경우, 완성된 연구 프로토콜은 독립적인 플랫폼 (예를 들어 clinicaltrials.gov 또는 OHDSI의 studyProtocols sandbox)에 이상적으로 기록되며, 버전 및 수정 사항은 타임 스탬프 timestamp와 독립적으로 추적할 수 있다. 기관이나 데이터 원천 소유자가 연구 실행 전에 프로토콜을 검토하고 승인할 기회를 요구하는 경우가 종종 있다.

19.1.4 표준화된 분석

OHDSI의 독특한 장점은 관찰 연구에서 반복적으로 묻는 몇몇 주요 질문을 인식함으로써 도구가 계획, 문서화 및 보고를 지원하는 방식이다 (2장, 7장, 11장, 12장, 13장). 따라서, 반복되는 측면의 자동화로 프로토콜 개발 및 연구 구현 프로세스를 간소화한다. 대부분의 도구는 발생할 수 있는 대부분의 사용 사례를 다루는 몇몇 연구 설계 또는 측정 항목을 매개 변수화하도록 설계되었다. 예를 들어, 연구자는 연구 집단과 몇몇 추가 매개 변수를 지정하고 다른 약물 및/또는 결과를 반복하는 수많은 비교 연구를 수행한다. 연구자의 질문이 일반 템플릿에 적합한 경우, 프로토콜에 필요한 연구 집단 및 기타 매개 변수에 대한 많은 기본 설명을 자동으로 생성하는 방법이 있다. 역사적으로, 이러한 접근 방식은 OMOP 실험을 통해 이루어졌으며, 이는 관찰 연구 설계가 다양한 연구 설계 및 매개 변수를 반복하여 약물과 부작용 사이의 알려진 인과 관계를 얼마나 잘 재현할 수 있는지 평가하고자 하였다.

OHDSI 접근 방식은 이러한 단계를 공통 프레임워크 및 도구 내에서 비교적 간단하게 수행할 수 있도록 하여 프로토콜에 실행 가능성 및 연구 진단이 포함되는 것을 지원한다 (아래 19.2.4절 참조).

19.1.5 연구 패키지

표준화된 템플릿 및 디자인에 대한 또 다른 동기는 연구자가 프로토콜 형태로 연구가 상세하게 설명되어 있다고 생각하더라도 연구를 실행하기 위해 전체 컴퓨터 코드를 생성하기에 실제로 충분히 지정되지 않은 요소가 있을 수 있다는 것이다. OHDSI 프레임워크에 의해 가능한 관련 기본 원칙은 종종 “연구 패키지”라고 하는 컴퓨터 코드 형태로 문서화된, 완벽하게 추적 및 재현 가능한 프로세스를 생성하는 것이다. OHDSI 모범 사례는 이러한 연구 패키지를 Git) 환경에 기록하는 것이다.

이러한 연구 패키지는 코드 기반에 대한 모든 매개 변수 및 버전 지정 스탬프를 포함한다. 앞에서 언급한 바와 같이, 관찰 연구는 종종 공중 보건 결정 및 정책에 영향을 미칠 수 있는 질문을 한다. 따라서, 결과를 찾기 전에 관찰 연구는 이상적으로 다른 연구자에 의해 여러 환경에서 반복되어야 한다. 이러한 목적을 달성하는 유일한 방법은 연구를 완전히 재현하는데 필요한 모든 세부 사항을 명시적으로 매핑하고 추측이나 오해의 여지가 없도록 하는 것이다. 이러한 모범 사례를 지원하기 위해 OHDSI 도구는 문서 형태의 프로토콜에서 컴퓨터 또는 기계 판독 가능 연구 패키지로의 변환을 지원하도록 설계되었다. 이 프레임워크의 한 가지 단점은 기존 OHDSI 도구만으로 모든 사용 사례나 사용자 지정 분석을 쉽게 처리할 수는 없다는 것이다. 그러나 커뮤니티가 성장하고 발전함에 따라, 더 많은 사용 사례를 처리할 수 있는 더 많은 기능이 추가되고 있다. 커뮤니티에 참여하는 사람은 새로운 사용 사례를 통해 새로운 기능에 대해 제안할 수 있다.

19.1.6 CDM의 기초가 되는 데이터

OHDSI 연구는 관찰 데이터베이스가 OMOP 공통 데이터 모델(CDM)로 변환되는 것을 전제로 한다. 모든 OHDSI 도구 및 다운스트림 분석 단계는 데이터 표현이 CDM 사양을 준수한다고 가정한다 (4장 참조). 따라서, 이렇게 하기 위한 ETL 프로세스(6장 참조)는 변환 과정에서 인공물 artifact 혹은 다른 사이트 데이터베이스 간에 차이를 발생시킬 수 있어, 특정 데이터 원천에 대해 잘 문서화하는 것도 중요하다. OMOP CDM의 목적은 사이트별 특정 데이터 표현을 감소하는 것이지만, 이는 완벽한 프로세스와는 거리가 있으며 여전히 커뮤니티가 개선하고자 하는 도전적인 영역으로 남아 있다. 따라서, OMOP CDM으로 변환된 모든 원천 데이터에 친숙한 네트워크 연구를 수행할 때 사이트 또는 외부 사이트에서 개인과 공동으로 연구하기 위해 연구를 실행할 때는 여전히 중요하다.

CDM 외에도 OMOP 표준 용어 시스템(standardized vocabulary system) (5장 참조)은 다양한 데이터 원천에서 상호운용성 interoperability을 얻기 위해 OHDSI 프레임워크와 함께 작업하는 데 중요한 요소이다. 표준화된 용어는 다른 모든 원천 용어 시스템이 매핑되는 각 용어 영역 vocabulary domain 내에서 일련의 표준 개념을 정의하려고 한다. 이런 방식으로, 약물, 진단 또는 절차에 다른 원천 용어 시스템을 사용하는 두 개의 서로 다른 데이터베이스는 CDM으로 변환될 때 비교될 것이다. OMOP 용어는 특정 코호트 정의에 적절한 코드를 식별하는데 유용한 계층 구조 hierarchy도 포함한다. 다시 말하지만, 데이터베이스를 OMOP CDM에 ETL하고 OMOP 용어를 사용하는 이점을 얻으려면 다운스트림 쿼리에서 용어 매핑을 구현하고 OMOP 표준화된 용어 코드를 사용하는 것을 권장한다.

19.2 세부 연구 단계

19.2.1 질문 정의

첫 번째 단계는 연구 관심사를 관찰 연구를 통해 해결할 수 있는 정확한 질문으로 변환하는 것이다. 당신은 임상 당뇨병 연구자이며 제2형 당뇨병 type 2 diabetes mellitus(T2DM) 환자에게 제공되는 치료의 질을 조사하려 한다고 가정해 보자. 이러한 큰 목적을 처음 7장에서 설명한 세 가지 유형의 질문 중 하나에 해당하는 훨씬 더 구체적인 질문으로 나눌 수 있다.

특성화 연구에서, “약물 처방이 지정된 의료 환경에서 경증 T2DM 환자와 중증 T2DM 환자에게 권장되는 사항을 준수하는가?”라고 물을 수 있다. 이러한 질문은 또 다른 치료와 비교해 주어진 치료의 유효성에 관한 인과 관계적 질문을 하는 것이 아니라, 단순히 기존 임상 가이드라인과 관련하여 데이터베이스에 약물 처방을 특성화하는 것이다.

T2DM 치료에 대한 처방 지침이 T2DM과 심장병을 함께 진단받은 환자와 같은 특정 환자 집단에 가장 적합한지 여부에 대해서 회의적인 의견이 있을 수 있다. 이러한 질문은 PLE 연구로 번역될 수 있다. 특히, 심부전과 같은 심혈관 질환을 예방하는데 2가지 다른 T2DM 약물 계열의 효과 비교에 대해 질문할 수 있다. 다른 약물을 복용하는 T2DM과 심장 질환을 함께 진단받은 환자를 포함하는 2개의 코호트에서 심부전으로 입원할 상대적 위험도를 조사하는 연구를 설계할 수 있다.

또는, 경증 T2DM에서 중증 T2DM으로 진행할 환자를 예측하는 모델을 개발할 수 있다. 이것은 PLP 질문으로 만들어질 수 있으며, 더욱 신중한 치료를 위해 중증 T2DM으로 발전할 위험이 있는 환자를 지정하기 위해 활용될 수 있다.

순수하게 실용적인 관점에서, 연구 질문을 정의하려면 질문에 대답하는데 필요한 접근 방식이 OHDSI 도구 모음 내에서 사용 가능한 기능을 준수하는지 여부도 평가해야 한다 (현재 도구로 해결할 수 있는 질문 유형에 대한 자세한 설명은 7장 참조). 물론, 항상 자신만의 분석 도구를 설계하거나 현재 사용 가능한 도구를 수정하여 다른 질문에 대답할 수 있다.

19.2.2 데이터 가능성 및 품질 검토

특정 연구 질문에 전념하기 전에, 데이터 품질 (15장 참조)을 검토하고, 특정 필드가 채워지고 데이터가 다루는 치료 설정의 관점에서 특정 관찰 의료 데이터베이스의 특성을 실제로 이해하는 것이 권장된다. 이를 통해 특정 데이터베이스에서 연구 질문을 실현할 수 없는 문제를 신속하게 파악할 수 있다. 아래에서, 발생할 수 있는 몇 가지 일반적인 문제를 지적한다.

경증 T2DM에서 중증 T2DM으로 진행을 예측하는 모델을 개발하는 위의 예로 돌아가자. 이상적으로 T2DM의 중증도는 당화혈색소(HbA1c) 수준을 검사함으로써 평가될 수 있는데, 이는 이전 3개월

동안 환자의 평균 혈당 수치를 반영하는 실험실 측정치이다. 모든 환자가 이 수치를 가지고 있거나 그렇지 않을 수도 있다. 전체 또는 일부 환자가 이 수치를 가지고 있지 않다면, T2DM의 중증도에 대한 다른 임상 기준을 확인하여 대신 사용할 수 있는지를 고려해야 한다. 또는, 일부 환자만 HbA1c 수치를 가지고 있다면, 이 하위 모음 환자에게만 초점을 맞추는 것이 연구에서 원치 않는 비뚤림을 유발하는지 여부도 평가해야 한다. 결측 자료에 대한 추가 논의는 7장을 참조하라.

또 다른 일반적인 문제는 특정 의료 환경에 대한 정보가 부족하다는 것이다. 위에 설명한 인구 수준 추정 예시에서, 제안된 결과는 심부전으로 인한 입원이었다. 주어진 데이터베이스에 입원 환자 정보가 없는 경우, 다른 T2DM 치료 방법의 효과를 비교 평가하기 위해 다른 결과를 고려해야 할 수도 있다. 다른 데이터베이스에서, 외래 환자 진단 데이터를 사용할 수 없으므로, 코호트 설계를 고려해야 할 것이다.

19.2.3 연구 집단

연구 집단을 정의하는 것은 모든 연구의 기본 단계이다. 관찰 연구에서, 관심 있는 연구 집단을 대표하는 개인의 그룹은 종종 코호트로 지칭된다. 코호트로 선택하기 위해 필요한 환자 특성은 현재 임상 질문과 관련된 연구 집단에 의해 결정될 것이다. 간단한 코호트 예는 18세 이상이며 의료 기록에 T2DM 진단 코드가 있는 환자이다. 이 코호트 정의에는 AND 논리로 연결된 두 가지 기준이 있다. 종종 코호트 정의에는 더 복잡한 중첩된 부울 boolean 논리와 특정 연구 기간 또는 환자의 기저 기간 baseline period에 필요한 시간과 같은 추가 시간 기준으로 연결된 더 많은 기준이 포함된다.

정제된 일련의 코호트 정의는 적절한 환자 그룹을 식별하기 위해 적절한 과학 문헌 및 특정 데이터베이스를 해석하는데 어려움을 이해하는 임상 및 기술 전문가의 조언과 검토가 요구된다. 관찰 데이터로 작업할 때 이러한 데이터는 환자의 병력에 대한 완전한 그림을 제공하지 않고 정보의 기록에 도입되었을 수 있는 사람의 실수와 편견이 있는 시간에 대한 스냅샷임을 명심해야 한다. 주어진 환자는 관찰 기간이라고 하는 제한된 시간 동안만 추적 할 수 있다. 연구 중인 특정 데이터베이스 또는 의료 환경 및 질병 또는 치료에 대해, 임상 연구자는 가장 일반적인 오류의 원천을 피하고자 제안을 할 수 있다. 간단한 예를 들어, T2DM 환자를 식별할 때 흔히 발생하는 문제는 T1DM 환자가 때때로 T2DM 진단으로 잘못 코딩된다는 것이다. T1DM을 가진 환자는 근본적으로 다른 그룹이기 때문에, T2DM 환자를 검사하려는 연구에 T1DM 환자 그룹을 의도치 않게 포함하면 결과가 왜곡될 수 있다. T2DM 코호트의 확고한 정의를 갖기 위해, T1DM 환자가 잘못 대표되는 것을 피하고자 당뇨병 치료제로서 인슐린만을 처방받은 환자를 제거하고자 할 수 있다. 그러나 동시에 의료 기록에 T2DM 진단 코드가 있는 모든 환자의 특성에 관심이 있는 상황일 수도 있다. 이 경우, 잘못 코딩된 T1DM 환자를 제거하기 위해 추가 자격 기준을 적용하는 것이 적절하지 않을 수 있다.

연구 집단의 정의가 기술되면, OHDSI 도구인 ATLAS는 관련 코호트를 생성하기 위한 좋은 출발점

이다. ATLAS 및 코호트 생성 프로세스는 8장 및 10장에 자세히 설명되어 있다. 간단히 말해, ATLAS는 상세한 포함 기준으로 코호트를 정의하고 생성하기 위한 사용자 인터페이스 user interface(UI)를 제공한다. 코호트가 ATLAS에서 정의되면, 사용자는 프로토콜에 통합하기 위해 세부 정의를 사람이 읽을 수 있는 형식으로 직접 내보낼 수 있다. 어떤 이유로 ATLAS 인스턴스가 관찰 의료 데이터베이스에 연결되지 않은 경우에도, ATLAS를 사용하여 코호트 정의를 작성하고 연구 패키지에 통합하기 위해 기본 SQL 코드를 직접 내보내서 SQL 데이터베이스 서버에서 별도로 실행할 수 있다. ATLAS는 코호트 정의를 위한 SQL 코드 작성 이상의 이점을 제공하므로 가능하면 ATLAS를 직접 사용하는 것이 권장된다 (아래 참조). 마지막으로, ATLAS UI로 코호트 정의를 구현할 수 없고, 수동 사용자 지정 SQL 코드가 필요한 드문 상황이 있을 수 있다.

ATLAS UI는 다양한 선택 기준으로 코호트를 정의할 수 있다. 기초 기준 baseline criteria뿐만 아니라 코호트 진입 entry 및 종료 exit 기준은 각 도메인에 대해 표준 코드 standard code를 지정해야 하는 질병 condition, 약물 drug, 시술 procedure 등과 같은 OMOP CDM의 모든 도메인을 기준으로 정의할 수 있다. 또한, 이러한 도메인을 기반으로 하는 논리 필터와 연구 기간을 정의하는 시간 기반 time-based 필터 및 기저 시간 프레임 baseline timeframe을 ATLAS 내에서 정의할 수 있다. ATLAS는 각 기준에 대한 코드를 선택할 때 특히 유용하다. ATLAS에는 코호트 정의에 필요한 일련의 코드를 작성하는데 사용할 수 있는 용어 탐색 기능 vocabulary-browsing feature이 통합되어 있다. 이 기능은 OMOP 표준 용어에만 의존하며 용어 계층에 모든 하위 요소 descendant를 포함하는 옵션이 있다 (5장 참조). 따라서, 이 기능을 사용하려면 ETL 프로세스 중에 모든 코드가 표준 코드에 적절하게 매핑되어 있어야 한다 (6장 참조). 포함 기준에 사용할 최상의 코드 모음 codeset이 명확하지 않은 경우, 코호트 정의에서 일부 탐색적 분석이 필요한 곳일 수 있다. 대안적으로, 상이한 코드 모음을 사용하여 코호트의 다른 가능한 정의를 설명하기 위해 더욱 공식적인 민감도 분석이 고려될 수 있다.

ATLAS가 데이터베이스에 연결되도록 적절하게 구성되어 있다고 가정하면, 정의된 코호트를 생성하기 위한 SQL 쿼리를 ATLAS 내에서 직접 실행할 수 있다. ATLAS는 각 코호트에 고유한 ID를 자동으로 할당하여 나중에 사용할 수 있도록 백엔드 backend 데이터베이스에서 코호트를 직접 참조하는데 사용될 수도 있다. 코호트는 발생률 연구를 수행하기 위해 ATLAS 내에서 직접 사용되거나 PLE 또는 PLP 연구 패키지의 코드로 백엔드 데이터베이스에서 직접 지시될 수 있다. 주어진 코호트에 대해 ATLAS는 코호트에 있는 개인 환자 ID, 색인 날짜 및 코호트 종료 날짜만 저장한다. 이 정보는 특성화, PLE 또는 PLP 연구를 위한 환자의 기본 공변량과 같이 환자에게 필요할 수 있는 다른 모든 속성 또는 공변량을 도출하기에 충분하다.

코호트가 생성되면, 환자 인구통계의 특성 요약과 가장 빈번한 관찰된 약물 및 상태 빈도가 기본적으로 ATLAS 내에서 직접 생성되고 볼 수 있다.

실제로 대부분의 연구에서는 여러 코호트 또는 여러 모음의 코호트를 지정해야 하며, 그런 다음 새로운 임상적 통찰을 얻기 위해 다양한 방식으로 비교된다. PLE 및 PLP의 경우, OHDSI 도구는 이러한 여러 코호트를 정의하기 위한 구조화된 프레임워크를 제공한다. 예를 들어, PLE 효과 비교 연구에서는 일반적으로 최소 3개의 코호트 (대상 코호트 target cohort, 대조 코호트 comparator cohort, 결과 코호트 outcome cohort)를 정의한다 (12장 참조). 또한, 전체 PLE 효과 비교 연구를 수행하려면, 음성 대조군 결과 negative control outcome 및 양성 대조군 결과 positive control outcome를 가진 여러 코호트가 필요하다. OHDSI 도구 모음 toolset은 18장에서 자세히 설명한 대로 이러한 음성 및 양성 대조군 코호트의 생성을 가속화하고 자동화하는 방법을 제공한다.

마지막으로, 연구에 대한 코호트를 정의하면 표현형 phenotype이 본질적으로 추출 가능한 코호트 정의인 확고하고 검증된 표현형의 라이브러리를 정의하기 위해 OHDSI 커뮤니티에서 진행 중인 작업의 이점이 있을 수 있다. 기존 코호트 정의 중 어느 것이 연구에 적합한 경우, JSON 파일을 ATLAS 인스턴스로 가져와 정확한 정의를 얻을 수 있다.

19.2.4 실행 가능성 및 진단

코호트가 정의되고 생성되면, 사용 가능한 데이터 원천에서 연구의 실행 가능성을 조사하기 위한 보다 공식적인 프로세스를 수행하고 결과를 최종 프로토콜에 요약할 수 있다. 연구의 실행 가능성에 대한 평가는 다수의 탐색적이며 때로는 반복적인 활동을 포함할 수 있다. 여기에 몇 가지 일반적인 측면을 설명한다.

이 단계의 주요 활동은 생성된 코호트가 원하는 임상적 특성과 일치하고 예기치 않은 특성을 표시하는지 확인하기 위해 코호트 내의 특성 분포를 철저히 검토하는 것이다. 위의 T2DM 예로 돌아가서, 다른 모든 진단의 빈도를 검토하여 이 간단한 T2DM 코호트를 특성화함으로써, T1DM 환자를 포착하는 문제 또는 다른 예상치 못한 문제를 표시할 수 있다. 코호트 정의의 임상적 타당성의 품질 검사로써 연구 프로토콜에 초기에 새로운 코호트를 특성화하는 단계를 작성하는 것은 바람직하다. 구현 측면에서 첫 번째 패스를 수행하는 가장 쉬운 방법은 코호트가 ATLAS에서 생성될 때 기본적으로 생성될 수 있는 코호트 인구통계와 주요 약물 및 상태를 검사하는 것이다. ATLAS 내에서 직접 코호트를 작성하는 옵션을 사용할 수 없는 경우, 수동 SQL 또는 R 기능 추출 패키지 feature extraction package를 사용하여 코호트를 특성화할 수 있다. 실제로, 대규모 PLE 연구 또는 PLP 연구에서, 이러한 단계는 기능 추출 단계를 가진 연구 패키지에 내장될 수 있다.

PLE 또는 PLP의 실행 가능성을 평가하기 위한 또 하나의 일반적이고 중요한 단계는 코호트 크기와 대상 코호트와 대조 코호트의 결과 수를 평가하는 것이다. ATLAS의 발생률 기능은 다른 위치에서 설명한 대로 검정력 계산을 수행하는데 사용될 수 있는 이러한 수를 찾는 데 활용될 수 있다.

PLE 연구에 권장되는 또 다른 옵션은 대상군과 대조군 집단 사이에 충분한 중복이 있는지 확인하기

위해 성향 점수 propensity score(PS) 매칭 단계 및 관련 진단을 완료하는 것이다. 이 단계는 12장에서 자세히 설명된다. 또한, 이러한 최종 일치된 코호트를 사용하여 통계적 검정력을 계산할 수 있다.

때에 따라, OHDSI 커뮤니티의 작업은 사용 가능한 샘플 크기가 주어지면 최소 검출 가능한 상대 위험 minimal detectable relative risk(MDRR)을 보고하여 연구가 실행된 후에만 통계적 검정력을 검사한다. 이 방법은 많은 데이터베이스와 사이트에서 높은 처리량, 자동화된 연구를 실행할 때 더 유용할 수 있다. 이 시나리오에서는, 사전 필터링보다는 모든 분석을 수행한 후 주어진 데이터베이스에서 연구 검정력을 더 잘 탐색할 수 있다.

19.2.5 프로토콜 및 연구 패키지 마무리

이전의 모든 단계에 대한 작업이 완료되면, 세부 코호트 정의와 이상적으로 ATLAS에서 추출한 연구 설계 정보가 포함된 최종 프로토콜이 완성되어야 한다. 부록 D에 PLE 연구를 위한 전체 프로토콜에 대한 샘플 목차를 제공한다. 이것은 OHDSI GitHub에서도 찾을 수 있다. 이 샘플은 포괄적인 가이드 및 체크리스트로 제공되지만, 일부 섹션은 여러분의 연구와 관련이 있을 수도 있고 그렇지 않을 수도 있다.

그림 19.1에 도시된 바와 같이, 최종 연구 프로토콜을 사람이 읽을 수 있는 형태로 완성하는 것은 최종 연구 패키지에 통합된 모든 기계 판독 가능한 연구 코드를 준비하는 것과 병행하여 수행되어야 한다. 이러한 후자 단계는 아래 그림에서 연구 구현 study implementation이라고 한다. 이것에는 ATLAS에서 최종 연구 패키지를 내보내거나 필요한 사용자 정의 코드를 개발하는 것이 포함된다.

완성된 연구 패키지는 차례로 프로토콜에 설명될 수 있는 예비 진단 단계만 실행하기 위해 사용될 수 있다. 예를 들어, 두 치료의 효과 비교를 조사하기 위한 새로운 사용자 코호트 PLE 연구의 경우, 대상 및 대조 집단이 연구가 실행할 수 있도록 충분한 중첩을 가지고 있다는 것을 확인하기 위해 코호트 생성, 성향 점수 생성 및 매칭이 요구될 것이다. 이것이 결정되면, 결과 수를 얻기 위해 결과 코호트와 교차된 일치된 대상 및 대조 코호트로 검정력 계산을 수행할 수 있고, 이러한 계산 결과는 프로토콜에 기술할 수 있다. 이러한 진단 결과를 바탕으로, 최종 결과 모델을 실행하여 계속 연구를 진행할지 여부를 결정할 수 있다. 특성화 또는 PLP 연구와 관련하여, 여기서는 모든 시나리오를 간략하게 설명하지는 않지만, 이 단계에서 완료해야 하는 유사한 단계가 있을 수 있다.

중요한 것은, 이 단계에서 공동 연구자와 이해 관계자 stakeholder가 최종 프로토콜을 검토하도록 하는 것이다.

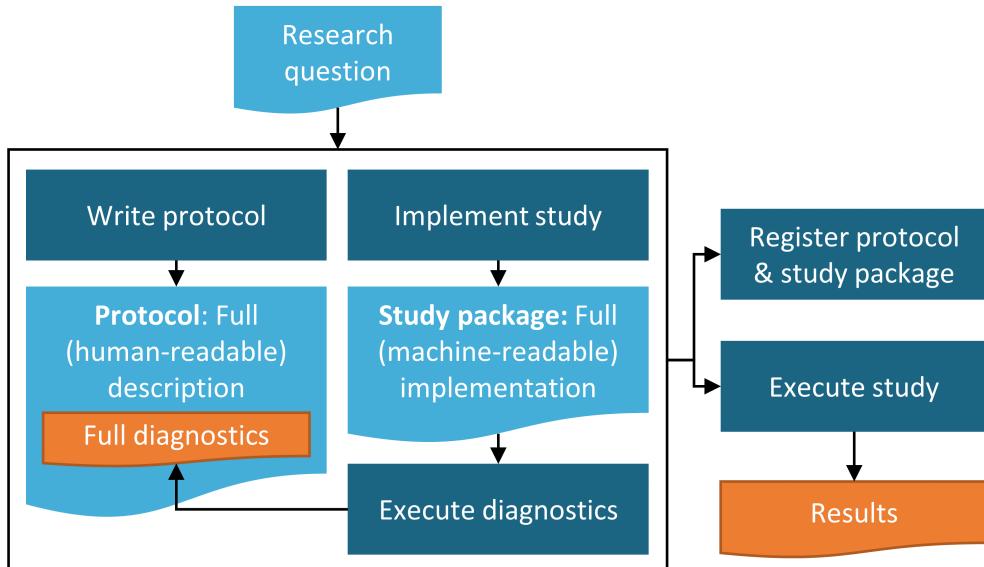


Figure 19.1: Diagram of the study process.

19.2.6 연구 수행

모든 이전 단계가 완료되면, 연구 실행은 이상적으로 간단해야 한다. 물론, 코드 또는 프로세스는 프로토콜에 요약된 방법 및 매개 변수에 대한 충실도 fidelity를 검토해야 한다. 연구 패키지가 사용자 환경에서 올바르게 실행되는 것을 확인하기 위해 연구 패키지를 테스트하고 디버깅해야 할 수도 있다.

19.2.7 해석 및 작성

표본 크기가 충분하고 데이터 품질이 합리적인 잘 정의된 연구에서, 결과 해석은 종종 간단하다. 마찬가지로, 최종 결과를 작성하는 것 이외의 최종 보고서를 작성하는 대부분의 작업은 프로토콜 계획 및 작성에서 수행되기 때문에, 보고서 또는 출판하려는 논문의 최종 작성 역시 종종 간단하다.

그러나 해석이 더 어려워지고 조심스럽게 해석에 접근해야 하는 몇 가지 일반적인 상황이 있다:

1. 표본 크기가 유의미한 경계선에 있으며 신뢰 구간이 커진다.
2. PLE에 특수한 경우: 음성 대조군을 사용한 p-value 교정은 상당한 비뚤림을 나타낼 수 있다.
3. 연구를 진행하는 과정에서 예상치 못한 데이터 품질 문제가 밝혀진다.

어떤 주어진 연구에 대해서도, 위의 우려 사항을 보고하고 그에 따른 연구 결과의 해석을 조율하는 것은 연구 저자의 재량에 달려 있다. 프로토콜 개발 프로세스와 마찬가지로, 최종 보고서를 발표하

거나 출판하려고 원고를 제출하기 전에 임상 전문가와 이해 관계자가 연구 결과와 해석을 검토할 것을 권장한다.

19.3 요약



- 연구는 잘 정의된 질문을 조사해야 한다.
- 사전에 데이터 품질, 완전성 및 타당성에 대한 적절한 점검을 수행한다.
- 가능하면 프로토콜 개발 프로세스에 원천 데이터베이스 전문가를 포함하는 것을 권장한다.
- 프로토콜로 제안된 연구를 미리 문서화한다.
- 작성된 프로토콜과 함께 연구 패키지 코드를 생성하고 최종 연구를 실행하기 전에 실행 가능성 및 진단을 수행하고 설명한다.
- 연구는 실행 전에 (필요한 경우) 등록하고 승인을 받아야 한다.
- 완성된 보고서 또는 원고는 임상 전문가 및 기타 이해 관계자가 검토해야 한다.

Chapter 20

OHDSI 네트워크 리서치

Chapter leads: Kristin Kostka, Greg Klebanov & Sara Dempster

OHDSI의 사명은 관찰 연구를 통해 높은 수준의 근거를 도출하는 것이다. 이것은 공동 연구를 통해 달성할 수 있는데, 이전 장에서 후향적 데이터베이스 연구를 수행하기 위해, OHDSI 커뮤니티가 OMOP 표준화 용어, 공통 데이터 모델(CDM), 분석 방법 패키지, ATLAS 및 연구 단계 19장을 포함하여 높은 수준의 재현 가능한 연구를 용이하게 하는 표준과 도구를 어떻게 작성했는지 확인할 수 있었다. OHDSI 네트워크 연구는 공간적으로 분산된 여러 데이터에서 연구를 수행하는 투명하고 일관되며 재현 가능한 최상의 방법을 보여준다. 이 장에서는 OHDSI 네트워크 연구를 구성하는 요소, 네트워크 연구를 실행하는 방법 및 ARACHNE 연구 네트워크와 같은 기술에 대해 알아보고자 한다.

20.1 연구 네트워크로서의 OHDSI

OHDSI 연구 네트워크는 의료 분야에서 관찰 데이터 연구를 발전시키려는 연구자의 국제적인 모임이다. 현재 네트워크는 OMOP 공통 데이터 모델로 표준화된 150개가 넘는 데이터베이스로 구성되어 있고, 20억 건 이상의 환자 기록이 포함되어 있다. OHDSI는 전 세계 의료기관이 데이터를 OMOP 공통 데이터 모델로 변환하고 다기관 네트워크 연구에 참여하도록 유도하고, 누구나 참여할 수 있는 개방형 네트워크로 운영되고 있다. 기관의 데이터 변환이 완료되면 OHDSI 프로그램 관리자 (<mailto:contact@ohdsi.org>)는 데이터 네트워크 인구 조사 현황에 기관의 정보를 게시하도록 알려준다. 각 OHDSI 네트워크 기관은 자발적으로 참여하고, 의무는 없다. 각 기관은 각각의 네트워크 연구를 선택할 수 있고, 각 연구에서 데이터는 기관의 방화벽 안에 위치하며, 네트워크 기관에서 환자 수준

데이터는 수집되지 않는다. 단지, 연구 최종 결과만 공유할 수 있다.



OHDSI 네트워크에 참여하는 기관의 이점

- **무료 도구 이용 :** OHDSI는 데이터 특성 분석 및 표준화된 분석 (OMOP concept 탐색, 코호트 정의 및 특성 분석, 인구 수준 추정 및 환자 수준 예측 연구 실행)을 위한 오픈 소스 도구를 무료로 제공한다.
- **최고의 연구 커뮤니티 참여 :** 네트워크 연구를 작성 및 게시하고 다양한 분야 및 관계자 그룹의 리더와 협력할 수 있다.
- **벤치마크 관리 기회 :** 네트워크 연구를 통해 데이터 파트너 간에 임상 특성 및 품질 개선 벤치마크를 할 수 있다.

20.2 OHDSI 네트워크 연구

이전 19장에서 CDM을 사용하여 연구를 수행하기 위한 일반적인 고려 사항에 대해 알아보았다. 일반적으로 한 개의 CDM 또는 여러 CDM에서 연구를 수행할 수 있고, 단일 기관의 CDM 데이터 뿐만 아니라 여러 기관에서도 실행할 수 있다. 이번 장에서는 여러 기관의 분석을 네트워크 연구로 확장하려는 이유를 알아보고자 한다.

20.2.1 OHDSI 네트워크 연구가 필요한 이유

관찰 연구의 전형적인 사용 사례는 “실세계”에서 치료의 비교 효과 또는 안전성을 조사하는 것이다. 구체적으로 이야기하면, 임상 시험에서 얻어지는 일반적 결과에 대한 확인을 위해 약물이 시판된 후 시행되는 임상 시험을 복제해서 동일하게 시행하는 연구를 해야 하는 경우, 임상 시험이 이루어지지 않은 상태에서는 약물의 사용 시 적응증외 사용 off label이 되는데, 이러한 상황에서 허용된 약물 과의 효과 비교연구를 시행해야 하는 경우, 임상 시험에서 관찰하기 어려운 매우 희귀한 부작용에 대한 시판 후 안전성 결과를 연구가 필요할 수 있는 경우를 들 수 있다. 만약 이러한 연구에서 하나 또는 두 개의 데이터베이스에서 단일 관측 연구를 시행하면 특정 환자그룹에 제한된 상황에서만 결과를 얻기 때문에 정확한 결과를 얻는데 충분하지 않을 수 있다.

관찰 연구의 결과는 순응도, 유전적 다양성 또는 환경적 요인, 전반적인 건강 상태와 같이 데이터 원천의 위치에 따라 달라지는 많은 요인에 의해 영향을 받을 수 있다. 따라서 네트워크에서 관측 연구를 실행하려는 일반적인 이유는 데이터 원천의 다양성을 늘리고 잠재적으로 연구를 통해 일 반화된 결과를 얻기 위함이다. 바꿔 말하면, 연구 결과가 여러 기관에서도 같은 결과를 보이는지 여부와 만약 다른 결과를 보이는 경우 원인을 확인할 수 있는지를 알아보기 위함이다.

따라서 네트워크 연구를 통해 광범위한 설정과 데이터 원천을 조사하여 관측 연구 결과에 “실제” 요인의 영향을 조사할 수 있다.

20.2.2 OHDSI 네트워크 연구의 정의



어떤 연구를 네트워크 연구라고 할 수 있는가? OHDSI 연구는 다른 기관의 여러 CDM에서 실행될 때 OHDSI 네트워크 연구라고 할 수 있다.

네트워크 연구에 대한 OHDSI 접근 방식은 OMOP CDM과 표준화된 도구 및 연구 실행을 위해 모든 매개 변수를 지정할 수 있는 연구 패키지를 사용하여 시행된다. OHDSI 표준화 분석은 불필요한 혼란 변수를 줄이고 네트워크 연구의 효율성과 확장성을 향상하도록 설계되었다.

네트워크 연구는 OHDSI 연구 커뮤니티의 중요한 부분이다. 그러나 OHDSI 연구를 전체 OHDSI 네트워크에 패키지를 반드시 공유할 의무는 없다. 단일 기관 내에서 OMOP CDM 및 OHDSI 분석법 라이브러리를 사용하여 연구를 수행하거나 선택된 기관에서만 시행하도록 제한할 수 있다. 연구가 단일 데이터베이스에서 실행되도록 설계되었는지, 제한된 파트너 집합을 대상으로 연구를 수행하거나, OHDSI 네트워크에 완전히 참여하기 위해 연구를 시작하는지 여부는 각 연구자의 재량에 달려있다. 이 장에서는 OHDSI 커뮤니티가 수행하는 개방형 네트워크 연구에 대해 다루도록 하겠다.

개방형 OHDSI 네트워크 연구 요소 개방형 OHDSI 네트워크 연구를 수행할 때는 완전히 투명한 연구를 수행하는 것이다. OHDSI 연구에서 다음과 같은 특징적인 몇 가지 구성요소가 있다.

- 모든 문서, 연구 코드 및 후속 결과는 OHDSI GitHub에서 공개적으로 제공된다.
- 연구자는 수행할 분석의 범위와 의도를 자세히 설명하는 공개 학습 프로토콜을 작성하고 게시해야 한다.
- 연구자는 CDM을 준수하는 코드로 연구 패키지 (일반적으로 R 또는 SQL)를 작성해야 한다.
- 연구자는 OHDSI 네트워크 연구를 위해 공동 작업자를 홍보하고 모집하기 위해 OHDSI 커뮤니티 콜에 참석하도록 권장된다.
- 분석이 끝나면 OHDSI GitHub에서 종합 연구 결과를 제공한다.
- 가능하면 연구자는 연구 R Shiny Applications를 data.ohdsi.org에 게시하도록 권장된다.

다음 장에서는 네트워크 연구를 구현하기 위한 고유한 설계 및 논리적 고려 사항뿐만 아니라 자체 네트워크 연구를 만드는 방법에 관해 설명한다.

20.2.3 OHDSI 네트워크 연구 설계를 위한 고려 사항

OHDSI 네트워크에서 실행할 연구를 설계하려면 연구 코드를 설계하고 조립하는 방법에 대한 패러다임 전환이 필요하다. 일반적으로 목표 데이터 모음을 염두에 두고 연구를 설계하게 되는데, 연구 분석에 이용되는 데이터 중에서 자신의 연구에 유리한 결과가 나오도록 코드를 작성할 가능성이 있다. 예를 들어, 혈관 부종 코호트를 구성하는 경우 보통 개별 연구자들은 자신의 CDM 데이터 상의 혈관 부종에 대한 concept 1 가지만 선택할 수 있는데, 그렇게 하는 경우 연구용 데이터가 특정 의료 환경 (1차 의료, 외래 환경) 또는 특정 지역 (미국 중부)에만 있는 경우 문제가 될 수 있다. 결국 이렇게 선택된 연구 코드는 코호트 정의에 있어서 선택비뚤림 selection bias이 발생하게 된다. (역자 주: OHDSI 연구를 설계할 때는, 반드시 다른 의료 용어 체계 (예를 들어 ICD-9) 상에서도 나의 phenotype이 잘 작동할 수 있을지 면밀히 고민하면서 설계해야 한다.)

OHDSI 네트워크 연구에서는 본인의 데이터만을 위한 연구 패키지를 설계 및 구축하지 않고, 전 세계 여러 기관에서 실행할 연구 패키지를 구축한다. 기관 외부의 참여 기관에 대한 기본 데이터를 검색하거나 공유하는 것은 불가능하고, 결과 파일만 공유한다. 연구 패키지는 CDM의 도메인에서 사용 가능한 데이터만 수집할 수 있다. 관찰 의료 연구 데이터가 확인되는 기관은 매우 다양하기 때문에 연구자는 이런 다양한 연구 기관에서 concept set을 적용할 수 있도록 포괄적인 접근법이 필요하다. OHDSI 연구 패키지는 보통 모든 기관에서 동일한 코호트 정의를 사용한다. 다시 말하면, 적격한 데이터 (보험 청구 자료 또는 전자 의무 기록(EHR) 자료)의 하부 구조에서만 적용되는 코호트 정의를 함으로써 편중이 발생하지 않도록 주의해야 한다. 따라서, 코호트 정의를 작성할 때는 여러 CDM에서 적용 가능한 코호트 정의를 작성하도록 신경 써야 한다. OHDSI 연구 패키지에서는 데이터베이스 연결이나 저장 위치와 같은 일부 부분만 각 기관에서 변경하고, 연구에 관련된 변수는 모두 같은 매개 변수를 사용하고 있다. 나중에 다양한 데이터 모음에서 임상적 소견을 해석하는 데 미치는 영향에 대해 알아보도록 한다.

임상 용어 코딩 변형 외에도 로컬 기술 인프라에서 변형을 예상하여 설계해야 한다. 연구자가 작성한 학습 코드는 단일 기술 환경에서 실행되지 않을 것이다. 각 OHDSI 네트워크 기관은 데이터베이스를 독립적으로 선택할 수 있다. 이는 연구 패키지를 특정 데이터베이스 용어로만 하드 코딩할 수 없음을 의미한다. 연구 코드는 해당 데이터베이스의 운영자가 쉽게 수정할 수 있는 SQL 유형으로 매개 변수화되어있어야 한다. 다행히 OHDSI 커뮤니티에는 ATLAS, DatabaseConnector (<https://ohdsi.github.io/DatabaseConnector/>) 및 SqlRender (<https://ohdsi.github.io/SqlRender/>) 와 같은 솔루션이 있어, 각자의 데이터베이스 용어로 변환하여 연구자의 연구 패키지를 실행시킬 수 있다. OHDSI 연구자는 여러 환경에서 연구 패키지를 테스트하고 검증할 수 있도록 다른 네트워크 연구 기관의 도움을 요청하는 것이 필요하다. 코딩 오류가 발생하면 연구자는 OHDSI 포럼 (<http://forums.ohdsi.org>) 을 사용하여 패키지를 다른 연구자와 논의하고 디버깅할 수 있다.

많은 방법론적인 혁신이 있었지만, 그럼에도 불구하고 OHDSI 연구를 진행하는 것은 결코 쉽지 않다. 만약, 당신이 적절한 연구 주제를 가지고 있고, 공개 ATLAS 등을 통해 연구 패키지를 만들어보았음에도 불구하고 OHDSI 네트워크 연구 진행에 어려움을 겪는다면, OHDSI 연구 육성 커뮤니티 study nurture committee 에 문의하여 도움을 청하기 바란다.

20.2.4 OHDSI 네트워크 연구를 위한 물류적 관점의 고려사항

OHDSI는 개방형 연구 커뮤니티이며 OHDSI 중앙 조정 센터는 공동 연구자가 커뮤니티 연구를 이끌고 참여할 수 있도록 커뮤니티 인프라를 제공하는 역할을 한다. 모든 OHDSI 네트워크 연구에는 연구 책임자가 필요하며 OHDSI 커뮤니티의 참여자의 누구라도 될 수 있다. OHDSI 네트워크 연구는 연구 책임자, 공동 연구자 및 참여 네트워크 데이터 퍼트너 간의 긴밀한 협업이 필요하다. 기관마다 각자의 CDM에서 연구가 실행될 수 있도록 필요시 자체 승인 절차를 수행해야 한다. 데이터 분석가는 연구를 수행할 수 있는 적절한 권한을 부여하기 위해 현지 IT 팀의 지원을 받아야 할 수도 있다. 각 기관에서 연구팀의 규모와 범위는 OMOP CDM과 OHDSI 패키지의 숙련도뿐만 아니라 제안된 네트워크 연구의 크기와 복잡성에 따라 결정되어야 한다. 또한, OHDSI 네트워크 연구를 수행하는 기관의 숙련도에 따라 필요한 인력이 결정될 수 있다.

각각의 연구에 따라 초기 준비 절차는 다음과 같을 수 있다 (기관별 상이).

- 연구에 대해 기관생명윤리위원회 (또는 동등한 기관)에 신청한다
- 기관생명윤리위원회의 승인을 얻은 후 연구를 시행한다.
- 승인된 CDM의 스키마를 읽고 쓸 수 있는 권한을 획득한다.
- 연구 패키지를 실행할 수 있도록 RStudio 환경을 조정한다.
- 기술적인 문제가 없는지 연구 패키지 코드를 검토한다.
- 연구 패키지 실행을 위한 연관된 R 패키지를 설치와 실행을 승인받도록 각 기관의 IT 팀에 업무 협조를 구한다.



데이터 품질 및 네트워크 연구: 6장에서 논의한 것처럼 품질 관리는 ETL 프로세스의 기본 적이고 반복적인 부분이다. 이는 네트워크 연구 프로세스와 관계없이 정기적으로 수행해야 한다. 네트워크 연구의 경우, 연구 책임자는 참여 기관의 데이터 품질 보고서를 검토하거나 사용자 지정 SQL 쿼리를 작성 및 배포하여 데이터 원천 간의 차이점을 확인할 수 있다. OHDSI 내에서 진행되는 데이터 품질 노력에 대한 자세한 내용은 15장을 참조한다.

각 기관에는 연구 패키지를 실행하는 데이터 분석가가 있을 것이다. 이 인원은 환자의 민감한 정보가 전송되지 않는지 연구 패키지의 결과를 검토해야 한다. 인구 수준 추정 Population-Level Effect Estimation(PLE) 및 환자 수준 예측 Patient Level Prediction(PLP)과 같은 사전 구축된 OHDSI

패키지를 사용하는 경우 지정된 분석에 대한 최소 환자 수를 정할 수 있는 설정이 있다. 데이터 분석가는 이러한 임계값을 검토하고 각 기관의 정책을 준수하는지 확인해야 한다.

연구 결과를 공유할 때 데이터 분석가는 결과 전송 방법을 포함하여 모든 사항에 대해 각 기관의 정책을 준수해야 하며 결과의 외부 반출을 위한 승인 프로세스를 준수해야 한다. **OHDSI 네트워크 연구는 환자 수준 데이터를 공유하지 않는다.** 즉, 각 기관의 환자 수준 데이터는 중앙에 저장되지 않는다. 연구 패키지는 집계 결과 (통계 결과 요약, 포인트 추정치, 진단 플롯 등)로 설계된 결과 파일을 작성하며 환자 수준 정보는 공유되지 않는다. 따라서, 많은 기관에서는 참여 연구팀 구성원 간에 데이터 공유 계약을 실행할 필요가 없다. 그러나 관련 기관 및 데이터 원천에 따라 특정 연구팀 원이 확인하고 보다 공식적인 데이터 공유 계약을 체결해야 할 수도 있다. 네트워크 연구에 관심이 있는 CDM 데이터 소유 연구기관 연구자인 경우 각 기관의 관련 팀과 협의하여 OHDSI 커뮤니티 연구에 참여하기 위해 충족해야 하는 정책을 확인하는 것이 필요하다.

20.3 OHDSI 네트워크 연구 수행하기

OHDSI 네트워크 연구를 수행하기 위한 세 가지 단계는 다음과 같다.

- 연구 설계와 타당성
- 연구 수행
- 결과 배포 및 출판

20.3.1 연구 설계와 타당성

연구 타당성 단계 (또는 사전 학습 단계)는 연구 주제를 정의하고 연구 프로토콜에 따른 주제의 결과를 도출하는 프로세스를 의미한다. 이 단계는 참여 기관에서 연구 프로토콜을 실행할 수 있는 가능성을 평가하는 데 중점을 둔다.

타당성 단계의 결과는 네트워크 실행 준비가 완료된 최종 프로토콜 및 연구 패키지를 생성하는 것이다. 공식 프로토콜은 지정된 연구 책임자 (논문에서는 책임저자) 및 연구 일정에 대한 정보를 포함한 내용으로 연구팀을 자세히 설명한다. 이 프로토콜은 추가로 연구에 참여하는 네트워크 기관이 CDM 데이터에서 전체 연구 패키지를 검토, 승인 및 실행하는 데 중요한 구성 요소가 되고 있다. 임상 시험 계획서에는 연구 모집단, 사용되는 방법, 결과 저장 및 분석 방법, 완료 후 연구 결과 배포 방법 (논문, 프레젠테이션 등)이 포함되어야 한다.

타당성 검증 단계는 정립된 절차는 없다. 이 과정은 연구의 종류에 따라 달라진다. 최소한 연구 책임자는 필요로 하는 약물 노출, 치치 정보, 진단명 또는 환자의 인구학적 정보가 있는 네트워크 기관을 알아보는 데 시간을 할애한다. 가능한 경우, 연구 책임자는 자신의 CDM을 이용하여 연구

대상을 설계해야 한다. 그러나, 연구 책임자는 네트워크 연구를 실행하기 위해 실제 환자 데이터가 있는 OMOP CDM에 접속할 필요는 없다. 책임자는 가상의 데이터 (CMS synthetic Public Use Files, Mitre 또는 Synthea의 syntheticMass)를 사용하여 대상 코호트 정의를 설계하고, OHDSI 네트워크 기관 공동 연구자에게 코호트의 타당성을 검증하도록 요청하는 방식으로도 할 수 있다. 타당성 조사 단계는 공동 연구자에게 ATLAS에서 만들어진 코호트 정의 JSON file을 이용하여 코호트를 생성하도록 요청하거나, 19장에서 설명한 것처럼 R 패키지를 실행하여 초기 진단을 실행해 보도록 요청하는 방식으로 할 수 있다. 동시에 연구 책임자는 기관생명윤리위원회와 같은 조직에서 OHDSI 연구를 승인받기 위한 절차를 진행한다. 타당성 조사 단계에서는 이러한 조직별 절차를 완료하는 작업은 연구 책임자의 역할로 진행되어야 한다.

20.3.2 연구 실행

타당성 연구를 완료한 후에는 실행의 단계로 진행한다. 이 단계는 OHDSI 네트워크 기관이 분석에 참여하는 시기이다. 이 단계는 우리가 이전에 논의하였던 연구 설계와 논리적 고려 사항이 가장 중요한 시기이다.

연구의 실행은 연구 책임자가 새로운 OHDSI 네트워크 연구에 대해 공식적으로 소개하고 참여 기관을 공식적으로 모집하는 것으로부터 시작한다. 연구 책임자는 연구 프로토콜을 OHDSI GitHub에 공개하고, 매주 열리는 OHDSI 커뮤니티 원격 회의나 OHDSI 포럼에 연구에 대해 소개하고, 참여하는 센터와 공동 연구자를 모집하도록 한다. 연구에 기관이 참여하려고 하면, 연구 책임자는 각 기관에 직접 연락하여 연구 프로토콜과 코드뿐만 아니라 연구 패키지를 실행하는 방법 안내서를 저장하고 있는 GitHub 저장소를 알려주도록 한다. 모든 기관에서 동시에 진행하여 각 기관의 결과가 동시에 공유되어 다른 기관의 결과값에 의해 자신의 기관 결과에 영향을 주지 않도록 하는 것 이상적이다.

각 기관 연구팀은 각자의 기관에서 패키지를 실행하고 외부로 결과를 공유할 수 있는 절차를 확인해서 진행해야 한다. 어떤 부분에서는 기관생명윤리위원회의 승인을 받거나 동등한 승인을 받는 것과 같은 절차일 것이다. 연구 실행이 승인되면 각자 기관의 연구자나 통계학자가 연구 책임자의 안내서대로 패키지를 실행하고, OHDSI 가이드라인에 따라서 표준화된 형태의 결과를 생성한다. 각자의 기관은 기관생명윤리위원회의 절차에 따라 데이터를 공유하도록 한다. 만약, 기관생명윤리위원회의 승인을 얻지 못한 상태에서는 결과를 공유해서는 안 된다.

연구 책임자는 연구 결과를 받을 방법 (예를 들어, SFTP나 Amazon S3 bucket)을 결정해야 하고 결과를 전환하는 시간표를 결정해야 한다. 각 기관에서는 전송 방법이 내부 규약에 맞지 않는 경우에는 새로운 해결방법을 개발해야 할 수도 있다.

실행 단계에서는 통합된 연구팀 (연구 책임자와 참여 기관 구성원 포함)은 합당한 조정이 필요한

경우, 실행을 반복해야 할 수 있다. 만약 이러한 과정에서 수정된 연구 프로토콜이 승인된 연구 내용을 벗어난 경우 각 참여 기관에서는 업데이트된 프로토콜을 받아서 각자의 기관생명윤리위원회의 검토 및 재승인 절차를 진행해야 한다.

연구 책임자와 데이터 사이언티스트나 통계학자는 여러 기관에서 공유된 데이터를 모으고, 적절하게 메타분석을 시행하는 최종적인 역할을 한다. OHDSI 커뮤니티에는 단일 결과를 얻기 위해 여러 네트워크 기관에서 생성된 결과를 모으고 분석하는 겸증된 방법이 있다. EvidenceSynthesis는 여러 기관에서 생성된 결과로 증거를 통합하고 진단을 실행할 수 있는 공개된 R 패키지이다. 이것은 메타분석과 포레스트 플롯을 작성할 수 있는 함수를 포함하고 있다.

연구 책임자는 참여 기관의 상황을 모니터하고 정기적으로 확인함으로써 패키지 실행 시 문제점을 해결할 수 있도록 도와줘야 한다. 연구 패키지가 각 기관에서 일괄적으로 실행되지 않을 수 있다. 데이터베이스 측면 (권한 설정/스키마 승인)과 연관된 문제점과 각자의 환경이 달라서 생기는 (필요한 패키지가 설치되지 않거나, R에서 데이터베이스에 접속이 되지 않는 등) 분석 도구 실행에 관계된 문제점이 생긴다. 참여 기관은 여러 상황을 직접적으로 처리하게 되고, 결국 연구를 실행할 때 발생하는 문제점을 해결하는 방법에 대해 의견을 나눌 것이다. 궁극적으로는 각자 CDM에서 발생하는 문제점을 해결하는 적절한 리소스를 찾는 절차는 참여하는 기관에서 고려해야 하는 사항이다.

OHDSI 연구의 실행이 신속하게 진행될 수 있더라도, 모든 참여 기관이 연구를 실행하고, 결과를 배포할 때 적절한 승인을 얻을 수 있는 충분한 시간을 주는 것이 바람직하다. 처음으로 OHDSI 네트워크에 참여하는 기관은 다양한 환경적인 요인, 예를 들어 데이터베이스 권한이나 분석 라이브러리 업데이트 문제와 같은 요인에 의해 처음 시행하는 연구에 참여하는 데 많은 시간이 소요된다. OHDSI 커뮤니티를 통해 여러 문제에 대한 지원을 받을 수 있다. OHDSI 포럼에 이슈를 제출할 수 있다.

연구 책임자는 연구 마일스톤을 프로토콜에 정하고 전반적인 연구 일정을 원활하게 하기 예상되는 마감일에 대해 의견을 나눠야 한다. 만약 연구 일정을 준수하지 않을 경우, 연구 책임자는 참여기관에 연구 일정 업데이트를 알려주고 연구 실행의 전반적인 상황을 관리하도록 한다.

20.3.3 결과의 보급과 출판

결과의 보급과 출판 단계에서는 연구 책임자는 보고서 작성과 데이터 시각화와 같은 다양한 업무에 대해 다른 참여자와 협력한다. 일단 연구가 시행되면, 연구 결과는 중앙에 저장되고 연구 책임자는 추가적인 분석을 할 수 있다. 연구 책임자는 참여기관의 연구 결과 검토를 위해 전체 연구 결과(Shiny 앱)를 작성하고 배포하도록 한다. 연구 책임자가 Atlas에서 생성된 코드 또는 GitHub code를 수동으로 수정한 OHDSI study skeleton를 사용하고 있으면, Shiny 앱이 자동으로 생성된다. 연구 책임자가 임의로 코드를 작성한 경우, 자신의 연구 패키지에 대한 shiny 앱을 생성하는데 문의하거나

도움을 얻기 위해 OHDSI 포럼에 문의할 수 있다.



자신의 OHDSI 네트워크 연구를 어디에 게재할지 결정하기 어렵다면, 초록과 출판물을 검색해서 가장 적절한 저널을 추천해주는 JANE(Journal/Author Name Estimator)을 사용하라.

¹

일단 논문이 작성되면, 모든 연구 참여자가 내용을 검토하고 외부 출판 과정에 이르는 결과를 확인하도록 한다. 최소한 개별 참여한 기관에서는 출판 책임자를 결정해야 한다(이 인원은 논문의 준비와 투고 과정에 내부적인 조정을 담당하게 된다). 어느 저널에 투고할지는 시작 단계에서 연구 참여자와 논의하는 것이 바람직하지만 연구자의 재량에 달려있다. OHDSI 연구에서 모든 공저자는 ICMJE 저자 가이드라인에 충족해야 한다.² 결과의 발표는 OHDSI 심포지엄, 다른 학술 심포지엄이나 논문 게재 등의 다양한 방법을 사용할 수 있다. 연구자는 OHDSI 네트워크 연구를 매주 열리는 OHDSI 커뮤니티 회의나 국제 OHDSI 심포지엄에서 발표하기 바란다.

20.4 미래의 모습: 네트워크 연구의 자동화

현재 네트워크 연구 방식은 수동이다. 즉, 연구팀 구성원이 다양한 방법 (wiki, GitHub, email)을 이용하여 연구 디자인, 코드와 결과 공유를 시행하고 있다. 이러한 방법은 일관적이지 못하고 확장성이 낮아, OHDSI 커뮤니티에서는 연구 프로세스를 체계화하기 위해 노력하고 있다.

ARACHNE는 네트워크 연구 과정을 간소화하고 자동화할 수 있도록 고안된 플랫폼이다. ARACHNE는 OHDSI 표준을 사용하여 여러 조직에서 일관되고, 투명하고, 안전하며, 준수하는 관찰 연구 프로세스를 설정한다. ARACHNE는 데이터 접근과 분석 결과 교환을 위한 통신 규약을 표준화하고 제한된 컨텐츠에 대한 인증 및 권한 부여를 가능하게 한다. 이것은 데이터 제공자, 연구자, 지원업체, 데이터 사이언티스트는 모든 참여 조직을 하나의 협동 연구 조직으로 만들 수 있고, 관찰 연구의 모든 단계에서 조정하는 역할을 하게 된다. 이 도구를 사용하면 데이터 관리자가 제어하는 작업을 포함하여 R, Python, SQL 기반 실행 환경을 만들 수 있다.

ARACHNE는 ACHILLES 보고서 및 ATLAS 디자인 인공물 artifact 가져오기, 자체 포함된 패키지 작성 및 여러 기관에서 자동으로 실행하는 기능을 포함하여 다른 OHDSI 도구와 완벽하게 통합되도록 설계되었다. 미래 비전은 단일 네트워크 내의 조직뿐만 아니라 여러 네트워크의 조직 간에도 연구를 수행할 목적으로 여러 네트워크를 서로 연결하는 것이다.

²<http://www.icmje.org/recommendations/browse/roles-and-responsibilities/defining-the-role-of-authors-and-contributors.html>

Network Study Workflow

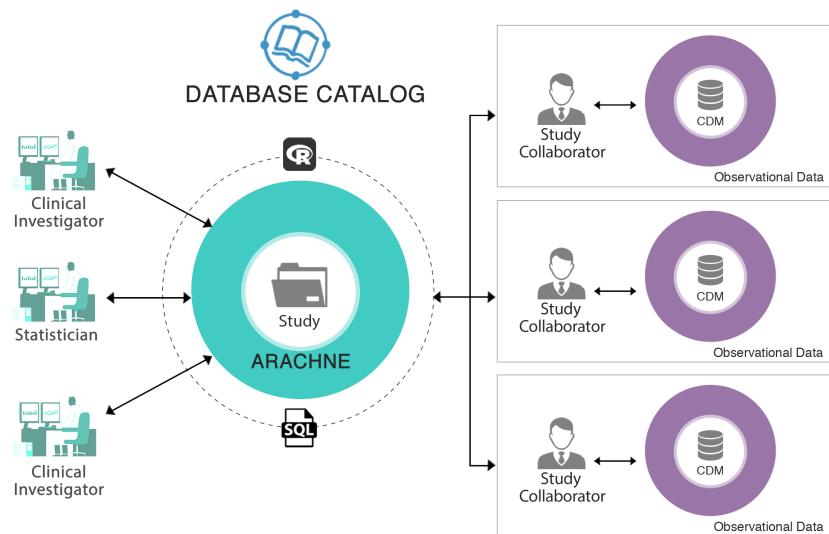


Figure 20.1: ARACHNE 네트워크 연구 과정.

20.5 OHDSI 네트워크 연구의 정석

네트워크 연구를 계획하고 있으며, 연구자가 OHDSI 네트워크 연구를 잘 진행할 수 있도록 OHDSI 커뮤니티는 도움을 줄 수 있다.

연구 설계와 타당도 네트워크 연구를 시행할 때, 자신의 연구가 한 형태의 데이터에 편향되어 있지 않은지 확인해야 한다. 모든 기관에서 동일한 대상을 표현할 수 있는 코호트 정의를 조율하는 과정은 데이터 형태의 이질성의 정도와 연구 기관에서 데이터를 OMOP CDM으로 변환할 때, 표준규약을 얼마나 잘 따랐는지에 따라 그 난이도가 결정될 것이다. 이 작업이 중요한 이유는 각 네트워크 기관과 실제 임상적으로 의미 있는 데이터 선택, 표현, 변환 간의 차이를 조정해야 하기 때문이다. 특히, 효과를 비교하는 연구에서는 각 데이터 기관 간에 일치된 노출 코호트와 결과 코호트 정의가 문제 될 수 있다. 예를 들어, 약물 노출 정도는 분류가 잘못되었을 가능성이 존재하는 데이터 원천에서 수집될 수 있다. 약국에서 수집된 약물 처방전의 경우, 약물에 대한 청구가 있을 때 환자가 처방을 받았을 가능성이 높다는 것을 의미한다. 그러나, EHR에 입력된 처방전은 약물의 실제 소비 여부를 확인하는 데이터와 연결이 되지 않는다. 또한, 의사가 처방전을 발급한 시간, 약사가 처방전에 따라 약을 조제한 시간, 약국에서 환자가 약을 수령한 시간, 실제로 약의 첫 복용이 일어난 시간 간의 차이가 존재한다. 이러한 측정 오류는 어떠한 연구를 하더라도 편향될 수 있다. 따라서, 연구 계획서를 개발할 때에는 데이터 참여 적절성을 고려하여 타당성 연구를 시행하는 것이 중요하다.

연구 실행 가능하면 연구 책임자가 ATLAS나 OHDSI 연구방법론 라이브러리 Method Library,

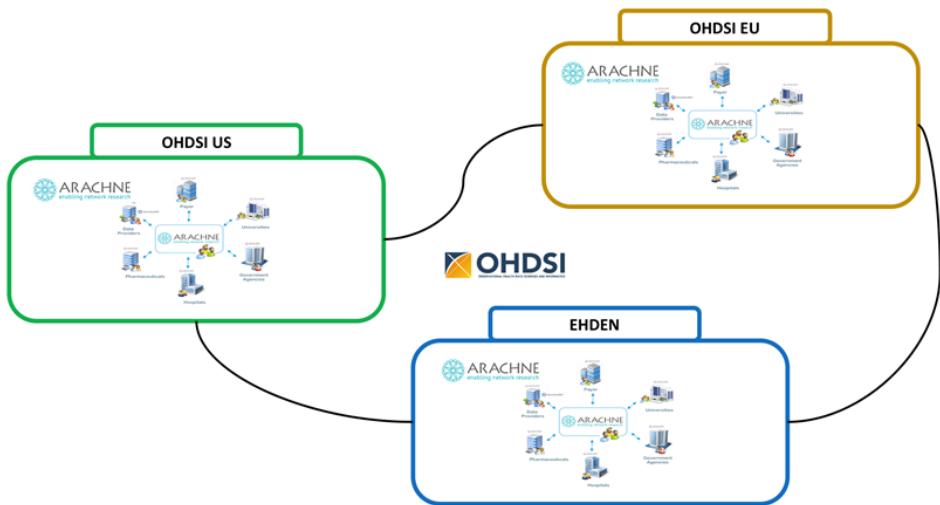


Figure 20.2: ARACHNE 네트워크의 네트워크.

OHDSI Study Skeleton을 이용하여 표준화된 분석 패키지를 사용하여 연구 코드를 작성하는 것을 권장한다. 연구 코드는 OHDSI 패키지를 이용하여 CDM에 호환성을 유지하고, 데이터베이스 레이어 규약에 따라 작성되어야 한다. 모든 기능과 변수는 매개 변수화해야 한다 (데이터베이스 연결 정보, 로컬 드라이브 경로, 운영체제를 지정하지 않는다). 참여 기관을 모집할 때는 연구 책임자는 각 참여 기관이 CDM 규약에 맞는지, 최신 OMOP 표준 용어집에 따라 업데이트되어 있는지 확인해야 한다. 연구 책임자는 각 네트워크 기관에서 CDM에 대해 데이터 품질 검사를 수행하고 문서화하도록 하고 이에 대한 점검을 해야 한다 (ETL 수행이 THEMIS 규약과 규칙에 따라서 올바른 CDM 테이블과 필드로 데이터가 배치되었는지 확인). 각 데이터 분석가는 연구 패키지는 실행하기 전에 R 패키지를 최신 OHDSI 패키지 버전으로 업데이트하도록 한다.

결과와 배포 연구 책임자는 결과를 공유하기 전에 각 기관이 각 기관의 규칙을 준수하도록 해야 한다. 연구가 개방적이고 재현 가능하다는 의미는 설계되고 실행되는 모든 것이 가능하다는 의미이다. OHDSI 네트워크 연구는 모든 문서와 결과가 OHDSI GitHub 저장소나 data.ohdsi.org R Shiny server에 게시되어 투명하게 관리된다. 논문을 준비할 때는 연구 책임자는 저널에서 OHDSI 네트워크 기관 간에 데이터가 어떻게 달라질 수 있는지 이해시킬 수 있도록 OMOP CDM과 표준화된 용어 원칙에 대해 언급을 해야 한다. 예를 들어, Claim 데이터베이스와 EHR을 이용한 네트워크 연구를 진행할 때에 저널 리뷰어는 다양한 데이터 형태에서 코호트 정의의 일관성을 유지할 수 있는지 설명을 요청할 수 있다. 리뷰어는 OMOP 관찰 기간 4장에서 언급된 바와 같이 자격 파일 (환자가 보험 자격 유지 기간에 있거나 있지 않은 상황에서 보험청구 데이터베이스에 존재하는 파일)과 비교하는 방법에 대해 궁금해할 수 있다. 이것은 본질적으로 데이터베이스 자체의 인위적인 요소에 중점을 두고 CDM이 자료를 관찰로 변환하는 방법의 ETL에 중점을 둔다. 이러한 경우 네트워크

연구 책임자는 OMOP CDM OBSERVATION PERIOD 작성 방법을 참조하고 원천 시스템에서 확인되는 상황을 이용하여 관찰기록이 작성되는 방법을 설명하는 것이 도움이 될 수 있다. 논문의 고찰 부분에서는 보험 기간에 모든 청구 내용을 반영하는 보험청구 데이터와는 달리 EHR 데이터의 경우는 환자가 다른 EHR을 사용하는 병원의 기록은 기록되지 않아서 관찰 기간의 중단이 발생할 수 있는 제한점에 대해 기술해야 한다. 이것은 데이터가 수집된 시스템에서 데이터가 존재하는 방식의 결과이다. 이것은 임상적으로 의미 있는 차이를 보이지는 않지만 OMOP에서 observation period table을 추출하는 방식에 익숙하지 않으면 혼동될 수 있다. 이러한 생소한 분야에 대해서 고찰 부분에서 언급하는 것이 필요하다. 비슷하게, 연구 책임자는 OMOP 표준 용어에서 제공되는 용어를 기술하는 것이 유용하며, 수집되는 모든 부분이 동일할 수 있다. 원천 코드를 표준 개념으로 매핑할 때 항상 결정이 이루어지지만 THEMIS 규칙과 CDM 품질 검사로서 정보 위치와 데이터베이스가 해당 원칙을 얼마나 잘 준수하는지에 대한 정보를 제공하는 것이 도움이 될 수 있다.

20.6 요약



- OHDSI 연구는 서로 다른 기관의 여러 CDM에서 실행될 때 OHDSI 네트워크 연구가 된다.
- OHDSI 네트워크 연구는 개방되어 있다. 누구나 네트워크 연구를 주도할 수 있다. OMOP 호환 데이터베이스를 소유한 사람은 누구나 참여하고 결과를 제공할 수 있다.
- 네트워크 연구를 하는 데 도움이 필요하면 연구를 디자인하고 실행하는 데 도움을 줄 수 있는 OHDSI 연구 육성 커뮤니티 study nurture committee 와 상의한다.
- **공유는 조심스럽게 시행한다.** 모든 연구 문서, 코드 및 결과는 OHDSI GitHub 또는 R Shiny 앱에 게시된다. 연구 책임자는 OHDSI 행사에 자신의 연구를 발표할 수 있도록 한다.

Chapter A

Glossary

ACHILLES 데이터베이스 수준의 특성 보고서.

ARACHNE 연합 네트워크의 조정 및 실행을 위해 개발 중인 OHDSI 플랫폼.

ATLAS 환자 수준 임상 데이터로부터 실제 근거를 생성하기 위한 관찰 분석의 실행과 설계를 지원하기 위하여 참여 사이트에 설치된 웹 기반 애플리케이션.

비뚤림 Bias 오류의 예상값 (실제 값과 예상값의 차이).

부울 Boolean 오직 두 개의 값만 가지는 변수 (참 혹은 거짓).

진료 장소 Care site 의료 서비스 제공을 실천하는 고유하게 식별된 기관 (물리적 혹은 조직적) 단위 (사무실 병동, 병원, 클리닉 등).

환자 대조군 Case control 인구-수준 효과 추정을 위한 회고 연구 설계 retrospective study design의 유형. 환자 대조군 연구는 표적 결과와의 “사례 cases”를 표적 결과가 없는 “대조 controls”로 일치시킨다. 그 후 이전의 시간을 살펴보고 사례와 대조의 노출 오즈 odds를 비교한다.

인과적 영향 Causal effect 인구 수준 추정이 그 자체로 얼마나 영향을 미치는가에 대한 것. 한 가지 정의는 “인과적 영향”을 대상 모집단 안 “단위 수준 unit-level 인과적 영향”의 평균으로 일치시킨다. 단위 수준 인과적 영향은 노출된 개인의 결과와 노출되지 않은 개인의 결과 간의 대조이다 (혹은 B에 대응하여 A에 노출된 적이 있는지).

특성 Characterization 코호트 혹은 전체 데이터베이스의 서술적 연구. 11장 참조.

청구 자료 Claims data 의료 보험을 회사에 청구하기 위한 목적으로 생성된 데이터

임상 시험 Clinical trial 중재 Interventional 임상 연구.

코호트 Cohort 특정 기간 동안 하나 혹은 다수 기준의 포함을 만족시키는 사람들의 집합. 10장 참조.

개념 Concept 의학 전문 용어에 정의된 용어 (코드 포함) (예를 들어, SNOMED CT). 5장 참조.

개념 모음 Concept set 개념 모음은 다양한 분석에서 재사용 가능한 요소들로 사용될 수 있는 개념의 목록을 나타내는 표현. 10장 참조.

공통 데이터 모델 Common Data Model(CDM) 분석의 이식성을 허용하는 의료 데이터를 대표하는 조약 (수정되지 않은 동일한 분석을 여러 데이터 모음에서 실행할 수 있다). 4장 참조.

비교 효과 Comparative Effectiveness 관심 결과에 대한 두 개의 다른 노출의 영향 비교. 12장 참조.

조건 Condition 제공자가 진찰하거나 환자가 보고한 진단, 기호, 혹은 증상.

교란 Confounding 교란은 주요 관심 노출이 결과와 관련된 몇 가지 다른 사실이 섞일 때 일어나는 연관성 예상 측정 안의 애곡 (불확실성)이다.

변수 Covariate 독립적 변수로 통계적 모델에 사용되는 데이터 요소 (예를 들어, 체중).

데이터 질 Data quality 데이터를 특정 용도에 적합하게 만드는 완전성, 유효성, 일관성, 적시성 및 정확성의 상태.

의료 기기 Device 화학 작용을 넘어서는 메커니즘을 통해 진단 또는 치료 목적으로 사용되는 대외 물리적 물체 혹은 기구. 의료 기기는 이식할 수 있는 물체 (예를 들면, 심박조정기, 스텐트, 인공 관절 등), 의료 장비 및 보급품 (예를 들어, 봉대, 목발, 주사기), 의료 절차에 사용되는 기구 (예를 들어, 봉함, 제세동기) 그리고 임상 치료에 사용되는 재료 (예를 들어, 접착제, body 재료, 치과용 재료, 수술용 재료)를 포함한다.

약물 Drug 약물은 사람에게 투여할 때 특정한 화학 반응을 발휘 할 수 있도록 조제된 생화학 물질이다. 약물은 처방전 및 처방전 없이 구매 가능한 약물, 백신, 그리고 대용량 molecule 생물학적 치료법을 포함한다. 국소 부위에 섭취되거나 접촉되는 방사선 기기는 약물로 간주하지 않는다.

도메인 Domain 도메인은 CDM 표의 표준화된 필드에 대해 허용하는 개념의 모음을 정의한다. 예를 들어, “조건” 도메인은 환자의 컨디션을 설명하는 개념을 포함하고, 이 개념은 CONDITION_OCCURRENCE와 CONDITION_ERA 테이블의 condition_concept_id 필드에만 저장된다.

전자 의무기록 Electronic Health Record(EHR) 전자기록 시스템에 기록되고 치료 기간 동안 생성된 데이터를 의미한다.

역학 Epidemiology 정의된 모집단의 건강 및 질병 조건의 분포, 패턴, 결정의 연구.

근거 중심 의학 Evidence-based medicine 개인의 환자 관리에 대한 의사 결정의 경험적, 과학적 근거의 사용.

추출 변환 적재 Extract-Transform-Load(ETL) 한 형식에서 다른 형식으로 데이터를 변환하는 과정, 예를 들어 원천 포맷을 CDM으로 변환한다. 6장 참조.

짝짓기 Matching 많은 인구-수준 효과 추정은 노출의 인과적 영향을 노출된 환자의 결과를 노출되지 않은 환자의 동일한 결과로 비교함으로써 식별하는 것을 시도한다 (혹은 A를 B의 대립으로 노출시킨다). 이 두 환자 집단은 노출 이외의 방식이 다를 수 있으므로, “짝짓기”는

최소한 측정된 환자 특성과 관련하여 최대한 비슷한 노출된 환자 그룹 및 노출되지 않은 환자 그룹을 생성하기 위해 시도한다.

측정 Measurement 개인 혹은 개인의 표본에 대한 체계적이고 표준화된 검사 혹은 학습을 통하여 얻은 구조적 값 (숫자 혹은 범주형).

측정 오차 Measurement error 기록된 측정 (혈압, 환자 연령, 치료 기간)이 해당 참 측정과 다를 때 발생한다.

메타데이터 Metadata 다른 데이터에 대한 정보를 제공하고 기술하는 데이터의 집단이다. 이것은 구체적 메타데이터, 구조적 메타데이터, 관리 메타데이터, 참조 메타데이터, 통계적 메타데이터를 포함한다.

연구방법론 라이브러리 Methods Library 관찰 연구를 수행하기 위하여 OHDSI 공동체에 의해 개발된 R 패키지의 집단.

Model misspecification 많은 OHDSI 방법은 비례 위험 회귀 또는 무작위 forest와 같은 통계적 모델을 채택한다. 데이터를 생성한 메커니즘이 가정된 모델에서 벗어나는 한, 모델은 “불특정”이 된다.

음성 통제 결과 Negative control 노출이 결과를 방지하거나 유발하지 않는다고 믿어지는 노출 결과 쌍. 효과 추정 방법이 생성하는 결과가 진실인지 여부를 판단하는데 사용될 수 있다. 18장 참조.

관찰 Observation 검사, 질의 혹은 절차의 내용에서 얻어진 개인에 대한 임상적 사실.

관찰 기간 Observation period 실제로 어떤 사건도 기록되지 않더라도 원천 시스템 내에서 임상 사건을 가질 위험이 있는 사람을 기록하는 시간 (보건의료 상호작용이 없는 건강한 환자).

관찰 연구 Observational study 연구자가 개입에 대한 통제력이 없는 연구.

OHDSI SQL SqlRender R 패키지를 사용하여 다른 다양한 SQL 언어로 자동 변환이 가능한 SQL 언어. OHDSI SQL은 대부분 SQL 서버 SQL의 하위 모음이지만, 추가적인 파라미터화를 허용한다. 9장 참조.

오픈 사이언스 Open science 과학적 연구와 (간행물, 데이터, 물리적 표본, 소프트웨어) 그것을 사회, 아마추어, 프로페셔널의 모든 계층이 접근 가능할 수 있게 보급할 수 있는 운동. 3장 참조.

결과 Outcome 분석의 초점을 제공하는 관찰. 예를 들어, 환자 수준 예측 모델은 결과 “stroke”를 예측할 수도 있다. 혹은 인구 수준 추정은 약물이 결과 “두통”에 미치는 인과적 영향을 예측할 수 있다.

환자 수준 예측 Patient-level prediction(PLP) 기저 특성에 기반한 몇 미래 결과의 경험의 환자 특정 가능성을 생성하기 위한 예측 모델의 애플리케이션과 개발.

표현형 Phenotype 물리적 특성에 대한 설명. 이것은 체중과 머리색 뿐만 아니라 전체 건강, 질병 기록 그리고 행동과 같은 시각적인 특성을 포함한다.

인구 수준 추정 Population-level estimation(PLE) 인과적 영향에 대한 연구. 평균 (인구 수준) 영향 크기를 추정한다.

양성 통제 결과 Negative control 노출이 결과를 유발하거나 방지하는 것으로 믿어지는 노출 결과 쌍. 효과 추정 방법이 생성하는 결과가 진실인지 여부를 판단하는데 사용될 수 있다. 18장 참조.

절차 Procedure 진단 혹은 치료 목적으로 의료 제공자가 환자에게 지시하거나 수행하는 활동 혹은 과정.

성향 점수 Propensity score(PS) 인구의 균형을 맞추기 위한 인구 수준 추정에 사용되는 단일 행렬. 이는 관찰 연구의 두 치료 그룹 간의 무작위화를 모방하기 위함이다. PS는 관심 치료를 받은 환자의 가능성을 관찰된 기저 공변량 모음의 기능으로 나타낸다. 이는 대부분 로지스틱 회귀분석을 사용하여 계산되는데, 이는 이항 결과가 표적 치료를 받는 집단은 1로, 대조군 치료의 경우 0으로 설정한다. 12장 참조.

프로토콜 Protocol 연구 설계를 완전히 지정하는 사람이 읽을 수 있는 문서.

Rabbit-in-a-Hat ETL을 원천 형식에서 CDM으로 정의하는 것을 도와주는 상호 소프트웨어 도구. White Rabbit으로 생성된 데이터베이스 개요를 입력값으로 사용한다. 7장 참조.

선택비뚤림 Selection bias 데이터 안의 환자 집합이 통계적 분석이 왜곡되는 방식으로 인구 안의 환자에게서 벗어날 때 발생하는 비뚤림.

자가제어 설계 Self-controlled designs 같은 환자 안에서 다른 노출이 진행되는 동안 결과를 비교하는 연구 설계.

민감도 분석 Sensitivity analysis 어떤 불확실성이 존재하는가에 대한 분석 선택의 영향을 평가하기 위한 연구에 사용되는 주요 분석의 변이

SNOMED 임상 문서와 보고에 사용되는 코드, 용어, 동의어와 정의를 제공하는 시스템적으로 조직적이고 체계적인 의학 용어 모음집.

연구 진단 Study diagnostics 주어진 분석 접근법이 주어진 연구 질문에 답하기 위해 사용될 수 있는가에 대한 여부를 결정하는 것이 목표인 분석 단계의 모음. 18장 참조.

연구 패키지 Study package 연구를 완전히 실행할 수 있는 컴퓨터 실행 프로그램. 17장 참조.

원천 코드 Source code 원천 데이터베이스에서 사용하는 코드. 예를 들어 ICD-10 코드가 있다.

표준 개념 Standard Concept 유효한 개념으로 설계되고 CDM에 나타날 수 있는 개념.

THEMIS CDM 모델 사양과 관련하여 더욱더 세밀하고 상세한 표적 데이터 형식을 설명하는 OHDSI 워크그룹.

방문 Visit 개인이 건강 관리 시스템 내 주어진 환경의 진료소에서 한 명 혹은 그 이상의 제공자로부터 의료 서비스를 지속해서 받는 기간.

용어 Vocabulary 단어의 목록과 종종 사용되는 구로, 보통 알파벳 순으로 정렬되거나 정의되고 혹은 번역된다. 5장 참조.

White Rabbit ETL을 CDM으로 정의하기 전에 데이터베이스를 수집하기 위한 소프트웨어 도구.
6장 참조.

Chapter B

Cohort definitions

이 책의 여러 장에 나온 코호트 정의를 부록에 상세히 기록하였다. ATLAS에서 구현을 따라하기 편하도록 가급적 영어 원문을 그대로 실었다.

B.1 ACE Inhibitor 사용 코호트

Initial Event Cohort

People having any of the following:

- a drug exposure of *ACE inhibitors* (Table B.1) for the first time in the person's history

with continuous observation of at least 365 days prior and 0 days after event index date, and limit initial events to: all events per person.

Limit qualifying cohort to: all events per person.

End Date Strategy

Custom Drug Era Exit Criteria This strategy creates a drug era from the codes found in the specified concept set. If the index event is found within an era, the cohort end date will use the era's end date. Otherwise, it will use the observation period end date that contains the index event.

Use the era end date of *ACE inhibitors* (Table B.1)

- allowing 30 days between exposures
- adding 0 days after exposure end

Cohort Collapse Strategy

Collapse cohort by era with a gap size of 30 days.

Concept Set Definitions

Table B.1: ACE inhibitors

Concept Id	Concept Name	Excluded	Descendants	Mapped
1308216	Lisinopril	NO	YES	NO
1310756	moexipril	NO	YES	NO
1331235	quinapril	NO	YES	NO
1334456	Ramipril	NO	YES	NO
1335471	benazepril	NO	YES	NO
1340128	Captopril	NO	YES	NO
1341927	Enalapril	NO	YES	NO
1342439	trandolapril	NO	YES	NO
1363749	Fosinopril	NO	YES	NO
1373225	Perindopril	NO	YES	NO

B.2 ACE inhibitor 단독 요법의 새로운 사용자 코호트 New User

Initial Event Cohort

People having any of the following:

- a drug exposure of *ACE inhibitors* (Table B.2) for the first time in the person's history

with continuous observation of at least 365 days prior and 0 days after event index date, and limit initial events to: earliest event per person.

Inclusion Rules

Inclusion Criteria #1: has hypertension diagnosis in 1 yr prior to treatment

Having all of the following criteria:

- at least 1 occurrences of a condition occurrence of *Hypertensive disorder* (Table B.3) where event starts between 365 days Before and 0 days After index start date

Inclusion Criteria #2: Has no prior antihypertensive drug exposures in medical history

Having all of the following criteria:

- exactly 0 occurrences of a drug exposure of *Hypertension drugs* (Table B.4) where event starts between all days Before and 1 days Before index start date

Inclusion Criteria #3: Is only taking ACE as monotherapy, with no concomitant combination treatments

Having all of the following criteria:

- exactly 1 distinct occurrences of a drug era of *Hypertension drugs* (Table B.4) where event starts between 0 days Before and 7 days After index start date

Limit qualifying cohort to: earliest event per person.

End Date Strategy

Custom Drug Era Exit Criteria. This strategy creates a drug era from the codes found in the specified concept set. If the index event is found within an era, the cohort end date will use the era's end date. Otherwise, it will use the observation period end date that contains the index event.

Use the era end date of *ACE inhibitors* (Table B.2)

- allowing 30 days between exposures
- adding 0 days after exposure end

Cohort Collapse Strategy

Collapse cohort by era with a gap size of 0 days.

Concept Set Definitions

Table B.2: ACE inhibitors

Concept Id	Concept Name	Excluded	Descendants	Mapped
1308216	Lisinopril	NO	YES	NO
1310756	moexipril	NO	YES	NO
1331235	quinapril	NO	YES	NO
1334456	Ramipril	NO	YES	NO
1335471	benazepril	NO	YES	NO
1340128	Captopril	NO	YES	NO
1341927	Enalapril	NO	YES	NO
1342439	trandolapril	NO	YES	NO
1363749	Fosinopril	NO	YES	NO
1373225	Perindopril	NO	YES	NO

Table B.3: Hypertensive disorder

Concept Id	Concept Name	Excluded	Descendants	Mapped
316866	Hypertensive disorder	NO	YES	NO

Table B.4: Hypertension drugs

Concept Id	Concept Name	Excluded	Descendants	Mapped
904542	Triamterene	NO	YES	NO
907013	Metolazone	NO	YES	NO
932745	Bumetanide	NO	YES	NO
942350	torsemide	NO	YES	NO
956874	Furosemide	NO	YES	NO
970250	Spiromolactone	NO	YES	NO
974166	Hydrochlorothiazide	NO	YES	NO
978555	Indapamide	NO	YES	NO
991382	Amiloride	NO	YES	NO
1305447	Methyldopa	NO	YES	NO
1307046	Metoprolol	NO	YES	NO

Concept Id	Concept Name	Excluded	Descendants	Mapped
1307863	Verapamil	NO	YES	NO
1308216	Lisinopril	NO	YES	NO
1308842	valsartan	NO	YES	NO
1309068	Minoxidil	NO	YES	NO
1309799	eplerenone	NO	YES	NO
1310756	moexipril	NO	YES	NO
1313200	Nadolol	NO	YES	NO
1314002	Atenolol	NO	YES	NO
1314577	nebivolol	NO	YES	NO
1317640	telmisartan	NO	YES	NO
1317967	aliskiren	NO	YES	NO
1318137	Nicardipine	NO	YES	NO
1318853	Nifedipine	NO	YES	NO
1319880	Nisoldipine	NO	YES	NO
1319998	Acebutolol	NO	YES	NO
1322081	Betaxolol	NO	YES	NO
1326012	Isradipine	NO	YES	NO
1327978	Penbutolol	NO	YES	NO
1328165	Diltiazem	NO	YES	NO
1331235	quinapril	NO	YES	NO
1332418	Amlodipine	NO	YES	NO
1334456	Ramipril	NO	YES	NO
1335471	benazepril	NO	YES	NO
1338005	Bisoprolol	NO	YES	NO
1340128	Captopril	NO	YES	NO
1341238	Terazosin	NO	YES	NO
1341927	Enalapril	NO	YES	NO
1342439	trandolapril	NO	YES	NO
1344965	Guanfacine	NO	YES	NO
1345858	Pindolol	NO	YES	NO
1346686	eprosartan	NO	YES	NO
1346823	carvedilol	NO	YES	NO

Concept Id	Concept Name	Excluded	Descendants	Mapped
1347384	irbesartan	NO	YES	NO
1350489	Prazosin	NO	YES	NO
1351557	candesartan	NO	YES	NO
1353766	Propranolol	NO	YES	NO
1353776	Felodipine	NO	YES	NO
1363053	Doxazosin	NO	YES	NO
1363749	Fosinopril	NO	YES	NO
1367500	Losartan	NO	YES	NO
1373225	Perindopril	NO	YES	NO
1373928	Hydralazine	NO	YES	NO
1386957	Labetalol	NO	YES	NO
1395058	Chlorthalidone	NO	YES	NO
1398937	Clonidine	NO	YES	NO
40226742	olmesartan	NO	YES	NO
40235485	azilsartan	NO	YES	NO

B.3 심근경색 Acute myocardial infarction(AMI) 코호트

Initial Event Cohort

People having any of the following:

- a condition occurrence of *Acute myocardial Infarction* (Table B.5)

with continuous observation of at least 0 days prior and 0 days after event index date, and limit initial events to: all events per person.

For people matching the Primary Events, include: Having any of the following criteria:

- at least 1 occurrences of a visit occurrence of *Inpatient or ER visit* (Table B.6) where event starts between all days Before and 0 days After index start date and event ends between 0 days Before and all days After index start date

Limit cohort of initial events to: all events per person.

Limit qualifying cohort to: all events per person.

End Date Strategy

Date Offset Exit Criteria. This cohort definition end date will be the index event's start date plus 7 days

Cohort Collapse Strategy

Collapse cohort by era with a gap size of 180 days.

Concept Set Definitions

Table B.5: Inpatient or ER visit

Concept Id	Concept Name	Excluded	Descendants	Mapped
314666	Old myocardial infarction	YES	YES	NO
4329847	Myocardial infarction	NO	YES	NO

Table B.6: Inpatient or ER visit

Concept Id	Concept Name	Excluded	Descendants	Mapped
262	Emergency Room and Inpatient Visit	NO	YES	NO
9201	Inpatient Visit	NO	YES	NO
9203	Emergency Room Visit	NO	YES	NO

B.4 혈관부종 *Angioedema*

Initial Event Cohort

People having any of the following:

- a condition occurrence of *Angioedema* (Table B.7)

with continuous observation of at least 0 days prior and 0 days after event index date, and limit initial events to: all events per person.

For people matching the Primary Events, include: Having any of the following criteria:

- at least 1 occurrences of a visit occurrence of *Inpatient or ER visit* (Table B.8) where event starts between all days Before and 0 days After index start date and event ends between 0 days Before and all days After index start date

Limit cohort of initial events to: all events per person.

Limit qualifying cohort to: all events per person.

End Date Strategy

This cohort definition end date will be the index event's start date plus 7 days

Cohort Collapse Strategy

Collapse cohort by era with a gap size of 30 days.

Concept Set Definitions

Table B.7: Angioedema

Concept Id	Concept Name	Excluded	Descendants	Mapped
432791	Angioedema	NO	YES	NO

Table B.8: Inpatient or ER visit

Concept Id	Concept Name	Excluded	Descendants	Mapped
262	Emergency Room and Inpatient Visit	NO	YES	NO
9201	Inpatient Visit	NO	YES	NO
9203	Emergency Room Visit	NO	YES	NO

B.5 Thiazide 계 이뇨제 Thiazide-Like Diuretics 단독 요법 새로운 사용자 New User 코호트

Initial Event Cohort

People having any of the following:

- a drug exposure of *Thiazide or thiazide-like diuretic* (Table B.9) for the first time in the person's history

with continuous observation of at least 365 days prior and 0 days after event index date, and limit initial events to: earliest event per person.

Inclusion Rules

Inclusion Criteria #1: has hypertension diagnosis in 1 yr prior to treatment

Having all of the following criteria:

- at least 1 occurrences of a condition occurrence of *Hypertensive disorder* (Table B.10) where event starts between 365 days Before and 0 days After index start date

Inclusion Criteria #2: Has no prior antihypertensive drug exposures in medical history

Having all of the following criteria:

- exactly 0 occurrences of a drug exposure of *Hypertension drugs* (Table B.11) where event starts between all days Before and 1 days Before index start date

Inclusion Criteria #3: Is only taking ACE as monotherapy, with no concomitant combination treatments

Having all of the following criteria:

- exactly 1 distinct occurrences of a drug era of *Hypertension drugs* (Table B.11) where event starts between 0 days Before and 7 days After index start date

Limit qualifying cohort to: earliest event per person.

End Date Strategy

Custom Drug Era Exit Criteria. This strategy creates a drug era from the codes found in the specified concept set. If the index event is found within an era, the cohort end date will use the era's end date. Otherwise, it will use the observation period end date that contains the index event.

Use the era end date of *Thiazide or thiazide-like diuretic* (Table B.9)

- allowing 30 days between exposures

- adding 0 days after exposure end

Cohort Collapse Strategy

Collapse cohort by era with a gap size of 0 days.

Concept Set Definitions

Table B.9: Thiazide or thiazide-like diuretic

Concept Id	Concept Name	Excluded	Descendants	Mapped
907013	Metolazone	NO	YES	NO
974166	Hydrochlorothiazide	NO	YES	NO
978555	Indapamide	NO	YES	NO
1395058	Chlorthalidone	NO	YES	NO

Table B.10: Hypertensive disorder

Concept Id	Concept Name	Excluded	Descendants	Mapped
316866	Hypertensive disorder	NO	YES	NO

Table B.11: Hypertension drugs

Concept Id	Concept Name	Excluded	Descendants	Mapped
904542	Triamterene	NO	YES	NO
907013	Metolazone	NO	YES	NO
932745	Bumetanide	NO	YES	NO
942350	torsemide	NO	YES	NO
956874	Furosemide	NO	YES	NO
970250	Spironolactone	NO	YES	NO
974166	Hydrochlorothiazide	NO	YES	NO
978555	Indapamide	NO	YES	NO
991382	Amiloride	NO	YES	NO
1305447	Methyldopa	NO	YES	NO

Concept Id	Concept Name	Excluded	Descendants	Mapped
1307046	Metoprolol	NO	YES	NO
1307863	Verapamil	NO	YES	NO
1308216	Lisinopril	NO	YES	NO
1308842	valsartan	NO	YES	NO
1309068	Minoxidil	NO	YES	NO
1309799	eplerenone	NO	YES	NO
1310756	moexipril	NO	YES	NO
1313200	Nadolol	NO	YES	NO
1314002	Atenolol	NO	YES	NO
1314577	nebivolol	NO	YES	NO
1317640	telmisartan	NO	YES	NO
1317967	aliskiren	NO	YES	NO
1318137	Nicardipine	NO	YES	NO
1318853	Nifedipine	NO	YES	NO
1319880	Nisoldipine	NO	YES	NO
1319998	Acebutolol	NO	YES	NO
1322081	Betaxolol	NO	YES	NO
1326012	Isradipine	NO	YES	NO
1327978	Penbutolol	NO	YES	NO
1328165	Diltiazem	NO	YES	NO
1331235	quinapril	NO	YES	NO
1332418	Amlodipine	NO	YES	NO
1334456	Ramipril	NO	YES	NO
1335471	benazepril	NO	YES	NO
1338005	Bisoprolol	NO	YES	NO
1340128	Captopril	NO	YES	NO
1341238	Terazosin	NO	YES	NO
1341927	Enalapril	NO	YES	NO
1342439	trandolapril	NO	YES	NO
1344965	Guanfacine	NO	YES	NO
1345858	Pindolol	NO	YES	NO
1346686	eprosartan	NO	YES	NO

Concept Id	Concept Name	Excluded	Descendants	Mapped
1346823	carvedilol	NO	YES	NO
1347384	irbesartan	NO	YES	NO
1350489	Prazosin	NO	YES	NO
1351557	candesartan	NO	YES	NO
1353766	Propranolol	NO	YES	NO
1353776	Felodipine	NO	YES	NO
1363053	Doxazosin	NO	YES	NO
1363749	Fosinopril	NO	YES	NO
1367500	Losartan	NO	YES	NO
1373225	Perindopril	NO	YES	NO
1373928	Hydralazine	NO	YES	NO
1386957	Labetalol	NO	YES	NO
1395058	Chlorthalidone	NO	YES	NO
1398937	Clonidine	NO	YES	NO
40226742	olmesartan	NO	YES	NO
40235485	azilsartan	NO	YES	NO

B.6 고혈압 일차 요법 사용자 코호트

Initial Event Cohort

People having any of the following:

- a drug exposure of *First-line hypertension drugs* (Table B.12) for the first time in the person's history

with continuous observation of at least 365 days prior and 365 days after event index date, and limit initial events to: earliest event per person.

Inclusion Rules

Having all of the following criteria:

- exactly 0 occurrences of a drug exposure of *Hypertension drugs* (Table B.13) where event starts between all days Before and 1 days Before index start date

- and at least 1 occurrences of a condition occurrence of *Hypertensive disorder* (Table B.14) where event starts between 365 days Before and 0 days After index start date

Limit cohort of initial events to: earliest event per person. Limit qualifying cohort to: earliest event per person.

End Date Strategy

No end date strategy selected. By default, the cohort end date will be the end of the observation period that contains the index event.

Cohort Collapse Strategy

Collapse cohort by era with a gap size of 0 days.

Concept Set Definitions

Table B.12: First-line hypertension drugs

Concept Id	Concept Name	Excluded	Descendants	Mapped
907013	Metolazone	NO	YES	NO
974166	Hydrochlorothiazide	NO	YES	NO
978555	Indapamide	NO	YES	NO
1307863	Verapamil	NO	YES	NO
1308216	Lisinopril	NO	YES	NO
1308842	valsartan	NO	YES	NO
1310756	moexipril	NO	YES	NO
1317640	telmisartan	NO	YES	NO
1318137	Nicardipine	NO	YES	NO
1318853	Nifedipine	NO	YES	NO
1319880	Nisoldipine	NO	YES	NO
1326012	Isradipine	NO	YES	NO
1328165	Diltiazem	NO	YES	NO
1331235	quinapril	NO	YES	NO
1332418	Amlodipine	NO	YES	NO
1334456	Ramipril	NO	YES	NO
1335471	benazepril	NO	YES	NO

Concept Id	Concept Name	Excluded	Descendants	Mapped
1340128	Captopril	NO	YES	NO
1341927	Enalapril	NO	YES	NO
1342439	trandolapril	NO	YES	NO
1346686	eprosartan	NO	YES	NO
1347384	irbesartan	NO	YES	NO
1351557	candesartan	NO	YES	NO
1353776	Felodipine	NO	YES	NO
1363749	Fosinopril	NO	YES	NO
1367500	Losartan	NO	YES	NO
1373225	Perindopril	NO	YES	NO
1395058	Chlorthalidone	NO	YES	NO
40226742	olmesartan	NO	YES	NO
40235485	azilsartan	NO	YES	NO

Table B.13: Hypertension drugs

Concept Id	Concept Name	Excluded	Descendants	Mapped
904542	Triamterene	NO	YES	NO
907013	Metolazone	NO	YES	NO
932745	Bumetanide	NO	YES	NO
942350	torsemide	NO	YES	NO
956874	Furosemide	NO	YES	NO
970250	Spironolactone	NO	YES	NO
974166	Hydrochlorothiazide	NO	YES	NO
978555	Indapamide	NO	YES	NO
991382	Amiloride	NO	YES	NO
1305447	Methyldopa	NO	YES	NO
1307046	Metoprolol	NO	YES	NO
1307863	Verapamil	NO	YES	NO
1308216	Lisinopril	NO	YES	NO
1308842	valsartan	NO	YES	NO
1309068	Minoxidil	NO	YES	NO

Concept Id	Concept Name	Excluded	Descendants	Mapped
1309799	eplerenone	NO	YES	NO
1310756	moexipril	NO	YES	NO
1313200	Nadolol	NO	YES	NO
1314002	Atenolol	NO	YES	NO
1314577	nebivolol	NO	YES	NO
1317640	telmisartan	NO	YES	NO
1317967	aliskiren	NO	YES	NO
1318137	Nicardipine	NO	YES	NO
1318853	Nifedipine	NO	YES	NO
1319880	Nisoldipine	NO	YES	NO
1319998	Acebutolol	NO	YES	NO
1322081	Betaxolol	NO	YES	NO
1326012	Isradipine	NO	YES	NO
1327978	Penbutolol	NO	YES	NO
1328165	Diltiazem	NO	YES	NO
1331235	quinapril	NO	YES	NO
1332418	Amlodipine	NO	YES	NO
1334456	Ramipril	NO	YES	NO
1335471	benazepril	NO	YES	NO
1338005	Bisoprolol	NO	YES	NO
1340128	Captopril	NO	YES	NO
1341238	Terazosin	NO	YES	NO
1341927	Enalapril	NO	YES	NO
1342439	trandolapril	NO	YES	NO
1344965	Guanfacine	NO	YES	NO
1345858	Pindolol	NO	YES	NO
1346686	eprosartan	NO	YES	NO
1346823	carvedilol	NO	YES	NO
1347384	irbesartan	NO	YES	NO
1350489	Prazosin	NO	YES	NO
1351557	candesartan	NO	YES	NO
1353766	Propranolol	NO	YES	NO

Concept Id	Concept Name	Excluded	Descendants	Mapped
1353776	Felodipine	NO	YES	NO
1363053	Doxazosin	NO	YES	NO
1363749	Fosinopril	NO	YES	NO
1367500	Losartan	NO	YES	NO
1373225	Perindopril	NO	YES	NO
1373928	Hydralazine	NO	YES	NO
1386957	Labetalol	NO	YES	NO
1395058	Chlorthalidone	NO	YES	NO
1398937	Clonidine	NO	YES	NO
40226742	olmesartan	NO	YES	NO
40235485	azilsartan	NO	YES	NO

Table B.14: Hypertensive disorder

Concept Id	Concept Name	Excluded	Descendants	Mapped
316866	Hypertensive disorder	NO	YES	NO

B.7 3년 이상 follow-up 된 고혈압 일차 요법 사용자 코호트

B.6 코호트 정의에서 continuous observation 조건만 “continuous observation of at least 365 days prior and 1095 days after event index date” 으로 바꾼다.

B.8 ACE Inhibitor 사용자 코호트

Initial Event Cohort

People having any of the following:

- a drug exposure of *ACE inhibitors* (Table B.15)

with continuous observation of at least 0 days prior and 0 days after event index date, and limit initial events to: all events per person.

Limit qualifying cohort to: all events per person.

End Date Strategy

This strategy creates a drug era from the codes found in the specified concept set. If the index event is found within an era, the cohort end date will use the era's end date. Otherwise, it will use the observation period end date that contains the index event.

Use the era end date of *ACE inhibitors* (Table B.15)

- allowing 30 days between exposures
- adding 0 days after exposure end

Cohort Collapse Strategy

Collapse cohort by era with a gap size of 30 days.

Concept Set Definitions

Table B.15: ACE inhibitors

Concept Id	Concept Name	Excluded	Descendants	Mapped
1308216	Lisinopril	NO	YES	NO
1310756	moexipril	NO	YES	NO
1331235	quinapril	NO	YES	NO
1334456	Ramipril	NO	YES	NO
1335471	benazepril	NO	YES	NO
1340128	Captopril	NO	YES	NO
1341927	Enalapril	NO	YES	NO
1342439	trandolapril	NO	YES	NO
1363749	Fosinopril	NO	YES	NO
1373225	Perindopril	NO	YES	NO

B.9 Angiotensin Receptor Blocker (ARB) 사용자 코호트

B.8 코호트 정의에서 *ACE inhibitors* (표 B.15)를 *Angiotensin Receptor Blockers (ARBs)* (표 B.16)으로 바꾼다.

Concept Set Definitions

Table B.16: Angiotensin Receptor Blockers (ARBs)

Concept Id	Concept Name	Excluded	Descendants	Mapped
1308842	valsartan	NO	YES	NO
1317640	telmisartan	NO	YES	NO
1346686	eprosartan	NO	YES	NO
1347384	irbesartan	NO	YES	NO
1351557	candesartan	NO	YES	NO
1367500	Losartan	NO	YES	NO
40226742	olmesartan	NO	YES	NO
40235485	azilsartan	NO	YES	NO

B.10 Thiazide 이뇨제 사용자 코호트

B.8 코호트 정의에서 ACE inhibitors (표 B.15)를 Thiazide or thiazide-like diuretic (표 B.17)으로 바꾼다.

Concept Set Definitions

Table B.17: Thiazide or thiazide-like diuretic

Concept Id	Concept Name	Excluded	Descendants	Mapped
907013	Metolazone	NO	YES	NO
974166	Hydrochlorothiazide	NO	YES	NO
978555	Indapamide	NO	YES	NO
1395058	Chlorthalidone	NO	YES	NO

B.11 Dihydropyridine Calcium Channel Blocker (dCCB) 사용자 코호트

B.8 코호트 정의에서 ACE inhibitors (표 B.15)를 dihydropyridine Calcium Channel Blocker (dCCB) (표 B.18)으로 바꾼다.

Concept Set Definitions

Table B.18: Dihydropyridine Calcium channel blockers (dCCB)

Concept Id	Concept Name	Excluded	Descendants	Mapped
1318137	Nicardipine	NO	YES	NO
1318853	Nifedipine	NO	YES	NO
1319880	Nisoldipine	NO	YES	NO
1326012	Isradipine	NO	YES	NO
1332418	Amlodipine	NO	YES	NO
1353776	Felodipine	NO	YES	NO

B.12 Non-Dihydropyridine Calcium Channel Blocker (ndCCB) 사용자 코호트

B.8 코호트 정의에서 ACE inhibitors (표 B.15)를 non-dihydropyridine Calcium channel blockers (ndCCB) (표 B.19)으로 바꾼다.

Concept Set Definitions

Table B.19: non-dihydropyridine Calcium channel blockers (ndCCB)

Concept Id	Concept Name	Excluded	Descendants	Mapped
1307863	Verapamil	NO	YES	NO
1328165	Diltiazem	NO	YES	NO

B.13 Beta-Blocker Use

B.8 코호트 정의에서 ACE inhibitors (표 B.15)를 Beta blockers (표 B.20)으로 바꾼다.

Concept Set Definitions

Table B.20: Beta blockers

Concept Id	Concept Name	Excluded	Descendants	Mapped
1307046	Metoprolol	NO	YES	NO
1313200	Nadolol	NO	YES	NO
1314002	Atenolol	NO	YES	NO
1314577	nebivolol	NO	YES	NO
1319998	Acebutolol	NO	YES	NO
1322081	Betaxolol	NO	YES	NO
1327978	Penbutolol	NO	YES	NO
1338005	Bisoprolol	NO	YES	NO
1345858	Pindolol	NO	YES	NO
1346823	carvedilol	NO	YES	NO
1353766	Propranolol	NO	YES	NO
1386957	Labetalol	NO	YES	NO

B.14 Loop 이뇨제 사용자 코호트

B.8 코호트 정의에서 ACE inhibitors (표 B.15)를 Diuretics - Loop (표 B.21)으로 바꾼다.

Concept Set Definitions

Table B.21: Diuretics - Loop

Concept Id	Concept Name	Excluded	Descendants	Mapped
932745	Bumetanide	NO	YES	NO
942350	torsemide	NO	YES	NO
956874	Furosemide	NO	YES	NO

B.15 칼륨보존 이뇨제 사용자 코호트

B.8 코호트 정의에서 ACE inhibitors (표 B.15)를 Diuretics - potassium sparing (표 B.22)으로 바꾼다.

Concept Set Definitions

Table B.22: Diuretics - potassium sparing

Concept Id	Concept Name	Excluded	Descendants	Mapped
904542	Triamterene	NO	YES	NO
991382	Amiloride	NO	YES	NO

B.16 Alpha-1 길항제 사용자 코호트

B.8 코호트 정의에서 *ACE inhibitors* (표 B.15)를 *Alpha-1 blocker* (표 B.23)으로 바꾼다.

Concept Set Definitions

Table B.23: Alpha-1 blocker

Concept Id	Concept Name	Excluded	Descendants	Mapped
1341238	Terazosin	NO	YES	NO
1350489	Prazosin	NO	YES	NO
1363053	Doxazosin	NO	YES	NO

Chapter C

Negative controls

이 책의 여러 장에 나온 음성 대조군 Negative controls 예시이다.

C.1 ACE inhibitor 와 thiazide 계 이뇨제

Table C.1: ACE inhibitors (ACEi) 와 thiazide 계 이뇨제(THZ) 비교시 사용한 음성 대조군.

Concept ID	Concept Name
434165	Abnormal cervical smear
436409	Abnormal pupil
199192	Abrasions and/or friction burn of trunk without infection
4088290	Absence of breast
4092879	Absent kidney
44783954	Acid reflux
75911	Acquired hallux valgus
137951	Acquired keratoderma
77965	Acquired trigger finger
376707	Acute conjunctivitis
4103640	Amputated foot
73241	Anal and rectal polyp

Concept ID	Concept Name
133655	Burn of forearm
73560	Calcaneal spur
434327	Cannabis abuse
4213540	Cervical somatic dysfunction
140842	Changes in skin texture
81378	Chondromalacia of patella
432303	Cocaine abuse
4201390	Colostomy present
46269889	Complication due to Crohn's disease
134438	Contact dermatitis
78619	Contusion of knee
201606	Crohn's disease
76786	Derangement of knee
4115402	Difficulty sleeping
45757370	Disproportion of reconstructed breast
433111	Effects of hunger
433527	Endometriosis
4170770	Epidermoid cyst
4092896	Feces contents abnormal
259995	Foreign body in orifice
40481632	Ganglion cyst
4166231	Genetic predisposition
433577	Hammer toe
4231770	Hereditary thrombophilia
440329	Herpes zoster without complication
4012570	High risk sexual behavior
4012934	Homocystinuria
441788	Human papilloma virus infection
4201717	Ileostomy present
374375	Impacted cerumen
4344500	Impingement syndrome of shoulder region
139099	Ingrowing nail

Concept ID	Concept Name
444132	Injury of knee
196168	Irregular periods
432593	Kwashiorkor
434203	Late effect of contusion
438329	Late effect of motor vehicle accident
195873	Leukorrhea
4083487	Macular drusen
4103703	Melena
4209423	Nicotine dependence
377572	Noise effects on inner ear
40480893	Nonspecific tuberculin test reaction
136368	Non-toxic multinodular goiter
140648	Onychomycosis due to dermatophyte
438130	Opioid abuse
4091513	Passing flatus
4202045	Postviral fatigue syndrome
373478	Presbyopia
46286594	Problem related to lifestyle
439790	Psychalgia
81634	Ptotic breast
380706	Regular astigmatism
141932	Senile hyperkeratosis
36713918	Somatic dysfunction of lumbar region
443172	Splinter of face, without major open wound
81151	Sprain of ankle
72748	Strain of rotator cuff capsule
378427	Tear film insufficiency
437264	Tobacco dependence syndrome
194083	Vaginitis and vulvovaginitis
140641	Verruca vulgaris
440193	Wristdrop
4115367	Wrist joint pain

Chapter D

Protocol template

1. Table of contents
2. List of abbreviations
3. Abstract
4. Amendments and Updates
5. Milestones
6. Rationale and Background
7. Study Objectives
 - Primary Hypotheses
 - Secondary Hypotheses
 - Primary Objectives
 - Secondary Objectives
8. Research methods
 - Study Design

- Data Source(s)
- Study population
- Exposures
- Outcomes
- Covariates

9. Data Analysis Plan

- Calculation of time-at risk
- Model Specification
- Pooling effect estimates across databases
- Analyses to perform
- Output
- Evidence Evaluation

10. Study Diagnostics

- Sample Size and Study Power
- Cohort Comparability
- Systematic Error Assessment

11. Strengths and Limitations of the Research Methods

12. Protection of Human Subjects

13. Management and Reporting of Adverse Events and Adverse Reactions

14. Plans for Disseminating and Communicating Study Results
15. Appendix: Negative controls
16. References

Chapter E

Suggested Answers

이 부록은 이 책의 예제에 대한 제안된 답변을 포함한다.

E.1 공통 데이터 모델

예제 4.1

예제의 설명에 기반하여, John의 기록은 표 E.1처럼 보여야 한다.

Table E.1: The PERSON table.

Column name	Value	Explanation
PERSON_ID	2	A unique integer.
GENDER_CONCEPT_ID	8507	The concept ID for male gender is 8507.
YEAR_OF_BIRTH	1974	
MONTH_OF_BIRTH	8	
DAY_OF_BIRTH	4	
BIRTH_DATETIME	1974-08-04 00:00:00	When the time is not known midnight is used.
DEATH_DATETIME	NULL	
RACE_CONCEPT_ID	8516	The concept ID for black or African American is 8516.

Column name	Value	Explanation
ETHNICITY_	38003564	38003564 refers to “Not hispanic”.
CONCEPT_ID		
LOCATION_ID		His address is not known.
PROVIDER_ID		His primary care Provider is not known.
CARE_SITE		His primary Care Site is not known.
PERSON_SOURCE_	NULL	Not provided.
VALUE		
GENDER_SOURCE_	Man	The text used in the description.
VALUE		
GENDER_SOURCE_	0	
CONCEPT_ID		
RACE_SOURCE_	African	The text used in the description.
VALUE	American	
RACE_SOURCE_	0	
CONCEPT_ID		
ETHNICITY_SOURCE_	NULL	
VALUE		
ETHNICITY_SOURCE_	0	
CONCEPT_ID		

예제 4.2

예제의 설명에 기반하여, John의 기록은 표 E.2 처럼 보여야 한다.

Table E.2: The OBSERVATION_PERIOD table.

Column name	Value	Explanation
OBSERVATION_	2	A unique integer.
PERIOD_ID		
PERSON_ID	2	This is a foreign key to John’s record in the PERSON table.
OBSERVATION_PERIOD_	2015-01-01	The date of enrollment.
START_DATE		

Column name	Value	Explanation
OBSERVATION_PERIOD_	2019-07-01	No data can be expected after the data extraction date.
END_DATE		
PERIOD_TYPE_	44814722	44814724 refers to “Period while enrolled in insurance”.
CONCEPT_ID		

예제 4.3

예제의 설명에 기반하여, John의 기록은 표 E.3처럼 보여야 한다.

Table E.3: The DRUG_EXPOSURE table.

Column name	Value	Explanation
DRUG_EXPOSURE_ID	1001	Some unique integer
PERSON_ID	2	This is a foreign key to John’s record in the PERSON table.
DRUG_CONCEPT_ID	19078461	The provided NDC code maps to Standard Concept 19078461.
DRUG_EXPOSURE_	2019-05-01	The start date of the exposure to the drug.
START_DATE		
DRUG_EXPOSURE_	2019-05-01	Midnight is used as the time is not known.
START_DATETIME	00:00:00	
DRUG_EXPOSURE_	2019-05-31	Based on start date + days supply.
END_DATE		
DRUG_EXPOSURE_	2019-05-31	Midnight is used as time is unknown.
END_DATETIME	00:00:00	
VERBATIM_END_DATE	NULL	Not provided.
DRUG_TYPE_	38000177	38000177 indicates “Prescription written”.
CONCEPT_ID		
STOP_REASON	NULL	
REFILLS	NULL	
QUANTITY	NULL	Not provided.
DAYS_SUPPLY	30	As described in the exercise.
SIG	NULL	Not provided.

Column name	Value	Explanation
ROUTE_CONCEPT_ID	4132161	4132161 indicates "Oral".
LOT_NUMBER	NULL	Not provided.
PROVIDER_ID	NULL	Not provided.
VISIT_OCCURRENCE_ID	NULL	No information on the visit was provided..
VISIT_DETAIL_ID	NULL	
DRUG_SOURCE_ID	76168009520	This is provided NDC code.
VALUE		
DRUG_SOURCE_ID	583945	583945 represents the drug source value (NDC code "76168009520").
ROUTE_SOURCE_ID	NULL	
VALUE		

예제 4.4

기록의 모음을 찾기 위해서, 우리는 테이블 CONDITION_OCCURRENCE를 쿼리할 수 있다:

```
library(DatabaseConnector)

connection <- connect(connectionDetails)

sql <- "SELECT *
FROM @cdm.condition_occurrence
WHERE condition_concept_id = 192671;"

result <- renderTranslateQuerySql(connection, sql, cdm = "main")

head(result)
```

예제 4.5

기록의 모음을 찾기 위해서, CONDITION_SOURCE_VALUE 필드를 사용하여 테이블 CONDITION_OCCURRENCE를 쿼리할 수 있다:

```
sql <- "SELECT *
FROM @cdm.condition_occurrence
WHERE condition_source_value = 'K92.2';"

result <- renderTranslateQuerySql(connection, sql, cdm = "main")
head(result)
```

```
##   CONDITION_OCCURRENCE_ID PERSON_ID CONDITION_CONCEPT_ID ...
## 1                   4657      273           192671 ...
## 2                   1021       61           192671 ...
## 3                   5978      351           192671 ...
## 4                   9798      579           192671 ...
## 5                   9301      549           192671 ...
## 6                   1997      116           192671 ...
```

예제 4.6

이 정보는 테이블 OBSERVATION_PERIOD에 저장되어 있다:

```
library(DatabaseConnector)
connection <- connect(connectionDetails)
sql <- "SELECT *
FROM @cdm.observation_period
WHERE person_id = 61;"

renderTranslateQuerySql(connection, sql, cdm = "main")
```

```
##   OBSERVATION_PERIOD_ID PERSON_ID OBSERVATION_PERIOD_START_DATE ...
## 1                   61          61           1968-01-21 ...
```

E.2 OMOP 표준 용어

예제 5.1

Concept ID 192671 (“Gastrointestinal hemorrhage”)

예제 5.2

ICD-10CM 코드:

- K29.91 “Gastroduodenitis, unspecified, with bleeding”
- K92.2 “Gastrointestinal hemorrhage, unspecified”

ICD-9CM 코드:

- 578 “Gastrointestinal hemorrhage”
- 578.9 “Hemorrhage of gastrointestinal tract, unspecified”

예제 5.3

MedDRA의 선호되는 용어:

- “Gastrointestinal haemorrhage” (Concept ID 35707864)
- “Intestinal haemorrhage” (Concept ID 35707858)

E.3 추출 변환 적재

예제 6.1

- A) 데이터 전문가와 CDM 전문가가 함께 ETL을 설계할 것
- B) 의학 지식이 있는 사람들이 코드 매핑을 할 것
- C) 엔지니어가 ETL을 수행할 것
- D) 모든 사람이 질 관리에 참여할 것

예제 6.2

Column	Value	Answer
PERSON_ID	A123B456	This column has a data type of integer so the source record value needs to be translated to a numeric value.
GENDER_CONCEPT_ID		
YEAR_OF_BIRTH	NULL	If we do not know the month or day of birth, we do not guess. A person can exist without a month or day of birth. If a person lacks a birth year that person should be dropped. This person would have to be dropped due to now year of birth.
MONTH_OF_BIRTH	NULL	
DAY_OF_BIRTH	NULL	
RACE_CONCEPT_ID	0	The race is WHITE which should be mapped to 8527.
ETHNICITY_CONCEPT_ID	8527	No ethnicity was provided, this should be mapped to 0.
ID		
PERSON_SOURCE_ID	A123B456	
VALUE		
GENDER_SOURCE_ID	F	
VALUE		
RACE_SOURCE_VALUE	WHITE	
ETHNICITY_SOURCE_ID	NONE	
VALUE	PROVIDED	

예제 6.3

Column	Value
VISIT_OCCURRENCE_ID	1

Column	Value
PERSON_ID	11
VISIT_START_DATE	2004-09-26
VISIT_END_DATE	2004-09-30
VISIT_CONCEPT_ID	9201
VISIT_SOURCE_VALUE	inpatient

E.4 데이터 분석 이용 사례

예제 7.1

1. 임상적 특성 분석
2. 환자 수준 예측
3. 인구 수준 추정

예제 7.2

아마 아닐 것이다. 디클로페낙 diclofenac 노출 코호트와 비교할 수 있는 비노출 코호트를 정의한다는 것은 보통 불가능한데 이는 사람들이 각각의 이유가 있어 디클로페낙을 복용하기 때문이다. 이것은 사람 간의 비교를 금지한다. 사람 내 비교가 가능할 수는 있어서, 디클로페낙 코호트 안의 각각의 환자는 그들이 노출되지 않았을 때의 시간을 정의할 수 있다. 그러나, 비슷한 문제가 여기서 발생한다: 이런 시기는 비교가 되지 않는다. 왜냐하면 어떤 때 누군가는 노출되고, 누군가는 노출이 되지 않기 때문이다.

E.5 SQL과 R

예제 9.1

간단하게 테이블 PERSON을 쿼리하여 사람의 수를 계산하기 위해서는:

```
library(DatabaseConnector)
connection <- connect(connectionDetails)
sql <- "SELECT COUNT(*) AS person_count
FROM @cdm.person;"
```

```
renderTranslateQuerySql(connection, sql, cdm = "main")
```

```
## PERSON_COUNT
## 1      2694
```

예제 9.2

최소 한 번이라도 celecoxib의 처방을 받은 사람의 수를 계산하기 위해서, 테이블 DRUG_EXPOSURE를 쿼리할 수 있다. 성분 celecoxib을 포함하는 모든 약물을 찾기 위해서, 우리는 테이블 CONCEPT_ANCESTOR와 CONCEPT를 조인해야 한다:

```
library(DatabaseConnector)
connection <- connect(connectionDetails)
sql <- "SELECT COUNT(DISTINCT(person_id)) AS person_count
FROM @cdm.drug_exposure
INNER JOIN @cdm.concept_ancestor
    ON drug_concept_id = descendant_concept_id
INNER JOIN @cdm.concept ingredient
    ON ancestor_concept_id = ingredient.concept_id
WHERE LOWER(ingredient.concept_name) = 'celecoxib'
    AND ingredient.concept_class_id = 'Ingredient'
    AND ingredient.standard_concept = 'S';"

renderTranslateQuerySql(connection, sql, cdm = "main")
```

```
## PERSON_COUNT
## 1      1844
```

개인이 두 개 이상의 처방을 가질 수 있다는 점을 고려하여, 겹치지 않는 개인의 수를 찾기 위해 COUNT(DISTINCT(person_id))를 사용하는 것을 명심한다. 또한 대소문자 구별 없이 “celecoxib”을 찾기 위하여 LOWER 기능을 사용하는 것도 명심한다.

대신에, 우리는 성분 레벨까지 이미 롤업 된 테이블 DRUG_ERA를 사용할 수 있다.

```
library(DatabaseConnector)
connection <- connect(connectionDetails)
```

```

sql <- "SELECT COUNT(DISTINCT(person_id)) AS person_count
FROM @cdm.drug_era
INNER JOIN @cdm.concept ingredient
  ON drug_concept_id = ingredient.concept_id
WHERE LOWER(ingredient.concept_name) = 'celecoxib'
  AND ingredient.concept_class_id = 'Ingredient'
  AND ingredient.standard_concept = 'S';"

renderTranslateQuerySql(connection, sql, cdm = "main")

```

```

##    PERSON_COUNT
## 1      1844

```

예제 9.3

노출되는 동안 증상의 수를 계산하기 위해서는 이전의 쿼리를 테이블 CONDITION_OCCURRENCE를 조인해서 확장한다. 위장 출혈을 암시하는 모든 condition concept을 찾기 위해서는 테이블 CONCEPT_ANCESTOR를 조인한다:

```

library(DatabaseConnector)
connection <- connect(connectionDetails)
sql <- "SELECT COUNT(*) AS diagnose_count
FROM @cdm.drug_era
INNER JOIN @cdm.concept ingredient
  ON drug_concept_id = ingredient.concept_id
INNER JOIN @cdm.condition_occurrence
  ON condition_start_date >= drug_era_start_date
  AND condition_start_date <= drug_era_end_date
INNER JOIN @cdm.concept_ancestor
  ON condition_concept_id = descendant_concept_id
WHERE LOWER(ingredient.concept_name) = 'celecoxib'
  AND ingredient.concept_class_id = 'Ingredient'
  AND ingredient.standard_concept = 'S'
  AND ancestor_concept_id = 192671;"

renderTranslateQuerySql(connection, sql, cdm = "main")

```

The screenshot shows the 'Cohort Entry Events' interface. At the top, it says 'Events having any of the following criteria:' with a 'Add Initial Event' button and a question mark icon. Below this, there are two main sections:

- Initial Event Criteria:** A dropdown menu shows 'a drug era of diclofenac'. To its right are buttons for '+ Add attribute...' and 'Delete Criteria'. Below this are two red X icons with conditions: 'for the first time in the person's history' and 'with age in years at era start Greater or Equal To 16'.
- Observation Period:** It specifies 'with continuous observation of at least 365 days before and 0 days after event index date'.
- Limiting:** It says 'Limit initial events to: earliest event per person.'
- Buttons:** A green 'Restrict initial events' button is at the bottom left, and a red 'Delete Criteria' button is at the top right.

Figure E.1: diclofenac을 복용하기 시작한 환자를 위한 코호트 입력 사례 설정

```
##   DIAGNOSE_COUNT
## 1      41
```

이런 경우에는 테이블 DRUG_EXPOSURE 대신에 테이블 DRUG_ERA를 사용하는 것이 중요하다는 것을 명심하자. 왜냐하면 같은 성분을 가진 약물 노출이 다수 겹치면서, 이중 계산으로 이어질 수도 있기 때문이다. 예를 들어, 한 개인이 동시에 celecoxib성분을 포함하는 두 개의 약물을 받았다고 상상해 보라. 이것은 두 개의 약물 노출로 기록될 것이며, 그러므로 노출 중에 일어나는 모든 증상이 두 번으로 집계될 것이다. 하지만 DRUG_ERA를 사용하면 이 두 개의 노출이 하나의 기록으로 합쳐진다.

E.6 코호트 만들기

예제 10.1

아래의 요구사항을 암호화하는 초기 사례 기준을 생성한다:

- 디클로페낙 diclofenac을 복용하기 시작한 환자
- 16세 이상의 환자
- 노출 전 최소 365일의 계속된 관찰이 있던 환자

마무리했다면, 코호트 입력 사례 섹션은 그림 E.1과 같아야 한다.

디클로페낙의 개념 모음 concept set 표현은 그림 E.2과 비슷해야 할 것이며, ‘디클로페낙’ 성분과 ‘디클로페낙’의 모든 하위목록도 모두 포함하여 디클로페낙 성분이 포함된 모든 약물을 포함한다.

다음으로, 그림 E.3에서 보이는 것과 같이, 모든 NSAID에 대한 이전의 노출이 없는 것을 필요로 한다.

The screenshot shows a search interface for a concept set expression. At the top, there are tabs for 'Concept Set Expression' (highlighted in blue), 'Included Concepts' (11473), 'Included Source Codes', 'Export', and 'Import'. Below the tabs, the 'Name:' field contains 'diclofenac'. A 'Show' dropdown set to '25' and a 'Search' input field are also visible. The main area displays a table of search results:

	Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	Exclude	Descendants	Mapped
	1124300	3355	Diclofenac	Drug	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Below the table, there are color-coded legends: Classification (purple square), Non-Standard (dark red square), and Standard (blue square).

Figure E.2: 디클로페낙의 개념 모음 표현.

The screenshot shows the 'Inclusion Criteria' section of a study configuration. It includes a 'New inclusion criteria' button and a list of existing criteria:

- 1. Without prior exposure to any NSAID** (Excluding subjects with prior exposure to any NSAID)
 - having all of the following criteria:
 - with exactly 0 using all occurrences of: a drug exposure of **NSAIDs** where **event starts** between All days **Before** and 1 days **Before** **index start date** **add additional constraint**
 - restrict to the same visit occurrence
 - allow events from outside observation period

At the bottom, there is a note: 'Limit qualifying events to: earliest event per person.'

Figure E.3: 모든 NSAID에 대한 이전의 노출이 없는 것이 필요하다.

Figure E.4: NSAID의 개념 모음 표현

Figure E.5: 이전의 암 증상이 없는 것이 필요하다.

NSAID의 개념 모음 표현은 그림 E.4와 비슷해야 할 것이며, NSAID 클래스와 NSAID의 모든 하위요소도 모두 포함하여 NSAID이 포함된 모든 약물을 포함한다.

추가로, 그림 E.5에서 보이는 것과 같이, 이전에 기록된 악성 암이 없도록 한다.

“Broad malignancies”의 개념 모음 표현은 그림 E.6과 비슷해야 할 것이며, 높은 레벨 개념의 “Malignant neoplastic disease”와 그의 모든 하위요소도 포함해야 한다.

마지막으로, 그림 E.7에서 보이는 것과 같이, 코호트 종료 기준을 노출의 중단 (30일 간격 허용)으로 정의한다.

Concept Set Expression		Included Concepts 4401	Included Source Codes	Export	Import			
Name:								
Broad malignancies								
Show	25 ▾	entries	Search:					
Showing 1 to 1 of 1 entries								
	Previous	1	Next					
Actions	Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	Exclude	Descendants	Mapped
	443392	363346000	Malignant neoplastic disease	Condition	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figure E.6: broad malignancies의 개념 모음 표현

Cohort Exit ?

Event Persistence:
 Event will persist until: ▾

Continuous Exposure Persistence:
 Specify a concept set that contains one or more drugs. A drug era will be derived from all drug exposure events for any of the drugs within the concept set, using the specified persistence window as a maximum allowable gap in days between successive exposure events and adding a specified surveillance window to the final exposure event. If no exposure event end date is provided, then an exposure event end date is inferred to be event start date + days supply in cases when days supply is available or event start date + 1 day otherwise. This event persistence assures that the cohort end date will be no greater than the drug era end date.

Concept set containing the drug(s) of interest: ▾

- Persistence window: allow for a maximum of days between exposure records when inferring the era of persistence exposure
- Surveillance window: add days to the end of the era of persistence exposure as an additional period of surveillance prior to cohort exit.

Censoring Events:

Exit Cohort based on the following criteria:

No censoring events selected.

Figure E.7: 코호트 종료 날짜 설정하기.

예제 10.2

가독성을 위하여 SQL을 두 개의 과정으로 나누었다. 첫 번째로, 심근경색의 모든 발생 기록을 찾고, 이를 “#diagnoses”로 불리는 임시 테이블에 저장한다:

```
library(DatabaseConnector)
connection <- connect(connectionDetails)
sql <- "SELECT person_id AS subject_id,
           condition_start_date AS cohort_start_date
      INTO #diagnoses
     FROM @cdm.condition_occurrence
    WHERE condition_concept_id IN (
        SELECT descendant_concept_id
          FROM @cdm.concept_ancestor
         WHERE ancestor_concept_id = 4329847 -- Myocardial infarction
    )
        AND condition_concept_id NOT IN (
        SELECT descendant_concept_id
          FROM @cdm.concept_ancestor
         WHERE ancestor_concept_id = 314666 -- Old myocardial infarction
    );"

renderTranslateExecuteSql(connection, sql, cdm = "main")
```

그리고 몇몇 특별한 COHORT_DEFINITION_ID (우리는 '1'을 선택하였다)를 사용하여 입원 중이거나 응급실에 방문한 환자들에게 일어난 것만 선택한다:

```
sql <- "INSERT INTO @cdm.cohort (
           subject_id,
           cohort_start_date,
           cohort_definition_id
       )
      SELECT subject_id,
             cohort_start_date,
             CAST (1 AS INT) AS cohort_definition_id
     FROM #diagnoses
    INNER JOIN @cdm.visit_occurrence
```

```

ON subject_id = person_id
AND cohort_start_date >= visit_start_date
AND cohort_start_date <= visit_end_date
WHERE visit_concept_id IN (9201, 9203, 262); -- Inpatient or ER;"

renderTranslateExecuteSql(connection, sql, cdm = "main")

```

대개의 접근 방식은 방문 시작과 종료 날짜 사이에 부합하는 condition 날짜를 요구하는 대신, VISIT_OCCURRENCE_ID에 근거하여 방문의 condition에 조인할 수 있음을 명심하자. 이는 입원 혹은 응급실 방문 시 기록된 condition 만을 확인하여, 더 정확할 수 있다. 하지만, 많은 관찰 데이터베이스에서는 방문과 증상의 관계를 기록하지 않기 때문에, 날짜를 대신 사용하는 것을 선택하여 특이도는 낮을 수 있으나 민감도를 높일 수도 있다.

또한 코호트 종료 날짜를 무시했다는 것을 명심한다. 종종, 결과를 정의하기 위해 코호트가 사용되었을 때 우리는 코호트 시작 날짜만 염두하고, (병명이 정의된) 코호트 종료 날짜를 생성하는 것은 의미가 없다.

임시 테이블은 더는 필요가 없다면 정리하는 것을 추천한다:

```

sql <- "TRUNCATE TABLE #diagnoses;
DROP TABLE #diagnoses;"

renderTranslateExecuteSql(connection, sql)

```

E.7 임상적 특성 분석

예제 11.1

아틀라스에서  Data Sources를 클릭하고, 관심 있는 데이터 원천을 선택한다. 약물 노출 기록을 선택할 수 있고, “Table” 탭을 선택할 수 있으며, 그림 E.8과 같이 “celecoxib”를 찾을 수 있다. 여기에서 이 특정한 데이터베이스가 celecoxib의 다양한 제형들의 노출을 포함하는 것을 볼 수 있다. 더욱 자세히 보기 위해서는 여기에서의 아무 약물을 클릭할 수 있다. 예를 들어, 이 약물에 대한 나이나 성별 분포를 나타내기 위함이다.

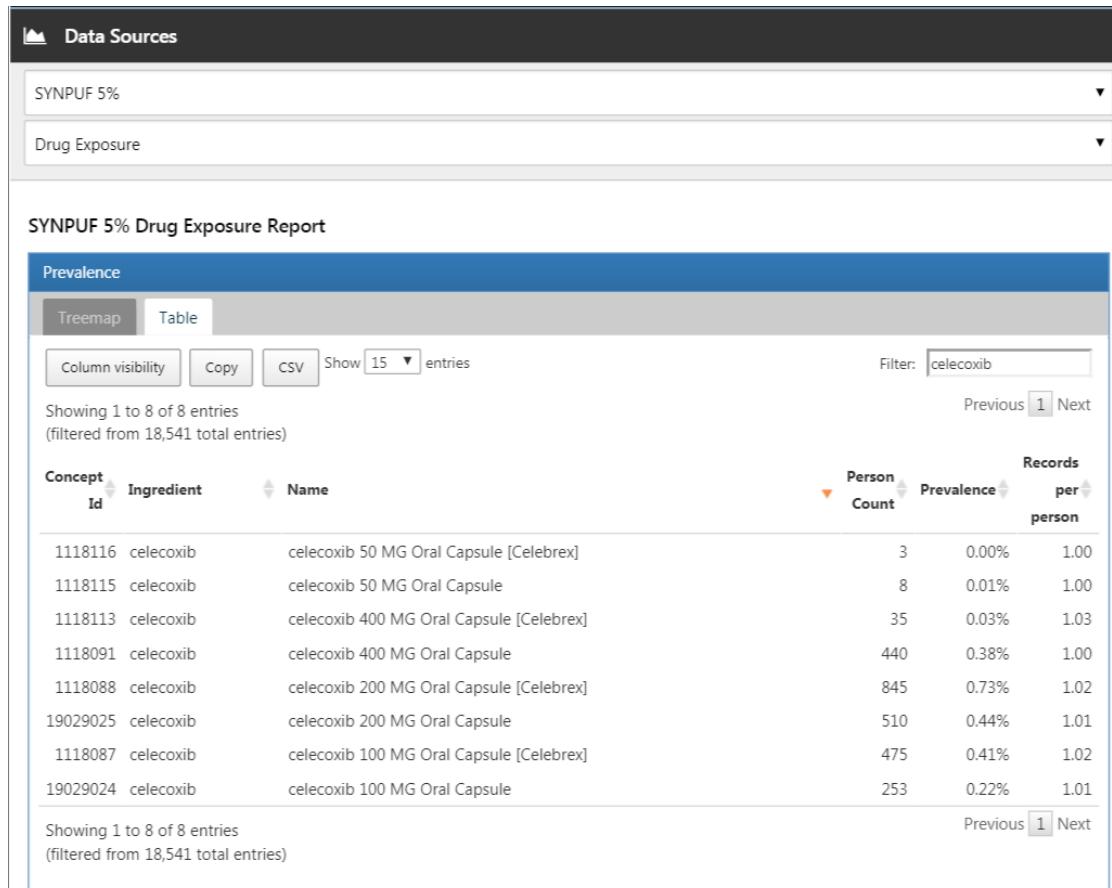


Figure E.8: 데이터 원천 특성.

	Id	Code	Name	Class	RC	DRC	Domain	Vocabulary
	1118084	140587	celecoxib	Ingredient	2,587	5,184	Drug	RxNorm

Figure E.9: "celecoxib" 성분의 표준 개념 선택하기.

예제 11.2

Cohort Definitions를 클릭하고 새로운 코호트를 생성하기 위해 “New cohort”를 클릭한다. 코호트에 의미 있는 이름을 부여하고 (예를 들어, “Celecoxib new users”가 있다) “Concept Sets” 탭으로 이동한다. “New Concept Set”를 클릭하고, 개념 모음 concept set에 의미있는 이름을 부여한다 (예를 들어, “Celecoxib”). 모듈을 열고, “celecoxib”를 검색하여, 클래스를 “Ingredient”로, 표준 개념을 “Standard”로 제한한 후, 그림 E.9와 같이 개념을 당신의 개념 모음에 추가하기 위해 를 클릭한다.

코호트 정의로 돌아가기 위해서는 그림 E.9의 상위 왼편에 보이는 왼쪽 화살표를 클릭한다. “+Add Initial Event”를 클릭한 후 “Add Drug Era”를 클릭한다. 이미 생성된 약물 범위 기준의 개념 모음을 선택한다. “Add attribute...”를 클릭하고 “Add First Exposure Criteria.”를 선택한다. 발생 시점 전으로부터 최소 365일이 요구되는 지속적인 관찰을 설정한다. 결과는 그림 E.10와 비슷해야 한다. 포함 기준, 코호트 종료, 코호트 범위 세션을 그대로 두고 나간다. 를 클릭하여 코호트 정의를 저장하고, 를 클릭하여 확실히 닫는다.

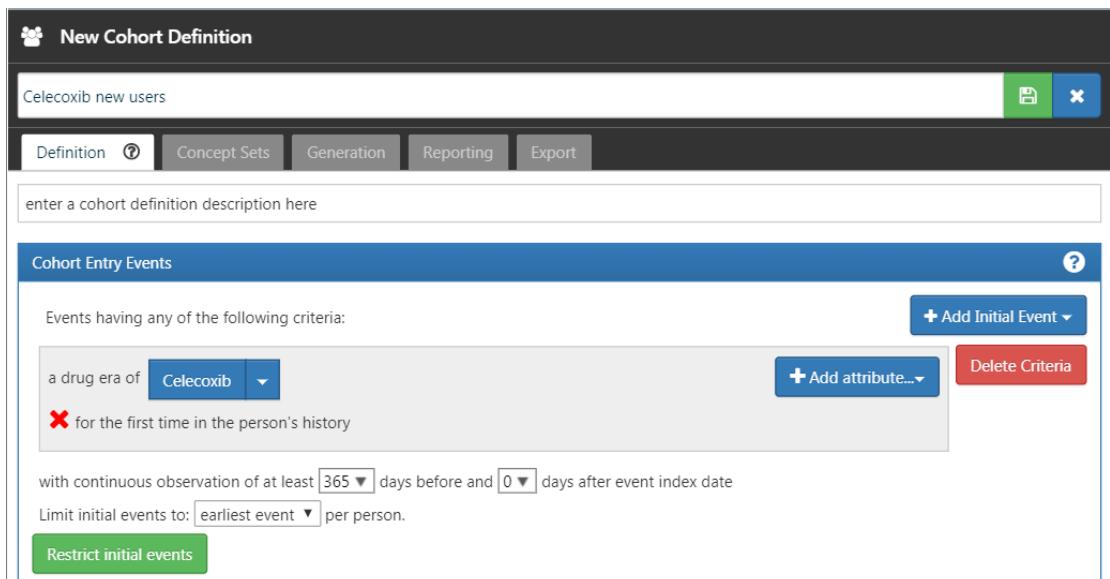


Figure E.10: 간단한 celecoxib 새 사용자의 코호트 정의.

이제 코호트가 정의되었으니, 특성화 할 수 있다. **Characterizations**를 클릭한 후 “New Characterization”을 클릭한다. 임상적 특성에 의미 있는 이름을 부여한다 (예를 들어, “Celecoxib 새 사용자의 임상적 특성”). 코호트 정의 아래에, “Import”를 클릭하고 최근에 생성한 코호트 정의를 선택한다. “Feature Analyses” 아래에, “Import”를 클릭한 후 최소 하나의 condition 분석과 하나의 약물 분석을 선택한다. 예를 들면 “Drug Group Era Any Time Prior”와 “Condition Group Era Any Time Prior”이다. 임상적 특성 정의는 그림 E.11과 비슷해야 한다. **Save**를 클릭하여 임상적 특성 설정을 확실히 저장한다.

“Exections” 탭을 클릭한 후, 데이터 베이스 중 하나에 “Generate”를 클릭한다. 시간이 꽤나 걸릴 수 있다. 끝나면, “View latest results”를 클릭할 수 있다. 결과 화면은 그림 E.12와 비슷하고, 이는 예를 들어 pain과 arthropathy가 흔히 관찰되는 것으로 보이는데, 진통제(celecoxib) 사용자 코호트의 특성으로 자연스러운 일이다. 목록의 아래쪽에 기대하지 않은 condition이 보일 수 있다.

예제 11.3

Cohort Definitions를 클릭한 후 새로운 코호트를 생성하기 위해 “New cohort”를 클릭한다. 코호트에 의미 있는 이름을 부여한다 (예를 들어 “GI bleed”). **Search** 모듈을 열고, “Gastrointestinal hemorrhage”를 검색한 후, 그림 E.13와 같이 개념을 개념 모음에 추가하기 위해 상위 개념 옆의 **Add**를 클릭한다.

당신의 코호트 정의로 돌아가기 위해서는 그림 E.13의 상위 왼쪽에 위치한 왼쪽 화살표를 클릭한다.

New Characterization

Celecoxib new users characterization

Design Executions Utilities

Cohort characterization is defined as the process of generating cohort level descriptive summary statistics from person level covariate data. Summary statistics of these person level covariates may be count, mean, sd, var, min, max, median, range, and quantiles. In addition, covariates during a period may be stratified into temporal units of time for time-series analysis such as fixed intervals of time relative to cohort_start_date (e.g. every 7 days, every 30 days etc.), or in absolute calendar intervals such as calendar-week, calendar-month, calendar-quarter, calendar-year.

Cohort definitions

Import

Show 10 entries Search:

ID	Name		
1771701	Celecoxib new users	Edit cohort	Remove

Showing 1 to 1 of 1 entries Previous 1 Next

Feature analyses

Import

Show 10 entries Search:

ID	Name	Description	Actions
15	Drug Group Era Any Time Prior	One covariate per drug rolled up to ATC groups in the drug_era table overlapping with any time prior to index.	Remove
27	Condition Group Era Any Time Prior	One covariate per condition era rolled up to groups in the condition_era table overlapping with any time prior to index.	Remove

Showing 1 to 2 of 2 entries Previous 1 Next

Figure E.11: 임상적 특성 설정하기.

The screenshot shows the Characterization #69 interface. At the top, there is a title bar with the text "Characterization #69" and a subtitle "Celecoxib new users characterization". Below the title bar is a toolbar with icons for saving, closing, and other operations. A navigation bar at the bottom of the title bar includes tabs for "Design", "Executions", and "Utilities", with "Executions" being the active tab. The main content area displays the results of the execution, specifically "Reports for SYNPUF 5%". It shows the date (08/23/2019 12:53 PM), design ID (-1840810470), and the number of results (2 reports). A filter panel is visible, containing sections for Cohorts, Analyses, and Domains, with dropdown menus for "Celecoxib new users", "Condition Group Era Any Time P", and "Condition, Drug".

CONDITION / Condition Group Era Any Time Prior

Covariate	Explore	Concept ID	Count	Pct
Pain	Explore	4329041	1,140	78.62%
Pain finding at anatomical site	Explore	4132926	1,135	78.28%
Inflammation of specific body systems	Explore	4178818	1,135	78.28%
Arthropathy	Explore	73553	1,122	77.38%

Figure E.12: 임상적 특성 설명.

The screenshot shows a search interface with a header "GI bleed > GI bleed". Below the header is a search bar with the query "Gastrointestinal hemorrhage" and a search button. There are also "Search" and "Import" buttons. To the right of the search bar are "Advanced Options" and a "Filter" input field. Below the search bar is a table displaying search results. The table has columns: Vocabulary, Id, Code, Name, Class, RC, DRC, Domain, and Vocabulary. The results show three entries related to "Gastrointestinal hemorrhage":

Vocabulary	Id	Code	Name	Class	RC	DRC	Domain	Vocabulary
SNOMED (17)	192671	74474003	Gastrointestinal hemorrhage	Clinical Finding	919	37,144	Condition	SNOMED
ICD10CM (2)	4338544	87763006	Lower gastrointestinal hemorrhage	Clinical Finding	0	15,617	Condition	SNOMED
ICD9CM (2)	4100660	27719009	Acute gastrointestinal hemorrhage	Clinical Finding	0	9,852	Condition	SNOMED

Figure E.13: "Gastrointestinal hemorrhage"의 표준 개념 선택하기.

Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	Exclude	Descendants	Mapped
192671	74474003	Gastrointestinal hemorrhage	Condition	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figure E.14: "Gastrointestinal hemorrhage"의 하위요소 추가하기.

Figure E.15: 간단한 gastrointestinal bleed 코호트 정의.

“Concept Sets” 템을 다시 열고, GI hemorrhage concept 옆에 있는 “Descendants”를 그림 E.14처럼 체크한다.

“Definition” 템으로 돌아간 후, “+Add Initial Event”를 클릭하고, “Add Condition Occurrence”를 클릭한다. 이전에 생성한 condition 발생 기준의 개념 모음을 선택한다. 결과는 E.15와 비슷해야 한다. 포함 기준, 코호트 종료, 코호트 범위 세션을 그대로 두고 나간다. 를 클릭하여 코호트 정의를 저장하고, 를 클릭하여 확실히 닫는다.

이제 코호트가 정의되었으면, 발생률을 계산할 수 있다. **Incidence Rates**를 클릭하고, “New Analysis”를 클릭한다. 분석에 의미 있는 이름을 부여한다 (예를 들어 “Incidence of GI bleed after celecoxib initiation”). “Add Target Cohort”를 클릭한 후 celecoxib 새로운 사용자 코호트를 선택한다. “Add Outcome Cohort”를 클릭한 후 새로운 GI bleed 코호트를 추가한다. 위험 노출 기간을 시작일 이후로 1,095일이 지난 시점을 종료일로 지정한다. 분석은 그림 E.16과 비슷해야 한다.

The screenshot shows the 'New Incidence Rate Analysis' interface. At the top, it displays 'Incidence of GI bleed after celecoxib initiation'. Below this are tabs for 'Definition', 'Concept Sets', 'Generation', and 'Utilities'. The 'Definition' tab is selected.

Study Cohorts:

- Target Cohorts:** #1771701:Celecoxib new users
- Outcome Cohorts:** #1771702:GI bleed

Buttons for 'Add Target Cohort' (green) and 'Add Outcome Cohort' (red) are visible.

Time At Risk: Time at risk defines the time window relative to the cohort start or end date with an offset to consider the person 'at risk' of the outcome.

- Time at risk starts with **start date** plus **0** days.
- Time at risk ends with **start date** plus **1095** days.

No study window defined. **Add Study Window** button.

Stratify Criteria: You can provide optional stratification criteria to the analysis that will divide the population into unique groups based on their satisfied criteria.

New stratify criteria button. A note says 'Please select a qualifying inclusion criteria to edit.'

Figure E.16: 발생률 분석.

Showing target cohort: Celecoxib new users and outcome cohort: GI bleed

Source Name	Persons	Cases	Proportion [+/-] per 1k persons	Time At Risk (years)	Rate [+/-] per 1k years	Started	Duration
C Rerun SYNPUF 5%	1,205	95	78.84	1,052	90.30	08/23/2019 1:59 PM	00:00:22

Reports and **CSV** export buttons.

Figure E.17: 발생 결과.

클릭해서 분석 설정을 확실히 저장한다.

“Generation” 탭을 클릭한 후, “Generate”를 클릭한다. 데이터 원천 중 하나를 선택하고 “Generate”를 클릭한다. 끝났다면, 그림 E.17에 보이는 것과 같이 계산된 발생률과 분율을 볼 수 있다.

E.8 인구 수준 추정

예제 12.1

공변량 covariate의 기본 모음을 명시하고 있지만, 반드시 비교하는 두 개의 약물은 제외하고, 그것의 하위 목록은 포함해야 한다. 그렇지 않으면 성향 모델은 완벽하게 예측이 가능해질 것이다:

```
library(CohortMethod)
nsaids <- c(1118084, 1124300) # celecoxib, diclofenac
covSettings <- createDefaultCovariateSettings(
  excludedCovariateConceptIds = nsaids,
```

```

addDescendantsToExclude = TRUE)

# Load data:
cmData <-getDbCohortMethodData(
  connectionDetails = connectionDetails,
  cdmDatabaseSchema = "main",
  targetId = 1,
  comparatorId = 2,
  outcomeIds = 3,
  exposureDatabaseSchema = "main",
  exposureTable = "cohort",
  outcomeDatabaseSchema = "main",
  outcomeTable = "cohort",
  covariateSettings = covSettings)
summary(cmData)

```

```

## CohortMethodData object summary
##
## Treatment concept ID: 1
## Comparator concept ID: 2
## Outcome concept ID(s): 3
##
## Treated persons: 1800
## Comparator persons: 830
##
## Outcome counts:
##   Event count Person count
## 3           479          479
##
## Covariates:
## Number of covariates: 389
## Number of non-zero covariate values: 26923

```

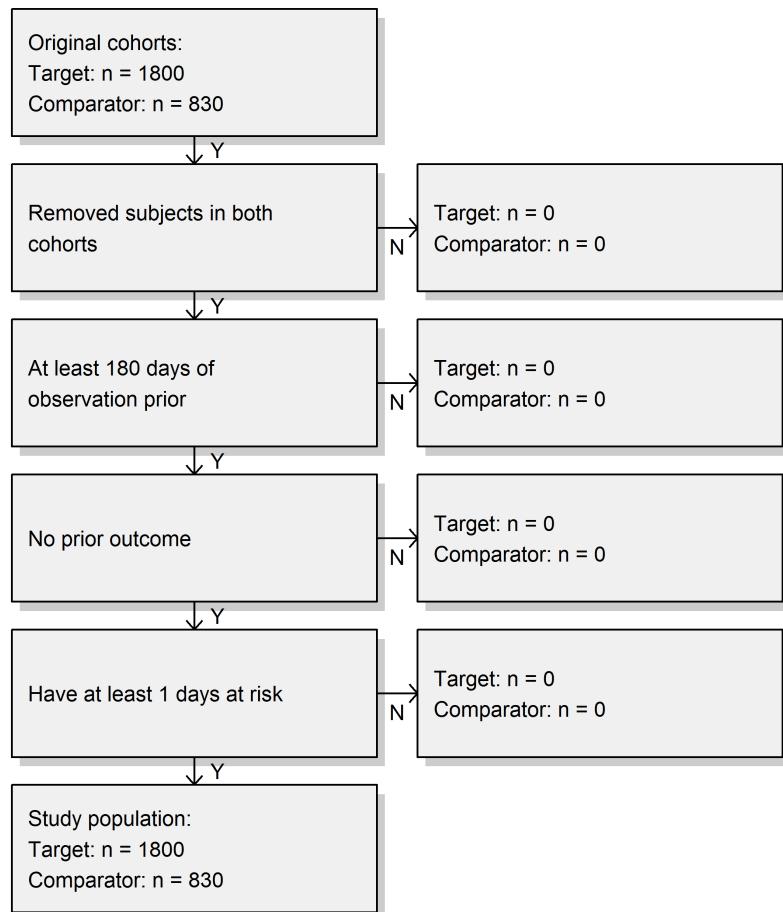
예제 12.2

사양 specification을 따르는 인구 모집단을 생성하고, attrition 도표를 출력한다:

```

studyPop <- createStudyPopulation(
  cohortMethodData = cmData,
  outcomeId = 3,
  washoutPeriod = 180,
  removeDuplicateSubjects = "remove all",
  removeSubjectsWithPriorOutcome = TRUE,
  riskWindowStart = 0,
  startAnchor = "cohort start",
  riskWindowEnd = 99999)
drawAttritionDiagram(studyPop)

```



기존의 코호트와 비교하여 대상을 전혀 잊지 않은 것을 볼수 있는데, 여기서 사용한 제한이 이미

코호트 정의에서 사용된 것이기 때문이다.

예제 12.3

콕스 회귀를 사용하여 간단한 결과 모델을 적합한다:

```
model <- fitOutcomeModel(population = studyPop,
                           modelType = "cox")
model

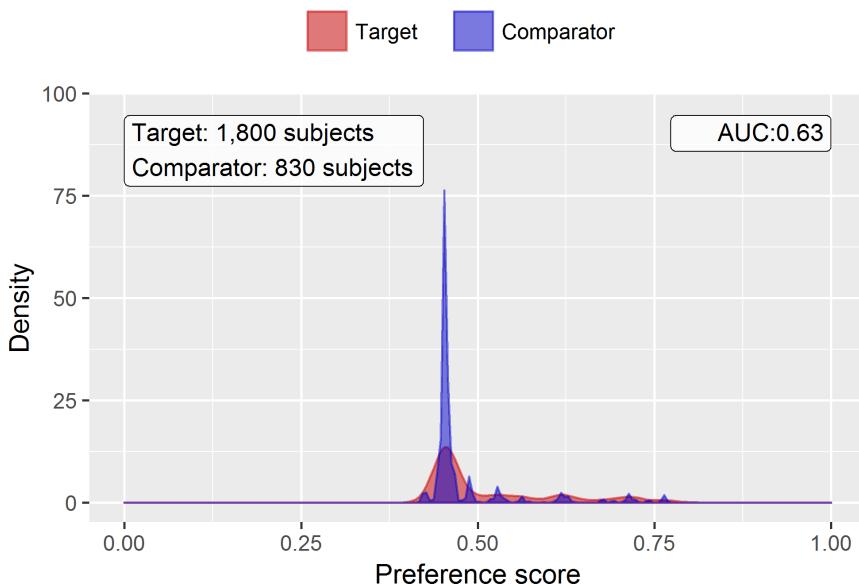
## Model type: cox
## Stratified: FALSE
## Use covariates: FALSE
## Use inverse probability of treatment weighting: FALSE
## Status: OK
##
##           Estimate lower .95 upper .95   logRr seLogRr
## treatment  1.34612   1.10065   1.65741 0.29723  0.1044
```

celecoxib 사용자가 diclofenac 사용자와 교환 가능하지 않을 가능성이 있고, 이러한 기저 과거력의 차이는 이와 같이 outcome 상의 차이가 있는 것처럼 결과가 나올 수도 있다. 이 차이를 조절하지 않으면, 이 분석과 같이 편향된 측정을 생성할 수 있다.

예제 12.4

추출한 모든 공변량을 사용하여 연구 모집단에 성향 모델을 적합했다. 그 후 선호 점수 분포도를 보여준다:

```
ps <- createPs(cohortMethodData = cmData,
                population = studyPop)
plotPs(ps, showCountsLabel = TRUE, showAucLabel = TRUE)
```



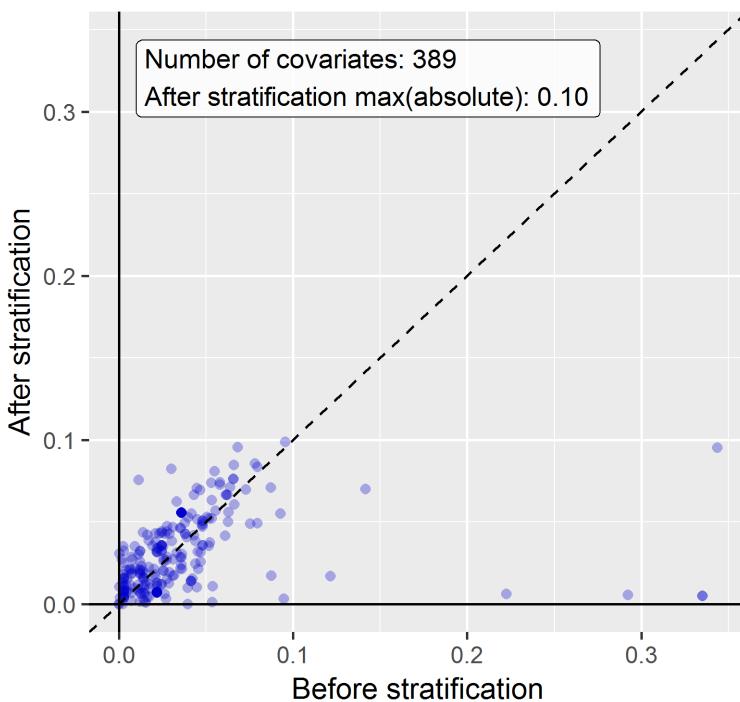
몇 개의 spike가 있는 이 분포가 조금 이상해 보일 수 있다. 예제에서는 시뮬레이션 된 정말 작은 데이터 모음을 사용했기 때문이다. 실제의 선호 점수 분포는 더 매끄러운 경향이 있다.

성향 모델은 0.63의 AUC를 달성하는데, 대상 target과 대조 comparator 코호트 간을 구분할 수 있다는 것을 의미한다. 두 집단 간의 꽤 많은 교차를 볼 수 있어, 성향 점수 조정으로 두 코호트를 더 동일하게 만들었음을 시사한다.

예제 12.5

모집단을 성향 점수에 근거하여 계층화하고, 층화 전과 후의 공변량 균형을 계산한다:

Standardized difference of mean



다양한 기저 공변량은 총화 전의(x-axis) 큰(>0.3) 표준화된 평균의 차이를 보여준다. 총화 후에, 최대 표준화 차이(≤ 0.1)와 같이 균형 balance가 좋아졌다.

예제 12.6

PS strata 기준 총화하여 콕스 회귀 적합을 시행한다:

```
adjModel <- fitOutcomeModel(population = strataPop,
                             modelType = "cox",
                             stratified = TRUE)

adjModel
```

```
## Model type: cox
## Stratified: TRUE
## Use covariates: FALSE
## Use inverse probability of treatment weighting: FALSE
## Status: OK
##
##           Estimate lower .95 upper .95    logRr seLogRr
```

```
## treatment 1.13211 0.92132 1.40008 0.12409 0.1068
```

조정된 추정치는 조정되지 않은 추정치보다 낮고, 95% 신뢰 구간은 현재 1을 포함하는 것을 볼 수 있다. 기저 과거력의 차이를 보정함으로써, 비뚤림이 감소함을 볼 수 있다.

E.9 환자 수준 예측

예제 13.1

공변량 설정의 모음을 지정하고, `getPlpData` 기능을 데이터베이스에서 데이터를 추출하기 위해 사용한다:

```
library(PatientLevelPrediction)
covSettings <- createCovariateSettings(
  useDemographicsGender = TRUE,
  useDemographicsAge = TRUE,
  useConditionGroupEraLongTerm = TRUE,
  useConditionGroupEraAnyTimePrior = TRUE,
  useDrugGroupEraLongTerm = TRUE,
  useDrugGroupEraAnyTimePrior = TRUE,
  useVisitConceptCountLongTerm = TRUE,
  longTermStartDays = -365,
  endDays = -1)

plpData <- getPlpData(connectionDetails = connectionDetails,
  cdmDatabaseSchema = "main",
  cohortDatabaseSchema = "main",
  cohortTable = "cohort",
  cohortId = 4,
  covariateSettings = covSettings,
  outcomeDatabaseSchema = "main",
  outcomeTable = "cohort",
  outcomeIds = 3)

summary(plpData)

## plpData object summary
```

```

## 
## At risk cohort concept ID: -1
## Outcome concept ID(s): 3
##
## People: 2630
##
## Outcome counts:
##   Event count Person count
## 3          479          479
##
## Covariates:
## Number of covariates: 245
## Number of non-zero covariate values: 54079

```

예제 13.2

관심 결과의 연구 모집단을 생성하고 (이 경우에는 추출한 데이터의 유일한 결과만), 364일의 위험 노출 시간 time-at-risk이 필요하며, NSAID를 시작하기 전의 결과를 경험한 피험자를 제거한다:

```

population <- createStudyPopulation(plpData = plpData,
                                      outcomeId = 3,
                                      washoutPeriod = 364,
                                      firstExposureOnly = FALSE,
                                      removeSubjectsWithPriorOutcome = TRUE,
                                      priorOutcomeLookback = 9999,
                                      riskWindowStart = 1,
                                      riskWindowEnd = 365,
                                      addExposureDaysToStart = FALSE,
                                      addExposureDaysToEnd = FALSE,
                                      minTimeAtRisk = 364,
                                      requireTimeAtRisk = TRUE,
                                      includeAllOutcomes = TRUE)

nrow(population)

## [1] 2578

```

이 경우에 사전의 결과를 가진 피험자를 제거하고, 최소 364일의 위험 노출 기간을 요구하기 때문에

몇 사람들을 잃게 된다.

예제 13.3

먼저 모델 설정 객체를 만든 후에, runPlp 기능을 호출함으로써 LASSO 모델을 실행한다. 이 경우에는 person split을 행하고, 75%의 데이터를 모델에 학습 train시키며, 25%의 데이터로 평가 test 한다:

```
lassoModel <- setLassoLogisticRegression(seed = 0)

lassoResults <- runPlp(population = population,
                        plpData = plpData,
                        modelSettings = lassoModel,
                        testSplit = 'person',
                        testFraction = 0.25,
                        nfold = 2,
                        splitSeed = 0)
```

이 예시에서는 LASSO 교차 검증과 학습-검증 데이터 분할을 위한 random seed를 지정하여 여러 번 실행해도 결과가 동일한지 확인한다.

Shiny 앱을 사용하여 결과를 볼 수 있다:

```
viewPlp(lassoResults)
```

이것은 그림 E.18에서 보이는 것과 같이 앱을 실행할 것이다. 여기 0.645의 테스트 모음의 AUC가 있는데 이는 무작위 추측보다 더 나을 수 있으나 임상적으로 사용할 만큼의 성능은 아니다.

E.10 데이터의 질

예제 15.1

ACHILLES을 실행하기 위해서는:

```
library(ACHILLES)
result <- achilles(connectionDetails,
                      cdmDatabaseSchema = "main",
```

The screenshot shows the 'PatientLevelPrediction Explorer' Shiny application interface. At the top, there are tabs for 'Internal Validation' and 'External Validation'. Below the tabs, a navigation bar includes 'Evaluation Summary' (which is active), 'Characterization', 'ROC', 'Calibration', 'Demographics', 'Preference', 'Box Plot', and 'Settings'. The main content area is titled 'Evaluation Summary' and contains a table. The table has a header row with 'Metric' and two columns: 'test' and 'train'. The data rows are as follows:

Metric	test	train
1 AUC	0.645	0.7112
2 AUC_lb95ci	0.589	0.6815
3 AUC_ub95ci	0.700	0.7409
4 AUPRC	0.286	0.3615
5 BrierScaled	0.062	0.0860
6 BrierScore	0.144	0.1382

Figure E.18: 환자 수준 예측 Shiny 앱.

```
resultsDatabaseSchema = "main",
sourceName = "Eunomia",
cdmVersion = "5.3.0")
```

예제 15.2

데이터의 질 Dashboard를 실행하기 위해서는:

```
DataQualityDashboard::executeDqChecks(
  connectionDetails,
  cdmDatabaseSchema = "main",
  resultsDatabaseSchema = "main",
  cdmSourceName = "Eunomia",
  outputFolder = "C:/dataQualityExample")
```

예제 15.3

데이터의 질 검사 목록을 보기 위해서는:

```
DataQualityDashboard::viewDqDashboard(
  "C:/dataQualityExample/Eunomia/results_Eunomia.json")
```

Bibliography

- Allison, D. B., Brown, A. W., George, B. J., and Kaiser, K. A. (2016). Reproducibility: A tragedy of errors. *Nature*, 530(7588):27–29.
- Arnold, B. F., Ercumen, A., Benjamin-Chung, J., and Colford, J. M. (2016). Brief Report: Negative Controls to Detect Selection Bias and Measurement Bias in Epidemiologic Studies. *Epidemiology*, 27(5):637–641.
- Austin, P. C. (2011). Optimal caliper widths for propensity-score matching when estimating differences in means and differences in proportions in observational studies. *Pharmaceutical statistics*, 10(2):150–161.
- Banda, J. M., Halpern, Y., Sontag, D., and Shah, N. H. (2017). Electronic phenotyping with APHRODITE and the Observational Health Sciences and Informatics (OHDSI) data network. *AMIA Jt Summits Transl Sci Proc*, 2017:48–57.
- Boland, M. R., Parhi, P., Li, L., Miotto, R., Carroll, R., Iqbal, U., Nguyen, P. A., Schuemie, M., You, S. C., Smith, D., Mooney, S., Ryan, P., Li, Y. J., Park, R. W., Denny, J., Dudley, J. T., Hripcsak, G., Gentine, P., and Tatonetti, N. P. (2017). Uncovering exposures responsible for birth season - disease effects: a global study. *J Am Med Inform Assoc*.
- Botsis, T., Hartvigsen, G., Chen, F., and Weng, C. (2010). Secondary use of ehr: data quality issues and informatics opportunities. *Summit on Translational Bioinformatics*, 2010:1.
- Burn, E., Weaver, J., Morales, D., Prats-Uribe, A., Delmestri, A., Strauss, V. Y., He, Y., Robinson, D. E., Pinedo-Villanueva, R., Kolovos, S., Duarte-Salles, T., Sproviero, W., Yu, D., Speybroeck, M. V., Williams, R., John, L. H., Hughes, N., Sena, A. G., Costello, R., Birlie, B., Culliford, D., O’Leary, C., Morgan, H., Burkard, T., Prieto-Alhambra, D., and Ryan, P. (2019). Opioid use,

- postoperative complications, and implant survival after unicompartmental versus total knee replacement: a population-based network study. *The Lancet Rheumatology*, 1(4):e229–e236.
- Byrd, J. B., Adam, A., and Brown, N. J. (2006). Angiotensin-converting enzyme inhibitor-associated angioedema. *Immunol Allergy Clin North Am*, 26(4):725–737.
- Callahan, T. J., Bauck, A. E., Bertoche, D., Brown, J., Khare, R., Ryan, P. B., Staab, J., Zozus, M. N., and Kahn, M. G. (2017). A comparison of data quality assessment checks in six data sharing networks. *eGEMs*, 5(1).
- Cepeda, M. S., Reps, J., Fife, D., Blacketer, C., Stang, P., and Ryan, P. (2018). Finding treatment-resistant depression in real-world data: How a data-driven approach compares with expert-based heuristics. *Depress Anxiety*, 35(3):220–228.
- Chen, X., Dallmeier-Tiessen, S., Dasler, R., Feger, S., Fokianos, P., Gonzalez, J. B., Hirvonsalo, H., Kousidis, D., Lavasa, A., Mele, S., Rodriguez, D. R., Šimko, T., Smith, T., Trisovic, A., Trzcinska, A., Tsanaktsidis, I., Zimmermann, M., Cranmer, K., Heinrich, L., Watts, G., Hildreth, M., Iglesias, L. L., Lassila-Perini, K., and Neubert, S. (2018). Open is not enough. *Nature Physics*, 15(2):113–119.
- Cicardi, M., Zingale, L. C., Bergamaschini, L., and Agostoni, A. (2004). Angioedema associated with angiotensin-converting enzyme inhibitor use: outcome after switching to a different treatment. *Arch. Intern. Med.*, 164(8):910–913.
- Dasu, T. and Johnson, T. (2003). *Exploratory data mining and data cleaning*, volume 479. John Wiley & Sons.
- Defalco, F. J., Ryan, P. B., and Soledad Cepeda, M. (2013). Applying standardized drug terminologies to observational healthcare databases: a case study on opioid exposure. *Health Serv Outcomes Res Methodol*, 13(1):58–67.
- DerSimonian, R. and Laird, N. (1986). Meta-analysis in clinical trials. *Control Clin Trials*, 7(3):177–188.
- Duke, J. D., Ryan, P. B., Suchard, M. A., Hripcsak, G., Jin, P., Reich, C., Schwalm, M. S., Khoma, Y., Wu, Y., Xu, H., Shah, N. H., Banda, J. M., and Schuemie, M. J. (2017). Risk of angioedema associated with levetiracetam compared with phenytoin: Findings of the observational health data sciences and informatics research network. *Epilepsia*, 58(8):e101–e106.
- Farrington, C. P. (1995). Relative incidence estimation from case series for vaccine safety evaluation. *Biometrics*, 51(1):228–235.

- Farrington, C. P., Anaya-Izquierdo, K., Whitaker, H. J., Hocine, M. N., Douglas, I., and Smeeth, L. (2011). Self-controlled case series analysis with event-dependent observation periods. *Journal of the American Statistical Association*, 106(494):417–426.
- Fuller, W. A. (2009). *Measurement error models*, volume 305. John Wiley & Sons.
- Garza, M., Del Fiol, G., Tenenbaum, J., Walden, A., and Zozus, M. N. (2016). Evaluating common data models for use with a longitudinal community registry. *J Biomed Inform*, 64:333–341.
- Hernan, M. A., Hernandez-Diaz, S., Werler, M. M., and Mitchell, A. A. (2002). Causal knowledge as a prerequisite for confounding evaluation: an application to birth defects epidemiology. *Am. J. Epidemiol.*, 155(2):176–184.
- Hernan, M. A. and Robins, J. M. (2016). Using Big Data to Emulate a Target Trial When a Randomized Trial Is Not Available. *Am. J. Epidemiol.*, 183(8):758–764.
- Hersh, W. R., Weiner, M. G., Embi, P. J., Logan, J. R., Payne, P. R., Bernstam, E. V., Lehmann, H. P., Hripcsak, G., Hartzog, T. H., Cimino, J. J., et al. (2013). Caveats for the use of operational electronic health record data in comparative effectiveness research. *Medical care*, 51(8 0 3):S30.
- Higgins, J. P., Thompson, S. G., Deeks, J. J., and Altman, D. G. (2003). Measuring inconsistency in meta-analyses. *BMJ*, 327(7414):557–560.
- Hill, A. B. (1965). THE ENVIRONMENT AND DISEASE: ASSOCIATION OR CAUSATION? *Proc. R. Soc. Med.*, 58:295–300.
- Hripcsak, G. and Albers, D. J. (2017). High-fidelity phenotyping: richness and freedom from bias. *J Am Med Inform Assoc*.
- Hripcsak, G., Duke, J. D., Shah, N. H., Reich, C. G., Huser, V., Schuemie, M. J., Suchard, M. A., Park, R. W., Wong, I. C. K., Rijnbeek, P. R., van der Lei, J., Pratt, N., Norén, G. N., Li, Y.-C., Stang, P. E., Madigan, D., and Ryan, P. B. (2015). Observational Health Data Sciences and Informatics (OHDSI): Opportunities for Observational Researchers. *Studies in health technology and informatics*, 216:574–578.
- Hripcsak, G., Levine, M. E., Shang, N., and Ryan, P. B. (2018). Effect of vocabulary mapping for conditions on phenotype cohorts. *J Am Med Inform Assoc*, 25(12):1618–1625.

- Hripcsak, G., Ryan, P. B., Duke, J. D., Shah, N. H., Park, R. W., Huser, V., Suchard, M. A., Schuemie, M. J., DeFalco, F. J., Perotte, A., Banda, J. M., Reich, C. G., Schilling, L. M., Matheny, M. E., Meeker, D., Pratt, N., and Madigan, D. (2016). Characterizing treatment pathways at scale using the OHDSI network. *Proceedings of the National Academy of Sciences*, 113(27):7329–7336.
- Hripcsak, G., Shang, N., Peissig, P. L., Rasmussen, L. V., Liu, C., Benoit, B., Carroll, R. J., Carrell, D. S., Denny, J. C., Dikilitas, O., Gainer, V. S., Marie Howell, K., Klann, J. G., Kullo, I. J., Lingren, T., Mentch, F. D., Murphy, S. N., Natarajan, K., Pacheco, J. A., Wei, W. Q., Wiley, K., and Weng, C. (2019). Facilitating phenotype transfer using a common data model. *J Biomed Inform*, page 103253.
- Huser, V., DeFalco, F. J., Schuemie, M., Ryan, P. B., Shang, N., Velez, M., Park, R. W., Boyce, R. D., Duke, J., Khare, R., Utidjian, L., and Bailey, C. (2016). Multisite Evaluation of a Data Quality Tool for Patient-Level Clinical Data Sets. *EGEMS (Washington, DC)*, 4(1):1239.
- Huser, V., Kahn, M. G., Brown, J. S., and Gouripeddi, R. (2018). Methods for examining data quality in healthcare integrated data repositories. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 23:628–633.
- Johnston, S. S., Morton, J. M., Kalsekar, I., Ammann, E. M., Hsiao, C. W., and Reps, J. (2019). Using Machine Learning Applied to Real-World Healthcare Data for Predictive Analytics: An Applied Example in Bariatric Surgery. *Value Health*, 22(5):580–586.
- Kahn, M. G., Brown, J. S., Chun, A. T., Davidson, B. N., Meeker, D., Ryan, P. B., Schilling, L. M., Weiskopf, N. G., Williams, A. E., and Zozus, M. N. (2015). Transparent reporting of data quality in distributed data networks. *EGEMS (Washington, DC)*, 3(1):1052.
- Kahn, M. G., Callahan, T. J., Barnard, J., Bauck, A. E., Brown, J., Davidson, B. N., Estiri, H., Goerg, C., Holve, E., Johnson, S. G., Liaw, S.-T., Hamilton-Lopez, M., Meeker, D., Ong, T. C., Ryan, P. B., Shang, N., Weiskopf, N. G., Weng, C., Zozus, M. N., and Schilling, L. (2016). A Harmonized Data Quality Assessment Terminology and Framework for the Secondary Use of Electronic Health Record Data. *EGEMS (Washington, DC)*, 4(1):1244.
- Kahn, M. G., Raebel, M. A., Glanz, J. M., Riedlinger, K., and Steiner, J. F. (2012). A pragmatic framework for single-site and multisite data quality assessment in electronic health record-based clinical research. *Medical care*, 50.

- Liaw, S.-T., Rahimi, A., Ray, P., Taggart, J., Dennis, S., de Lusignan, S., Jalaludin, B., Yeo, A., and Talaei-Khoei, A. (2013). Towards an ontology for data quality in integrated chronic disease management: a realist review of the literature. *International journal of medical informatics*, 82(1):10–24.
- Lipsitch, M., Tchetgen Tchetgen, E., and Cohen, T. (2010). Negative controls: a tool for detecting confounding and bias in observational studies. *Epidemiology*, 21(3):383–388.
- MacLure, M. (1991). The case-crossover design: a method for studying transient effects on the risk of acute events. *Am. J. Epidemiol.*, 133(2):144–153.
- Madigan, D., Ryan, P. B., and Schuemie, M. (2013a). Does design matter? Systematic evaluation of the impact of analytical choices on effect estimates in observational studies. *Ther Adv Drug Saf*, 4(2):53–62.
- Madigan, D., Ryan, P. B., Schuemie, M., Stang, P. E., Overhage, J. M., Hartzema, A. G., Suchard, M. A., DuMouchel, W., and Berlin, J. A. (2013b). Evaluating the impact of database heterogeneity on observational study results. *Am. J. Epidemiol.*, 178(4):645–651.
- Magid, D. J., Shetterly, S. M., Margolis, K. L., Tavel, H. M., O'Connor, P. J., Selby, J. V., and Ho, P. M. (2010). Comparative effectiveness of angiotensin-converting enzyme inhibitors versus beta-blockers as second-line therapy for hypertension. *Circ Cardiovasc Qual Outcomes*, 3(5):453–458.
- Makadia, R. and Ryan, P. B. (2014). Transforming the Premier Perspective Hospital Database into the Observational Medical Outcomes Partnership (OMOP) Common Data Model. *EGEMS (Wash DC)*, 2(1):1110.
- Matcho, A., Ryan, P., Fife, D., and Reich, C. (2014). Fidelity assessment of a clinical practice research datalink conversion to the OMOP common data model. *Drug Saf*, 37(11):945–959.
- Noren, G. N., Caster, O., Juhlin, K., and Lindquist, M. (2014). Zoo or savannah? Choice of training ground for evidence-based pharmacovigilance. *Drug Saf*, 37(9):655–659.
- Norman, J. L., Holmes, W. L., Bell, W. A., and Finks, S. W. (2013). Life-threatening ACE inhibitor-induced angioedema after eleven years on lisinopril. *J Pharm Pract*, 26(4):382–388.
- Oliveira, J. L., Trifan, A., and Silva, L. A. B. (2019). EMIF catalogue: A collaborative platform for sharing and reusing biomedical data. *International Journal of Medical Informatics*, 126:35–45.

- Olsen, L., Aisner, D., McGinnis, J. M., et al. (2007). *The learning healthcare system: workshop summary*. Natl Academy Pr.
- O'Mara, N. B. and O'Mara, E. M. (1996). Delayed onset of angioedema with angiotensin-converting enzyme inhibitors: case report and review of the literature. *Pharmacotherapy*, 16(4):675–679.
- Overhage, J. M., Ryan, P. B., Reich, C. G., Hartzema, A. G., and Stang, P. E. (2012). Validation of a common data model for active safety surveillance research. *J Am Med Inform Assoc*, 19(1):54–60.
- Perkins, N. J., Cole, S. R., Harel, O., Tchetgen Tchetgen, E. J., Sun, B., Mitchell, E. M., and Schisterman, E. F. (2017). Principled approaches to missing data in epidemiologic studies. *American journal of epidemiology*, 187(3):568–575.
- Powers, B. J., Coeytaux, R. R., Dolor, R. J., Hasselblad, V., Patel, U. D., Yancy, W. S., Gray, R. N., Irvine, R. J., Kendrick, A. S., and Sanders, G. D. (2012). Updated report on comparative effectiveness of ACE inhibitors, ARBs, and direct renin inhibitors for patients with essential hypertension: much more data, little new information. *J Gen Intern Med*, 27(6):716–729.
- Prasad, V. and Jena, A. B. (2013). Prespecified falsification end points: can they validate true observational associations? *JAMA*, 309(3):241–242.
- Ramcharan, D., Qiu, H., Schuemie, M. J., and Ryan, P. B. (2017). Atypical Antipsychotics and the Risk of Falls and Fractures Among Older Adults: An Emulation Analysis and an Evaluation of Additional Confounding Control Strategies. *J Clin Psychopharmacol*, 37(2):162–168.
- Rassen, J. A., Shelat, A. A., Myers, J., Glynn, R. J., Rothman, K. J., and Schneeweiss, S. (2012). One-to-many propensity score matching in cohort studies. *Pharmacoepidemiol Drug Saf*, 21 Suppl 2:69–80.
- Reps, J. M., Rijnbeek, P. R., and Ryan, P. B. (2019). Identifying the DEAD: Development and Validation of a Patient-Level Model to Predict Death Status in Population-Level Claims Data. *Drug Saf*.
- Reps, J. M., Schuemie, M. J., Suchard, M. A., Ryan, P. B., and Rijnbeek, P. R. (2018). Design and implementation of a standardized framework to generate and evaluate patient-level prediction models using observational healthcare data. *Journal of the American Medical Informatics Association*, 25(8):969–975.

- Roebuck, K. (2012). *Data quality: high-impact strategies-what you need to know: definitions, adoptions, impact, benefits, maturity, vendors*. Emereo Publishing.
- Rosenbaum, P. (2005). *Sensitivity Analysis in Observational Studies*. American Cancer Society.
- Rosenbaum, P. and Rubin, D. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70:41–55.
- Rubbo, B., Fitzpatrick, N. K., Denaxas, S., Daskalopoulou, M., Yu, N., Patel, R. S., Hemingway, H., Danesh, J., Allen, N., Atkinson, M., Blaveri, E., Brannan, R., Brayne, C., Brophy, S., Chaturvedi, N., Collins, R., deLusignan, S., Denaxas, S., Desai, P., Eastwood, S., Gallacher, J., Hemingway, H., Hotopf, M., Landray, M., Lyons, R., O’Neil, T., Pringle, M., Sprosen, T., Strachan, D., Sudlow, C., Sullivan, F., Zhang, Q., and Flaig, R. (2015). Use of electronic health records to ascertain, validate and phenotype acute myocardial infarction: A systematic review and recommendations. *Int. J. Cardiol.*, 187:705–711.
- Rubin, D. B. (2001). Using propensity scores to help design observational studies: application to the tobacco litigation. *Health Services and Outcomes Research Methodology*, 2(3-4):169–188.
- Ryan, P. B., Buse, J. B., Schuemie, M. J., DeFalco, F., Yuan, Z., Stang, P. E., Berlin, J. A., and Rosenthal, N. (2018). Comparative effectiveness of canagliflozin, SGLT2 inhibitors and non-SGLT2 inhibitors on the risk of hospitalization for heart failure and amputation in patients with type 2 diabetes mellitus: A real-world meta-analysis of 4 observational databases (OBSERVE-4D). *Diabetes Obes Metab*, 20(11):2585–2597.
- Ryan, P. B., Madigan, D., Stang, P. E., Overhage, J. M., Racoosin, J. A., and Hartzema, A. G. (2012). Empirical assessment of methods for risk identification in healthcare data: results from the experiments of the Observational Medical Outcomes Partnership. *Stat Med*, 31(30):4401–4415.
- Ryan, P. B., Schuemie, M. J., and Madigan, D. (2013a). Empirical performance of a self-controlled cohort method: lessons for developing a risk identification and analysis system. *Drug Saf*, 36 Suppl 1:95–106.
- Ryan, P. B., Schuemie, M. J., Ramcharan, D., and Stang, P. E. (2017). Atypical Antipsychotics and the Risks of Acute Kidney Injury and Related Outcomes Among Older Adults: A Replication Analysis and an Evaluation of Adapted Confounding Control Strategies. *Drugs Aging*, 34(3):211–219.

- Ryan, P. B., Stang, P. E., Overhage, J. M., Suchard, M. A., Hartzema, A. G., DuMouchel, W., Reich, C. G., Schuemie, M. J., and Madigan, D. (2013b). A comparison of the empirical performance of methods for a risk identification system. *Drug Saf*, 36 Suppl 1:S143–158.
- Sabroe, R. A. and Black, A. K. (1997). Angiotensin-converting enzyme (ACE) inhibitors and angio-oedema. *Br. J. Dermatol.*, 136(2):153–158.
- Schneeweiss, S. (2018). Automated data-adaptive analytics for electronic healthcare data to study causal treatment effects. *Clin Epidemiol*, 10:771–788.
- Schuemie, M. J., Hripcsak, G., Ryan, P. B., Madigan, D., and Suchard, M. A. (2016). Robust empirical calibration of p-values using observational data. *Stat Med*, 35(22):3883–3888.
- Schuemie, M. J., Hripcsak, G., Ryan, P. B., Madigan, D., and Suchard, M. A. (2018a). Empirical confidence interval calibration for population-level effect estimation studies in observational healthcare data. *Proc. Natl. Acad. Sci. U.S.A.*, 115(11):2571–2577.
- Schuemie, M. J., Ryan, P. B., DuMouchel, W., Suchard, M. A., and Madigan, D. (2014). Interpreting observational studies: why empirical calibration is needed to correct p-values. *Stat Med*, 33(2):209–218.
- Schuemie, M. J., Ryan, P. B., Hripcsak, G., Madigan, D., and Suchard, M. A. (2018b). Improving reproducibility by using high-throughput observational studies with empirical calibration. *Philos Trans A Math Phys Eng Sci*, 376(2128).
- Sherman, R. E., Anderson, S. A., Dal Pan, G. J., Gray, G. W., Gross, T., Hunter, N. L., LaVange, L., Marinac-Dabic, D., Marks, P. W., Robb, M. A., et al. (2016). Real-world evidence—what is it and what can it tell us. *N Engl J Med*, 375(23):2293–2297.
- Simpson, S. E., Madigan, D., Zorych, I., Schuemie, M. J., Ryan, P. B., and Suchard, M. A. (2013). Multiple self-controlled case series for large-scale longitudinal observational databases. *Biometrics*, 69(4):893–902.
- Slater, E. E., Merrill, D. D., Guess, H. A., Roylance, P. J., Cooper, W. D., Inman, W. H. W., and Ewan, P. W. (1988). Clinical Profile of Angioedema Associated With Angiotensin Converting-Enzyme Inhibition. *JAMA*, 260(7):967–970.
- Stang, P. E., Ryan, P. B., Racoosin, J. A., Overhage, J. M., Hartzema, A. G., Reich, C., Welebob, E., Scarneccchia, T., and Woodcock, J. (2010). Advancing the science for active surveillance: rationale and design for the Observational Medical Outcomes Partnership. *Ann. Intern. Med.*, 153(9):600–606.

- Suchard, M. A., Schuemie, M. J., Krumholz, H. M., You, S. C., Chen, R., Pratt, N., Reich, C. G., Duke, J., Madigan, D., Hripcak, G., and Ryan, P. B. (2019). Comprehensive comparative effectiveness and safety of first-line antihypertensive drug classes: a systematic, multinational, large-scale analysis. *The Lancet*, 0(0).
- Suchard, M. A., Simpson, S. E., Zorych, I., Ryan, P. B., and Madigan, D. (2013). Massive parallelization of serial inference algorithms for a complex generalized linear model. *ACM Trans. Model. Comput. Simul.*, 23(1):10:1–10:17.
- Suissa, S. (1995). The case-time-control design. *Epidemiology*, 6(3):248–253.
- Swerdel, J. N., Hripcak, G., and Ryan, P. B. (2019). PheEvaluator: Development and Evaluation of a Phenotype Algorithm Evaluator. *J Biomed Inform*, page 103258.
- Thompson, T. and Frable, M. A. (1993). Drug-induced, life-threatening angioedema revisited. *Laryngoscope*, 103(1 Pt 1):10–12.
- Tian, Y., Schuemie, M. J., and Suchard, M. A. (2018). Evaluating large-scale propensity score performance through real-world and synthetic data experiments. *Int J Epidemiol*, 47(6):2005–2014.
- Toh, S., Reichman, M. E., Houstoun, M., Ross Southworth, M., Ding, X., Hernandez, A. F., Levenson, M., Li, L., McCloskey, C., Shoaibi, A., Wu, E., Zornberg, G., and Hennessy, S. (2012). Comparative risk for angioedema associated with the use of drugs that target the renin-angiotensin-aldosterone system. *Arch. Intern. Med.*, 172(20):1582–1589.
- van der Lei, J. (1991). Use and abuse of computer-stored medical records. *Methods of information in medicine*, 30(02):79–80.
- Vandenbroucke, J. P. and Pearce, N. (2012). Case-control studies: basic concepts. *Int J Epidemiol*, 41(5):1480–1489.
- Vashisht, R., Jung, K., Schuler, A., Banda, J. M., Park, R. W., Jin, S., Li, L., Dudley, J. T., Johnson, K. W., Shervey, M. M., Xu, H., Wu, Y., Natrajan, K., Hripcak, G., Jin, P., Van Zandt, M., Reckard, A., Reich, C. G., Weaver, J., Schuemie, M. J., Ryan, P. B., Callahan, A., and Shah, N. H. (2018). Association of Hemoglobin A1c Levels With Use of Sulfonylureas, Dipeptidyl Peptidase 4 Inhibitors, and Thiazolidinediones in Patients With Type 2 Diabetes Treated With Metformin: Analysis From the Observational Health Data Sciences and Informatics Initiative. *JAMA Netw Open*, 1(4):e181755.

- von Elm, E., Altman, D. G., Egger, M., Pocock, S. J., Gøtzsche, P. C., and Vandenbroucke, J. P. (2008). The strengthening the reporting of observational studies in epidemiology (strobe) statement: guidelines for reporting observational studies. *Journal of Clinical Epidemiology*, 61(4):344 – 349.
- Voss, E. A., Boyce, R. D., Ryan, P. B., van der Lei, J., Rijnbeek, P. R., and Schuemie, M. J. (2016). Accuracy of an Automated Knowledge Base for Identifying Drug Adverse Reactions. *J Biomed Inform.*
- Voss, E. A., Ma, Q., and Ryan, P. B. (2015a). The impact of standardizing the definition of visits on the consistency of multi-database observational health research. *BMC Med Res Methodol*, 15:13.
- Voss, E. A., Makadia, R., Matcho, A., Ma, Q., Knoll, C., Schuemie, M., DeFalco, F. J., Londhe, A., Zhu, V., and Ryan, P. B. (2015b). Feasibility and utility of applications of the common data model to multiple, disparate observational health databases. *J Am Med Inform Assoc*, 22(3):553–564.
- Walker, A. M., Patrick, A. R., Lauer, M. S., Hornbrook, M. C., Marin, M. G., Platt, R., Roger, V. L., Stang, P., and Schneeweiss, S. (2013). A tool for assessing the feasibility of comparative effectiveness research. *Comp Eff Res*, 3:11–20.
- Wang, Y., Desai, M., Ryan, P. B., DeFalco, F. J., Schuemie, M. J., Stang, P. E., Berlin, J. A., and Yuan, Z. (2017). Incidence of diabetic ketoacidosis among patients with type 2 diabetes mellitus treated with SGLT2 inhibitors and other antihyperglycemic agents. *Diabetes Res. Clin. Pract.*, 128:83–90.
- Weinstein, R. B., Ryan, P., Berlin, J. A., Matcho, A., Schuemie, M., Swerdel, J., Patel, K., and Fife, D. (2017). Channeling in the Use of Nonprescription Paracetamol and Ibuprofen in an Electronic Medical Records Database: Evidence and Implications. *Drug Saf*, 40(12):1279–1292.
- Weiskopf, N. G. and Weng, C. (2013). Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *Journal of the American Medical Informatics Association: JAMIA*, 20(1):144–151.
- Whelton, P. K., Carey, R. M., Aronow, W. S., Casey, D. E., Collins, K. J., Dennison Himmelfarb, C., DePalma, S. M., Gidding, S., Jamerson, K. A., Jones, D. W., MacLaughlin, E. J., Muntner, P., Ovbiagele, B., Smith, S. C., Spencer, C. C., Stafford, R. S., Taler, S. J., Thomas, R. J., Williams, K. A., Williamson, J. D., and Wright, J. T. (2017). 2017

- ACC/AHA/AAPA/ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA Guideline for the Prevention, Detection, Evaluation, and Management of High Blood Pressure in Adults: Executive Summary: A Report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines. *Circulation*, 138(17):e426–e483.
- Whitaker, H. J., Farrington, C. P., Spiessens, B., and Musonda, P. (2006). Tutorial in biostatistics: the self-controlled case series method. *Stat Med*, 25(10):1768–1797.
- Who, A. (2013). Global brief on hypertension. *World Health Organization*.
- Wickham, H. (2015). *R Packages*. O'Reilly Media, Inc., 1st edition.
- Wikipedia (2019a). Open science — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Open%20science&oldid=900178688>. [Online; accessed 24-June-2019].
- Wikipedia (2019b). Science 2.0 — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Science%202.0&oldid=887565958>. [Online; accessed 09-July-2019].
- Wikiquote (2019). Ronald fisher — wikiquote,. [Online; accessed 2-August-2019].
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J. W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 't Hoen, P. A., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S. A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., and Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*, 3:160018.
- Yoon, D., Ahn, E. K., Park, M. Y., Cho, S. Y., Ryan, P., Schuemie, M. J., Shin, D., Park, H., and Park, R. W. (2016). Conversion and Data Quality Assessment of Electronic Health Record Data at a Korean Tertiary Teaching Hospital to a Common Data Model for Distributed Network Research. *Healthc Inform Res*, 22(1):54–58.
- You, S. C., Park, H., Yoon, D., Park, S., Joung, B., and Park, R. W. (2019). Olmesartan is not associated with the risk of enteropathy: a Korean nationwide observational cohort study. *The Korean Journal of Internal Medicine*, 34(1):90–98.

- Yuan, Z., DeFalco, F. J., Ryan, P. B., Schuemie, M. J., Stang, P. E., Berlin, J. A., Desai, M., and Rosenthal, N. (2018). Risk of lower extremity amputations in people with type 2 diabetes mellitus treated with sodium-glucose co-transporter-2 inhibitors in the USA: A retrospective cohort study. *Diabetes Obes Metab*, 20(3):582–589.
- Zaadstra, B. M., Chorus, A. M., van Buuren, S., Kalsbeek, H., and van Noort, J. M. (2008). Selective association of multiple sclerosis with infectious mononucleosis. *Mult. Scler.*, 14(3):307–313.
- Zaman, M. A., Oparil, S., and Calhoun, D. A. (2002). Drugs targeting the renin-angiotensin-aldosterone system. *Nat Rev Drug Discov*, 1(8):621–636.

Index

- accuracy, 279
- ACE inhibitors, 283
- ACHILLES, 330
- adaboost, 275
- agnostic SQL, *see* SqlRender
- analysis implementation, 117
- angioedema, 283
- APHRODITE, 168
- arachne, 417
- area under the precision-recall curve, 282
- ATHENA, 62
- ATLAS, 120, 285
 - characterization features, 207
 - cohort characterization, 121
 - cohort definitions, 121
 - cohort pathways, 121
 - concept sets, 121
 - configuration, 121
- Data Sources, 120
- documentation, 122
- feedback, 122
- incidence rates, 121
- installation, 122
- jobs, 121
- patient level prediction, 121
- population level estimation, 121
- profiles, 121
- security, 122
- vocabulary search, 121
- attrition diagram, 261
- AUC, 281
- back-propagation, 277
- balance, *see* covariate balance
- baseline time, 194
- best practice for network research, 418
- between-database heterogeneity, 381
- bigknn, 275
- Bill of Mortality, 59
- calibration, 282
- caliper, 228
 - scale, 242
- case-control design, 229
- case-crossover design, 230
- case-time-control design, 230
- CDM , *see* Common Data Model
- chapters, 13, 20
- characterization, 109, 194
 - cohort, 194
 - database level, 194
 - treatment pathways, 195
- classes, 270

- classification concept, 66
- Clem McDonald, 323
- clinical decision making, 267
- clinical equipoise, 228
- clinical validity, 345
- code set, 164
- cohort, 164
 - entry event, 166
 - exit criteria, 167
 - inclusion criteria, 166
 - probabilistic design, 168
 - rule-based design, 165
- cohort definition, 164
- cohort method, 224
- colliders, 227
- Common Data Model, 33
 - backwards compatibility, 36
 - conventions, 36
 - data loss prevention, 35
 - data model diagram, 34
 - data protection, 35
 - design principles, 34
 - domains, 35
 - scalability, 36
 - Source Codes, 35
 - standardized tables, 42
 - suitability for purpose, 35
 - technology neutrality, 35
- common data model, 365
- community, 13, 323, 324
 - community calls, 17
- comparative effect estimation, 223
- comparative effectiveness, *see* comparative effect estimation
- comparator cohort, 224
- concept, 62
 - ancestor, 75
 - class, 65
 - code, 68
 - hierarchy, 73
 - mapping, 71
 - relationship, 71
- concept set, 167
- conditioned model, 245
- confidence interval calibration, 380
- confounder, 226
- control hypotheses, 124
- convolutional neural network, 278
- counterfactual, 223
- covariate balance, 228
- Cox proportional hazards model, *see* Cox regression
- Cox regression, 225
- cross-validation, 273, 279
- Cyclops, 273
- data profiling, *see* White Rabbit
- data quality, 323, 327
 - checks, 330
 - completeness, 329
 - conformance, 329
 - data quality check, 330
 - plausibility, 329
 - study-specific checks, 334
 - validation, 330
 - verification, 330
- Data Quality Dashboard, 330
- DatabaseConnector, 134
 - creating a connection, 144

- querying, 146
- decision boundary, 272
- decision tree, 276
- deep learning, 278
- descriptive statistics, *see* characterization
- design considerations for network research, 412
- diagnostic outcome, 267
- direct effect estimation, 223
- discrimination, 281
- disease natural history, *see* characterization
- domain
 - concept, 64
- drug utilization, 194
- empirical calibration, 378
- empirical evaluation, 378
- ETL, *see* extract, transform and load (ETL)
 - quality control, 100
 - unit tests, 332
- ETL design, *see* Rabbit-In-A-Hat
- evaluating prediction models, 278
- evidence quality, 321, 323
- FAIR, 27
- false negative, 280
- false positive, 280
- feature analyses, 202
- FeatureExtraction, 207
- forum, 15
- gradient boosting, 274
- high correlation, 248
- hyper-parameter, 273
- incidence, 196
- proportion, 196
- rate, 197
- index date, 194, 270, 282
- instrumental variables, 227
- join the journey, 13
- k-nearest neighbors, 275
- Kaplan-Meier plot, 262
- labels, 270
- LASSO, 273
- limitations of observational research, 114
- logistic regression, 225, 273
- logistics of network research, 413
- machine learning, 268
- method validity, 373
- methods library, 122
- minimum detectable relative risk (MDRR), 261
- missing data, 114, 271
- mission, 7
- model viewer app, 305
- naive bayes, 275
- native data, *see* raw data
- negative controls, 375
- nesting cohort
 - case-control design, 229
- network study, 410
- neural network, 277
- no free lunch, 273
- objectives, 8
- OHDSI Methods Benchmark, 390
- OHDSI SQL, *see* SqlRender
- open science, 23
- open data, 26

- open discourse, 27
- open source, 26
- open standards, 25
- orphan codes, 336
- outcome cohort, 268
 - case-control design, 229
 - case-crossover design, 230
 - cohort method, 224
 - SCCS design, 231
 - self-controlled cohort design, 228
- outcome status, 270
- p-value calibration, 378
- Pallas system, 61
- patient-level prediction, 112, 267
- perceptron, 277
- performance metrics, 279
- person-time, 197
- phenotype, 164
- phenotype library, 169
- PheEvaluator, 352
- Poisson regression, 225
- population-level estimation, 111, 223
- positive controls, 376
 - synthesis, 377
- positive predictive value, 279
- post-index time, 194
- power, 261
- prediction model, 267
- preference score, 228
- prognostic outcome, 267
- programming best practices, 366
- propensity model, 226
 - example, 258
- propensity score, 226
- matching, 226
- stratification, 226
- trimming, 242
- weighting, 226
- protocol, 398
- python, 274–276
- quality improvement, *see* characterization
- Query Library, 134
- QueryLibrary, 152
- R, 134
 - installation, 126
- Rabbit-In-A-Hat, 86
- random forest, 274
- randomized trial, 226
- raw data, 81
- recurrent neural networks, 278
- regularization, 273
- regulatory decision-making, 323
- relational data model, *see* Common Data Model
- reliable evidence, 324
- research network, 409
- ROC, 281
- running network research, 414
- safety surveillance, 224
- self-controlled case series (SCCS) design, 231
- self-controlled cohort design, 228
- sensitivity, 279
- sensitivity analysis, 381
- sklearn, 274
- software development process, 365
- software validity, 365
- source code mapping, *see* Usagi
- source data, *see* raw data

- source record verification, 349
specificity, 279
SQL, 133
SQL Query Library, *see* Query Library
SqlRender, 134
 debugging, 143
 parameterization, 135
 supported functions, 137
 translation, 137
standard concept, 66
Standard SQL Dialect, *see* SqlRender
standardized vocabularies, 59
 download, 62
 search, 62
stratified model, *see* conditioned model
strongly ignorable, 227
structured query language, *see* SQL
study code validity, 366
study diagnostics, 374, 404
study feasibility
 single study, 404
study package, 399
study-a-thon, 25
supervised learning, 270
survival plot, *see* Kaplan-Meier plot
system requirements, 125
- target cohort, 268
 case-control design, 229
 case-crossover design, 230
 cohort method, 224
 SCCS design, 231
 self-controlled cohort design, 228
time-at-risk, 268
tools deployment, 130
- Amazon AWS, 130
Broadsea, 130
treatment utilization, *see* characterization
TRIPOD, 268
true negative, 280
true positive, 280
- Usagi, 93
- validation
 external validation, 278
 internal validation, 278
 spatial validation, 279
 temporal validation, 279
- variable ratio matching, 226
variance, 273
vignette, 125
vision, 7
vocabulary, 64
- White Rabbit, 82
workgroups, 13, 17
- xgboost, 274
xSens cohort, 354
xSpec cohort, 354