

-翻訳作業中- OHDSI の本

Observational Health Data Sciences and Informatics (OHDSI)

2025-03-10

Contents

序章	ix
この本の目標	ix
本書の構成	ix
貢献者	x
ソフトウェアのバージョン	x
ライセンス	xi
本書が作成された方法	xi
 I OHDSI コミュニティ	 1
1 OHDSI コミュニティ	3
1.1 データからエビデンスへの旅	3
1.2 観察医療アウトカムパートナーシップ (OMOP)	5
1.3 オープンサイエンスの協働組織としての OHDSI	6
1.4 OHDSI の進展	7
1.5 OHDSI における協力	9
1.6 まとめ	9
 2 どこから始めようか	 11
2.1 旅に参加しよう	11
2.2 どこにフィットするか	20
2.3 まとめ	22
 3 オープンサイエンス	 23
3.1 オープンサイエンス	23
3.2 オープンサイエンスの実践: the Study-a-Thon	25
3.3 オープンスタンダード	25
3.4 オープンソース	26
3.5 オープンデータ	26
3.6 オープンな議論	27
3.7 OHDSI と FAIR ガイディングプリンシブルズ	27

II 共通データモデル	31
4 共通データモデル	33
4.1 デザインの原則	34
4.2 データモデルの規約	35
4.3 CDM 標準化テーブル	42
4.4 追加情報	59
4.5 まとめ	59
4.6 演習	59
5 標準化ボキャブラリ	61
5.1 なぜボキャブラリが必要で、なぜ標準化が必要なのか	61
5.2 コンセプト	64
5.3 関係	72
5.4 階層	75
5.5 内部参照テーブル	77
5.6 特別な状況	77
5.7 まとめ	79
5.8 演習	80
6 ETL (抽出-変換-読込)	81
6.1 はじめに	81
6.2 ステップ 1: ETL のデザイン	81
6.3 ステップ 2: コードマッピングの作成	92
6.4 ステップ 3: ETL の実装	100
6.5 ステップ 4: 品質管理	101
6.6 ETL の規約と THEMIS	102
6.7 CDM および ETL のメンテナンス	103
6.8 ETL に関する最終的な考察	104
6.9 まとめ	104
6.10 演習	105
III データ解析	107
7 データ解析の使用例	109
7.1 特性評価	109
7.2 集団レベルの推定	110
7.3 患者レベルの予測	111
7.4 高血圧症におけるユースケース	112
7.5 観察研究の限界	113
7.6 まとめ	114
7.7 演習	114
8 OHDSI 分析ツール	115

8.1 分析の実装	115
8.2 分析戦略	116
8.3 ATLAS	117
8.4 Methods Library	120
8.5 展開戦略	128
8.6 まとめ	129
9 SQL と R	131
9.1 SqlRender	132
9.2 DatabaseConnector	141
9.3 CDM へのクエリ	144
9.4 クエリ実行時にボキャブラリを使用する	147
9.5 QueryLibrary	148
9.6 簡単な研究のデザイン	149
9.7 SQL と R を使用した研究の実施	150
9.8 まとめ	156
9.9 演習	156
10 コホートの定義	159
10.1 コホートとは?	160
10.2 ルールベースのコホート定義	161
10.3 コンセプトセット	163
10.4 確率的コホート定義	163
10.5 コホート定義の妥当性	164
10.6 高血圧のコホート定義	165
10.7 ATLAS を用いたコホートの実装	165
10.8 SQL を使用したコホートの実装	176
10.9 要約	184
10.1 演習	184
11 特性評価	187
11.1 データベースレベルの特性評価	187
11.2 コホート特性評価	188
11.3 治療経路	188
11.4 発生率	189
11.5 高血圧症患者の特性評価	190
11.6 ATLAS におけるデータベースの特性評価	191
11.7 ATLAS におけるコホート特性分析	194
11.8 R でのコホートの特性評価	201
11.9 ATLAS におけるコホート経路分析	204
11.10 ATLAS における発生率分析	208
11.1 まとめ	212
11.1 演習	212
12 集団レベルの推定	215

12.1 コホートメソッドの設計	216
12.2 自己対照コホートデザイン	220
12.3 症例対照デザイン	220
12.4 ケース・クロスオーバーデザイン	221
12.5 自己対照症例シリーズデザイン	222
12.6 高血圧症研究のデザイン	223
12.7 ATLAS を使用した研究の実施	225
12.8 R を使用した研究の実施	239
12.9 研究の結果	248
12.1 まとめ	253
12.1 演習	254
13 患者レベルの予測	257
13.1 予測課題	258
13.2 データ抽出	260
13.3 モデルの適合	261
13.4 予測モデルの評価	266
13.5 患者レベル予測研究のデザイン	270
13.6 ATLAS での研究の実装	273
13.7 R での研究実施	286
13.8 結果の公表	293
13.9 患者レベルの予測に関する追加の機能	301
13.1 まとめ	303
13.1 演習	303
IV エビデンスの質	305
14 エビデンスの質	307
14.1 信頼できるエビデンスの属性	307
14.2 エビデンスの質の理解	309
14.3 エビデンスの質の伝達	310
14.4 まとめ	310
15 データ品質	313
15.1 データ品質問題の原因	314
15.2 一般的なデータ品質	314
15.3 研究特有のチェック	320
15.4 実践における ACHILLES	322
15.5 Data Quality Dashboard の実践	325
15.6 特定の研究チェックの実践	326
15.7 まとめ	329
15.8 演習	329
16 臨床的妥当性	331

16.1 医療データベースの特性	331
16.2 コホートバリデーション	332
16.3 ソースレコード検証	335
16.4 PheEvaluator	338
16.5 エビデンスの一般化可能性	349
16.6 まとめ	349
17 ソフトウェアの妥当性	351
17.1 研究コードの妥当性	351
17.2 Methods Library のソフトウェア開発プロセス	354
17.3 Methods Library のテスト	356
17.4 まとめ	357
18 方法の妥当性	359
18.1 デザイン特有の診断	360
18.2 推定のための診断	360
18.3 実践におけるメソッド検証	368
18.4 OHDSI メソッド評価ベンチマーク	377
18.5 まとめ	378
V OHDSI 研究	381
19 研究の段階	383
19.1 一般的なベストプラクティスガイドライン	384
19.2 詳細な研究手順	387
19.3 まとめ	393
20 OHDSI ネットワーク研究	395
20.1 OHDSI 研究ネットワークとして	395
20.2 OHDSI ネットワーク研究	396
20.3 OHDSI ネットワーク研究の実行	400
20.4 展望: ネットワーク研究の自動化を利用する	403
20.5 OHDSI ネットワーク研究のベストプラクティス	404
20.6 まとめ	406
Appendix	407
A 用語集	409
B コホート定義	415
B.1 ACE 阻害薬	415
B.2 ACE 阻害薬単剤療法新規ユーザー	416
B.3 急性心筋梗塞 (AMI)	419
B.4 血管性浮腫	420
B.5 サイアザイド様利尿薬単剤療法の新規ユーザー使用者	421

B.6 高血圧のための第一選択治療を開始する患者	424
B.7 追跡期間が 3 年以上ある高血圧のための第一選択治療を開始する患者	428
B.8 ACE 阻害薬の使用	428
B.9 アンジオテンシン受容体拮抗薬 (ARB) の使用	429
B.10 サイアザイドおよびサイアザイド様利尿薬の使用	429
B.11 ジヒドロピリジン系カルシウムチャネル遮断薬 (DCCB) の使用	430
B.12 非ジヒドロピリジン系カルシウムチャネル遮断薬 (NDCCB) の使用	430
B.13 ベータ遮断薬使用	431
B.14 ループ利尿薬使用	431
B.15 カリウム保持性利尿薬使用	431
B.16 アルファ 1 遮断薬使用	432
C ネガティブコントロール	433
C.1 ACE 阻害薬とサイアザイド・サイアザイド様利尿薬	433
D プロトコルテンプレート	437
E 回答例	439
E.1 共通データモデル	439
E.2 標準化ボキャブラリ	443
E.3 ETL (Extract-Transform-Load)	443
E.4 データ分析のユースケース	445
E.5 SQL と R	445
E.6 コホートの定義	447
E.7 特性評価	452
E.8 集団レベルの推定	459
E.9 患者レベルの予測	465
E.10 データ品質	467
E.11	468
Bibliography	469
Index	481

序章

これは、OHDSI コラボレーションについての本です。この本は、OHDSI コミュニティにより作成され、OHDSI に関するすべての知識の中心的なリポジトリとして役立つことを目指しています。この本はオープンソース開発ツールを通じてコミュニティにより維持される生きた文書であり、絶えず進化しています。オンライン版は無料で <http://book.ohdsi.org> から利用でき、常に最新バージョンを表示します。物理的なコピー（訳者注：英語版）はAmazon で原価価格で入手可能です。

この本の目標

この本は、OHDSI の中心的な知識リポジトリとなることを目的としており、OHDSI コミュニティ、OHDSI データ標準、および OHDSI ツールについて説明します。本書は、OHDSI の初心者とベテランの両方を対象としており、必要な理論とそれに続く手順を提供する実用的な内容を目指しています。本書を読んだ後には、OHDSI が何であり、どのようにしてその旅に参加できるかを理解できます。共通データモデルおよび標準ボキャブラリとは何か、また、それらが観察医療データベースを標準化のためにどのように使用されるかを学びます。これらのデータの主なユースケースである特性評価、集団レベルの推定、および患者レベルの予測について学びます。これらの 3 つすべての活動をサポートする OHDSI のオープンソースツールとその使用方法についても読みます。データ品質、臨床的妥当性、ソフトウェアの適切性、および方法の適切性に関する章は、創生されたエビデンスの質をどのように確立するかを説明します。最後に、分散された研究ネットワークでこれらの研究を実行するための OHDSI ツールの使用方法を学びます。

本書の構成

この本は 5 つの主要な部に分かれています：

- I. OHDSI コミュニティ
- II. 統一されたデータ表現
- III. データ分析

IV. エビデンスの質

V. OHDSI 研究

各部には複数の章があり、各章は次の順序に従います：導入、理論、実践、要約、演習。

貢献者

各章には 1 名または複数の章の著者がリストされています。これらは章の執筆を主導した人々です。しかし、本書に貢献した他の多くの人々もあり、ここで感謝の意を表したいと思います：

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		

ソフトウェアのバージョン

この本の大部分は OHDSI のオープンソースソフトウェアについてであり、このソフトウェアは時間とともに進化します。開発者はユーザーに一貫して安定した体験を提供するよう最善を尽くしていますが、時間の経過とともにソフトウェアの改善により、本書の一部の指示が時代遅れになるのは避けられません。コミュニティはそれらの変更を反映するためにオンラインの本書を更新し、時間の経過とともにハードコピーの新しい版（エディション）をリリースします。参考までに、本書のこのバージョンで使用されているソフトウェアのバージョンは以下の通りです：

- ACHILLES: バージョン 1.6.6

Table 1: 本書で使用されている Methods Library のパッケージのバージョン

パッケージ	バージョン
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

- ATLAS: バージョン 2.7.3
- EUNOMIA: バージョン 1.0.0
- 方法ライブラリパッケージ: 表 1を参照

ライセンス

この本は Creative Commons Zero v1.0 Universal license に基づいてライセンスされています。



本書が作成された方法

この本は RMarkdown を使用して bookdown パッケージで書かれています。オンラインバージョンはソースリポジトリ <https://github.com/OHDSI/TheBookOfOhdsilnJapanese/> から自動的に再構築され、継続的統合システム “travis” によって管理されます。定期的に本の状態のスナップショットが取得され、「版（エディション）」としてマークされます。これらの版は Amazon から物理コピーとして入手できます（訳者注：物理コピーは英語版のみ入手可能）。

第Ⅰ部

OHDSI コミュニティ

第 1 章

OHDSI コミュニティ

著者 : Patrick Ryan & George Hripcsak

集まることは始まりであり、共にいることは進歩であり、共に働くことが成功である。ヘンリー・フォード

1.1 データからエビデンスへの旅

世界中のあらゆる医療現場、大学の医療センターや診療所、規制当局や医療製品メーカー、保険会社や政策センター、そしてすべての患者と医療従事者との対話の中心には共通の課題があります。過去から学んだことをどのようにして将来のより良い意思決定に生かすのかということです。

10年以上もの間、多くの人々が「患者と医療従事者が協力して医療を選択するための最善のエビデンスを生成し、適用すること、患者ケアの自然な結果として発見のプロセスを推進すること、そして医療における革新、品質、安全性、価値を確保すること」を目的とした学習型医療システムのビジョンを主張してきました (Olsen et al., 2007)。この大志の主たる要素は、日常診療の過程で収集された患者レベルのデータを分析し、リアルワールドのエビデンスを導き出し、それを医療システム全体に広めて実臨床に役立てるという、非常に魅力的な見通しに基づいています。2007年、米国医学研究所のエビデンスに基づく医療に関する会議が「2020年までに、臨床判断の90%は正確かつタイムリーで最新の情報に裏付けられ、最良のエビデンスを反映したものとなる」という目標を掲げた報告書を発行しました (Olsen et al., 2007)。多くの分野で目覚ましい進歩が遂げられている一方で、私たちはこうした素晴らしい目標にはまだ遠く及ばないのが現状です。

なぜでしょうか？その理由の一つとして、患者レベルのデータから信頼性の高いエビデンスを導き出すまでの道のりが困難であることが挙げられます。データからエビデンスに至るまでの明確な道筋は一つではなく、その道筋をたどる

のに役立つ地図も一つではありません。実際、「データ」という概念は一つではなく、「エビデンス」という概念も一つではありません。



Figure 1.1: データからエビデンスへの旅

ソースシステムには、さまざまな患者レベルのデータを収集するさまざまなタイプの観察データベースがあります。これらのデータベースは、医療システム自体と同様に多様であり、異なる集団、医療環境、データ収集プロセスを反映しています。意思決定に役立つエビデンスにもさまざまな種類があり、臨床的特性、集団レベルの推定、患者レベルの予測などの分析のユースケースによって分類することができます。出発点（ソースデータ）と目的の目的地（エビデンス）とは別に、この課題はそのプロセスに必要とされる臨床、科学、技術的な能力の幅広さによってさらに複雑化しています。医療情報学を徹底的に理解する必要があります。これには、患者と医療従事者との診療現場でのやり取りから、管理システムや臨床システムを経て最終的な保存場所に至るまでのソースデータの完全な由来、データ収集や管理プロセスに関する医療政策や行動インセンティブの一部として生じる可能性のある偏りへの理解が含まれます。臨床上の疑問を、適切な回答を得るために適した観察研究のデザインに変換するための疫学の原則と統計的手法を習得する必要があります。何年にもわたる縦断的追跡調査で得られた何億件もの臨床の観察結果を含む、何百万人もの患者データセットに対して、計算効率の高いデータサイエンスアルゴリズムを実装し、実行する技術的能力が必要です。また、観察データネットワークで得られた結果と他の情報源からのエビデンスを統合し、この新しい知識が医療政策や実臨床にどのような影響を与えるべきかを判断する臨床的知識も必要です。したがって、データからエビデンスを導くために必要なスキルとリソースとを備えて

いる人物は非常にまれです。むしろ、このプロセスには、すべての利害関係者が信頼し意思決定プロセスに活用できるエビデンスを生成するために、最良のデータが最適な方法で分析されるよう、複数の個人や組織が協力することが必要となる場合がほとんどです。

1.2 観察医療アウトカムパートナーシップ（OMOP）

観察研究におけるコラボレーションの顕著な例として、Observational Medical Outcomes Partnership (OMOP) が挙げられます。OMOP は官民パートナーシップで、米国食品医薬品局 (FDA) が主導し、米国立衛生研究所 (NIH) 財団が運営し、製薬会社からなるコンソーシアムが資金を提供しました。製薬会社は学術研究者や医療データパートナーと協力し、観察医療データを使用した積極的な医薬品安全性監視の科学を推進する研究プログラムを確立しました (Stang et al., 2010)。OMOP は、多様なステークホルダーによるガバナンス体制を確立し、真の医薬品安全性の関連性を特定し、偽陽性所見と区別するという課題に対して、さまざまな医療請求データや EHR データベースに適用した場合の、代替となる疫学デザインや統計的手法のパフォーマンスを実証的に検証するための一連の methodological 実験を設計しました。

チームは集中型環境と分散型研究ネットワークの両方で、異なる観察データベースにまたがって研究を行うことの技術的な難しさを認識し、観察データの構造、内容、意味を標準化し、統計分析コードを一度作成すればすべてのデータサイトで再利用できるようにする仕組みとして、OMOP 共通データモデル (CDM) を設計しました (Overhage et al., 2012)。OMOP の実験により、異なる医療現場から得られた異なるデータタイプを、異なるソース用語で表現し、施設間の連携と計算効率の高い分析を促進する方法で取り込むことができる共通データモデルと標準ボキャブラリを確立できることができました。

OMOP は設立当初からオープンサイエンスのアプローチを採用し、研究デザイン、データ標準、分析コード、実証結果など、すべての成果物をパブリックドメインに置くことで透明性を高め、OMOP が実施している研究に対する信頼を構築するとともに、他者の研究目的の推進に再利用可能なコミュニティリソースを提供してきました。OMOP の当初の焦点は医薬品の安全性でしたが、OMOP CDM は、医療介入や医療制度政策の比較効果など、より広範な分析事例をサポートするために継続的に進化してきました。

OMOP は、大規模な実証実験の完了 (Ryan et al., 2012, 2013b)、方法論の革新 (Schuemie et al., 2014)、安全性に関する意思決定のための観察データの適切な利用に役立つ知識の創出 (Madigan et al., 2013b,a) に成功しましたが、OMOP の遺産は、オープンサイエンスの原則を早期に採用し、OHDSI コミュニティの形成を促したという点で、より記憶されるかもしれません。

OMOP プロジェクトが完了し、FDA のアクティブサーベイランス活動に情報を提供するための methodological 研究という使命を果たしたとき、チームは OMOP の旅路が終わり、新たな旅路が始まったことを認識しました。OMOP の方法論的研究は、観察データから生成されるエビデンスの質を明らかに改善できる科

学的ベストプラクティスに関する具体的な洞察を提供しましたが、それらのベストプラクティスの採用は遅々として進みませんでした。いくつかの障壁が特定されました。1) 分析の革新よりも優先して取り組むべきであると考えられていた観察データの品質に関する根本的な懸念、2) 方法論上の問題と解決策に対する概念的理解の不足、3) 各自のローカル環境で独自に解決策を実行できること、4) これらのアプローチが各自の関心のある臨床問題に適用できるかどうかといった不確実性、などです。すべての障壁に共通する要素は、自分一人だけでは変化をもたらすために必要なすべてを持っているわけではないという感覚であり、しかし、何らかの協力的な支援があれば、すべての問題を克服できるというものでした。とはいっても、いくつかの分野でのコラボレーションが必要でした。

- オープンコミュニティのデータ標準、標準化ボキャブラリ、ETL（抽出-変換-読込）規約の確立に向けたコラボレーション。これにより、基礎となるデータ品質に対する信頼性が高まり、構造、内容、意味論の一貫性が促進され、標準化された分析が可能になります。
- 医薬品の安全性に留まらず、臨床的特性、集団レベルの推定、患者レベルの予測など、より広範なベストプラクティスを確立するための方法論的研究におけるコラボレーション。方法論的研究により実証された科学的ベストプラクティスを体系化し、研究コミュニティが容易に採用できる公開ツールとして利用可能にするためのオープンソース分析開発におけるコラボレーション。
- コミュニティ全体で関心のある重要な健康問題に対処する臨床応用に関するコラボレーション。データからエビデンスへの道のりを共にたどる。

このような洞察から、OHDSI は誕生しました。

1.3 オープンサイエンスの協働組織としての OHDSI

Observational Health Data Sciences and Informatics (OHDSI、発音は「オデッセイ」) は、コミュニティが協力してより良い医療判断とケアを促進するエビデンスを生成することで、健康の改善を目指すオープンサイエンスのコミュニティです (Hripcsak et al., 2015)。OHDSI は、観察医療データの適切な利用に関する科学的ベストプラクティスを確立するための方法論的研究を実施し、これらのプラクティスを一貫性があり、透明性が高く、再現可能なソリューションに体系化するオープンソースの分析ソフトウェアを開発し、臨床上の疑問に適用してエビデンスを生成し、医療政策と患者ケアの指針となることを目指しています。

1.3.1 我々の使命

健康に関する意思決定とケアを向上させるエビデンスを協力して生成することにより、コミュニティをエンパワーメントし、健康を改善する。

1.3.2 我々のビジョン

観察研究によって健康と疾病に関する包括的な理解が得られる世界。

1.3.3 我々の目標

- 革新性: 観察研究は、革新的な恩恵を得ることができる分野です。我々の仕事において、新しい方法論的アプローチを積極的に探求し、奨励します。
- 再現性: 正確で再現可能な、適切に調整されたエビデンスが健康の改善に不可欠です。
- コミュニティ: 患者、医療従事者、研究者、そして私たちの活動に賛同する方など、誰もが OHDSI に積極的に参加いただけます。
- コラボレーション: 私たちは協力して、コミュニティの参加者の現実的なニーズを優先し、対処するために協力して取り組んでいます。
- 開放性: 私たちは私たちが生み出す方法、ツール、生成されたエビデンスなど、コミュニティの成果をすべて公開し、一般にアクセスできるよう努めています。
- 有益性: コミュニティ内の個人や組織の権利を常に保護するよう努めています。

1.4 OHDSI の進展

OHDSI は 2014 年の発足以来、学術界、医療製品業界、規制当局、政府、保険者、技術提供者、医療システム、臨床医、患者など、さまざまなステークホルダーから 2,500 人以上のコラボレーターをオンラインフォーラムに迎え入れてきました。また、コンピュータサイエンス、疫学、統計学、生物医学情報学、医療政策、臨床科学など、さまざまな分野を代表する参加者もいます。OHDSI のコラボレーターのリストは、OHDSI のウェブサイトで閲覧できます¹。OHDSI の協力者マップ（図 1.2）は、国際的なコミュニティの広さと多様性を示しています。

2019 年 8 月現在、OHDSI は 20 か国以上から 100 以上の異なる医療データベースのデータネットワークを構築し、OMOP CDM という OHDSI が維持するオープンコミュニティデータ標準を用いた分散型ネットワークアプローチを適用することで 10 億件以上の患者レコードを収集しています。分散型ネットワークとは、患者レベルのデータを組織間で共有する必要がないことを意味します。代わりに、研究に関する問い合わせはコミュニティ内の個人によって研究プロトコルの形で提起され、エビデンスを生成する分析コードが添付されます。生成されたデータは要約統計として共有され、研究に参加するパートナー間でのみ

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



Figure 1.2: 2019 年 8 月現在の OHDSI 協力者の地図

共有されます。OHDSI の分散型ネットワークを通じて、各データパートナーは患者レベルのデータの使用について完全な自主性を維持し、それぞれの機関のデータガバナンス方針を遵守し続けます。

OHDSI の開発者コミュニティは、OMOP CDM を基盤として、以下の 3 つのユースケースをサポートする堅牢なオープンソース分析ツールのライブラリを作成しました：1) 疾病の自然史、治療実態、品質向上のための臨床的特性評価；2) 医療製品の安全性監視と比較効果のための因果推論法を適用した集団レベルの効果推定；3) 精密医療や疾病予防のための機械学習アルゴリズムを適用する患者レベルの予測。OHDSI の開発者らは、OMOP CDM の採用、データの品質評価、OHDSI ネットワーク研究の促進を支援するアプリケーションも開発しています。これらのツールには、R と Python で作成されたバックエンドの統計パッケージや、HTML と Javascript で開発されたフロントエンドのウェブアプリケーションが含まれます。すべての OHDSI ツールはオープンソースであり、GitHub を通じて一般公開されています²。

OHDSI のオープンサイエンスコミュニティアプローチとオープンソースツールにより、観察研究は飛躍的に進歩しました。OHDSI ネットワーク分析の初期の成果の一つとして、糖尿病、うつ病、高血圧という 3 つの慢性疾患の治療経路に関する調査が挙げられます。これは National Academy of Science に掲載され、2 億 5000 万人以上の患者データを対象とした 11 のデータソースから得られた結果を分析し、これまでに観察されたことのない治療選択に関する地理的な違いや患者の異質性を明らかにしました (Hripcsak et al., 2016)。OHDSI は交絡因子調整のための新しい統計的手法 (Tian et al., 2018) や因果推論のた

²<https://github.com/OHDSI>

めの観察的エビデンスの妥当性評価 (Schuemie et al., 2018a) など、複数の分野でこれらのアプローチを適用しています。てんかんの安全性監視に関する問題 (Duke et al., 2017) から第二選択の糖尿病治療薬の比較効果 (Vashisht et al., 2018) や、うつ病治療の安全性比較に関する大規模な集団レベルの効果推定研究 (Schuemie et al., 2018b) に至るまで、さまざまな分野で適用されています。OHDSI コミュニティは、観察医療データに機械学習アルゴリズムを適用する方法の枠組みも確立しており (Reps et al., 2018)、さまざまな治療領域で適用されています (Johnston et al., 2019; Cepeda et al., 2018; Reps et al., 2019)。

1.5 OHDSI における協力

OHDSI はエビデンスを生成するためのコラボレーションを促進することを目的としたコミュニティです。OHDSI のコラボレーターになることにはどういう意味があるのでしょうか？もしあなたが OHDSI のミッションに賛同し、データからエビデンスを生む出すまでの過程のどこかに貢献したいと思うなら、OHDSI は最適なコミュニティです。コラボレーターには、患者レベルのデータにアクセスでき、そのデータがエビデンス生成に活用されることに興味を持つ人も含まれます。コラボレーターには、科学的ベストプラクティスを確立し、代替アプローチを評価したいという方法論者も含まれます。コラボレーターには、プログラミングスキルを活かしてコミュニティ全体が利用できるツール開発に関心を持つソフトウェア開発者も含みます。コラボレーターには、重要な公衆衛生上の疑問を持ち、それに対するエビデンスを広範な医療コミュニティに提供したいと考える臨床研究者も含まれます。コラボレーターには、この共通の公衆衛生のための目的を信じ、コミュニティが自立し、そのミッションを継続できるリソースを提供したいと思う個人や組織が含まれます。また、世界中でコミュニティ活動やトレーニングセッションを主催することも含まれます。OHDSI は専門分野やステークホルダーの所属に関わらず、共通の目的に向かった個人が協力し、それぞれが貢献することで、医療の進歩に寄与できる場となることを目指しています。この取り組みに参加したい方は第 2 章（「どこから始めようか」）を参照し、参加方法をご確認ください。

1.6 まとめ



- OHDSI のミッションは、健康に関する意思決定とケアを向上させるエビデンスを協力して生成することにより、コミュニティをエンパワーメントし、健康を改善させることです。
- 私たちのビジョンは、観察研究が健康と疾患に関する包括的な理解をもたらす世界であり、これを革新、再現性、コミュニティ、コラボレーション、開放性、有益性の目標を通じて達成します。
- OHDSI の協力者は、オープンコミュニティのデータ標準、方法論的

研究、オープンソース分析の開発、臨床応用に重点的に取り組み、データからエビデンスへの旅を改善することに取り組んでいます。

第 2 章

どこから始めようか

著者 : Hamed Abedtash & Kristin Kostka

「千里の道も一步から」 - 老子

OHDSI コミュニティは、学術界、産業界、政府機関といった多くの利害関係者で構成されています。私たちの仕事は患者、医療提供者、研究者、医療システム、産業界、政府機関など、さまざまな個人や組織に利益をもたらします。この利益は、医療データ分析の質を向上させるだけでなく、これらの利害関係者にとっての医療データの有用性を向上させることによって実現されます。私たちは、観察研究が革新的な思考から大いに恩恵を受ける分野だと考えており、積極的に新しい方法論的アプローチを模索し、奨励しています。

2.1 旅に参加しよう

OHDSI には、患者、医療専門家、研究者、あるいは私たちの活動に賛同する人など、誰もが積極的に参加できます。OHDSI は包括的なメンバーシップモデルを維持しています。OHDSI のコラボレーターになるために会費は必要ありません。コラボレーションは手を挙げるだけの簡単なもので、毎年の OHDSI 会員数に含まれます。参加は完全に任意です。コラボレーターは、毎週のコミュニティコールに参加するだけの人から、ネットワーク研究や OHDSI ワーキンググループを率いる人まで、さまざまなレベルの貢献が可能です。データ保持者でなくても、活発なコミュニティメンバーとして参加できます。OHDSI コミュニティは、データ保持者、研究者、医療提供者、患者や消費者に同様にサービスを提供することを目的としています。コラボレーターのプロファイルの記録は OHDSI ウェブサイトで維持され、定期的に更新されます。メンバーシップは OHDSI コミュニティコール、ワーキンググループや地域支部によって促進されます。

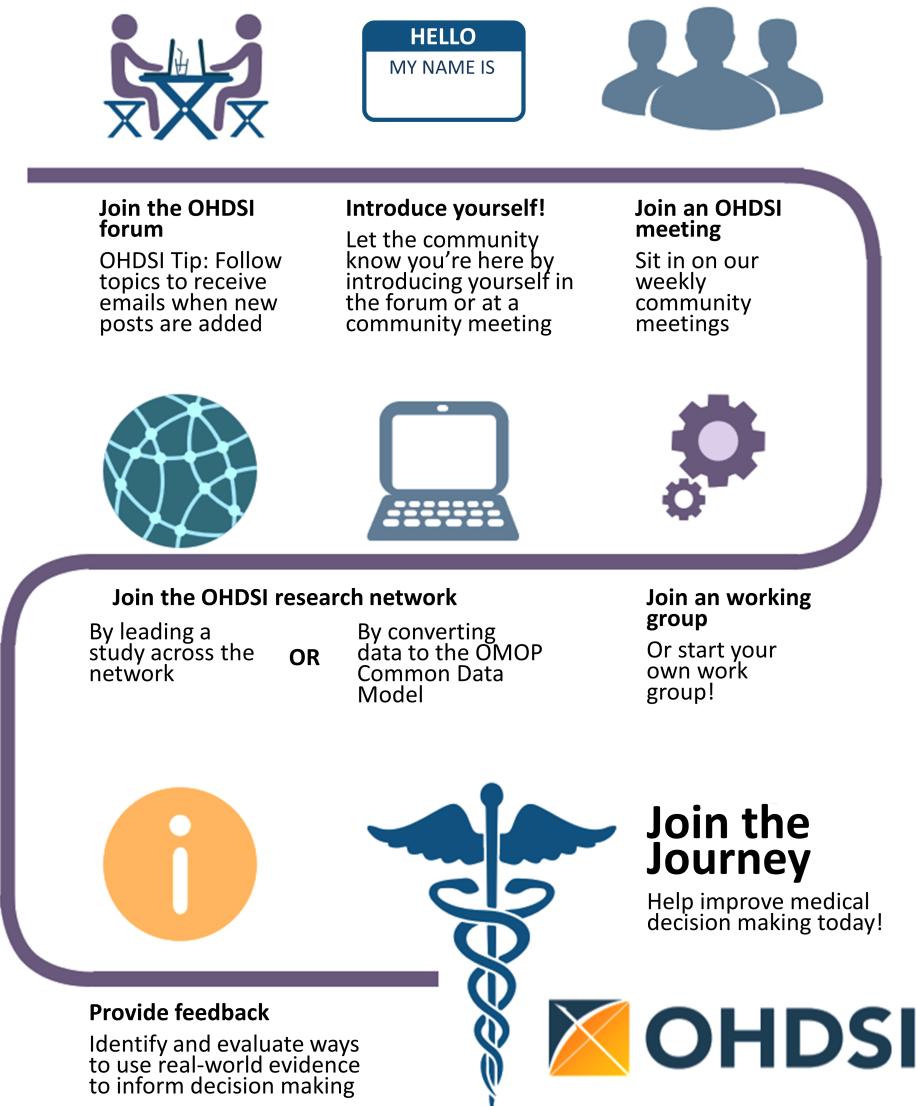


Figure 2.1: 旅に参加しよう — OHDSI のコラボレーターになるには

2.1.1 OHDSI フォーラム

OHDSI フォーラム¹ は、OHDSI コミュニティのコラボレーターが投稿メッセージの形で会話ができるオンラインのディスカッションサイトです。フォーラムはツリー状のディレクトリ構造で構成されています。最上部は「カテゴリ」です。フォーラムは、関連するディスカッションのカテゴリで分けることができます。カテゴリの下にはサブフォーラムがあり、これらのサブフォーラムにはさらにサブフォーラムがあります。トピック（一般にはスレッドと呼ばれる）はサブフォーラムの最下層にあり、ここでフォーラムメンバーはディスカッションや投稿を始められます。

OHDSI フォーラムには、次のようなコンテンツのカテゴリがあります：

- ・一般: OHDSI コミュニティに関する一般的なディスカッションと参加方法
- ・実装者: 共通データモデルと OHDSI 分析フレームワークをローカル環境に実装する方法についてのディスカッション
- ・開発者: OHDSI アプリケーションや他の OMOP CDM を活用するツールのオープンソース開発についてのディスカッション
- ・研究者: CDM ベースの研究に関するディスカッション（エビデンス生成、共同研究、統計手法や OHDSI 研究ネットワークに関するその他のトピックを含む）
- ・CDM ビルダー: 要件、ボキャブラリ、技術的側面を含む進行中の CDM 開発に関するディスカッション
- ・語彙ユーザー: ボキャブラリコンテンツに関するディスカッション
- ・地域支部（例：韓国、中国、ヨーロッパ）: OMOP 実装や OHDSI コミュニティ活動に関する、母国語での地域のディスカッション

自分のトピックを投稿するには、アカウントにサインアップする必要があります。フォーラムのアカウントを取得し、一般的なトピックの「Welcome to OHDSI! - Please introduce yourself (OHDSI へようこそ！-自己紹介をお願いします」というスレッドで自己紹介することをお勧めします。返信では 1) 自己紹介と自分の仕事について簡単に教えてください、2) コミュニティでどのように貢献したいか教えてください（例：ソフトウェア開発、研究実施、研究論文の執筆など）。これで OHDSI の旅が始まります！これから、ディスカッションに参加することをお勧めします。質問したり、新しいアイデアを議論したり、コラボレーションしたりするための手段として、OHDSI コミュニティはフォーラムを使用することを推奨しています。



トピックを選択して「ウォッチ」することができます。ウォッチしているトピックに新しい投稿が追加されるたびにメールが届き、その投稿にメールで直接返信できるようになります。一般的なスレッドをウォッチして、今後の会議の議題や、コラボレーションの機会に関する詳細を受け取ったり、毎週の OHDSI ダイジェストを直接受信トレイに配信しましょう！

¹<https://forums.ohdsi.org>

2.1.2 OHDSI イベント

OHDSI は、コラボレーター同士が学び合い、将来の協力を促進する機会を提供するため、定期的に対面イベントを開催しています。これらのイベントは OHDSI ウェブサイトで通知され、参加を希望する人は誰でも無料で参加できます。

OHDSI シンポジウムは、米国、ヨーロッパ、アジアで毎年開催される学術会議で、コラボレーターが全体会議、ポスター発表、ソフトウェアのデモを通じて最新の研究を発表できます。OHDSI シンポジウムはネットワーキングのための素晴らしい場であり、コミュニティ全体での最新の進歩について学ぶ機会です。OHDSI シンポジウムには通常、OHDSI チュートリアルが併設されており、OHDSI 共同研究者が OHDSI 共同研究者がコースの講師となって教えるもので、コミュニティの新規参加者に、データ標準や分析のベストプラクティスに関するトピックについて実践的な取り組みを行う機会を提供します。これらのチュートリアルは通常ビデオ録画され、イベント後に OHDSI ウェブサイトで公開され、イベントに参加できなかった人も利用できます。

OHDSI コラボレーターの対面イベントは、通常は共通の関心事の問題に焦点を当てた小規模なフォーラムです。過去のイベントには、フェノタイプ・ハッカソン、データ品質ハッカソン、オープンソースソフトウェアのドキュメンテーション・ハッカソンなどがあります。OHDSI は、複数の Study-a-thon イベントを主催してきました。この複数日にわたるセッションの目的は、適切な観察分析を設計・実装し、OHDSI ネットワーク全体で研究を実行し、一般への普及のためにエビデンスを統合することにより、特定の研究問題に関してチームとして協力することです。これらのすべてのイベントにおいて、共通の問題を解決したいという共通の願いがあるだけでなく、共同で問題解決を行うプロセスを学び、継続的な改善を促す歓迎的な環境を提供することにも共通の関心があります。

OHDSI コミュニティのパワーをもっと学びましょう。過去のシンポジウム、対面会議を検索し、OHDSI チュートリアルを視聴ください。OHDSI ウェブサイトの過去のイベントセクションから視聴できます。過去のイベントは定期的に更新され、コミュニティイベントはアーカイブされています。

2.1.3 OHDSI コミュニティコール

OHDSI コミュニティコールは、OHDSI コミュニティ内で進行中の活動にスポットライトを当てる機会です。毎週火曜日の午前 11 時から 12 時(東部標準時)に開催される電話会議は、OHDSI コミュニティが集まり、最近の開発を共有し、個々のコラボレーター、ワーキンググループ、コミュニティ全体の成果を認識する時間です。毎週の会議は記録され、プレゼンテーションは OHDSI ウェブサイトのリソースにアーカイブされます。

OHDSI コラボレーターは、毎週の電話会議への参加を歓迎され、コミュニティディスカッションのトピックを提案することが奨励されています。OHDSI コミュニティコールは、研究成果を共有し、進行中の作業に対するフィードバック

を求める、開発中のオープンソースソフトウェアツールをデモンストレーションし、データモデリングと分析のためのコミュニティベストプラクティスを議論し、助成金/出版物/会議ワークショップのための将来のコラボレーションの機会をブレインストーミングするフォーラムとなります。OHDSI コラボレーター会議のトピックを持っているコラボレーターは、OHDSI フォーラムに意見を投稿ください。

OHDSI コミュニティの新規参加者として、OHDSI ネットワーク全体で何が起こっているかを把握するために、このコールシリーズをカレンダーに追加することをお勧めします。OHDSI コールに参加する場合は、OHDSI フォーラムでアナウンスを確認してください。OHDSI フォーラムコミュニティコールのトピックは週ごとに異なります。OHDSI フォーラムの OHDSI ウィークリーダイジェストで、毎週のプレゼンテーションのトピックの詳細情報を確認することができます。新規参加者は、初回のコールで自己紹介し、自分自身、経歴、OHDSI に参加した理由についてコミュニティに話すよう求められます。

2.1.4 OHDSI ワークグループ[¶]

OHDSI には、ワークグループチームが主導するさまざまな進行中のプロジェクトがあります。各ワークグループにはそれぞれリーダーシップチームがあり、プロジェクトの目的、目標、コミュニティに提供される成果物を決定します。ワークグループには、プロジェクトの目的と目標に貢献することに関心のあるすべての人が参加できます。ワークグループは、長期にわたる戦略的目的の場合もあれば、コミュニティの特定のニーズを満たすための短期プロジェクトである場合もあります。ワークグループの会議の頻度は、プロジェクトリーダーシップによって決定され、グループごとに異なります。アクティブなワークグループのリストは、OHDSI Wiki で管理されています。

表 2.1 は、アクティブな OHDSI 作業グループのクイックリファレンスです。是非コールに参加して、より多くを学ぶことをお勧めします。

Table 2.1: 注目すべき OHDSI 作業グループ

ワークグループ名	目的	対象参加者
Atlas & WebAPI	Atlas と WebAPI は、OMOP 共通データモデルを基盤として構築し、標準化された分析機能を提供する OHDSI オープンソースソフトウェアアーキテクチャの一部です。	オープンソースの Atlas/WebAPI プラットフォームを改善し、貢献することを目指す Java & JavaScript ソフトウェア開発者

ワークグループ名	目的	対象参加者
CDM & ボキャブラリ	臨床患者データに適用される体系的で、標準化された大規模な分析を目的として、OMOP 共通データモデルの開発を継続します。他のワーキンググループによって開発された標準化された分析をサポートするため、国際的なコーディングシステムと患者ケアの臨床的側面のカバレッジを拡大することで、標準化ボキャブラリの品質を向上させます。	OMOP 共通データモデルと標準ボキャブラリの改善に関心があり、すべてのニーズとユースケースに対応できる人
ゲノム解析	OMOP CDM を拡張して、患者のゲノムデータを組み込みます。グループは、さまざまなシーケンスプロセスからの遺伝子変異に関する情報を保存できる CDM 互換スキーマを定義します。	すべての人が参加可能
集団レベルの推定	正確で信頼性が高く、再現性のある集団レベルの効果推定につながる観察研究のための科学的手法を開発し、コミュニティによるこれらの手法の使用を促進します。	すべての人が参加可能
自然言語処理	OHDSI 傘下の観察研究で、EHR のテキスト情報の使用を促進します。この目的を促進するため、グループは OHDSI コミュニティによる研究に臨床テキストを利用するための実装できる方法とソフトウェアを開発します。	すべての人が参加可能

ワークグループ名	目的	対象参加者
患者レベルの予測	複数の対象とするアウトカムに用いることがで き、対象とするあらゆる 患者サブグループからの 観察医療データに適用し ます。正確で十分に調整 された患者中心の予測モ デルを開発するため、標 準化されたプロセスを確 立します。	すべての人が参加可能
ゴールドスタンダート表 現型ライブラリ	OHDSI コミュニティの メンバーが、コミュニティで検証されたコホート 定義を研究やその他の活 動のために見つけ、評価 し、利用できるようにし ます。	フェノタイプのキュレー ションと検証に関心のあ るすべての人が参加可能
FHIR ワークグループ	OHDSI FHIR 統合のロー ドマップを確立し、 OHDSI ベースの観察研 究のために EHR コミュ ニティの FHIR 実装とデ ータを活用し、FHIR ベ ースのツールと API を 通じて OHDSI データと 研究結果を普及するため の推奨事項を、より広範 なコミュニティに提供し ます。	相互運用性に関心のある すべての人が参加可能
GIS	OMOP CDM を拡張し、 患者の環境曝露の履歴を その臨床フェノタイプと 関連付けるために OHDSI ツールを活用し ます。	健康関連の地理属性に興 味のあるすべての人が参 加可能
臨床試験	OHDSI プラットフォー ムとエコシステムがあら ゆる面で試験を支援でき る臨床試験のユースケー スを理解し、サポートす る OHDSI ツールの更新 の推進を支援します。	臨床試験に興味のあるす べての人が参加可能

ワークグループ名	目的	対象参加者
THEMIS	THEMIS の目的は、OMOP CDM 規則を超える標準規則を開発し、各 OMOP サイトで設計された ETL（抽出-変換-読込）プロトコルが最高品質で、再現可能かつ効率的であることを保証することです。	ETL 標準化に関心のあるすべての人が参加可能
メタデータ & 注釈	私たちの目標は、人間と機械が作成したメタデータと注釈を共通データモデルに保存するための標準プロセスを定義し、研究者が観察データセットに関する有用なデータ成果物を利用して作成できるようにすることです。	すべての人が参加可能
患者生成医療データ(PGHD)	このワーキンググループの目標は、スマートフォン/アプリ/ウェアラブルデバイスから生成される PGHD の ETL 規則、臨床データとの統合プロセス、分析プロセスを開発することです。	すべての人が参加可能

ワークグループ名	目的	対象参加者
OHDSI 女性グループ	OHDSI コミュニティ内の女性が一堂に会し、科学、技術、工学、数学(STEM)で働く女性として直面する課題について話し合う場を提供すること。OHDSI コミュニティが STEM 分野の女性をどのようにサポートできるかについて、女性たちが自分たちの視点を共有し、懸念を提起し、アイデアを提案し、最終的にはコミュニティやそれぞれの分野でリーダーとなる女性たちを鼓舞することができるような議論を促進することを目指しています。	このミッションに賛同するすべての人が参加可能
運営委員会	OHDSI のすべての活動とイベントが、成長を続けるコミュニティのニーズに合致していることを確認することで、OHDSI の使命、ビジョン、価値観を維持します。さらに、このグループは、OHDSI の将来の方向性についてガイダンスを提供することで、コロンビアに拠点を置く OHDSI 調整センターの諮問グループとして機能します。	コミュニティ内のリーダー

2.1.5 OHDSI 地域支部

OHDSI 地域支部は、地理的な地域に所在し、地域特有の問題に対処するため、ローカルネットワークイベントや会議を開催したいと考えている OHDSI コラボレーターのグループです。現在、OHDSI 地域支部は、ヨーロッパ²、韓国³、中

²<https://www.ohdsi-europe.org/>

³<https://forums.ohdsi.org/c/For-collaborators-wishing-to-communicate-in-Korean>

国⁴にあります。ご自身の地域で OHDSI 地域支部を設立したい場合は、OHDSI Web サイトで説明されている OHDSI 地域支部のプロセスに従って設立できます。

2.1.6 OHDSI リサーチネットワーク

OHDSI のコラボレーターの多くは、データを OMOP 共通データモデルに変換することに関心を持っています。OHDSI 研究ネットワークは、ETL プロセスを経て OMOP に準拠した観測データベースの多様なグローバルコミュニティを表しています。OHDSI コミュニティでの取り組みにデータの変換が含まれる場合は、OMOP CDM とボキャブラリに関するチュートリアル、変換を支援する無料で利用可能なツール、特定のドメインやデータ変換の種類を対象とするワークグループなど、取り組みを支援する多数のコミュニティリソースがあります。OHDSI のコラボレーターは、OHDSI フォーラムを利用して、CDM 変換中に発生する課題について話し合い、トラブルシューティングすることを推奨されます。

2.2 どこにフィットするか

ここまで読んで、あなたは「私は OHDSI コミュニティのどこに属しているのだろう？」と疑問に思っているかもしれません。

私は臨床研究者で、研究を始めたいたいと思っています。特定の質問に答えるために OHDSI リサーチネットワークを使用したい臨床研究者であるなら、たとえば論文を発表したいと考えているなら、あなたは正しい場所にいます。OHDSI フォーラムの OHDSI リサーチャーズトピックにアイデアを投稿することから始めましょう。これにより、同様の関心を持つ研究者とつながることができます。OHDSI は論文の出版を好んでおり、リサーチクエスチョンを分析や迅速に分析や論文にしていくための多くのリソースを提供しています。詳細は第11、12、13章をご覧ください。

私は OHDSI コミュニティが発信する情報を読んで利用したいと思っています。患者、臨床医、医療の専門家のいずれであっても、OHDSI は健康アウトカムをよりよく理解するのに役立つ高品質のエビデンスを提供したいと考えています。コードを書くのは久しぶりかもしれません。プログラムを書いたことがないかもしれません。あなたにはこのコミュニティに居場所があります。私たちはあなたをエビデンスの消費者と呼びます。あなたは OHDSI の研究を行動に移す人です。あなたは OHDSI がどのようなエビデンスを生成したか、または生成中であるかを知るためにふるいにかけており、おそらく自分に関連する質問を提案したいとも思っているでしょう。私たちはあなたがディスカッションに参加することを歓迎します。OHDSI フォーラムで質問を始めましょう。コミュニティコールに参加し、最新の研究について聞いてください。OHDSI シンポジウムや対面ミーティングに参加してコミュニティと直接交流しましょう。あ

⁴<https://ohdsichina.org/>

あなたの質問は OHDSI コミュニティの重要な部分です。声を上げて、あなたが探しているエビデンスについて、私たちがさらに知る手助けをしてください！

私は医療のリーダーとして働いています。データ所有者、またはその代表者であるかもしれません。組織にとっての OMOP CDM と OHDSI 分析ツールの有用性を評価しています。組織の管理者/リーダーとして、あなたは OHDSI について聞いたことがあります。OMOP CDM があなたのユースケースにどのように役立つかを知りたいと思っているかもしれません。OHDSI の過去のイベント資料に目を通し、研究内容を確認ください。コミュニティコールに参加して、ただ聞くだけでも構いません。7 章（データ分析のユースケース）を読むと、OMOP CDM や OHDSI 分析ツールで実現できる研究の種類を理解するのに役立つかもしれません。OHDSI コミュニティは、あなたの旅をサポートします。興味のある具体的な分野がある場合は遠慮せずに発言し、事例を尋ねてください。世界中の 200 以上の組織が OHDSI で協力しており、コミュニティの価値を示すための多くの成功事例があります。

私はデータベース管理者で、私の機関のデータを ETL または OMOP CDM に変換したいと考えています。データを「OMOP」することは、斬新で価値のある取り組みです。ETL プロセスを始めたばかりの場合、OHDSI コミュニティの ETL チュートリアルスライドを参照するか、今後開催される OHDSI シンポジウムに登録したりしてください。THEMIS 作業グループのコールに参加し、OHDSI フォーラムで質問することも考えてみてください。OMOP CDM の実装を成功に導く支援となる知識がコミュニティには豊富にあります。遠慮しないでください！

私はバイオ統計学者かつ、またはメソッドの開発者で、OHDSI ツールスタックへの貢献に興味があります。R に精通しており、Git にコミットする方法を知っています。何よりも、OHDSI メソッドライブラリに専門知識を持ち込み、これらの方法論をさらに発展させたいと思っています。まずは、集団レベルの推定または患者レベルの予測のワークグループコールに参加し、現在のコミュニティの優先事項について詳しく聞くことをお勧めします。OHDSI ツールを使用する際、該当する GitHub リポジトリ（例：SQL Render パッケージの問題であれば、OHDSI/SqlRender の GitHub リポジトリに提出します）に問題を報告することができます。皆さんの貢献をお待ちしています！

私はソフトウェア開発者で、OHDSI ツールスタックを補完するツールの構築に关心があります。コミュニティへようこそ！OHDSI のミッションの一環として、私たちのツールは Apache ライセンスの下でオープンソースとして管理されています。OHDSI ツールスタックを補完するソリューションの開発を歓迎しています。ワーキンググループに参加し、アイデアを提案ください。OHDSI はオープンサイエンスとオープンコラボレーションに多大な投資をしていることに留意ください。独自のアルゴリズムとソフトウェアソリューションは歓迎しますが、私たちのソフトウェア開発の主な焦点ではありません。

私はコンサルタントで、OHDSI コミュニティに助言したいと考えています。コミュニティへようこそ！あなたの専門知識は貴重であり、高く評価されています。必要に応じて、OHDSI フォーラムでサービスを宣伝することができます。

OHDSI チュートリアルに参加ください。また、年間を通じてシンポジウムの議事録や OHDSI の対面ミーティングで専門知識を提供して貢献することを検討ください。

私は学生で、OHDSI についてもっと学びたいと思っています。あなたは正しい場所にいます！ OHDSI コミュニティコールに参加し、自己紹介することを考えみてください。OHDSI チュートリアルを詳しく調べたり、OHDSI シンポジウムや対面ミーティングに参加して OHDSI コミュニティが提供する方法とツールについてさらに学ぶことをお勧めします。特定の研究に関心がある場合は、OHDSI フォーラムの研究者トピックに投稿してお知らせください。多くの組織が OHDSI が後援する研究の機会（例：ポスドク、研究フェローシップ）を提供しています。OHDSI フォーラムでは、これらの機会などに関する最新情報を提供しています。

2.3 まとめ



- OHDSI コミュニティに参加するのは、挨拶するのと同じくらい簡単です。OHDSI フォーラムに投稿し、コミュニティコールに参加ください。
- 研究や ETL に関する質問を OHDSI フォーラムに投稿ください。

第 3 章

オープンサイエンス

著者 : Kees van Bochove

OHDSI コミュニティの発足当初から、オープンソースソフトウェアの利用、すべての会議の議事録や資料の公開、生成された医療的エビデンスの透明性あるオープンアクセスによる公開など、オープンサイエンスの価値観に基づいて国際的な共同研究体制を確立することが目標とされてきました。しかし、オープンサイエンスとは具体的にはどのようなものでしょうか？また、プライバシーへの配慮が非常に重要であり、通常は正当な理由から公開されない医療データに関して OHDSI はどのようにオープンサイエンスやオープンデータ戦略を構築できるのでしょうか。分析の再現性がなぜそれほど重要なのでしょうか。OHDSI コミュニティはこれをどのようにしてこれを実現しようとしているのでしょうか。本章ではこれらの疑問について触れていきます。

3.1 オープンサイエンス

「オープンサイエンス」という用語は 1990 年代から使われてきましたが実際に注目を集めるようになったのは 2010 年代で、OHDSI が誕生したのと同じ時期です。Wikipedia (Wikipedia, 2019a) ではこれを「科学的研究（出版物、データ、物理的サンプル、ソフトウェアを含む）とその普及を、アマチュアか専門家を問わず、探求心のあるあらゆるレベルの人々が利用できるようにする運動」と定義しており、通常は共同ネットワークを通じて開発されると述べています。OHDSI コミュニティは明確には「オープンサイエンス」集団またはネットワークとして位置づけられたことはありませんが、この用語は OHDSI の基本的な概念や原則を説明する際に頻繁に使われています。例えば、2015 年にはジョン・デュークが OHDSI を「医療エビデンス生成へのオープンサイエンスアプローチ」¹と表現し、2019 年には EHDEN コンソーシアムの紹介ウェビナーで OHDSI ネットワークアプローチを「21 世紀のリアルワールドオープンサイエンス」として位置づけました。

¹<https://ohdsi.github.io/TheBookOfOhdsi/OpenScience.html#fn17>

ンス」²として称賛しました。実際、この章で詳しく見ていくように、オープンサイエンスの実践の多くは今日の OHDSI コミュニティに見出すことができます。OHDSI コミュニティは、医療におけるエビデンス生成の透明性と信頼性を向上させるという共通の願いから生まれた草の根的なオープンサイエンスの集合体である、という見方もできるでしょう。

オープンサイエンスまたは「サイエンス 2.0」のアプローチ (Wikipedia, 2019b) は、現在の科学的手法における多くの認識された問題に対処することを意味します。情報技術はデータの生成と分析方法の爆発的な増加をもたらし、個々の研究者にとっては、専門分野で発表されるすべての文献を把握するのは非常に困難になっています。これは、本業として診療をしながらも最新の医学的エビデンスに遅れずについていく必要のある医師にとっては、なおさらのことです。さらに、多くの試験が統計上の設計不備、出版バイアス、p-hacking、その他の同様の統計的問題に直面し、再現は困難であるという懸念が高まっています。こうした懸念を修正する従来の方法である論文の査読では、このような問題を特定し、対処できないことがよくあります。2018 年の『Nature』誌の特集号「再現不可能な研究における課題」に関する 2018 年の Nature 特集版³には、この問題の例がいくつか紹介されています。ある分野の論文に系統的な査読を適用しようとした著者グループは、さまざまな理由により、彼らが指摘したエラーを修正してもらうのが非常に難しいことを発見しました。特に、最初から欠陥のあるデザインの試験は修正が難しかったのです。ロナルド・フィッシャーの言葉によると、「試験が終了してから統計学者に相談することは、単に死後解剖を依頼するようなものだ。おそらく、その試験がなぜ失敗したのかを教えてくれるだろう」(Wikiquote, 2019)。著者らは、ランダム化デザインの不備による統計的有意性についての誤った結論、メタ分析における誤算、不適切なベースライン比較など、一般的な統計上の問題に直面しました (Allison et al., 2016)。同じ論文集の別の論文では、物理学の経験を例に挙げ、完全な再現性を実現するには、基礎データへのアクセスを提供するだけでなく、データ処理と分析のスクリプトを公開し、適切に文書化することが重要であると主張しています (Chen et al., 2018)。

OHDSI コミュニティはこれらの課題に対して独自の方法で取り組んでおり、大規模な医療エビデンスの生成の重要性を強調しています。Schuemie et al. (2018b) によると、現在のパラダイムは「信頼性が不明な独自の研究デザインを用いて、1 つずつ推定値を生成し、1 つずつ推定値を公表（または不公表）することに重点を置いている」一方で、OHDSI コミュニティは「一貫性のある標準化された方法を用いた高スループットの観察研究を提唱し、評価、較正、偏りのない普及を可能にすることで、より信頼性が高く完全なエビデンスベースを生成する」としています。これは、OMOP 共通データモデルにデータをマッピングする医療データソースのネットワーク、誰もが利用・検証可能なオープンソース分析コード、howoften.org で公開されている疾患発生状況などの大規模なベースラインデータの組み合わせによって実現されます。以下では、具体的な例を挙げ、オープンスタンダード、オープンソース、オープンデータ、オ

²<https://www.ehdeneu/webinars/>

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

オープンディスカッションの4つの原則を指針として、OHDSIのオープンサイエンスのアプローチについてさらに詳しく説明します。本章の締めくくりとして、オープンサイエンスの観点から OHDSI の FAIR 原則と展望について簡単に言及します。

3.2 オープンサイエンスの実践: the Study-a-Thon

コミュニティにおける最近の動きとして、「study-a-thon」の出現が挙げられます。これは、OMOP データモデルと OHDSI ツールを使用して、臨床的に重要な研究課題の答えを導くことを目的とした、多分野にわたる科学者グループの短期集中型の対面式集会です。その好例が、2018 年のオックスフォード研究マラソンです。この研究マラソンについては、EHDEN のウェビナー⁴で説明されており、そのプロセスが詳しく紹介されているほか、公開されている結果も強調されています。研究マラソンに先立ち、参加者は医学的に関連性の高い研究課題を提案し、研究マラソンで研究する 1 つもしくは複数の研究課題が選定されました。OMOP 形式の患者レベルデータにアクセスでき、これらのデータソースでクエリを実行できる参加者にデータが提供されました。実際の study-a-thon の時間の多くは、統計的アプローチ（第2章参照）、データソースの適合性、インタラクティブに作成される結果、およびこれらの結果から必然的に生じる追加の質問について議論することに費やされます。オックスフォード大学での study-a-thon の場合は、さまざまな人工膝関節置換術の術後の有害作用の研究に焦点が当てられ、study-a-thon の期間中に OHDSI フォーラムとツールを使用してインタラクティブに結果が発表されました（Chapter 8 参照）。ATLAS などの OHDSI ツールは、コホート定義の迅速な作成、交換、議論、テストを可能にし、定義と方法の選択に関するコンセンサスを達成する初期プロセスを大幅にスピードアップさせます。関連するデータソースが OMOP 共通データモデルを使用し、OHDSI のオープンソース患者レベル予測パッケージ13が利用可能であったため、術後 90 日間の死亡率予測モデルを 1 日で作成し、翌日には複数の大規模データソースで外部検証を行うことができました。また、この研究マラソンは、従来の学術論文（「人工膝関節全置換術後の有害事象に対する患者レベル予測モデルの開発と検証」）の執筆にもつながりました。この論文は、査読に数ヶ月を要しました。しかし、数億件の患者記録を網羅する複数の医療データベースの分析スクリプトと結果が、わずか 1 週間でゼロから構想、作成、公開されたという事実は、OHDSI が医学にもたらす根本的な改善を示しています。これにより、エビデンスが利用可能になるまでの期間が数か月から数日に短縮されます。

3.3 オープンスタンダード

OHDSI コミュニティで維持されている非常に重要なコミュニティリソースは、OMOP 共通データモデル（第 4 章参照）と関連する標準ボキャブラリ（第 5 章

⁴<https://youtu.be/X5yuoJoL6xs>

参照) です。このモデル自体は観察医療データを収集することを目的としており、もともとは薬物、処置(プロシージャー)、デバイスなどの曝露と、コンディションやメジャーメント(測定)などのアウトカムとの関連性を分析することを目的としていました。様々な分析用途に合わせて拡張されてきました(詳しくは第7章参照)。しかし、世界中のさまざまなコーディングシステム、医療パラダイム、さまざまなタイプの医療ソースからヘルスケアデータを調和させるには、ソースコードとその最も近い標準化された対応コードとの間の膨大な「マッピング」が必要になります。OMOP 標準化ボキャブラリは第7章でさらに詳しく説明されており、世界中で使用されている数百の医療コーディングシステムからのマッピングを含み、OHDSI の Athena ツールを通じて閲覧可能です。これらのボキャブラリとマッピングを無料で利用可能なコミュニティリソースとして提供することにより、OMOP と OHDSI コミュニティは医療データ分析に多大な貢献を果たしています。また、世界中の約 12 億件の医療記録を代表する、この目的のための最も包括的なモデルとされています⁵(Garza et al., 2016)。

3.4 オープンソース

OHDSI コミュニティが提供するもう一つの重要なリソースはオープンソースのプログラムです。これらはいくつかのカテゴリーに分類することができ、例えば OMOP へのデータマッピング用のヘルパートール(第6章参照)、広く使用されている統計手法の強力なスイートを含む OHDSI メソッドライブラリ、公開された観察研究のオープンソースコード、ATLAS、Athena、その他 OHDSI エコシステムを支えるインフラ関連のソフトウェア(第8章参照)などがあります。オープンサイエンスの観点から、最も重要なリソースの一つは、OHDSI ネットワーク研究(第20章参照)の実行コードです。これらのプログラムは、GitHub を介して調査、レビュー、貢献ができる完全なオープンソースの OHDSI スタックを活用しています。例えば、ネットワーク研究は多くの場合メソッドライブラリに基づいて構築されており、分析のユースケース全体で統計手法の一貫した再利用を保証します。オープンソースソフトウェアの利用とコラボレーションが生成されたエビデンスの品質と信頼性をいかに支えているかに関する詳細な概要については、第17章を参照ください。

3.5 オープンデータ

医療データはプライバシーセンシティブな性質を持つため、完全にオープンで包括的な患者レベルのデータセットは通常入手できません。しかし、OMOP にマッピングされたデータセットを活用して、前述の <http://howoften.org> や <http://data.ohdsi.org> で公開されている、他の公開結果セットのような、重要な集計データや結果セットを公開することは可能です。また、OHDSI コミュニティは、テストや開発目的で SynPUF などのシミュレートデータセットを提供

⁵<https://www.ema.europa.eu/en/events/common-data-model-europe-why-which-how>

しており、OHDSI リサーチネットワーク（第20 章参照）を活用して、データを OMOP にマッピングした利用可能なデータソースのネットワーク上で研究を実行することもできます。ソースデータと OMOP CDM の間のマッピングを透明化するため、データソースが OHDSI ETL または「マッピング」ツールを再利用し、マッピングコードをオープンソースとして公開することが奨励されています。

3.6 オープンな議論

オープンスタンダード、オープンソース、オープンデータは素晴らしい資産ですが、それだけでは医療行為に影響を与えることはできません。オープンサイエンスの実践と OHDSI のインパクトの鍵となるのは、医療上のエビデンスの生成と科学の医療行為への応用です。OHDSI コミュニティは、米国、欧州、アジアで毎年開催される OHDSI シンポジウムを複数開催しているほか、中国や韓国などでも実践コミュニティを展開しています。これらのシンポジウムでは、統計的手法、データ、ソフトウェアツール、標準ボキャブラリ、OHDSI オープンソースコミュニティのその他のあらゆる側面における進歩について議論されています。OHDSI フォーラム⁶や Wiki⁷は、世界中の何千人もの研究者が観察研究を実施する上で役立っています。コミュニティコール⁸や GitHub のコード、問題、プルリクエスト⁹は、コードや CDM などのオープンコミュニティの資産を常に進化させており、OHDSI ネットワーク研究では、世界中の何億件もの患者レコードを用いて、グローバルな観察研究がオープンかつ透明性の高い方法で実施されています。

コミュニティ全体で開放性とオープンな議論が奨励されており、この本もまさに、OHDSI wiki、コミュニティコール、GitHub リポジトリによって促進されたオープンなプロセスを通じて執筆されています¹⁰。ただし、OHDSI のコラボレーターなしには、プロセスやツールは空虚な殻にすぎないことを強調しておく必要があります。実際、OHDSI コミュニティの真価は、第 1 で議論したように、コラボレーションとオープンサイエンスを通じて健康を改善するというビジョンを共有するメンバーにある、という主張も成り立ちます。

3.7 OHDSI と FAIR ガイディングプリンシブルズ

3.7.1 序論

この章の最後の段落では、Wilkinson et al. (2016) で発表された FAIR 原則で OHDSI コミュニティとツールの現状を概観します。

⁶<https://forums.ohdsi.org>

⁷<https://www.ohdsi.org/web/wiki>

⁸<https://www.ohdsi.org/web/wiki/doku.php?id=projects:overview>

⁹<https://github.com/ohdsi>

¹⁰<https://github.com/OHDSI/TheBookOfOhdsi>

3.7.2 検索可能性

OMOP にマッピングされ、分析に用いられる医療データベースは、科学的観点から、将来の参照と再現のために保存されるべきです。OMOP データベースの永続的な識別子の使用は、まだ広く普及しているとは言えません。その理由の一つとして、これらのデータベースはファイアウォールの内側や内部ネットワークに置かれていることが多く、必ずしもインターネットに接続されているわけではないことが挙げられます。しかし、データベースの概要を記述子レコードとして公開し、引用目的などで参照できるようにすることは十分に可能です。この方法は、例えば EMIF カタログ¹¹で採用されており、データ収集の目的、ソース、ボキャブラリや用語、アクセス制御の仕組み、ライセンス、同意など、データベースの包括的な記録を提供しています (Oliveira et al., 2019)。このアプローチは、IMI EHDEN プロジェクトでさらに発展しています。

3.7.3 アクセシビリティ

OMOP マッピングされたデータのオープンプロトコルを介したアクセスは、通常、OMOP CDM と組み合わせた SQL インターフェースを通じて実現され、OMOP データへのアクセス方法として標準化され、十分に文書化された方法を提供します。しかし、前述の通り、セキュリティ上の理由から、OMOP ソースはインターネット上で直接利用できないことがよくあります。研究者たちがアクセスできる安全な世界規模の医療データネットワークの構築は、IMI EHDEN のようなプロジェクトの活発な研究テーマであり、運営目標でもあります。しかし、LEGEND や <http://howoften.org>などの OHDSI イニシアティブを通じて示されているように、複数の OMOP データベースにおける分析結果は、公開することができます。

3.7.4 相互運用性

相互運用性は、OMOP データモデルと OHDSI ツールの強みであるといえるでしょう。エビデンスの生成に活用できる世界中の医療データソースの強固なネットワークを構築するには、医療データソース間の相互運用性を実現することが鍵となります。これは OMOP モデルと標準化ボキャブラリによって達成されます。しかし、コホート定義と統計的手法を共有することで、OHDSI コミュニティはコードマッピングを超えて、医療データの分析方法に関する相互運用可能な理解を構築するためのプラットフォームも提供しています。OMOP データの記録元となるのは病院などの医療システムであることが多いため、HL7 FHIR、HL7 CIMI、openEHR などの医療業務における相互運用性標準規格との整合により、OHDSI アプローチの相互運用性はさらに強化される可能性があります。CDISC や生物医学オントロジーなどの臨床相互運用性標準規格との整合についても同様です。特に腫瘍学などの分野では、これは重要なトピックであり、OHDSI コミュニティの腫瘍学ワーキンググループや臨床試験ワーキンググループは、これらの問題が活発に議論されるフォーラムの好例です。他のデー

¹¹<https://emif-catalogue.eu>

タ、特にオントロジー用語への参照という観点では、ATLAS と OHDSI Athena は重要なツールです。これらのツールは、他の利用可能な医療用コードシステムとの関連で OMOP 標準ボキャブラリの調査を可能にします。

3.7.5 再利用性

再利用に関する FAIR 原則は、データライセンス、データの由来（データの発生経緯の明確化）、関連するコミュニティ標準へのリンクなど、重要な問題に焦点を当てています。データライセンスは、特に管轄区域をまたぐ場合、複雑なトピックであり、本書で詳しく取り上げるには範囲を超えていています。しかし、もし自分のデータ（例えば分析結果）を他者に自由に利用してもらいたいのであれば、データライセンスを通じてこれらの許可を明示的に提供することが望ましい、と述べておくことは重要です。これは、インターネット上で見つかるほとんどのデータではまだ一般的な慣行ではなく、OHDSI コミュニティも残念ながら例外ではありません。OMOP データベースのデータ由来に関しては、メタデータを自動的に利用できるようにするといった改善の余地があります。例えば、CDM バージョン、標準化ボキャブラリのリリース、カスタムコードリストなどです。OHDSI ETL ツールは現在、この情報を自動的に生成していませんが、データ品質作業グループやメタデータ作業グループなどの作業グループが積極的に取り組んでいます。もう一つの重要な側面は、基礎となるデータベース自体の由来です。病院や一般開業医の情報システムが置き換えられたり変更されたりしたかどうか、また、既知のデータ欠落やその他のデータの問題がいつ発生したかを知ることは重要です。OMOP CDM にこのメタデータを体系的に添付する方法を検討することは、メタデータ作業部会の管轄となっています。



- OHDSI コミュニティは、医療におけるエビデンス生成の相互運用性と再現性を積極的に追求するオープンサイエンスのコミュニティと見なすことができます。
- また、単一の研究と単一の推定値による医療研究から、大規模な体系的なエビデンス生成へのパラダイムシフトを提唱しています。この大規模な体系的なエビデンス生成では、ベースライン発生率などが明らかになり、エビデンスは実際の医療情報源から介入や治療の効果を統計的に推定することに焦点を当てています。

第 II 部

共通データモデル

第 4 章

共通データモデル

著者: Clair Blacketer

観察データは、患者が医療を受ける際に起こる出来事を示すものです。このデータは世界中でますます多くの患者について収集し、保存されるため、ビッグヘルスデータと呼ばれことがあります。これらのデータ収集には3つの目的があります：1) 直接的に研究を支援するため（よくあるのは調査データや登録データの形で）、2) 医療の提供をサポートするため（いわゆる EHR - 電子的健康記録）、3) 医療の費用を管理するため（いわゆる保険請求データ）。この3つの目的はすべて臨床研究に日常的に使用されており、後者の二つは二次利用データとして使用され、すべての目的が独自のフォーマットやエンコーディングを持っています。

なぜ観察医療データに共通データモデルが必要なのでしょうか？

それぞれの主要なニーズに応じて、観察データベースが臨床イベントをすべて均等に捉えることはできません。そのため、潜在的な捕捉バイアスの影響を理解するには、多くの異なるデータソースから研究結果を導き出し、比較・対照する必要があります。さらに、統計的に有効な結論を導くには、多数の観察対象の患者が必要です。これが、複数のデータソースを同時に評価・分析する必要性を説明するものです。そのためには、データを共通のデータ標準に統合する必要があります。さらに、患者データの高度な保護も必要です。従来のように分析目的でデータを抽出するには、厳格なデータ利用契約と複雑なアクセス制御が必要です。共通のデータ標準により抽出ステップを省略し、標準化された分析をネイティブ環境のデータ上で実行できるようにすることで、このニーズを軽減することができます。分析はデータにアクセスするのではなく、データが分析にアクセスするのです。

この標準を提供するのが共通データモデル（CDM）です。標準化された内容（第5章参照）と組み合わせた CDM は、研究方法が体系的に適用され、意味のある比較可能な再現性のある結果を生成することを保証します。この章では、データモデル自体の概要、デザイン、規約、一部のテーブルについて概説します。

CDM のすべてのテーブルの概要は、図 4.1に示されています。

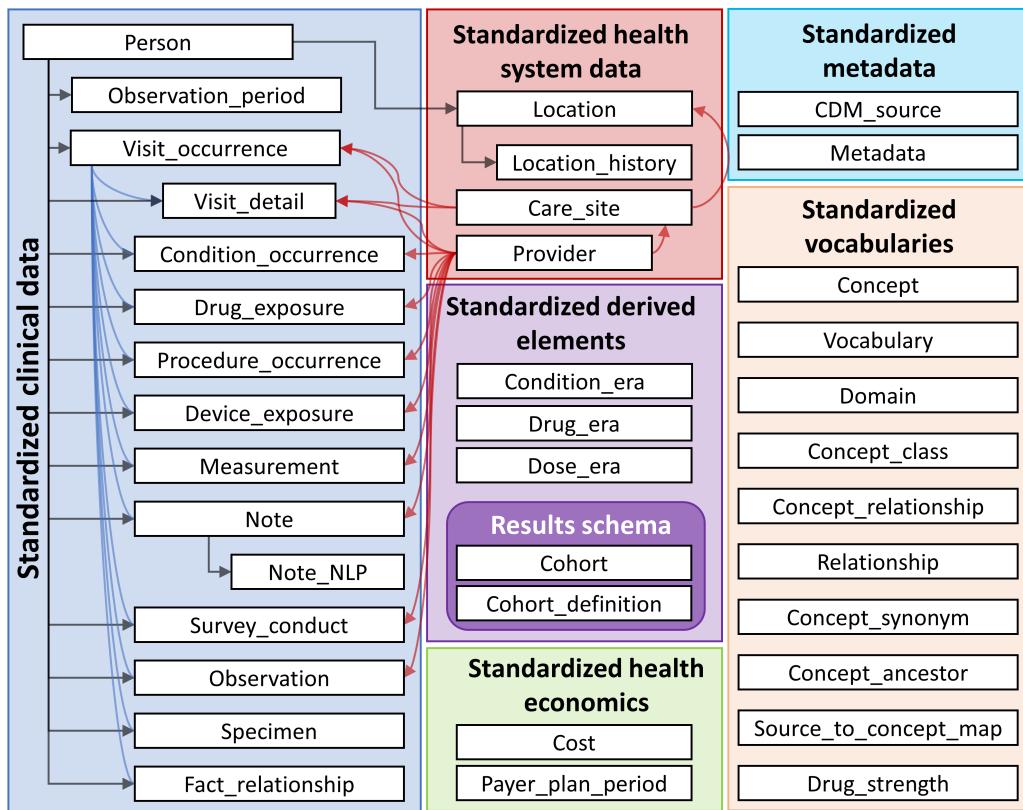


Figure 4.1: CDM バージョン 6.0 のすべてのテーブルの概要。テーブル間のすべての関係が示されているわけではありません。

4.1 デザインの原則

CDM は、以下の目的のために最適化されています。

- 特定の医療介入（薬物曝露、処置（プロシージャー）、医療政策の変更など）やアウトカム（コンディション、プロシージャー、他の薬物曝露など）を持つ患者集団を特定する。
- 人口統計情報、疾患の自然経過、医療提供、利用と費用、併存疾患、治療や治療の順序などさまざまなパラメータに関して患者集団の特性を評価する。
- 個々の患者でアウトカムが発生する可能性を予測する —— 第 13 章参照。
- これらの介入が集団に及ぼす影響を推定する —— 第 12 章参照。

この目標を達成するために、CDM の開発は以下のデザイン要素に従います：

- 目的適合性: CDM は、医療従事者または保険者の運用ニーズを満たす目的ではなく、分析のために最適な形でデータを提供することを目指しています。

ます。

- データ保護: 名前や正確な誕生日など、患者の身元や保護を危うくする可能性のあるすべてのデータは限定されています。乳児の研究のために正確な誕生日が必要な場合など、研究でより詳細な情報が明示的に必要な場合は例外となることがあります。
- ドメインの設計: ドメインは、各レコードに最低限、個人の識別情報と日付が記録される、個人中心のリレーションナルデータモデルでモデル化されています。ここでいうリレーションナルデータモデルとは、主キーと外部キーでリンクされたテーブルの集合としてデータが表現されるものを指します。
- ドメインの根拠: ドメインは、分析のユースケースがある場合（たとえばコンディション）で、そのドメインが他のものには適用されない特定の属性を持つ場合に、エンティティ-リレーションシップモデルで特定され、個別に定義されます。他のデータはすべて、エンティティ-属性-値構造のオブザーバーション（観察）テーブルに保持できます。
- 標準化ボキャブラリ: それらの記録の内容を標準化するために、CDM は、標準的な医療コンセプトのすべてに対応する、必要かつ適切な標準化ボキャブラリに依存しています。
- 既存のボキャブラリの再利用: 可能な場合、国立医学図書館、退役軍人省、疾病予防管理センターなどの国立または業界の標準化やボキャブラリ定義を行う組織やイニシアチブはこれらのコンセプトを活用しています。
- ソースコードの保持: すべてのコードが標準化ボキャブラリにマッピングされている場合でも、モデルは元のソースコードも保持して、情報が失われないようにしています。
- 技術の中立性: CDM は特定のテクノロジーを必要としません。Oracle、SQL Server などのあらゆるリレーションナルデータベース、または SAS 分析データセットとして実現できます。
- スケーラビリティ: CDM は、データベースに含まれる何億人もの人々や何十億件もの臨床観察データなど、さまざまな規模のデータソースに対応できるよう、データ処理と計算分析用に最適化されています。
- 後方互換性: これまでの CDM からの変更はすべて github リポジトリ(<https://github.com/OHDSI/CommonDataModel>)で明確に示されています。CDM の旧バージョンは現在のバージョンから簡単に作成でき、以前に存在していた情報が失われることはありません。

4.2 データモデルの規約

CDM では、暗黙的および明示的な規約が数多く採用されています。CDM に対応するメソッドの開発者は、これらの規約を理解する必要があります。

4.2.1 モデルの一般的な規約

CDM は「人中心」のモデルと見なされており、すべての臨床イベントのテーブルが PERSON テーブルにリンクされています。日付または開始日と組み合わ

せることで、人ごとに医療関連イベントをすべて縦断的に見ることができます。このルールに対する例外は、標準化された医療システム・データ・テーブルで、これはさまざまなドメインのイベントに直接リンクされています。

4.2.2 スキーマの一般的な規約

スキーマ（または一部のシステムではデータベースユーザー）により、読み取り専用テーブルと読み取り/書き込みテーブルを分離することができます。臨床イベントやボキャブラリテーブルは「CDM」スキーマにあり、エンドユーザー や分析ツールからは読み取り専用と見なされます。ウェブベースのツールやエンドユーザーが操作する必要のあるテーブルは、「Results」スキーマに格納されます。「Results」スキーマの 2 つのテーブルは、COHORT と COHORT_DEFINITION です。これらのテーブルは、ユーザーが定義する可能性のあるグループを記述することを目的としています。詳細は第 10 章を参照ください。これらのテーブルは書き込み可能であり、実行時に COHORT テーブルにコホートを保存することができます。すべてのユーザーに対して読み取り/書き込み可能なスキーマは 1 つだけなので、複数のユーザー アクセスをどのように構成し制御するかは、CDM の実装次第です。

4.2.3 データテーブルの一般的な規約

CDM はプラットフォームに依存しません。データ型は ANSI SQL データ型 (VARCHAR、INTEGER、FLOAT、DATE、DATETIME、CLOB) を用いて一般的に定義されます。VARCHAR には最小必要文字列長のみが指定され、具体的な CDM のインスタンス内で拡張できます。CDM は日付および日時の形式を規定しません。CDM に対する標準クエリは、ローカルインスタンスや日付/日時の設定によって異なる場合があります。

注意: データモデル自体はプラットフォームに依存しませんが、それに対応するために構築された多くのツールは特定の仕様が必要です。詳細については、第 8 章をご覧ください。

4.2.4 ドメインの一般的な規約

異なる性質のイベントはドメインに整理されています。これらのイベントはドメイン固有のテーブルやフィールドに格納され、標準化ボキャブラリで定義されたドメイン固有の標準コンセプトによって表現されます（セクション 5.2.3 参照）。各標準コンセプトには一意のドメイン割り当てがあり、どのテーブルに記録されるかを定義します。正しいドメイン割り当てはコミュニティ内で議論の余地がありますが、この厳格なドメインテーブルフィールドの対応規則により、コードやコンセプトの記録場所が常に明確であることが保証されます。例えば、徴候、症状、診断のコンセプトはコンディション・ドメインに属し、CONDITION_OCCURRENCE テーブルの CONDITION_CONCEPT_ID に記録されます。処置薬と呼ばれるものは通常、ソースデータのプロシージャーテーブルにプロシージャーコードとして記録されます。CDM では、対応する標準

コンセプトにドメイン割り当て「Drug」が設定されているため、これらのレコードは DRUG_EXPOSURE テーブルに記録されます。ドメインの総数は 30 で、表4.1に示されています。

Table 4.1: 各ドメインに属する標準コンセプトの数

コンセプト数	ドメイン ID	コンセプト数	ドメイン ID
1731378	薬剤 (Drug)	183	経路 (Route)
477597	デバイス (Device)	180	通貨 (Currency)
257000	プロシージャー (Procedure)	158	支払者 (Payer)
163807	コンディション (Condition)	123	ビジット (受診期間) (Visit)
145898	オブザベーション (Observation)	51	費用 (Cost)
89645	メジャーメント (測定) (Measurement)	50	人種 (Race)
33759	特定の解剖学的部位 (Spec Anatomic Site)	13	プランの中止理由 (Plan Stop Reason)
17302	測定値 (Meas Value)	11	プラン (Plan)
1799	試料 (Specimen)	6	エピソード (Episode)
1215	医療従事者専門 (Provider Specialty)	6	スポンサー (Sponsor)
1046	単位 (Unit)	5	測定値符号 (Meas Value Operator)
944	メタデータ (Metadata)	3	特定の疾患ステータス (Spec Disease Status)
538	収益コード (Revenue Code)	2	性別 (Gender)
336	タイプコンセプト (Type Concept)	2	民族性 (Ethnicity)
194	関係性 (Relationship)	1	オブザベーション タイプ (Observation Type)

4.2.5 コンテンツのコンセプトによる表現

CDM データテーブル内の各レコードのコンテンツは完全に正規化され、コンセプトを通じて表現されます。コンセプトは CONCEPT テーブルへの外部キーである CONCEPT_ID 値とともにイベントテーブルに格納され、CONCEPT テーブルは一般的な参照テーブルとして機能します。CDM インスタンスはすべて、コンセプトの参照として同じ CONCEPT テーブルを使用します。これにより、CDM とともに OHDSI 研究ネットワークの基盤となる相互運用性の主要なメカニズムが提供されます。標準コンセプトが存在しない場合や特定できない場合、CONCEPT_ID の値は 0 に設定されます。これは、存在しないコンセプト、不明またはマッピング不可能な値を表します。

CONCEPT テーブルのレコードには、各コンセプトの詳細情報（名前、ドメイン、クラスなど）が含まれています。コンセプト、コンセプトリレーションシップ、コンセプトの祖先など、コンセプトに関連するその他の情報は、標準化ボキャブラリのテーブルに含まれています（第 5 章を参照）。

4.2.6 フィールドの一般的な命名規約

すべてのテーブルの変数名は 1 つの規約に従います。

Table 4.2: フィールド名の規約

記法	説明
[Event]_ID	各レコードの固有の識別子で、イベントテーブル間の関係を確立する外部キーとして機能します。例えば、PERSON_ID は各個人を一意に識別します。VISIT_OCCURRENCE_ID はビジットを一意に識別します。
[Event]_CONCEPT_ID	CONCEPT 参照テーブルの標準コンセプトレコードへの外部キーです。これはイベントの主要な表現として機能し、標準化された分析の主要な基礎となります。たとえば、CONDITION_CONCEPT_ID = 31967 には SNOMED コンセプトの「吐き気」の参照値が含まれています。

記法	説明
[Event]_SOURCE_CONCEPT_ID	CONCEPT 参照テーブルのレコードへの外部キーです。このコンセプトは、以下のソース値に相当し、標準コンセプトの場合、それは [Event]_CONCEPT_ID と同一になります。例えば、CONDITION_SOURCE_CONCEPT_ID = 45431665は「吐き気」というコンセプトを、Read 用語で示し、同様の CONDITION_CONCEPT_ID は標準の SNOMED-CT コンセプト31967です。標準分析アプリケーションの場合、ソースコンセプトの使用は推奨されません。なぜなら、標準コンセプトだけがイベントの意味的内容を明確に示し、ソースコンセプトは相互運用可能ではないためです。
[Event]_TYPE_CONCEPT_ID	ソース情報の出所を示す標準化ボキャブラリ内で標準化されたコンセプト参照テーブルのレコードへの外部キーです。フィールド名とは異なり、これはイベントの種類やコンセプトの種類でもなく、このレコードを作成した取得メカニズムを宣言するものであることに留意ください。例として、DRUG_TYPE_CONCEPT_ID は、薬剤レコードが薬局での調剤イベント（「薬局での調剤」）から派生したものか、処方アプリケーション（「処方箋発行」）から派生したものかを区別します。

[Event]_SOURCE_VALUE

ソースデータでこのイベントがどのように表現されていたかを反映する、逐語的なコードまたは自由形式の文字列。これらのソース値はデータソース間で整合されていないため、標準的な分析アプリケーションでの使用は推奨されません。例えば、CONDITION_SOURCE_VALUE にはドットを省略した表記で書かれた ICD-9 コード 787.02 に対応する「78702」のレコードが含まれる可能性があります。

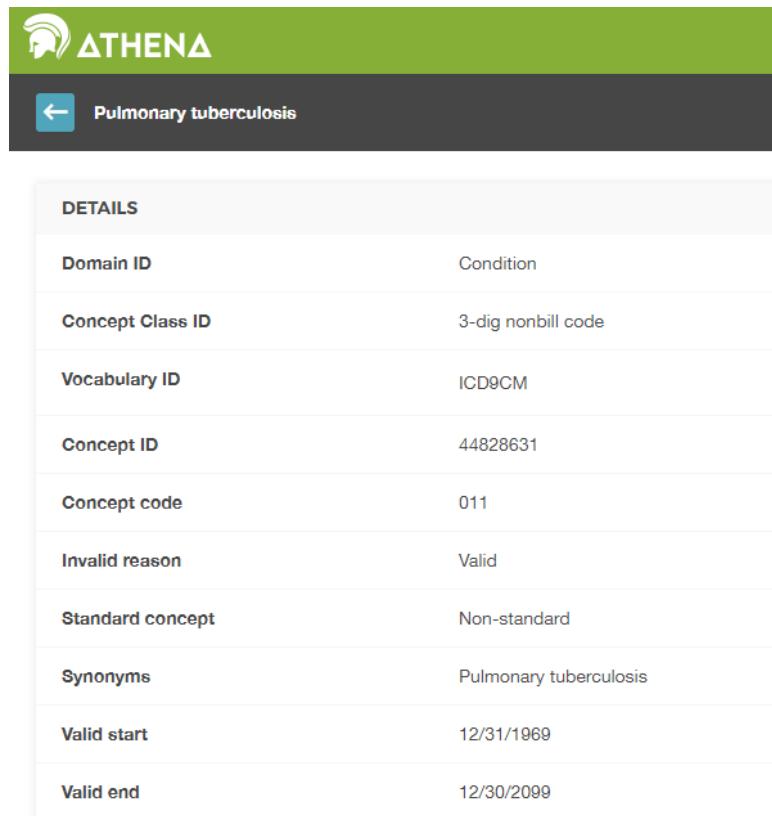
4.2.7 コンセプトとソース値の違い

多くのテーブルには、ソース値、ソースコンセプト、標準コンセプトとして、複数の場所に同等の情報が含まれています。

- ソース値は、ソースデータにおけるイベントコードのオリジナル表現です。これらは、ICD9CM、NDC、Read などのように、広く使用されているがパブリックドメインであることが多いコーディングシステムからのコード、CPT4、GPI、MedDRA などの独自のコーディングシステム、あるいは女性を F、男性を M のようにソースデータのみで使用される管理された用語集があります。また、標準化も管理もされていない短い自由テキストのフレーズもあります。ソース値は、データテーブルの [Event]_SOURCE_VALUE フィールドに格納されます。
- コンセプトは、CDM 特有のエンティティであり、臨床事実の意味を標準化します。ほとんどのコンセプトは、医療分野における既存の公開または独自仕様のコーディングシステムに基づきますが、中には新規に作成されたものもあります (CONCEPT_CODE が「OMOP」で始まる)。コンセプトには、すべてのドメインにわたって一意の ID が割り当てられています。
- ソースコンセプトは、ソースで使用されるコードを表すコンセプトです。ソースコンセプトは、既存の公開または独自のコード体系に対してのみ使用され、OMOP で生成されたコンセプトには使用されません。ソースコンセプトはデータテーブル内の [Event]_SOURCE_CONCEPT_ID フィールドに格納されます。
- 標準コンセプトは、ソースで使用されるコードシステムとは無関係に、すべてのデータベースで臨床実態の意味を一意に定義するために使用されるコンセプトです。標準コンセプトは通常、既存の公開または専有のボキャブラリソースから取得されます。標準コンセプトと同等な意味を持つ非標準コンセプトは、標準化ボキャブラリにおいて標準コンセプトにマッピングされます。標準コンセプトはデータテーブルの [Event]_CONCEPT_ID フィールドで参照できます。

ソース値は、便宜上、品質保証 (QA) の目的でのみ提供されます。ソース値に

は、特定のデータソースの文脈のみで意味を持つ情報が含まれる場合があります。ソース値やソースコンセプトの使用は任意ですが、ソースデータがコード体系を使用している場合には強く推奨されます。一方、標準コンセプトの使用は必須です。この標準コンセプトの使用が必須である理由は、すべての CDM インスタンスが同じ言語を話すことができるようになります。例えば、図 4.2 で示されているように、コンディション「肺結核」(TB) の ICD9CM コードは 011 です。



The screenshot shows the ATHENA interface with the title 'Pulmonary tuberculosis'. Below it is a table titled 'DETAILS' containing the following information:

Domain ID	Condition
Concept Class ID	3-dig nonbill code
Vocabulary ID	ICD9CM
Concept ID	44828631
Concept code	011
Invalid reason	Valid
Standard concept	Non-standard
Synonyms	Pulmonary tuberculosis
Valid start	12/31/1969
Valid end	12/30/2099

Figure 4.2: 肺結核の ICD9CM コード

文脈がなければ、コード 011 は、UB04 ボキャブラリでは「病院入院患者（メディケアパート A を含む）」、DRG ボキャブラリでは「神経系新生物（合併症、併発症なし）」と解釈される可能性があります。このような場合に、ソースと標準の両方のコンセプト ID が役立ちます。ICD9CM の 011 を表す CONCEPT_ID の値は 44828631 です。これにより、UBO4 と DRG と ICD9CM が区別されます。ICD9CM TB のソースコンセプトは、図 4.3 に示されているように、「非標準から標準へのマップ (OMOP)」という関係を通じて、SNOMED ボキャブラリーから Standard Concept 253954 にマップされます。この同じマッピング関係は、Read、ICD10、CIEL、MeSH コードなどにも存在するため、SNOMED 標準コンセプトを参照するあらゆる検索は、サポートされているすべてのソースコードを含みます。

TERM CONNECTIONS (82)			
RELATIONSHIP	RELATES TO	CONCEPT ID	VOCABULARY
ICD-9-CM to MedDRA (MSSO)	Pulmonary tuberculosis	36110777	MedDRA
Non-standard to Standard map (OMOP)	Pulmonary tuberculosis	253954	SNOMED
Subsumes	Other specified pulmonary tuberculosis	44830894	ICD9CM
	Other specified pulmonary tuberculosis, bacteriological or histological examination not done	44836741	ICD9CM
	Other specified pulmonary tuberculosis, bacteriological or histological examination unknown (at present)	44836742	ICD9CM
	Other specified pulmonary tuberculosis, tubercle bacilli found (in sputum) by microscopy	44821641	ICD9CM
	Other specified pulmonary tuberculosis, tubercle bacilli not found (in sputum) by microscopy, but found by bacterial culture	44833188	ICD9CM

Figure 4.3: 肺結核の SNOMED コード

標準コンセプトとソースコンセプトとの関係の例を表 4.7 に示します。

4.3 CDM 標準化テーブル

CDM には 16 の臨床イベントテーブル、10 のボキャブラリテーブル、2 つのメタデーターテーブル、4 つのヘルスシステムデーターテーブル、2 つの医療経済データーテーブル、3 つの標準化された派生要素、および 2 つの結果スキーマテーブルが含まれています。これらのテーブルは CDM Wiki で完全に指定されています¹。

これらのテーブルが実際にどのように使用されるかを説明するために、本章の残りの部分ではある 1 人のデータを一貫した例として使用します。

4.3.1 実行例: 子宮内膜症

子宮内膜症は、通常女性の子宮内膜にある細胞が体の他の場所に生じる痛みを伴う状態です。重症になると、不妊症、腸や膀胱の問題を引き起こすことがあります。次のセクションでは、この病気にかかった患者の経験と、それが共通データモデルでどのように表現される可能性があるかを詳しく説明します。

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



この痛みを伴う旅のすべての段階で、どれほど痛みを感じているかを皆に納得させなければなりませんでした。

Lauren は何年も子宮内膜症の症状に悩まされてきましたが、診断を受けるまでには卵巣囊腫の破裂を経験しています。Lauren についての詳細は<https://endometriosis-uk.org/laurens-story> をご覧ください。

4.3.2 PERSON テーブル

Lauren についてわかっていること

- ・彼女は 36 歳の女性です
- ・彼女の誕生日は 1982 年 3 月 12 日です
- ・彼女は白人です
- ・彼女はイギリス人です

これを踏まえると、彼女の PERSON テーブルは次のようにになります：

Table 4.3: PERSON テーブル

列名	値	説明
PERSON_ID	1	PERSON_ID はソースから直接あれ、ビルドプロセスの一部として生成されたものであれ、整数である必要があります。
GENDER_CONCEPT_ID	8532	女性性別を参照するコンセプト ID は8532です。
YEAR_OF_BIRTH	1982	
MONTH_OF_BIRTH	3	
DAY_OF_BIRTH	12	
BIRTH_DATETIME	1982-03-12 00:00:00	時間が不明の場合は真夜中が使用されます。
DEATH_DATETIME		

列名	値	説明
RACE_CONCEPT_ID	8527	白人を示すコンセプト ID は8527です。英国の 民族は4093769です。 どちらも正しいですが、 後者は前者に統合されま す。民族は、ETHNIC- ITY_CONCEPT_ID では なく、人種の一部として ここに格納されているこ とに留意ください。
ETHNICITY_CONCEPT_ ID	38003564	これはヒスパニック系の 人々を他の人々から区別 するための米国特有の表 記法です。この場合の 「English」という民族性 は、 RACE_CONCEPT_ID に 格納されます。米国以外 では使用されません。 38003564は「ヒスパニ ックではない」を表しま す。
LOCATION_ID		彼女の住所は不明です。
PROVIDER_ID		彼女のプライマリケア医 療従事者は不明です。
CARE_SITE		彼女の主な医療施設は不 明です。
PERSON_SOURCE_ VALUE	1	通常、これはソースデー タでの彼女の識別子です が、多くの場合、それは PERSON_ID と同じで す。
GENDER_SOURCE_ VALUE	F	ソースに表示されている 性別値がここに格納され ています。

列名	値	説明
GENDER_SOURCE_CONCEPT_ID	0	ソースの性別値が OHDSI がサポートする コーディングスキームで コード化されている場 合、そのコンセプトはこ こに格納されます。例え ば、ソースの性別が「性 別-F」であり、PCORNet ボキャブラリコンセプト に記載されている場合、 44814665がこのフィ ールドに入ります。
RACE_SOURCE_VALUE	white	ソースに表示されてい人 種値がここに格納されま す。
RACE_SOURCE_CONCEPT_ID	0	同様に GEN- DER_SOURCE_CONCEPT_ID の原則が適用されます。
ETHNICITY_SOURCE_VALUE	english	ソースに表示されている 民族値がここに格納され ます。
ETHNICITY_SOURCE_CONCEPT_ID	0	同様に GEN- DER_SOURCE_CONCEPT_ID の原則が適用されます。

4.3.3 OBSERVATION_PERIOD テーブル

OBSERVATION_PERIOD テーブルは、妥当な感度と特異度が期待されるソース システムにおいて、少なくとも患者の人口統計、コンディション、プロシージャー、薬剤が記録される期間を定義するために設計されています。保険請求デ タの場合は、通常、患者の加入期間となります。電子的健康記録 (EHR) の場 合は、より複雑です。ほとんどの医療システムでは、どの医療機関または医療 従事者を受診したかを特定しないためです。次善の策として、システム内の最 初のレコードが観察期間の開始日と見なされ、最新のレコードが終了日と見な されることがよくあります。

Lauren の観察期間はどのように定義されているのですか？

表 4.4に示される Lauren の情報が EHR に記録されているとしましょう。彼女 の観察期間の元となる彼女の受診 (Encounter) は：

Table 4.4: Lauren のヘルスケア受診

受診 ID	開始日	終了日	タイプ
70	2010-01-06	2010-01-06	外来患者
80	2011-01-06	2011-01-06	外来患者
90	2012-01-06	2012-01-06	外来患者
100	2013-01-07	2013-01-07	外来患者
101	2013-01-14	2013-01-14	歩行可能
102	2013-01-17	2013-01-24	入院患者

受診レコードに基づいて彼女の OBSERVATION_PERIOD テーブルは次のようになるかもしれません：

Table 4.5: OBSERVATION_PERIOD テーブル

列名	値	説明
OBSERVATION_PERIOD_ID	1	これは通常、自動生成された値で、テーブル内の各レコードに一意の識別子を生成します。
PERSON_ID	1	これは PERSON テーブルで Laura のレコードへの外部キーであり、PERSON を OBSERVATION_PERIOD テーブルにリンクします。
OBSERVATION_PERIOD_START_DATE	2010-01-06	これは記録上、彼女の最初の受診の開始日です。
OBSERVATION_PERIOD_END_DATE	2013-01-24	これは記録上、彼女の最後の受診の終了日です。
PERIOD_TYPE_CONCEPT_ID	44814725	“Obs Period Type (観察期間タイプ)” コンセプトクラスを持つボキャブラリにおける最良のオプション
		は44814724で、「ヘルスケア受診をカバーする期間」を表します。

4.3.4 VISIT_OCCURRENCE

VISIT_OCCURRENCE テーブルには、患者が医療システムを利用した際の情報が格納されています。OHDSI では、これらの情報を「ビジット」と呼び、個別

のイベントとして扱います。ビジットには 12 のトップカテゴリーがあり、医療が提供されるさまざまな状況を描写する広範な階層構造があります。最も多く記録されているビジットは、入院、外来、救急外来、医療機関以外の施設へのビジットです。

Lauren の受診がビジットとしてどのように表現されるか？

例として、入院受診を VISIT_OCCURRENCE テーブルで表現しましょう。

Table 4.6: VISIT_OCCURRENCE テーブル。

列名	値	説明
VISIT_OCCURRENCE_ID	514	これは通常、自動生成された値で、各レコードに一意の識別子を生成します。
PERSON_ID	1	これは PERSON テーブルで Lauren のレコードにリンクする外部キーです。
VISIT_CONCEPT_ID	9201	入院ビジットを参照するキーは9201です。
VISIT_START_DATE	2013-01-17	ビジットの開始日です。
VISIT_START_DATETIME	2013-01-17 00:00:00	ビジットの日付と時間です。時間が不明なため、0 時が使用されます。
VISIT_END_DATE	2013-01-24	ビジットの終了日です。これは 1 日のビジットである場合、終了日は開始日と一致します。
VISIT_END_DATETIME	2013-01-24 00:00:00	ビジットの終了日と時間です。時間が不明なため、0 時が使用されます。
VISIT_TYPE_CONCEPT_ID	32034	ビジットレコードの出所を示します。保険請求、病院請求、EHR など。これらのエンカウンターが EHR に似ている例として、32035（「EHR エンカウンターレコードから派生したビジット」）のコンセプト ID が使用されています。

列名	値	説明
PROVIDER_ID	NULL	エンカウンターレコードに医療従事者が関連付けられている場合、その医療従事者の ID がこのフィールドに格納されます。これが医療従事者テーブルの PROVIDER_ID フィールドの内容であるはずです。
CARE_SITE_ID	NULL	エンカウンターレコードに関連するケアサイトがある場合、そのケアサイトの ID がこのフィールドに入ります。これが CARE_SITE テーブルの CARE_SITE_ID であるはずです。
VISIT_SOURCE_VALUE	入院	ソースデータでどのように表示されるかに基づいてここに格納されます。Lauren のデータにはそれがありません。
VISIT_SOURCE_CONCEPT_ID	NULL	ソースデータが OHDSIによって認識されているボキャブラリを使用してコーディングされている場合、ソースコードを表すコンセプト ID がここに表示されます。Lauren のデータにはこの値はありません。
ADMITTED_FROM_CONCEPT_ID	NULL	既知の場合、患者が入院した場所を表すコンセプトが表示されます。このコンセプトのドメインは「ビジット」であるべきです。例えば、患者が自宅から病院に入院した場合には、コンセプト ID 8536 「自宅」が含まれます。

列名	値	説明
ADMITTED_FROM_SOURCE_CONCEPT_ID	NULL	患者が入院した元の場所を表すソース値が表示されます。上記の例では「自宅」です。
DISCHARGE_TO_CONCEPT_ID	NULL	既知の場合、患者が退院した先の場所を表すコンセプトが含まれます。このコンセプトのドメインは「ビジット」であるべきです。例えば、患者が介護付き生活施設に退院した場合、コンセプト ID 8615 「介護付き生活施設」となります。
DISCHARGE_TO_SOURCE_VALUE	NULL	患者が退院した場所を表すソース値が含まれます。上記の例では「介護付き生活施設」となります。
PRECEDING_VIS	NULL	現在のビジットの直前のビジットを示します。ADMITTED_FROM_CONCEPT_ID とは対照的に、ビジットコンセプトではなく、実際のビジット発生記録にリンクします。また、注意すべきは後続のビジットがないことです。ビジット発生記録は、このフィールドを通じてのみリンクされます。

- 患者は、入院患者の場合によくあるように、1回の来院中に複数の医療従事者とやりとりすることができます。これらのやりとりは、VISIT_DETAIL テーブルに記録することができます。この章では深く掘り下げませんが、VISIT_DETAIL テーブルの詳細については、CDM wikiを参照ください。

4.3.5 CONDITION_OCCURRENCE

CONDITION_OCCURRENCE テーブルのレコードは、医療従事者によって観察された、または患者によって報告された、コンディションの診断、徵候、または症状です。

Lauren のコンディションは何ですか？

彼女の記録を再確認すると、次のように述べられています。：

約 3 年前、それまでにも痛かった生理痛がますますひどくなっていることに気づきました。直腸のすぐ近くに鋭い突き刺すような痛みを感じ、尾骨と骨盤下部のあたりが圧痛と腫れを伴っていることに気づきました。生理痛がひどくなり、月に 1~2 日は仕事を休むほどでした。鎮痛剤で痛みを和らげられることもありましたが、あまり効果はありませんでした。

月経痛（月経困難症）の SNOMED コードは 266599000 です。表 4.7 は、それが CONDITION_OCCURRENCE テーブルでどのように表現されるかを示しています。

Table 4.7: CONDITION_OCCURRENCE テーブル

列名	値	説明
CONDITION_OCCURRENCE_ID	964	これは通常、自動生成された値で、各レコードに一意の識別子を生成します。
PERSON_ID	1	これは、PERSON テーブルの Laura のレコードへの外部キーであり、PERSON と CONDITION_OCCURRENCE をリンクしています。
CONDITION_CONCEPT_ID	194696	SNOMED コード 266599000 を表す外部キーは 194696 です。
CONDITION_START_DATE	2010-01-06	コンディションが記録された日付です。
CONDITION_START_DATETIME	2010-01-06 00:00:00	コンディションが記録された日時です。時刻は不明なので、0 時が使用されます。
CONDITION_END_DATE	NULL	コンディションが終了したと見なされる日付ですが、これはほとんど記録されていません。
CONDITION_END_DATETIME	NULL	既知の場合、コンディションが終了したと見なされる日時です。

列名	値	説明
CONDITION_TYPE_CONCEPT_ID	32020	この列は、レコードの由来に関する情報を提供することを目的としています。すなわち、保険請求、病院の請求記録、EHR などから取得されたものであることを示すものです。この例では、エンカウンターが電子カルテに類似しているため、32020 「EHR エンカウンター診断」) というコンセプトが使用されています。このフィールドのコンセプトは、“Condition Type (コンディションタイプ)” のボキャブラリに属するべきです。
CONDITION_STATUS_CONCEPT_ID		これが分かると、状況と理由がわかります。例えば、入院時の診断が条件である場合、コンセプト ID 4203942 が使用されました。
STOP_REASON	NULL	既知の場合、ソースデータに示されているコンディションが存在しなくなった理由。
PROVIDER_ID	NULL	コンディションレコードに診断を付けた医療従事者がリストされている場合、その医療従事者の ID がこのフィールドに入ります。これは、そのビットの医療従事者を表す PROVIDER テーブルの provider_id でなければなりません。

列名	値	説明
VISIT_OCCURRENCE_ID	509	コンディションが診断されたビジット(VISIT_OCCURRENCE テーブルの VISIT_OCCURRENCE_ID に対する外部キー)。
CONDITION_SOURCE_VALUE	266599000	これはコンディションを表す元のソース値です。Lauren の月経困難症の場合、そのコンディションの SNOMED コードはここに格納され、そのコードを表す概念は CONDITION_SOURCE_CONCEPT_ID に格納され、そこからマッピングされた標準コンセプトは CONDITION_CONCEPT_ID フィールドに格納されます。
CONDITION_SOURCE_CONCEPT_ID	194696	ソースからのコンディションの値が OHDSI で認識されるボキャブラリを使用してコード化されている場合、その値を表すコンセプト ID がここに入ります。月経困難症の例では、ソース値は SNOMED コードなので、そのコードを表すコンセプトは 194696 です。この場合、CONDITION_CONCEPT_ID フィールドと同じ値になります。

CONDITION_STATUS_	0	もしソースからのコンディション・ステータス値が OHDSI がサポートするコード化スキームでコード化されていれば、そのコンセプトはここに入ります。
SOURCE_VALUE		

4.3.6 DRUG_EXPOSURE

DRUG_EXPOSURE テーブルは、患者の体内への薬剤の意図的使用または実際の導入に関する記録を取得します。薬剤には、処方薬、市販薬、ワクチン、高分子生物学的製剤が含まれます。薬剤への曝露は、オーダーに関連する臨床イベント、記載された処方箋、薬局での調剤、処置薬としての投与、およびその他の患者報告情報から推測されます。

Lauren の薬物への曝露はどのように表現されますか？

月経困難症の痛みを改善するために、Lauren は 2010 年 01 月 06 日のビジット時に、375mg の経口投与のアセトアミノフェン（別名パラセタモール、例えば米国では NDC コード 69842087651 で販売）を 60 錠、30 日分が出されました。DRUG_EXPOSURE テーブルでは以下のようになります：

Table 4.8: DRUG_EXPOSURE テーブル

列名	値	説明
DRUG_EXPOSURE_ID	1001	通常、各レコードの一意な識別子を作成するために自動生成される値です。
PERSON_ID	1	PERSON テーブルの Lauren のレコードに対する外部キーで、PERSON と DRUG_EXPOSURE をリンクしています。
DRUG_CONCEPT_ID	1127433	薬剤のコンセプト。アセトアミノフェンの NDC コードは RxNorm コード 313782 に対応し、コンセプト 1127433 を表します。
DRUG_EXPOSURE_START_DATE	2010-01-06	薬剤曝露の開始日。

列名	値	説明
DRUG_EXPOSURE_START_DATETIME	2010-01-06 00:00:00	薬剤曝露の開始日時。時間が不明なため 0 時を使用。
DRUG_EXPOSURE_END_DATE	2010-02-05	薬剤曝露の終了日。様々な情報源によって、既知の日付または推測される日付となり、患者が薬物に曝露されていた最後の日を示します。この場合、Lauren が 30 日分を持っていたことが分かっているので、この日付が推測されます。
DRUG_EXPOSURE_END_DATETIME	2010-02-05 00:00:00	薬剤曝露の終了日時。 DRUG_EXPOSURE_END_DATE と同様のルールが適用されます。時刻が不明な場合は 0 時が使用されます。
VERBATIM_END_DATE	NULL	情報源が実際の終了日を明確に記録している場合。推定される終了日は、患者によって全日数分が使用されたという仮定に基づいています。
DRUG_TYPE_CONCEPT_ID	38000177	この欄は、記録の出所に関する情報（保険請求や処方箋の記録など）を提供するためのものです。この例では、コンセプト 38000177 (“Prescription written”) が使用されています。
STOP_REASON	NULL	薬剤の投与が中止された理由。理由にはレジメンの完了、変更、削除などが含まれます。この情報はほとんど記録されません。

列名	値	説明
REFILLS	NULL	多くの国で処方システムの一部となっている、初回処方後の自動再処方数。最初の処方はカウントされず、値は NULL から始まります。Lauren のアセトアミノフェンの場合、リフィルはなかったので、値は NULL です。
QUANTITY	60	最初の処方箋または調剤記録に記録された薬剤の量。
DAY_SUPPLY	30	処方された薬の処方日数。
SIG	NULL	元の処方箋または調剤記録に記録されている（米国の薬剤処方システムでは容器に印刷されている）薬剤処方箋の指示（「signetur」）。signetur は CDM ではまだ標準化されておらず、逐語的に提供されます。
ROUTE_CONCEPT_ID	4132161	このコンセプトは、患者が曝露された薬剤の投与経路を表すものです。Lauren はアセトアミノフェンを経口摂取したので、コンセプト ID 4132161 (“Oral (経口)”) が使用されています。
LOT_NUMBER	NULL	製造業者から薬剤の特定の数量またはロットに割り当てられた識別子。この情報はほとんど取得されません。

列名	値	説明
PROVIDER_ID	NULL	薬剤レコードに処方プロバイダがリストされている場合、そのプロバイダの ID がこのフィールドに入ります。その場合、このフィールドには PROVIDER テーブルの PROVIDER_ID が入ります。
VISIT_OCCURRENCE_ID	509	薬剤が処方された VISIT_OCCURRENCE テーブルへの外部キー。
VISIT_DETAIL_ID	NULL	薬剤が処方された VISIT_DETAIL テーブルへの外部キー。
DRUG_SOURCE_VALUE	69842087651	ソース・データに表示される薬剤のソースコードです。Lauren の場合、NDC コードがここに格納されています。
DRUG_SOURCE_CONCEPT_ID	750264	薬剤のソースデータでの値を表すコンセプトです。コンセプト 750264 NDC コードで”Acetaminophen 325 MG Oral Tablet (アセトアミノフェン 325 MG 経口錠) “を表します。
ROUTE_SOURCE_VALUE	NULL	情報源に詳述されている投与経路に関する逐語的な情報。

4.3.7 PROCEDURE_OCCURRENCE

PROCEDURE_OCCURRENCE テーブルには、医療従事者が診断または治療目的で患者に命じた、または実施した活動やプロセスの記録が含まれます。プロシージャーは様々なデータソースに様々な形で存在し、標準化のレベルも様々です。例えば、

- 医療保険請求データには、実施されたプロシージャーを含む、提供された医療サービスの請求の一部として提出されるプロシージャーコードが含まれます。

- オーダーとしてプロシージャを取り込む電子カルテ。

Lauren はどのプロシージャを受けたか?

彼女の記述から、2013-01-14 に左卵巣の超音波検査を受け、4x5cm の囊胞があることがわかりました。PROCEDURE_OCCURRENCE テーブルでは、このように表示されます：

Table 4.9: PROCEDURE_OCCURRENCE テーブル

Column name	Value	Explanation
PROCEDURE_OCCURRENCE_ID	1277	これは通常、各レコードの一意な識別子を作成するため自動生成される値です。
PERSON_ID	1	これは PERSON テーブルのローラのレコードに対する外部キーで、PERSON と PROCEDURE_OCCURRENCE をリンクしています。
PROCEDURE_CONCEPT_ID	4127451	骨盤超音波検査の SNOMED プロシージャコードは 304435002 で、コンセプト 4127451 で表されます。
PROCEDURE_DATE	2013-01-14	プロシージャが実施された日付。
PROCEDURE_DATETIME	2013-01-14 00:00:00	プロシージャが行われた日時。時刻が不明な場合は 0 時を使用します。
PROCEDURE_TYPE_CONCEPT_ID	38000275	このカラムはプロシージャの記録の由来に関する情報を提供することを目的としています。すなわち、保険請求、EHR オーダーなどによるものかどうかです。この例では、プロシージャの記録が EHR によるものであるため、コンセプト ID 38000275 (「EHR order list entry」) が使用されています。

Column name	Value	Explanation
MODIFIER_CONCEPT_ID	0	これは手技の修飾子を表すコンセプト ID を意味します。例えば、CPT4 のプロシージャーが両側で行われたと記録されている場合、コンセプト ID 42739579 (「両側プロシージャー」) が使用されます。
QUANTITY	0	オーダーされた、または実施されたプロシージャーの数。数がない場合、0 と 1 はすべて同じ意味です。
PROVIDER_ID	NULL	Procedure レコードに Provider がリストされている場合、その Provider の ID がこのフィールドに入ります。これは PROVIDER テーブルの PROVIDER_ID に対する外部キーでなければなりません。
VISIT_OCCURRENCE_ID	740	情報がある場合には、これはプロシージャーが施行されたビジットです (VISIT_OCCURRENCE テーブルから取得した VISIT_occurrence_id として表されます)。
VISIT_DETAIL_ID	NULL	情報がある場合、プロシージャーが実施されたビジット詳細です (VISIT_DETAIL テーブルから VISIT_detail_id として取得)。
PROCEDURE_SOURCE_VALUE	304435002	ソース・データに表示されているプロシージャーのコードまたは情報。
PROCEDURE_SOURCE_CONCEPT_ID	4127451	プロシージャーのソースデータの値を表すコンセプトです。

Column name	Value	Explanation
MODIFIER_SOURCE_VALUE	NULL	ソース・データに表示される修飾子のソース・コード。

4.4 追加情報

本章では、CDM に用意されている表の一部のみを取り上げ、データの表現方法の例として紹介しています。より詳しい情報については、Wiki²をご覧ください。

4.5 まとめ



- CDM は広範囲の観察研究活動をサポートするように設計されています。
- CDM は人中心のモデルです。
- CDM はデータの構造を標準化するだけでなく、標準化ボキャブラリを通じてコンテンツの表現も標準化します。
- 完全な追跡可能性を確保するために、ソースコードは CDM で管理されています。

4.6 演習

前提条件

これらの最初の練習問題のために、以前に議論された CDM テーブルを確認する必要があります、ATHENA³または ATLAS⁴を通じて語彙内のコンセプトを調べる必要があります。

演習 4.1. ジョンは 1974 年 8 月 4 日生まれのアフリカ系アメリカ人男性です。この情報をエンコードする PERSON テーブルのエントリを定義してください。

演習 4.2. ジョンは 2015 年 1 月 1 日に現在の保険に加入しました。彼の保険データは 2019 年 7 月 1 日に抽出されました。この情報をエンコードする OBSERVATION_PERIOD テーブルのエントリを定義してください。

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

³<http://athena.ohdsi.org/>

⁴<http://atlas-demo.ohdsi.org>

演習 4.3. ジョンは 2019 年 5 月 1 日にイブプロフェン 200 MG 経口錠剤 (NDC コード : 76168009520) の 30 日分の供給を処方されました。この情報をエンコードする DRUG_EXPOSURE テーブルのエントリを定義してください。

前提条件

最後の 3 つの課題には、セクション 8.4.5 で説明されているように R、R-Studio、および Java がインストールされていることが前提となります。また、SqlRender、DatabaseConnector、および Eunomia パッケージも必要で、以下のコマンドでインストールできます：

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

Eunomia パッケージは、ローカルの R セッション内で実行される CDM 内のシミュレートされたデータセットを提供します。接続の詳細は以下を使用して取得できます：

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

CDM データベーススキーマは「main」です。これは CONDITION_OCCURRENCE テーブルの一一行を取得するための SQL クエリの例です：

```
library(DatabaseConnector)
connection <- connect(connectionDetails)
sql <- "SELECT *
FROM @cdm.condition_occurrence
LIMIT 1;"
result <- renderTranslateQuerySql(connection, sql, cdm = "main")
```

演習 4.4. SQL と R を使用して、「消化管出血」(コンセプト ID192671) のすべてのレコードを取得してください。

演習 4.5. SQL と R を使用して、ソースコードを使用して「消化管出血」のすべてのレコードを取得してください。このデータベースは ICD-10 を使用しており、関連する ICD-10 コードは「K92.2」です。

演習 4.6. SQL と R を使用して、PERSON_ID 61 の人物の観察期間を取得してください。

提案される答えは付録 E.1 にあります。

第 5 章

標準化ボキャブラリ

著者: Christian Reich & Anna Ostropolets

OMOP 標準化ボキャブラリは、単に「ボキャブラリ」と呼ばれることが多く、データの内容を定義することで、手法、定義、結果の標準化を可能にし、真のリモート（ファイアウォール内）ネットワーク研究と分析への道を開きます。通常、コーディングスキームを使用した構造化データであるか、自由形式のテキストであるかに関わらず、観察医療データのコンテンツを見つけ、解釈することは、臨床イベントを記述する無数の異なる方法に直面する研究者へと引き継がれます。OHDSI では、標準化されたフォーマットだけでなく、厳格な標準コンテンツへの調和も必要とされています。本章では、まず標準化ボキャブラリの主な原則、その構成要素、関連する規則、慣例、および典型的な状況について説明します。これらはすべて、この基盤となるリソースを理解し活用するために必要なものです。また、継続的に改善していくためにコミュニティのサポートが必要な箇所についても指摘します。

5.1 なぜボキャブラリが必要で、なぜ標準化が必要なのか

医学ボキャブラリの歴史は、中世のロンドンでペストやその他の疾患の流行を管理するために作成された死亡報告書（“Bill of Mortality”）に遡ります（図 5.1 参照）。

それ以来、分類の規模と複雑性は大幅に拡大し、医療の他の側面、例えばプロセッジャー（処置）やサービス、薬剤、医療デバイスなどにも広がりました。主な原則は変わっていません：つまり、患者データを収集、分類、分析するためにいくつかの医療コミュニティが合意した管理されたボキャブラリ、専門用語、階層やオントロジーです。これらの多くのボキャブラリは、公的あるいは政府機関によって長期的に管理されます。例えば、世界保健機関（WHO）は最近第 11 版（ICD11）が追加された “International Classification of Disease (ICD)” を作成しています。各国の政府は、ICD10CM（米国）、ICD10GM（ド

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	Palsey	—	—	—	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	Quinify and sore Throat	—	—	—	21
	Blasted and Planet	—	Gout	—	4	Rickets	—	—	—	441
	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 themselves	{	11	Scurvy	—	—	—	82
	Cancer, Gangrene and Fistula	63	Head-ach and Headmouldshot	—	35	Shot	—	—	—	7
	Canker, sore Mouth and Thrush	73	Jaundies	—	102	Shingles	—	—	—	1
	Childbed	226	Imposthume	—	105	Sores, Ulcers, broken and bruised Limbs	—	{	61	
	Chrisomes and Infants	858	Killed by several Accidents	—	55	Spleen	—	—	—	7
	Cold, Cough and Hiccup	33	King's Evil	—	28	Spotted Fever and Purples	—	—	—	368
	Colick and Wind	116	Lethargy	—	6	Starved	—	—	—	7
	Consumption and Tisick	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	4	Overlaid and Starved at Nurse	—	46	Wen	—	—	—	1

Figure 5.1: 1660 年のロンドン死亡報告書には、その時代に知られていた 62 の疾患の分類システムを使用して住民の死因が示されています。

イツ）など、各国独自のバージョンを作成しています。政府はまた、薬品のマーケティングと販売を管理し、認証された薬剤の国家リポジトリを維持しています。ボキャブラリは民間部門でも、商業製品として、あるいはEHRシステムや保険請求報告書作成などの社内利用のために使用されています。

その結果、各国、各地域、医療制度、医療機関は、それぞれ独自の分類法を持つ傾向にあり、それは使用される場所でのみ関連性がある可能性が高いものです。こうした無数のボキャブラリが、使用されるシステムの相互運用性を妨げています。標準化は、患者データの交換を可能にし、世界レベルでの医療データ分析を可能にし、パフォーマンス特性や品質評価を含む体系化された標準化された研究を可能にする鍵となります。この問題に対処するため、多国籍の組織が設立され、前述のWHOや、Standard Nomenclature of Medicine (SNOMED)や、Logical Observation Identifiers Names and Codes (LOINC)などの幅広い標準の作成を開始しました。米国では、Health IT Standards Committee (HITAC)が、SNOMED、LOINC、および薬剤用ボキャブラリであるRxNormを、さまざまな組織間で全国規模の医療情報交換を行う共通プラットフォームで使用するための標準として、National Coordinator for Health IT (ONC)に推奨しています。

OHDSIは、観察研究のためのグローバルスタンダードであるOMOP CDMを開発しました。OMOP標準化ボキャブラリは、CDMの一部として、主に次の2つの目的で利用できます。

- コミュニティで使用されるすべてのボキャブラリの共通リポジトリ
- 研究使用のための標準化とマッピング

標準化ボキャブラリはコミュニティに無料で提供されており、OMOP CDMインスタンスでは必須の参照テーブルとして使用する必要があります。

5.1.1 標準化ボキャブラリの構築

標準化ボキャブラリのすべてのボキャブラリは、共通の形式に統合されています。これにより、研究者が元のボキャブラリの複数の異なる形式とライフサイクルの慣例を理解して扱う必要がなくなります。すべてのボキャブラリは定期的に更新され、Pallasシステムを使用して統合されます¹。これは、全体のOMOP CDMワークグループの一部であるOHDSIボキャブラリチームによって構築と運営がなされています。誤りを見つけた場合は、OHDSIフォーラム²またはCDM GitHubページ³に投稿して、私たちのリソースを改善するのにご協力ください。

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

²<https://forums.ohdsi.org>

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

5.1.2 標準化ボキャブラリへのアクセス

標準化ボキャブラリを得るために、自分で Pallas を実行する必要はありません。代わりに、ATHENA⁴から最新バージョンをダウンロードし、ローカルデータベースにロードできます。ATHENA では、ボキャブラリのファセット検索も可能です。

OMOP CDM のボキャブラリをすべて選んで、標準化ボキャブラリテーブルのすべてを含む zip ファイルをダウンロードします。標準コンセプトを持つボキャブラリ（セクション 5.2.6 参照）と非常に一般的な使用法は事前に選択されています。提供元データで使用されているボキャブラリを追加します。著作権のあるボキャブラリには選択ボタンがありません。「ライセンス必要」ボタンをクリックしてそのようなボキャブラリをリストに組み込みます。ボキャブラリチームが連絡し、ライセンスを提示するか、適切な人々と連絡を取り合うためのサポートを提供します。

5.1.3 ボキャブラリの元: 採用するか構築するか

OHDSI は一般に、既存のボキャブラリを採用することを優先します。なぜなら、1) 多くのボキャブラリがコミュニティ内で観察データに使用されているため、2) ボキャブラリの構築とメンテナンスは複雑で、成熟するためには多くの利害関係者の長期にわたる協力が必要だからです。このため、特定の組織がボキャブラリを提供しており、ボキャブラリは生成、廃止、統合、分割のライフサイクルの対象となります（セクション 5.2.10 参照）。現在、OHDSI はタイプコンセプト（例：コンディションタイプコンセプト）などの内部管理ボキャブラリのみを作成しています。唯一の例外は、米国以外でのみ使用される医薬品を対象とする RxNorm Extension ボキャブラリです（セクション 5.6.9 参照）。

5.2 コンセプト

OMOP CDM の臨床イベントはすべてコンセプトとして表現されます。これらはデータレコードの基本的な構成要素であり、ほとんどのテーブルは、いくつかの例外を除いて完全に正規化されています。コンセプトは CONCEPT テーブルに格納されます（図 5.2 を参照）。

このシステムは包括的であることを意味し、患者の医療体験に関連するすべてのイベント（例：コンディション、プロシージャー、薬剤曝露など）や医療システムの一部の管理情報（例：ビジット（受診期間）、医療施設など）をカバーするのに十分なコンセプトが存在します。

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 during time interval
VALID_END_DATE	31-Dec-2099	
INVALID_REASON		

Figure 5.2: OMOP CDM における標準化ボキャブラリコンセプトの標準的な表現。提示されている例は心房細動の SNOMED コードに対する CONCEPT テーブルのレコードです。

5.2.1 コンセプト ID

各コンセプトにはプライマリキーとして使用されるコンセプト ID が割り当てられます。この無意味な整数 ID は、CDM のイベントテーブルにデータを記録する際に使用され、元のボキャブラリコードではありません。

5.2.2 コンセプト名

各コンセプトには 1 つの名称が割り当てられます。名称は常に英語表記です。名称はボキャブラリのソースからインポートされます。ソースのボキャブラリに複数の名称がある場合は、最も表現力のある名称が選択され、残りの名称は同じ CONCEPT_ID キーの下にある CONCEPT_SYNONYM テーブルに保存されます。英語以外の名称も CONCEPT_SYNONYM に記録され、LANGUAGE_CONCEPT_ID フィールドに適切な言語のコンセプト ID が含まれます。名前の長さは 255 文字です。つまり、非常に長い名前は切り捨てられ、完全版は別の同義語として記録され、最大 1000 文字まで保持できます。

5.2.3 ドメイン

各コンセプトには DOMAIN_ID フィールドにドメインが割り当てられています。これは数値の CONCEPT_ID とは対照的に、ドメイン用の小文字表記の大文字小文字区別を区別する一意の英数字 ID です。例として、“Condition (コンディション)”、“Drug (薬剤)”、“Procedure (処置 (プロシージャー))”、“Visit (ビジット)”、“Device (デバイス)”、“Specimen (試料)”などのドメイン識別子があります。曖昧なコンセプトや事前にコード化された（組み合わせ）コンセプトは複合ドメインに属することがあります。標準コンセプト

(セクション 5.2.6 参照) は常に単一のドメインが割り当てられます。ドメインは、臨床イベントやイベント属性がどの CDM テーブルやフィールドに記録されるかを指示します。ドメインの割り当ては、Pallas に示されているヒューリスティックな手法を使用してボキャブラリの取り込み中に実行される OMOP 固有の機能です。ソースボキャブラリは、さまざまな程度で混合ドメインのコードを組み合わせる傾向があります (図 5.3 参照)。

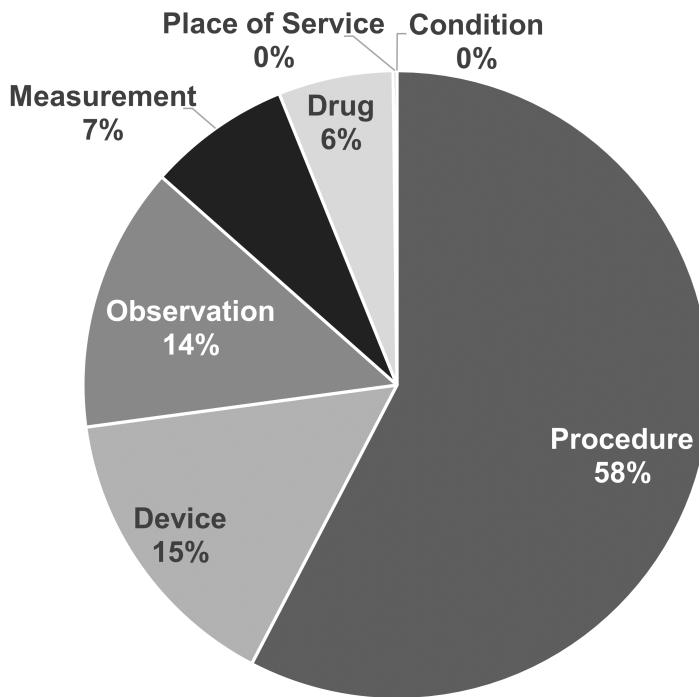


Figure 5.3: プロシージャーボキャブラリ CPT4 および HCPCS におけるドメインの割り当て。直感的には、これらのボキャブラリは単一のドメインのコードとコンセプトを含むべきですが、実際には混在しています。

ドメインのヒューリスティックは、ドメインの定義に従います。これらの定義は CDM のテーブルとフィールド定義から派生しています (セクション 4 参照)。ヒューリスティックは完全ではなく、グレーゾーンも存在します (セクション 5.6 「特別な状況」参照)。ドメインが誤って割り当てられているコンセプトがある場合、フォーラムまたは CDM 問題の投稿を通じて報告し、プロセスの改善に貢献してください。

5.2.4 ボキャブラリ

各ボキャブラリには短い大文字小文字区別のない一意の英数字 ID が割り当てられており、通常はダッシュを省略したボキャブラリの略称に続きます。例えば、ICD-9-CM のボキャブラリ ID は「ICD9CM」です。現在、OHDSI でサポートされているボキャブラリは 111 あり、そのうち 78 は外部ソースから採用されたもので、残りは OMOP 内部のボキャブラリです。これらのボキャブラリ

は通常、四半期ごとに更新されます。ボキャブラリのソースおよびバージョンは、ボキャブラリリファレンスファイルで定義されています。

5.2.5 コンセプトクラス

一部のボキャブラリでは、大文字と小文字を区別する固有の英数字 ID によって表されるコードまたはコンセプトを分類しています。例えば、SNOMED には 33 のこのようなコンセプトクラスがあり、SNOMED ではこれを「意味タグ」と呼んでいます。臨床所見、社会的背景、身体構造などです。これらはコンセプトの垂直的な区分です。MedDRA や RxNorm などの他のものには、階層化された階層構造の水平レベルを分類するコンセプトクラスがあります。HCPCS などのコンセプトクラスを持たないボキャブラリでは、ボキャブラリ ID をコンセプトクラス ID として使用します。

Table 5.1: コンセプトクラスにおける水平および垂直のサブ分類原則を持つボキャブラリと持たないボキャブラリ

コンセプトクラスの区分原則	ボキャブラリ
水平	すべての薬剤ボキャブラリ、ATC、CDT、Episode、HCPCS、HemOnc、ICDs、MedDRA、OSM、国勢調査
垂直	CIEL、HES 専門、ICDO3、MeSH、NAACCR、NDFRT、OPCS4、PCORNET、Plan、PPI、Provider、SNOMED、SPL、UCUM
混在	CPT4、ISBT、LOINC
なし	APC、すべてのタイプコンセプト、民族性、OXMIS、種族、収益コード、スポンサー、供給者、UB04、ビジット

水平コンセプトクラスにより、特定の階層レベルを決定することができます。たとえば、医薬品ボキャブラリの RxNorm におけるコンセプトクラス「Ingredient」は階層の最上位レベルを定義します。垂直モデルでは、コンセプトクラスのメンバーは最上位から最下位までの任意の階層レベルにすることができます。

5.2.6 標準コンセプト

各臨床イベントを表す 1 つのコンセプトが標準として指定されます。例えば、MESH コード D001281、CIEL コード 148203、SNOMED コード 49436004、ICD9CM コード 427.31、Read コード G573000 はすべて、コンディションドメインで「心房細動」を定義していますが、SNOMED の概念のみが標準であり、データ内のコンディションを表します。他のものは非標準またはソースコンセプトとして指定され、標準コンセプトにマッピングされています。標準コンセプトは STANDARD_CONCEPT フィールドに「S」で示されます。そして、

CDM フィールドの末尾が「_CONCEPT_ID」となっているデータ記録には、これらの標準コンセプトのみが使用されます。

5.2.7 非標準コンセプト

非標準コンセプトは臨床イベントを表現するためには使用されませんが、標準化ボキャブラリの一部であり、ソースデータに頻繁に見られます。そのため、それらは「ソースコンセプト」とも呼ばれます。ソースコンセプトを標準コンセプトに変換するプロセスは「マッピング」と呼ばれます（セクション 5.3.1 参照）。非標準コンセプトには STANDARD_CONCEPT フィールドに値がありません（NULL）。

5.2.8 分類コンセプト

これらのコンセプトは標準ではなく、したがってデータを表現するためには使用されませんが、標準コンセプトと階層的に関連しており、そのため階層クエリを実行するために使用できます。たとえば、MedDRA コード 10037908 のすべての下位層をクエリする場合（MedDRA ライセンスを取得していないユーザーには表示されません。アクセス制限についてはセクション 5.1.2 参照）では、標準の SNOMED コンセプト「心房細動」を取得します（CONCEPT_ANCESTOR テーブルを使用した階層クエリについてはセクション 5.4 を参照） - 図 5.4 を参照。

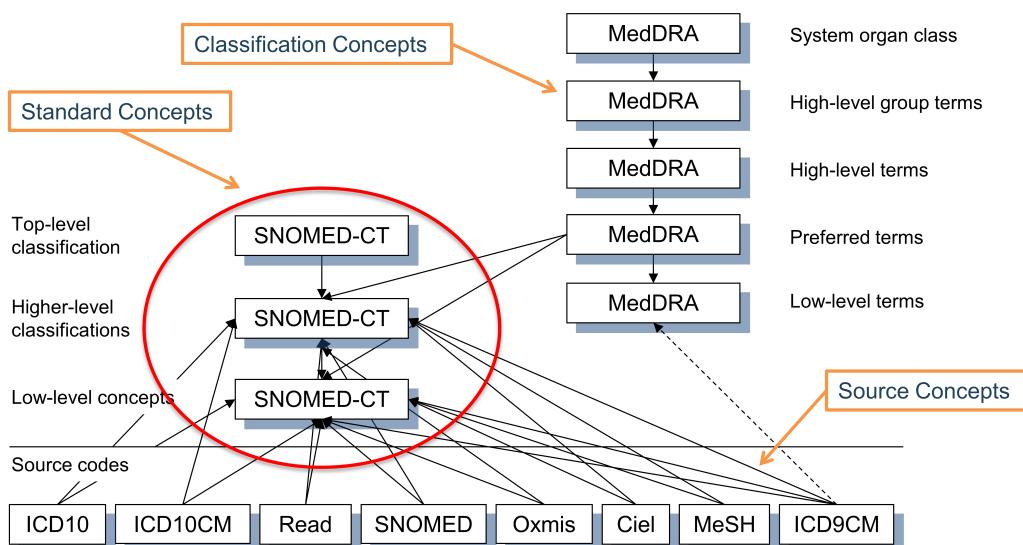


Figure 5.4: コンディションドメインにおける標準、非標準ソースおよび分類コンセプトとその階層関係。SNOMED はほとんどの標準コンディションコンセプトに使用されており（いくつかの腫瘍関連コンセプトは ICDO3 から派生）、MedDRA コンセプトは階層分類コンセプトに使用されており、他のすべてのボキャブラリは非標準またはソースコンセプトを含み、階層には含まれません。

標準、非標準、分類のコンセプトの選択は、通常各ドメインごとにボキャブラリレベルで行われます。これはコンセプトの質、組込みの階層、ボキャブラリが宣言された目的に基づいています。また、すべてのボキャブラリのコンセプトが標準コンセプトとして使用されているわけではありません。各ドメインごとに別々に指定されており、各コンセプトはアクティブである必要があります（セクション 5.2.10 参照）し、異なるボキャブラリから同じ意味を持つ複数のコンセプトが競合する場合には、優先順位が設定される場合もあります。つまり、「標準ボキャブラリ」というものは存在しません。例については表 5.2 を参照ください。

Table 5.2: 標準/非標準/分類コンセプトの割り当てに利用するボキャブラリのリスト

ドメイン	標準コンセプトのためのボキャブラリ	ソースコンセプトのためのボキャブラリ	分類コンセプトのためのボキャブラリ
コンディション	SNOMED, ICDO3	SNOMED Veterinary	MedDRA
プロシージャー	SNOMED, CPT4, HCPCS, ICD10PCS, ICD9Proc, OPCS4	SNOMED Veterinary, HemOnc, NAACCR	現時点ではなし
メジャーメント (測定)	SNOMED, LOINC	SNOMED Veterinary, NAACCR, CPT4, HCPCS, OPCS4, PPI	現時点ではなし
薬剤	RxNorm, RxNorm Extension, CVX	HCPCS, CPT4, HemOnc, NAACCR	ATC
デバイス	SNOMED	他のボキャブラリ、現在は標準化されていない	現時点ではなし
オブザベーション ビジット	SNOMED CMS Place of Service, ABMT, NUCC	他のボキャブラリ SNOMED, HCPCS, CPT4, UB04	現時点ではなし 現時点ではなし

5.2.9 コンセプトコード

コンセプトコードはソースボキャブラリで使用される識別子です。たとえば、ICD9CM または NDC コードはこのフィールドに保存され、OMOP テーブルは CONCEPT テーブルへの外部キーとしてコンセプト ID を使用します。その理

由は、ボキャブラリを超えて名前空間が重複するためです。つまり、同じコードが異なるボキャブラリに存在し、それぞれ全く異なる意味を持つ可能性があるためです（表 5.3 参照）。

Table 5.3: 同じコンセプトコード 1001 を持つが、異なる
ボキャブラリ、ドメイン、コンセプトクラスのコンセプト

コンセプト ID	コンセプトコード	コンセプト名	ドメイン ID	ボキャブラリ ID	コンセプトクラス
35803438	1001	顆粒球コロニー刺激因子	薬剤	HemOnc	コンポーネントクラス
35942070	1001	AJCC TNM Clin T	メジャー メント	NAACCR	NAACCR 変数
1036059	1001	アンチピリン	薬剤	RxNorm	成分
38003544	1001	レジデンシャル治療 - 精神科	収益コード	収益コード	収益コード
43228317	1001	アセプロメタジンマレイン酸塩	薬剤	BDPM	成分
45417187	1001	プロムフェニラミンマレイン酸塩、10 mg/ml 注射用溶液	薬剤	Multum	Multum
45912144	1001	血清	標本	CIEL	標本

5.2.10 ライフサイクル

ボキャブラリは、固定されたコードセットを持つ恒久的なコーパスであることはまれです。その代わり、コードやコンセプトは追加され、廃止されています。OMOP CDM は、患者の経時的データをサポートするモデルであり、過去に使用されていたが現在は使用されていないコンセプトをサポートする必要があるだけでなく、新しいコンセプトをサポートし、そのコンセプトを文脈に配置する必要があります。CONCEPT テーブルには、ライフサイクルのステータスを記述する 3 つのフィールドがあります。VALID_START_DATE、VALID_END_DATE、INVALID_REASON です。これらの値は、コンセプトのライフサイクルのステータスによって異なります。：

- アクティブまたは新しいコンセプト

- 説明: 使用中のコンセプト。
 - VALID_START_DATE: コンセプトの生成日。不明の場合はボキャブラリへの取り込み日。不明の場合は 1970-1-1。
 - VALID_END_DATE: 「将来、定義されていない時点で無効になる可能性があるが、現在はアクティブである」ことを示す慣例として、2099 年 12 月 31 日に設定。
 - INVALID_REASON: NULL
- 非推奨のコンセプトで後継なし
 - 説明: 非アクティブであり、標準として使用することはできない（セクション 5.2.6 参照）。
 - VALID_START_DATE: コンセプトの生成日。不明の場合はボキャブラリへの取り込み日。不明の場合は 1970-1-1。
 - VALID_END_DATE: 過去の廃止日。不明の場合はボキャブラリ内のコンセプトが欠落あるいは非アクティブに設定されたボキャブラリ更新日。
 - INVALID_REASON: “D”
 - 後継コンセプトとともにアップグレードされたコンセプト
 - 説明: コンセプトは非アクティブだが、後継コンセプトが定義されています。通常は、重複排除が行われたコンセプトです。
 - VALID_START_DATE: コンセプトの生成日。不明の場合はボキャブラリへの取り込み日、もしくは 1970-1-1。
 - VALID_END_DATE: アップグレードが行われた過去の年月日。不明の場合は、アップグレードが含まれたボキャブラリのリフレッシュ日。
 - INVALID_REASON: “U”
 - 別の新しいコンセプトで再利用されたコード
 - 説明: 非推奨のコンセプトコードが、新しいコンセプトで再利用されました。
 - VALID_START_DATE: コンセプトの生成日。不明の場合はボキャブラリへの取り込み日、もしくは 1970 年 1 月 1 日。
 - VALID_END_DATE: 非推奨であることを示す過去の日、またはそれがわからない場合は、ボキャブラリのコンセプトがなくなった、または非アクティブに設定されたボキャブラリ更新の日。
 - INVALID_REASON: “R”

一般に、コンセプトコードは再利用されません。しかし、特に HCPCS、NDC、DRG など、このルールから外れるボキャブラリがいくつかあります。これらのボキャブラリでは、同じコンセプトコードが同じボキャブラリの複数のコンセプトに現れます。CONCEPT_ID の値は一意です。これらの再使用されるコンセプトコードは、INVALID_REASON フィールドに「R」が付され、VALID_START_DATE から VALID_END_DATE の期間は、同じコンセプトコードを持つコンセプトを区別するために使用されるべきです。

5.3 関係

任意の 2 つのコンセプトは、そのドメインやボキャブラリーが同じであるかどうかに関係なく、定義された関係を持つことができます。関係の性質は、CONCEPT_RELATIONSHIP テーブルの RELATIONSHIP_ID フィールドにある、大文字小文字を区別する一意の英数字 ID で示されます。関係は対称的であり、各関係には同等の関係が存在し、フィールド CONCEPT_ID_1 と CONCEPT_ID_2 の内容が入れ替わり、RELATIONSHIP_ID はその逆に変更されます。たとえば、「Maps to」関係には反対の関係「Mapped from」があります。

CONCEPT_RELATIONSHIP テーブルのレコードには、ライフサイクルフィールド RELATIONSHIP_START_DATE、RELATIONSHIP_END_DATE、INVALID_REASON も含まれています。ただし、ATHENA を通じて利用可能なのは INVALID_REASON が NULL のアクティブなレコードのみです。非アクティブな関係は内部処理のために Pallas システムに保存されます。RELATIONSHIP テーブルは、全ての関係 ID およびその逆関係のリストを参照するためのものです。

5.3.1 マッピング関係

これらの関係は、非標準のコンセプトから標準コンセプトへの変換を提供し、2 つの関係 ID ペアによってサポートされています（表 5.4 を参照）。

Table 5.4: マッピング関係の種類

関係 ID ペア	目的
“Maps to” と “Mapped from”	標準コンセプトはそれ自身にマッピングされ、非標準コンセプトは標準コンセプトにマッピングされます。ほとんどの非標準コンセプトとすべての標準コンセプトは、標準コンセプトとの間にこの関係があります。前者は *_SOURCE_CONCEPT_ID フィールドに、後者は _CONCEPT_ID フィールド * に格納されます。分類コンセプトはマッピングされません。
“Maps to value” と “Value mapped from”	MEASUREMENT と OBSERVATION テーブルの VALUE_AS_CONCEPT_ID フィールドに配置する値を表すコンセプトへのマッピング。

これらのマッピング関係の目的は、同等のコンセプト間の相互参照を可能にし、臨床イベントが OMOP CDM でどのように表現されるかを統一することです。

これは標準化ボキャブラリの主要な成果です。

「同等のコンセプト」とは、同じ意味を持ち、さらに重要なことには、階層下位のコンセプトが同じ意味領域をカバーすることを意味します。同等のコンセプトが利用できず、コンセプトが標準でない場合、それはより広いコンセプトにマッピング（いわゆる「上方向マッピング」）されます。たとえば、ICD10CM W61.51「ガチョウに噛まれる」は、標準のコンディションコンセプトとして使用される SNOMED ボキャブラリには同等のものはありません。代わりに、それは SNOMED 217716004「鳥に突かれる」にマッピングされ、コンテキストとしての鳥がガチョウであるという情報が失われます。上方向マッピングは、情報の損失が標準的な研究用途には無関係であるとみなされる場合にのみ使用されます。

一部のマッピングでは、ソースコンセプトが複数の標準コンセプトにリンクされます。たとえば、ICD9CM 070.43「肝性昏睡を伴う E 型肝炎」は、SNOMED 235867002「急性 E 型肝炎」と SNOMED 72836002「肝性昏睡」の両方にマッピングされます。これは、元のソースコンセプトが肝炎と昏睡という 2 つの状態のあらかじめ組み合わせられたものであるためです。SNOMED にはその組み合わせがなく、その結果、ICD9CM レコードではマッピングされた標準コンセプトそれぞれに対して 2 つのレコードが作成されます。

「Maps to value」関係は、エンティティ-属性-値 (EAV) モデルに従って OMOP CDM テーブルの値を分割することを目的としています。これは次の状況で発生します：

- 検査と結果の値からなるメジャーメント
- 本人または家族の病歴
- 物質に対するアレルギー
- 予防接種の必要性

このような状況では、ソースコンセプトは属性（テストまたは履歴）と値（テスト結果または疾患）の組み合わせです。「Maps to」関係はこのソースを属性コンセプトにマッピングし、「Maps to value」は値コンセプトにマッピングします。例については図 5.5 を参照ください。

コンセプトのマッピングは、無料で提供され、ネットワーク研究を行うコミュニティの取り組みを支援する OMOP 標準化ボキャブラリのもう一つの中心的な機能です。マッピング関係は外部ソースから導出されるか、ボキャブラリチームによって手動で維持されます。つまり、それらは完璧ではないということです。誤ったマッピング関係や好ましくないマッピング関係を見つけた場合は、フォーラムや CDM の問題の投稿を通じて報告し、プロセスの改善に協力することが重要です。

マッピング規則の詳細な説明は、OHDSI Wiki で見つけることができます⁵。

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

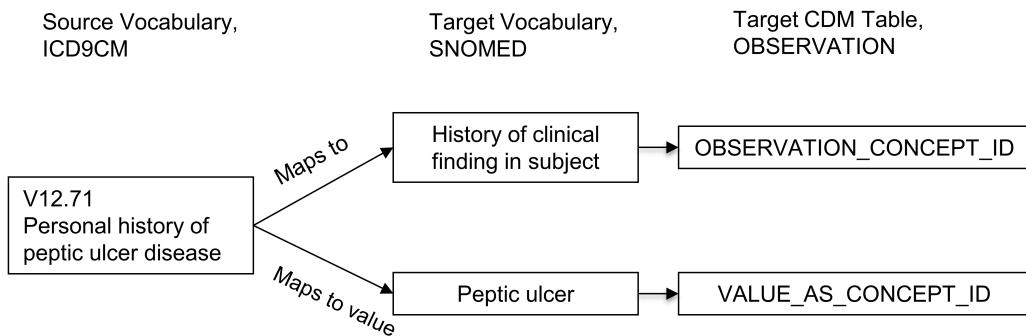


Figure 5.5: ソースコンセプトと標準コンセプト間の一対多のマッピング。事前に組み合わせられたコンセプトは 2 つのコンセプトに分割され、一つは属性（ここでは臨床所見の履歴）で、もう一つは値（消化性潰瘍）です。「Maps to」関係はメジャーメントまたはオブザベーションのドメインのコンセプトにマッピングされますが、「Maps to value」コンセプトにはドメインの制限はありません。

5.3.2 階層関係

階層関係は、「Is a」 - 「Subsumes」関係によって定義されます。階層関係は、子コンセプトが親コンセプトのすべての属性に加えて、1つ以上の追加属性またはより厳密に定義された属性を持つように定義されます。たとえば、SNOMED 49436004「心房細動」は、SNOMED 17366009「心房性不整脈」と「Is a」関係で関連しています。両コンセプトは、不整脈の種類（一方では細動と定義されているが、他方では定義されていない）を除いて、同一の属性セットを持っています。コンセプトは複数の親または複数の子コンセプトを持つことができます。この例では、SNOMED 49436004「心房細動」は SNOMED 40593004「細動」に対しても「Is a」に該当します。

5.3.3 異なるボキャブラリーのコンセプト間の関係

これらの関係は通常、「ボキャブラリ A - ボキャブラリ B は同等」というタイプであり、ボキャブラリのオリジナルソースから提供されるか、OHDSI ボキャブラリチームによって作成されます。それらは近似的なマッピングとして機能することが多いですが、より厳密に管理されたマッピング関係よりも制度が低い場合があります。高品質の同等関係（例えば、「ソース - RxNorm と同等」）は常に「Maps to」関係によって複製されます。

5.3.4 同一ボキャブラリーのコンセプト間の関係

内部ボキャブラリ間の関係は通常、ボキャブラリの提供者によって提示されます。OHDSI Wiki の個々のボキャブラリの個々のボキャブラリー文書に完全な説明が記載されています⁶。

⁶<https://www.ohdsi.org/web/wiki/doku.php?id=documentation:vocabulary>

これらの多くは、臨床イベント間の関係を定義しており、情報検索に使用することができます。例えば、尿道の障害は、「Finding site of (部位の検索)」関係に従うことで検索することができます（表 5.5 を参照）。

Table 5.5: 尿道の「Finding site of」関係で、すべてこの解剖学的構造に位置する状態を示しています。

CONCEPT_ID_1	CONCEPT_ID_2
4000504 “Urethra part”	36713433 “部分的尿道重複”
4000504 “Urethra part”	433583 “下部尿道裂孔”
4000504 “Urethra part”	443533 “男性下部尿道裂孔”
4000504 “Urethra part”	4005956 “女性下部尿道裂孔”

これらの関係の質と網羅性は、元のボキャブラリーの質によって異なります。一般に、SNOMED のような標準コンセプトを抽出するために使用されるボキャブラリーは、より優れた管理がされているという理由で選択されるため、内部関係もより質の高いものとなる傾向があります。

5.4 階層

ドメイン内では、標準および分類コンセプトは階層構造に整理され、CONCEPT_ANCESTOR テーブルに格納されます。これにより、コンセプトとその下位層に含まれるコンセプトをすべてクエリして取得することができます。これらの下位層は上位層と同じ属性を持ちますが、追加の属性や、より詳細に定義された属性も持ります。

CONCEPT_ANCESTOR テーブルは、階層関係を通じてつながっているすべてのコンセプトを網羅する CONCEPT_RELATIONSHIP テーブルから自動的に構築されます。これらは “Is a” - “Subsumes” のペア（図 5.6 参照）であり、ボキャブラリー間の階層を結びつけるその他の関係です。関係が階層構築に参加するかは、関係 ID ごとに RELATIONSHIP 参照テーブルの DEFINES_ANCESTRY フラグによって定義されます。

上位層の度合い、つまり上位層と下位層の間の階層数は、MIN_LEVELS_OF_SEPARATION および MAX_LEVELS_OF_SEPARATION フィールドに記録され、最短または最長の接続を定義します。すべての階層関係が分離レベルの計算に等しく寄与するわけではありません。この度合いにカウントされるステップは、各関係 ID に対して RELATIONSHIP 参照テーブルの IS_HIERARCHICAL フラグによって決まります。

現時点では、高品質で包括的な階層は薬剤とコンディションの 2 つのドメインにのみ存在します。プロシージャー、メジャーメント、およびオブザベーションのドメインは部分的にしかカバーされておらず、構築中です。祖先関係は、原産国、ブランド名、その他の属性に関係なく、指定された成分や薬効分類のすべての薬品を参照できるため、「薬」のドメインでは特に有用です。

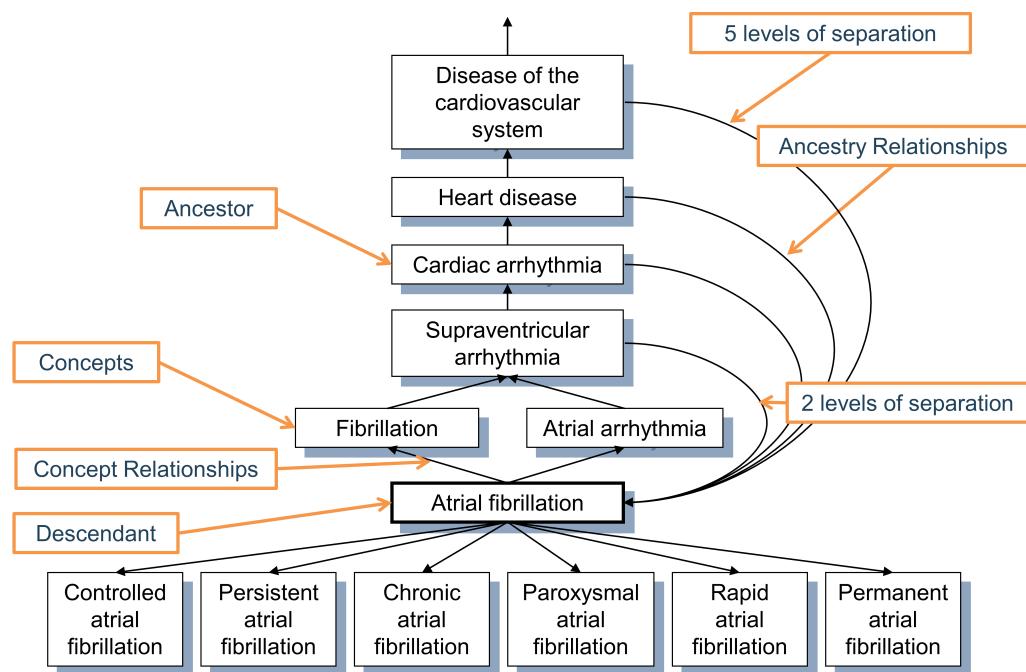


Figure 5.6: 「心房細動」という条件の階層。第 1 上位層関係は “Is a (~の一つです)” と ”Subsumes (包含)” 関係によって定義され、それより高次の関係はすべて推論され、CONCEPT_ANCESTOR テーブルに格納されます。各コンセプトは、それぞれ自身の下位層もあり、その間の分離階層数は 0 です。

5.5 内部参照テーブル

DOMAIN_ID、VOCABULARY_ID、CONCEPT_CLASS_ID（すべて CONCEPT レコード内）および CONCEPT_RELATIONSHIP_ID（CONCEPT_RELATIONSHIP 内）は、すべて独自の語彙によって制御されています。これらは、4 つの参照テーブル DOMAIN、VOCABULARY、CONCEPT_CLASS、RELATIONSHIP で定義されており、*_ID フィールドを主キーとして、より詳細な *_NAME フィールドと、CONCEPT テーブルへの参照を持つ *_CONCEPT_ID フィールドを含んでいます。CONCEPT テーブルには、参照テーブルのレコードそれぞれに対応するコンセプトが含まれています。これらの重複レコードの目的は、自動ナビゲーションエンジンを可能にする情報モデルをサポートすることです。また、VOCABULARY テーブルには、オリジナルのボキャブラリソースとバージョンを参照する VOCABULARY_REFERENCE と VOCABULARY_VERSION フィールドが含まれています。RELATIONSHIP テーブルには、追加のフィールドとして DEFINES_ANCESTRY、IS_HIERARCHICAL、REVERSE_RELATIONSHIP_ID があります。後者は、関係のペアのカウンター関係 ID を定義します。

5.6 特別な状況

5.6.1 性別

OMOP CDM と標準化ボキャブラリにおける性別は、出生時の生物学的性別を意味します。代替の性別をどのように定義するのかという質問がよく寄せられます。このようなケースは、OBSERVATION テーブルのレコードでカバーする必要があります。このテーブルには、本人が定義した性別が格納されます（データ資産にそのような情報が含まれている場合）。

5.6.2 人種と民族

これらは米国政府の定義に従います。民族はヒスパニック系または非ヒスパニック系の区別であり、人種は問いません。人種は一般的な上位 5 つの人種に分けられ、民族は階層的な下位層として含まれます。“Mixed races (混血)” は含まれていません。

5.6.3 診断コーディングシステムと OMOP 条件

ICD-9 や ICD-10 などの一般に使用されているコーディング体系は、適切な診断評価に基づいて、ある程度明確な診断を定義しています。コンディションドメインは、このセマンティックスペースと完全に一致するものではありませんが、部分的に重複しています。例えば、コンディションには診断が下される前に記録される徴候や症状も含まれます。また、ICD コードには他のドメイン（例えば、プロシージャー）に属するコンセプトも含まれます。

5.6.4 プロシージャーコードシステム

同様に、HCPCS や CPT4 のようなコーディングシステムは医療プロシージャーのリストであると考えられます。実際には、これらは医療サービスに対する支払い請求の根拠となるメニューのようなものです。これらのサービスの多くはプロシージャードメインに含まれますが、多くのコンセプトはこれに該当しません。

5.6.5 医療機器

医療機器のコンセプトには、標準コンセプトのソースとして使用できる標準化されたコーディングスキームがありません。多くのソースデータでは、医療機器はコード化されていないか、外部のコーディングスキームにも含まれていません。同じ理由により、現在利用可能な階層システムはありません。

5.6.6 ビジットとサービス

ビジットのコンセプトは、医療受診の性質を定義します。多くのソースシステムでは、これらはサービス提供場所として呼ばれており、病院などの組織や物理的構造を示します。他のソースシステムでは、サービスと呼ばれます。これらの用語の定義も国によって異なり、その定義入手するのは困難です。医療施設は、数少ない訪問の 1 つに特化していることが多い（XYZ 病院）ですが、それでもそれらによって定義されるべきではありません（XYZ 病院でも、患者は病院外のビジットをすることがある）。

5.6.7 医療従事者と専門分野

医療従事者は、医療従事者ドメインで定義されます。これには、医師や看護師などの医療専門家だけでなく、検眼医や靴職人などの医療以外の専門家も含まれます。専門分野は、医療従事者「医師」の下位層です。医療施設は専門分野を持つことはできませんが、主要スタッフの専門分野で定義されることによくあります（「外科」など）。

5.6.8 特別な要件を持つ治療領域

標準化ボキャブラリは包括的に医療のあらゆる側面をカバーしています。しかし、一部の治療領域では特別なニーズがあり、特別なボキャブラリが必要となります。例としては、腫瘍学、放射線医学、ゲノミクスが挙げられます。これらの拡張を開発するために、特別な OHDSI ワーキンググループが存在します。その結果、OMOP で標準化ボキャブラリは、異なる起源と目的を持つコンセプトがすべて同じドメイン固有の階層に存在する統合システムを構成しています。

5.6.9 薬剤ドメインにおける標準コンセプト

薬剤ドメインの多くのコンセプトは、米国国立医学図書館が作成した公的に利用可能なボキャブラリである RxNorm から引用されています。ただし、米国外の医薬品については、成分、形態、および強度の組み合わせが米国で市販されているかどうかに応じて、対象とならない場合があります。米国市場にない医薬品は、OHDSI ボキャブラリチームによって、唯一の大規模ドメインボキャブラリである RxNorm Extension というボキャブラリに追加されます。

5.6.10 NULL のバリエーション

多くのボキャブラリには、情報の欠如に関するコードが含まれています。例えば、5つの性別コンセプト 8507 「男性」、8532 「女性」、8570 「曖昧」、8551 「不明」、および 8521 「その他」のうち、標準コンセプトは最初の 2 つのみであり、他の 3 つはマッピングなしのソースコンセプトです。標準化ボキャブラリでは、なぜ情報が利用できないかの区別はなされません。それは患者による情報の積極的な撤回、欠落値、何らかの形で定義または標準化されていない値、または CONCEPT_RELATIONSHIP でのマッピング記録の欠如によるものである可能性があります。このようなコンセプトはマッピングされず、標準コンセプトのコンセプト ID=0 のデフォルトマッピングに対応します。

5.7 まとめ



- すべてのイベントと管理上の事実は、OMOP 標準化ボキャブラリでコンセプト、コンセプト関係、コンセプト祖先階層として表されます。
- これらのほとんどは既存のコーディングスキームやボキャブラリから採用されていますが、一部は OHDSI ボキャブラリチームによって新規にキュレーションされます。
- すべてのコンセプトにはドメインが割り当てられ、そのコンセプトが表す事象が CDM のどこに格納されるかが制御されます。
- 異なるボキャブラリにおける同等の意味を持つコンセプトは、そのうちの 1 つにマッピングされ、これが標準コンセプトとして指定されます。他のコンセプトはソースコンセプトです。
- マッピングは「Maps to」および「Maps to value」というコンセプト関係を通じて行われます。
- 分類コンセプトという追加のコンセプトクラスがあり、これらは非標準ですが、ソースコンセプトとは異なり、階層構造に参加します。
- コンセプトには時間の経過とともにライフサイクルがあります。
- ドメイン内のコンセプトは階層に整理されています。階層の質はドメインごとに異なり、階層システムの完成は継続的な作業です。
- 間違いや不正確さを発見した場合は、コミュニティに積極的に参加することを強くお勧めします。

5.8 演習

前提条件

最初の演習では、標準化ボキャブラリのコンセプトを検索する必要があります。これは ATHENA⁷ または ATLAS⁸を通じて行うことができます。

演習 5.1. “消化管出血” の標準コンセプト ID は何ですか？

演習 5.2. “消化管出血” の標準コンセプトに対応する ICD-10CM コードは何ですか？この標準コンセプトに対応する ICD-9CM コードはどれですか？

演習 5.3. “消化管出血” の標準コンセプトに相当する MedDRA の優先用語は何ですか？

回答例は付録 E.2 を参照のこと。

⁷<http://athena.ohdsi.org/>

⁸<http://atlas-demo.ohdsi.org>

第 6 章

ETL（抽出-変換-読込）

著者: Clair Blacketer & Erica Voss

6.1 はじめに

ネイティブ/生データから OMOP 共通データモデル (CDM) を作成するには、ETL (抽出-変換-読込) プロセスを作成する必要があります。このプロセスでは、データを CDM に再構築し、標準化ボキャブラリにマッピングを追加する必要があります。通常、このプロセスは、例えば SQL スクリプトのような自動化されたスクリプトのセットとして実装されます。この ETL プロセスは繰り返し実行できることが重要です。これにより、ソースデータが更新されるたびに再実行することができます。

ETL の作成は通常、大規模な取り組みとなります。長年にわたり、私たちは以下の 4 つの主要なステップからなるベストプラクティスを開発してきました：

1. データの専門家と CDM の専門家が共同で ETL をデザインする。
2. 医学的知識を持つ人がコードのマッピングをする。
3. 技術者が ETL を実装する。
4. 全員が品質管理に関与する。

本章では、これらのステップをそれぞれ詳しく説明します。OHDSI コミュニティでは、これらのステップの一部をサポートするツールがいくつか開発されており、それらについても説明します。本章の最後に、CDM と ETL のメンテナンスについて説明します。

6.2 ステップ 1: ETL のデザイン

ETL のデザインと実装を明確に区別することが重要です。ETL のデザインにはソースデータと CDM の両方に関する広範な知識が必要です。一方、ETL の実

装は通常、ETL を計算効率的に行うための技術的専門知識に大きく依存します。両方を同時に行おうとすると、細部にこだわってしまい、全体像に集中できなくなることがあります。

ETL デザインプロセスを支援するために密接に統合された 2 つのツールが開発されました：White Rabbit と Rabbit-in-a-Hat です。

6.2.1 White Rabbit

データベースで ETL プロセスを開始するには、データ（テーブル、フィールド、内容など）を理解する必要があります。そこで登場するのがWhite Rabbitツールです。White Rabbit は、縦断的な医療データベースの ETL をOMOP CDM用に準備するためのソフトウェアツールです。White Rabbit はデータをスキャンし、ETL のデザインを開始するために必要なすべての情報を含むレポートを作成します。全てのソースコードとインストール手順、マニュアルへのリンクは GitHub で入手可能です¹。

範囲と目的

White Rabbit の主な機能は、ソースデータをスキャンし、テーブル、フィールド、フィールドに表示される値に関する詳細な情報を提供することです。ソースデータは、カンマ区切りのテキストファイルやデータベース (MySQL, SQL Server, Oracle, PostgreSQL, Microsoft APS, Microsoft Access, Amazon Redshift) に保存される必要があります。スキャンによって、例えば Rabbit-In-a-Hat ツールと併用して、ETL デザイン時に参照用として使用できるレポートが作成されます。White Rabbit は標準的なデータプロファイリングツールとは異なり、生成された出力データファイルに個人識別情報 (PII) を表示しないようにします。

プロセス概要

ソフトウェアを使用してソースデータをスキャンする一般的な手順は以下の通りです：

1. 作業フォルダを設定します。作業フォルダは、結果がエクスポートされるローカルのデスクトップコンピュータ上の場所です。
2. ソースデータベースまたは CSV テキストファイルに接続し、接続をテストします。
3. スキャン対象のテーブルを選択し、テーブルをスキャンします。
4. White Rabbit がソースデータに関する情報をエクスポートします。

作業フォルダの設定

White Rabbit アプリケーションをダウンロードしてインストールした後、まず最初に作業フォルダを設定する必要があります。White Rabbit が作成するすべ

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

てのファイルはこのローカルフォルダにエクスポートされます。図6.1に示されている「Pick Folder (フォルダの選択)」ボタンを使用して、スキャン文書を保存するローカル環境をナビゲートします。

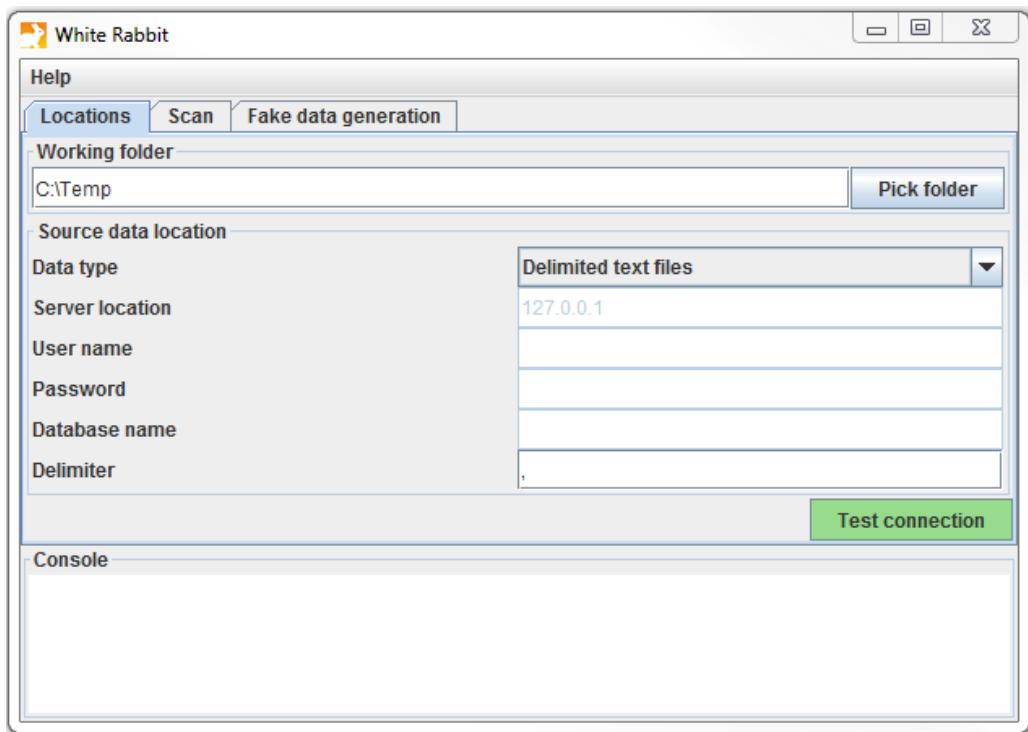


Figure 6.1: White Rabbit アプリケーションの作業フォルダを指定するための「Pick Folder」ボタン

データベースへの接続

White Rabbit は区切りテキストファイルとさまざまなデータベースプラットフォームをサポートしています。各フィールドにマウスカーソルを合わせると、必要な情報が表示されます。詳細についてはマニュアルをご覧ください。

データベース内のテーブルをスキャン

データベースに接続後、含まれるテーブルをスキャンできます。スキャンにより ETL のデザインに役立つ情報を含むレポートが生成されます。図 6.2 に示されたスキャンタブを使用して、「Add」(Ctrl + マウスクリック) をクリックして選択したソースデータベース内の個々のテーブルを選択するか、データベース内のすべてのテーブルを自動的に選択する「Add all in DB」ボタンをクリックできます。

スキャンにはいくつかの設定オプションもあります：

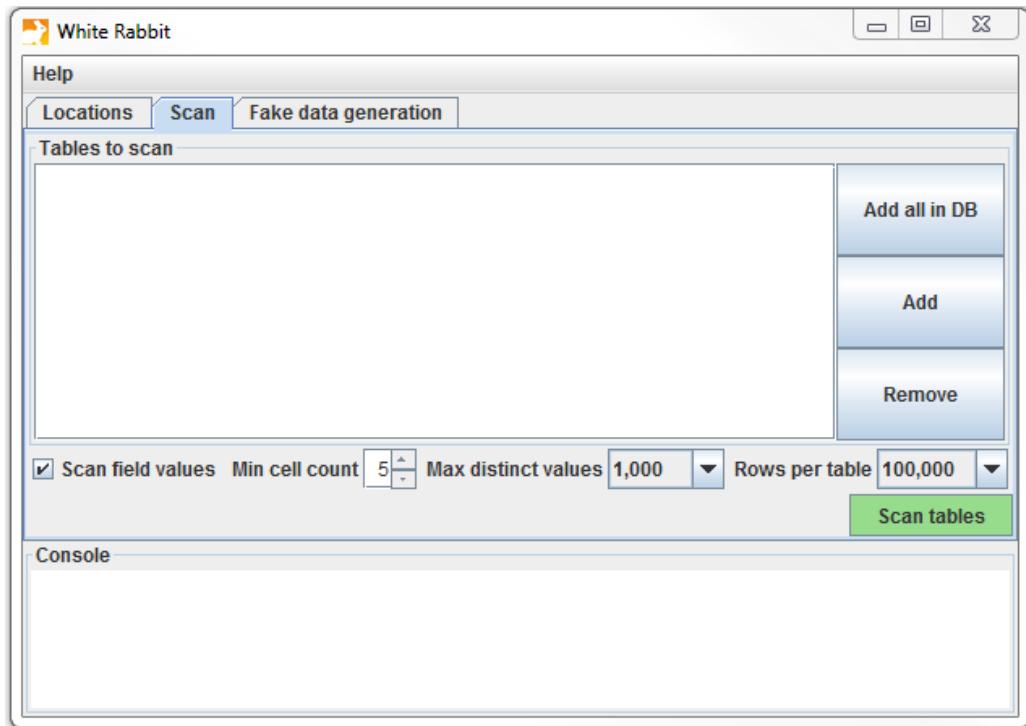


Figure 6.2: White Rabbit スキャンタブ

- 「フィールド値をスキャン」をチェックすると、列に表示される値を調査したいことを WhiteRabbit に通知します。
- 「最小セル数」はフィールド値のスキャン時のオプションです。デフォルトでは 5 に設定されており、ソースデータで 5 回未満しか表示されない値は報告に表示されません。個別のデータセットには、この最小セル数に関する独自のルールがある場合があります。
- 「テーブルあたりの行数」はフィールド値のスキャン時のオプションです。デフォルトでは、White Rabbit はテーブル内の 100,000 行をランダムに選択してスキャンします。

すべての設定が完了したら、「テーブルをスキャン」ボタンをクリックします。スキャンが完了すると、レポートが作業フォルダに書き込まれます。

スキャンレポートの解釈

スキャンが完了すると、選択したフォルダにスキャンしたテーブルごとにタブが設けられた Excel ファイルが作成されます。また、概要タブも用意されています。概要タブには、スキャンしたすべてのテーブル、各テーブルの各フィールド、各フィールドのデータタイプ、フィールドの最大長、テーブルの行数、スキャンした行数、各フィールドが空欄であることが判明した頻度が記載されています。図 6.3 は、概要タブの例を示しています。

The screenshot shows a Microsoft Excel-like interface for a 'Scan Report' sample summary tab. The grid has columns labeled A through G. Rows 1 through 8 show data for the 'dbo.allergies' table, and rows 9 through 18 show data for the 'dbo.careplans' table. The columns represent fields like 'Table', 'Field', 'Type', 'Max length', 'N rows', 'N rows checked', and 'Fraction empty'. The 'Fraction empty' column for the 'dbo.allergies' table shows values like 0.725188442 and 0.057849598.

	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: スキャンレポートのサンプル概要タブ

各テーブルのタブには、各フィールド、各フィールド内の値、各値の頻度が示されます。各ソーステーブルの列は、Excel に 2 つの列を生成します。1 つの列には、「最小セルカウント」がスキャン時に設定された値よりも大きいすべての異なる値がリストされます。一意の値のリストが切り捨てられた場合、リストの最後の値は「リストが切り捨てされました」となります。これは、「最小セルカウント」に入力された数値よりも少ない数の追加の一意のソース値が 1 つ以上存在することを示します。各一意の値の隣には、2 番目の列として頻度（サンプルに含まれるその値の回数）が表示されます。この 2 つの列（一意の値と頻度）は、ワークブックでプロファイルされたテーブル内のすべてのソース列に対して繰り返されます。

The screenshot shows a table with two columns, A and B. Column A contains the values 'Sex', '2', '1', and 'List truncated...'. Column B contains the frequencies '61491' and '35401'. Row 4 is highlighted with a red background.

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

Figure 6.4: 単一列のサンプル値

レポートはソースデータを理解するのに強力であり、存在するものを強調して表示します。例えば、図6.4に示された結果がスキャンされたテーブルの「Sex」列に戻された場合、2 つの共通値（1 と 2）がそれぞれ 61,491 回と 35,401 回出現したことが分かります。White Rabbit は 1 を男性、2 を女性として定義することではなく、データホルダーが通常、ソースシステムに固有のソースコードを定義する必要があります。しかし、このリストが切り捨てられていることから、データにはこの 2 つの値（1 と 2）だけが存在しているわけではないこと

が分かります。これらの他の値は、「最小セル数」で定義されるように、非常に低い頻度でしか現れず、不正確な値や非常に疑わしい値であることがよくあります。ETL を生成する際には、高頻度の性別コンセプト 1 と 2 だけでなく、この列に存在するその他の頻度の低い値も処理できるように計画する必要があります。例えば、低頻度の性別が「NULL」であった場合、ETL がそのデータを処理でき、その状況で何をすべきかを知っていることを確認する必要があります。

6.2.2 Rabbit-In-a-Hat

White Rabbit スキャンを手にすると、ソースデータの全体像や CDM の仕様が掴めます。次に、これら 2 つの間の論理を定義する必要があります。この設計作業には、ソースデータと CDM の両方にに関する深い知識が必要です。White Rabbit ソフトウェアと共に提供される Rabbit-in-a-Hat ツールは、これらの分野の専門家チームをサポートするように特別にデザインされています。典型的な設定では、ETL 設計チームが一堂に会し、Rabbit-in-a-Hat をスクリーンに映し出します。最初のラウンドでは、テーブル間のマッピングを共同で決定し、その後、フィールド間のマッピングを設計し、値が変換されるロジックを定義します。

範囲と目的

Rabbit-In-a-Hat は White Rabbit のスキャン文書を読み取り、表示するように設計されています。White Rabbit はソースデータに関する情報を生成し、Rabbit-In-a-Hat はその情報を使用して、グラフィカルユーザーインターフェイスを通じて、ユーザーがソースデータを CDM のテーブルとカラムに接続できるようにします。Rabbit-In-a-Hat は ETL プロセスのドキュメントを生成しますが、ETL を作成するコードは生成しません。

プロセス概要

このソフトウェアを使用して ETL のドキュメントを生成する一般的な手順は以下の通りです：

1. White Rabbit のスキャン結果を完了させます。
2. スキャン結果を開くと、インターフェースにソーステーブルと CDM テーブルが表示されます。
3. ソーステーブルが対応する CDM テーブルに情報を提供する場合は、ソーステーブルを CDM テーブルに接続します。
4. 各ソーステーブルから CDM テーブルへの接続について、ソース列と CDM 列の詳細でさらに定義します。
5. Rabbit-In-a-Hat の作業内容を保存し、MS Word 文書にエクスポートします。

ETL ロジックの記述

White Rabbit スキャンレポートを Rabbit-In-a-Hat で開くと、ソースデータを OMOP CDM に変換する方法のロジックの設計と記述する準備が整ったことになります。次のセクションでは、Synthea²データベースのいくつかのテーブルが変換中にどのように見えるかを例示します。

ETL の一般的なフロー

CDM は人を中心としたモデルであるため、PERSON テーブルのマッピングを最初に始めることが常に良い方法です。すべての臨床イベントテーブル (CONDITION_OCCURRENCE、DRUG_EXPOSURE、PROCEDURE_OCCURRENCE など) は、person_id を介して PERSON テーブルを参照するため、最初に PERSON テーブルのロジックを構築しておくと、後で作業が容易になります。PERSON テーブルの次に、OBSERVATION_PERIOD テーブルを変換するのが良い方法です。CDM データベースの各人には少なくとも 1 つの OBSERVATION_PERIOD があり、通常、ほとんどのイベントはこの期間内に発生します。PERSON テーブルと OBSERVATION_PERIOD テーブルが完了したら、次に PROVIDER、CARE_SITE、LOCATION などのディメンションナルテーブルが通常は続きます。臨床テーブルの前に作成すべき最後のテーブルロジックは VISIT_OCCURRENCE です。これは ETL 全体で最も複雑なロジックであることが多くまた、患者の旅の過程で発生するほとんどのイベントはビジット時に発生するため、最も重要なものの 1 つです。これらのテーブルが完了したら、どの CDM テーブルをどのような順序でマッピングするかは実施者の選択にゆだねられます。

CDM 変換中に中間テーブルの作成が必要になることがあります。これは、イベントに正しい VISIT_OCCURRENCE_ID を割り当てるため、またはソースコードを標準コンセプトにマッピングするためです（このステップをその場で実行すると非常に時間がかかることがあります）。中間テーブルは 100% 許可され推奨されています。ただし、変換が完了した後にこれらの中間テーブルを永続的に使用し続け、それらに依存することは推奨されません。

マッピング例：Person テーブル

Synthea データ構造には patients テーブルに 20 のカラムがありますが、図6.6 に示されているように、すべてが PERSON テーブルを埋めるために必要というわけではありません。これは非常に一般的であり、心配する必要はありません。この例では、CDM PERSON テーブルで使用されていない Synthea patients テーブルのデータポイントの多くは患者名、運転免許証番号、パスポート番号などの追加識別子です。

表 6.1 には、Synthea patients テーブルを CDM PERSON テーブルに変換するために適用されたロジックを示しています。『Destination Field』(変換先フィー

²Synthea™ は実際の患者をモデル化することを目的とした患者ジェネレーターです。データはアプリケーションに渡されたパラメータに基づいて作成されます。データの構造については、こちら：<https://github.com/synthetichealth/synthea/wiki> をご覧ください。

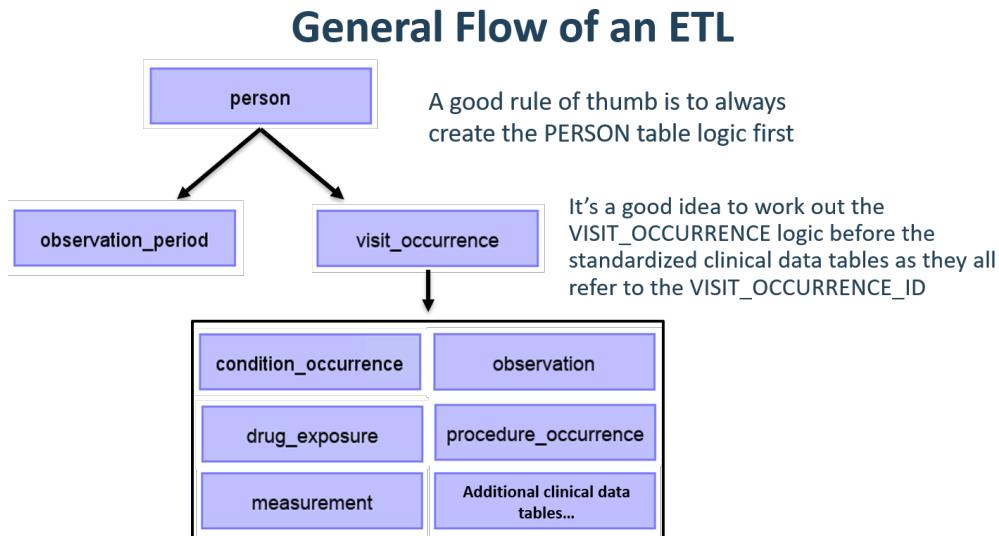


Figure 6.5: ETL の一般的なフローと、最初にマッピングするテーブル

ルド) は、CDM のどこにデータがマッピングされるかを示しています。『Source field』(変換元フィールド) では、CDM カラムにデータを入力するのに使用されるソーステーブル (この場合は patients) のカラムを強調して表示しています。最後に、『Logic & comments』(ロジックとコメント) カラムには、ロジックの説明が記載されています。

Table 6.1: Synthea Patients テーブルを CDM PERSON テーブルに変換するための ETL ロジック

目的フィールド	ソースフィールド	ロジックとコメント
PERSON_ID		自動生成。PERSON_ID は実装時に生成されます。これは、ソースの id 値が varchar 値であるのに対し、PERSON_ID は整数であるためです。ソースからの id フィールドは、その値を保持し、必要に応じてエラーチェックを行うために PERSON_SOURCE_VALUE として設定されます。

目的フィールド	ソースフィールド	ロジックとコメント
GENDER_CONCEPT_ID	gender	性別が「M」の場合、GENDER_CONCEPT_ID は 8507、性別が「F」の場合は 8532 に設定します。性別不明の行は削除します。これらの 2 つのコンセプトは、性別ドメインの唯一の標準コンセプトであるため選択されました。性別不明の患者を削除するかどうかの決定は、通常、施設で行われる傾向がありますが、性別不明の人は分析から除外されるため、削除することが推奨されます。
YEAR_OF_BIRTH	birthdate	生年月日から年を取得します。
MONTH_OF_BIRTH	birthdate	生年月日から月を取得します。
DAY_OF_BIRTH	birthdate	生年月日から日を取得します。
BIRTH_DATETIME	birthdate	0 時を 00:00:00 とします。ここでは、ソースが出生時間を指定していないため、深夜を出生時間として設定しました。
RACE_CONCEPT_ID	race	race = 'WHITE' の場合は 8527、race = 'BLACK' の場合は 8516、race = 'ASIAN' の場合は 8515、それ以外の場合は 0 として設定します。これらのコンセプトが選択されたのは、人種ドメインに属する標準コンセプトであり、ソース内の民族カテゴリーに最も近いためです。

目的フィールド	ソースフィールド	ロジックとコメント
ETHNICITY_CONCEPT_ID	race ethnicity	race = ‘HISPANIC’、または民族が(‘CENTRAL_AMERICAN’、‘DOMINICAN’、‘MEXICAN’、‘PUERTO_RICAN’、‘SOUTH_AMERICAN’)の場合、38003563と設定し、それ以外の場合は0に設定します。これは、複数のソース列が1つのCDM列にどのように影響するかを示す良い例です。CDMでは、民族はヒスパニックまたは非ヒスパニックとして表されるため、ソース列raceとソース列ethnicityの両方の値がこの値を決定します。
LOCATION_ID		
PROVIDER_ID		
CARE_SITE_ID		
PERSON_SOURCE_VALUE	id	
GENDER_SOURCE_VALUE	gender	
GENDER_SOURCE_CONCEPT_ID		
RACE_SOURCE_VALUE	race	
RACE_SOURCE_CONCEPT_ID		
ETHNICITY_SOURCE_VALUE	ethnicity	この場合、ETHNICITY_SOURCE_VALUEはETHNICITY_CONCEPT_IDよりも詳細な値となります。
ETHNICITY_SOURCE_CONCEPT_ID		

Synthea データセットが CDM にどのようにマッピングされたかについては、

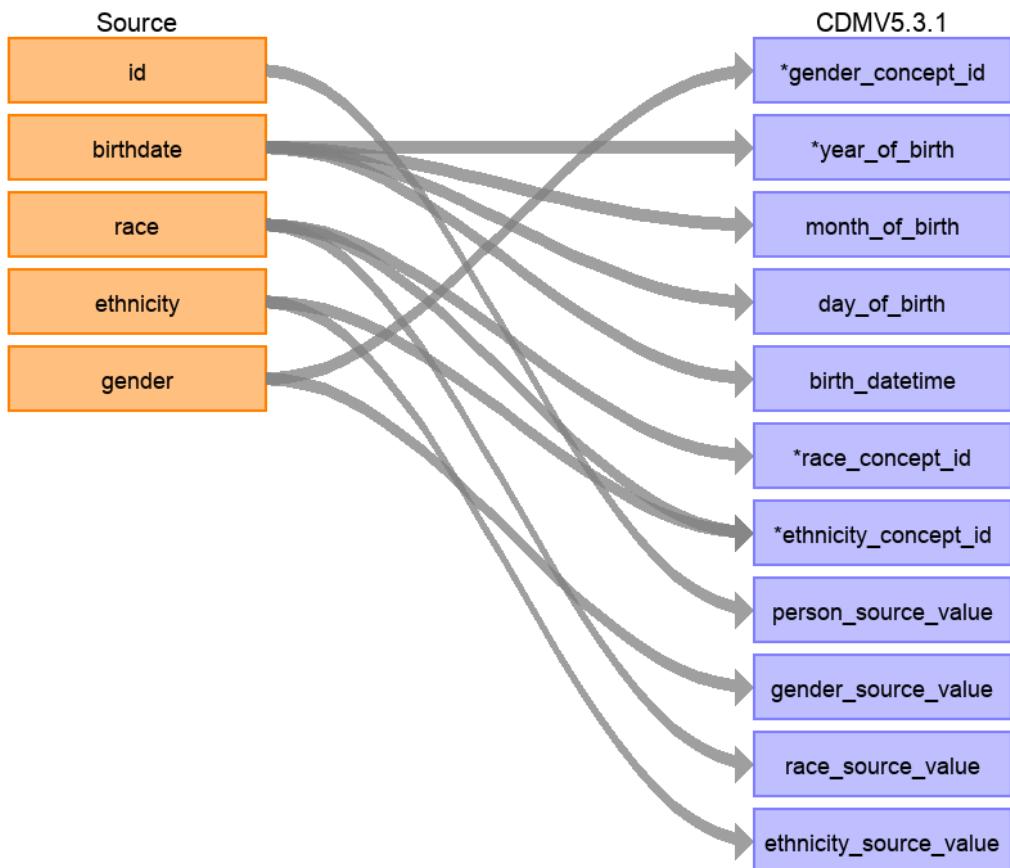


Figure 6.6: Synthea Patients テーブルから CDM PERSON テーブルへのマッピング

仕様書全文をご覧ください³。

6.3 ステップ 2: コードマッピングの作成

OMOP ボキャブラリには、常にソースコードが追加されています。これは、CDM にデータを変換する際のコーディングシステムが既に含まれており、マッピングされている可能性があることを意味します。含まれているボキャブラリは、OMOP ボキャブラリの VOCABULARY テーブルを確認ください。非標準のソースコード（例：ICD-10CM コード）から標準コンセプト（例：SNOMED コード）へのマッピングを抽出するには、relationship_id = 「Maps to」を持つ CONCEPT_RELATIONSHIP テーブルのレコードを使用できます。例えば、ICD-10CM コード「I21」（「急性心筋梗塞」）の標準コンセプト ID を特定するには、次の SQL を使用します：

```
SELECT concept_id_2 AS standard_concept_id
FROM concept_relationship
INNER JOIN concept AS source_concept
  ON concept_id = concept_id_1
WHERE concept_code = 'I21'
  AND vocabulary_id = 'ICD10CM'
  AND relationship_id = 'Maps to';
```

STANDARD_CONCEPT_ID
312327

残念ながら、ソースデータがボキャブラリに含まれていないコーディングシステムを使用している場合もあります。この場合、ソースコーディングシステムから標準コンセプトへのマッピングを作成する必要があります。コードマッピングは特にソースコーディングシステムに多くのコードが含まれている場合に、困難な作業となる可能性があります。作業を用意するために、以下の方法があります：

- 最も頻繁に使用されるコードに焦点を当てる。使用されることのないコードや使用頻度の低いコードは、実際の研究では使用されることがないため、マッピングする価値はありません。
- 可能な限り既存の情報を活用しましょう。例えば、多くの国の医薬品コードは ATC にマッピングされています。ATC は多くの目的に対して詳細さに欠けますが、ATC と RxNorm のコンセプトの関係性を利用することで適切な RxNorm コードを推測することができます。
- Usagi を使用しましょう。

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

6.3.1 Usagi

Usagi はコードマッピングを手動で作成するプロセスを支援するツールです。コードの記述のテキスト類似性に基づいて、マッピングの候補を作成することができます。ソースコードが外国語でしか利用できない場合、Google 翻訳⁴は驚くほど正確な翻訳結果を提示することが多いことがわかっています。Usagi は自動提案が適切でない場合に、適切なターゲットコンセプトを検索する機能を提供します。最終的に、ユーザーは ETL で使用することが承認されたマッピングを指定することができます。Usagi は GitHub で入手できます⁵。

範囲と目的

マッピングが必要なソースコードは Usagi に読み込みます（コードが英語でない場合は追加の翻訳列が必要です）。用語の類似性アプローチを使用してソースコードをボキャブラリコンセプトに紐づけします。ただし、これらのコードの紐づけは手動で確認する必要があります、Usagi はこれを容易にするためのインターフェースを提供します。Usagi はボキャブラリで標準コンセプトとしてマークされているコンセプトのみを提案します。

プロセス概要

このソフトウェアを使用する一般的な手順は次のとおりです：

1. マッピングしたいソースシステムからソースコードを読み込みます。
2. Usagi は用語の類似性アプローチを実行してソースコードをボキャブラリコンセプトにマッピングします。
3. Usagi のインターフェースを利用して、自動提案の正しさを確認し、必要に応じて改善します。コードシステムと医療用語に精通した担当者によるレビューが推奨されます。
4. ボキャブラリの SOURCE_TO_CONCEPT_MAP にマッピングをエクスポートします。

ソースコードを Usagi にインポート

ソースコードを CSV または Excel (.xlsx) ファイルにエクスポートします。これには、ソースコードと英語のソースコードの説明を含む列が必要ですが、コードに追加する情報（例：用量単位、翻訳されている場合は元の言語での説明）もインポートできます。さらに、コードの使用頻度も引き継ぐことが望ましく、これはマッピングに最も労力を割くべきコードの優先順位付けに役立つためです（（例：1000 件のソースコードがあっても、システム内で実際に使用されているのは 100 件だけかもしれません）。ソースコードを英語に翻訳する必要がある場合は、Google 翻訳を使用ください。

⁴<https://translate.google.com/>

⁵<https://github.com/OHDSI/Usagi>

注意事項: ソースコードの抽出はドメインごとに分けて行い、1つの大きなファイルにまとめないでください（例：薬剤、プロシージャー（処置）、コンディション、オブザベーション（観察））。

ソースコードは File → Import codes メニューから Usagi に読み込まれます。ここでは「Import codes …」が表示されます（図6.7）。この図では、ソースコードの用語はオランダ語で、英語にも翻訳されています。Usagi は英語の翻訳を利用して標準ボキャブラリにマッピングします。

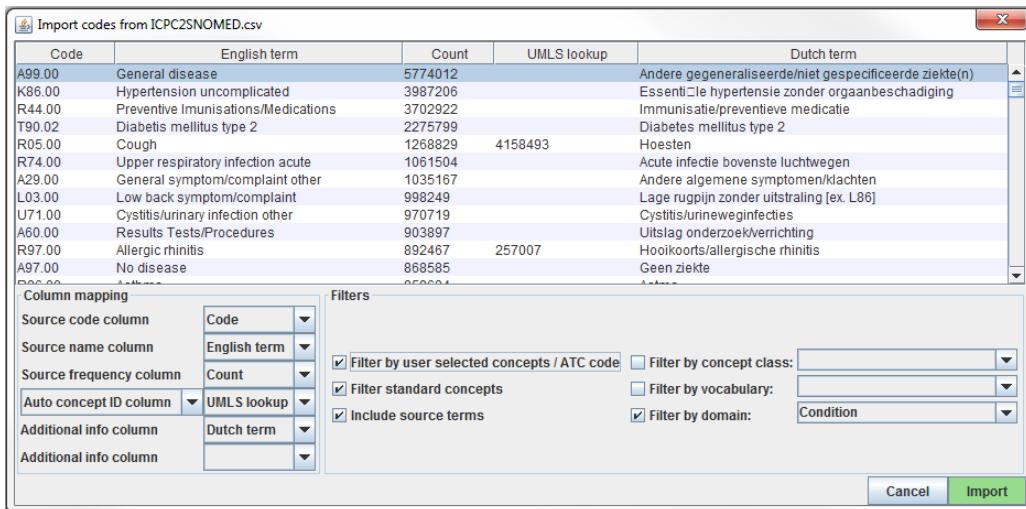


Figure 6.7: Usagi コード入力画面

「Column mapping」セクション（左下）では、インポートしたテーブルを Usagi に対してどのように使用するかを定義します。ドロップダウンメニューにマウスを合わせると、それぞれの列の定義が表示されるポップアップが表示されます。Usagi は「Additional info」列をソースコードとボキャブラリコンセプトコードを関連付けるための情報として使用しませんが、この追加情報はソースコードマッピングを確認する際に役立つ場合があるため、含めるべきでしょう。

最後に、「Filters」セクション（右下）では、Usagi がマッピングする際の制限をいくつか設定できます。例えば、図6.7では、ユーザーはソースコードをコンディションドメインのコンセプトのみにマッピングしています。デフォルトでは、Usagi は標準コンセプトのみにマッピングしますが、「Filter standard concepts」オプションをオフにすると、分類コンセプトも考慮されます。各フィルターについての追加情報は、異なるフィルター上にマウスカーソルを移動させます。

特別なフィルターとして「Filter by automatically selected concepts / ATC code」があります。検索を制限する情報がある場合、CONCEPT_IDS のリストまたは ATC コードを Auto concept ID 列で指定された列に記述することで、検索を制限することができます（セミコロンで区切ります）。例えば、医薬品の場合、各医薬品にすでに ATC コードが割り当てられている場合があります。ATC

コードは RxNorm の医薬品コードを一意に識別するものではありませんが、ボキャブラリの ATC コードに該当するコンセプトのみに検索対象を限定するのに役立ちます。ATC コードを使用するには、以下の手順に従います。：

1. Column mapping セクションで、“Auto concept ID column” から “ATC column” に切り替えます。
2. Column mapping セクションで、ATC コードを含む列を “ATC column” として選択します。
3. フィルターセクションで「ユーザーが選択したコンセプト/ATC コードによるフィルター」をオンにします。

ATC コード以外の情報源を使用して制限することもできます。上図の例では、UMLS から派生した部分的なマッピングを使用して Usagi の検索を制限しています。この場合、“Auto concept ID column” を使用する必要があります。

すべての設定が完了したら、“Import” ボタンをクリックしてファイルをインポートします。ファイルのインポートには、ソースコードをマッピングするためにボキャブラリの類似性アルゴリズムが実行されるため、数分かかります。

ソースコードからボキャブラリへのコンセプトマップの確認

ソースコードの入力ファイルをインポートすると、マッピング処理が始まります。図 6.8 では、Usagi の画面は、コンセプトテーブル、選択されているマッピング・セクション、検索を実行する場所の 3 つの主要な部分で構成されています。いずれのテーブルでも、右クリックで表示/非表示の列を選択でき、視覚的な複雑さを軽減できます。

The screenshot shows the Usagi application window with three distinct sections:

- Overview table:** A table showing the mapping between source codes and concepts. It includes columns for Source code, Source term, Frequency, Concept ID, Concept name, Domain, Concept class, Vocabulary, Concept code, Standard concept, Parents, and Children. A red box highlights the "Overview table" section.
- Selected mapping:** A table showing the selected mappings. It includes columns for Source code, Source term, Frequency, Concept ID, Concept name, Domain, Concept class, Vocabulary, Concept code, Standard concept, Parents, and Children. A red box highlights the "Selected mapping" section.
- Search facility:** A search interface with fields for "Source term" (set to "Cough"), "Concept ID" (set to "272039006"), and "Vocabulary" (set to "S"). It also includes checkboxes for "Use source term as query" and "Query:" (with a text input field). Below these are "Filters" for "Concept ID", "Concept name", "Domain", "Concept class", "Vocabulary", "Concept code", "Standard concept", "Parents", and "Children". A red box highlights the "Search facility" section.

At the bottom of the interface, there are buttons for "Comment:", "Approved", and "Vocabulary version: v5.0 23-APR-19".

Figure 6.8: Usagi でのソースコード入力画面

提案されたマッピングの承認

「コンセプトテーブル」には、ソース・コードとコンセプトの現在のマッピングが表示されます。ソース・コードをインポートした直後は、このマッピングには、ボキャブラリの類似性と任意の検索オプションに基づいて自動的に生成されたマッピング候補が含まれます。図 6.8 の例では、ユーザーがドメインをコンディションに限定しているため、オランダ語のコンディションコードの英語名がコンディションドメインの標準コンセプトにマッピングされています。Usagi は、ソースコードの説明とコンセプト名および同義語を比較して、最適な一致を見つけます。ユーザは “Include source terms”（ソース用語を含める）を選択していたため、Usagi は、特定のコンセプトにマッピングされるボキャブラリ内のすべてのソースコンセプトの名前と同義語も考慮しました。Usagi がマッピングできない場合は、CONCEPT_ID = 0 にマッピングされます。

ソース・コードを関連する標準ボキャブラリにマッピングする際には、コーディング・システムの経験がある人が支援することをお勧めします。その担当者は、“Overview Table”（概要テーブル）のコードごとに作業して、Usagi が提案したマッピングを受け入れるか、新しいマッピングを選択します。例えば、図 @ref:fig:usagiOverview では、オランダ語の “Hoesten” は英語の “Cough”（咳嗽）に翻訳されています。Usagi は “Cough” を使って、“4158493-C/O - cough” というボキャブラリコンセプトにマッピングしました。このマッチしたペアのマッチングスコアは 0.58 でした（マッチングスコアは通常 0~1 で、1 は確実に一致することを意味します）。一致スコア 0.58 は、Usagi がオランダ語のコードを SNOMED にどの程度うまくマッピングできたかについて、あまり確信がないことを意味します。このマッピングで問題ないと思われる場合は、画面右下の緑色の”Approve (承認)” ボタンを押すことで、このマッピングを承認することができます。

新しいマッピングの検索

Usagi がマップを提案した場合、ユーザはより良いマッピングを見つけるか、マップをコンセプトなし (CONCEPT_ID = 0) に設定する必要があります。図 6.8 の例では、オランダ語の「Hoesten」を「Cough」と訳しています。Usagi の提案は、UMLS から自動的に導出されたマッピングで識別されたコンセプトによって制限されており、結果は最適ではないかもしれません。検索機能では、実際のボキャブラリ自体または検索ボックスのクエリを使用して、他のコンセプトを検索することができます。

手動の検索ボックスを使用する場合、Usagi はあいまい検索を使用し、AND や OR のような論理演算子はサポートしていないことに留意する必要があります。

例を続けると、より適切なマッピングを見つけるために「Cough」という検索語を使用したとします。検索機能の「クエリ」セクションの右側には「フィルタ」セクションがあり、検索語を検索する際にボキャブラリから結果を絞り込むオプションがあります。この場合、標準コンセプトのみを検索したいことが分かっているため、標準コンセプトにマッピングされているボキャブラリ内の

ソースコンセプトの名前と同義語に基づいてコンセプトを検索することも可能です。

これらの検索条件を適用すると、「254761-Cough」が見つかり、これがオランダ語のコードにマッピングする適切なボキャブラリコンセプトであることが分かります。そのため、「Selected Source Code (選択されたソースコード)」セクションの更新後に表示される「Replace concept (コンセプトを置換する)」ボタンを押し、「Approve (承認)」ボタンを押します。また、「Add concept (コンセプトの追加)」ボタンもあり、複数の標準化ボキャブラリのコンセプトを1つのソースコードにマッピングすることができます。(例えば、ソースコードによっては、複数の疾患をひとまとめにしている場合がありますが、標準化ボキャブラリではひとまとめにしていない場合があります)。

コンセプト情報

マッピングする適切なコンセプトを探す場合、コンセプトの「社会性」を考慮することが重要です。コンセプトの意味は、そのコンセプトの階層における位置に部分的に依存することがあります。また、ボキャブラリの中には、階層関係がほとんどない、あるいは全くない「孤立したコンセプト」が存在することがあり、それらはターゲットにするコンセプトとしては適していません。また、Usagi は概念の親子関係の数を頻繁に報告しますが、ALT + C キーを押すか、上部メニューバーの view (閲覧) -> Concept information (コンセプト情報) を選択することで、より多くの情報を表示することもできます。

図 6.9 は、コンセプト情報パネルを示しています。このパネルには、コンセプトに関する一般的な情報、親、子、そのコンセプトにマッピングされる他のソースコードが表示されます。このパネルを使用して階層を移動し、別のターゲット・コンセプトを選択することができます。

すべてのコードがチェックされるまで、コードごとにこのプロセスを続行します。画面上部のソースコードのリストで、列見出しを選択すると、コードを並べ替えることができます。通常、頻度の高いコードから低いコードに並べ替えることをお勧めします。画面左下には、マッピングが承認されたコードの数と、そのコードの出現回数が表示されます。

マッピングにコメントを追加することもでき、マッピングがどのように決定されたのかを記録するのに役立ちます。

ベスト・プラクティス

- コーディングスキームの経験がある人に参加してもらってください。
- 列名をクリックすると、「コンセプトテーブル」の列を並べ替えることができます。“Match Score (マッチングスコア)” で並べ替えると、Usagi が最も信頼するコードを最初に確認でき、かなりの数のコードを迅速に除外できる可能性があります。また、“Frequency (頻度)” での並べ替えも重要です。頻度に利用されるコードとそうでないコードに重点的に取り組むことが重要です。

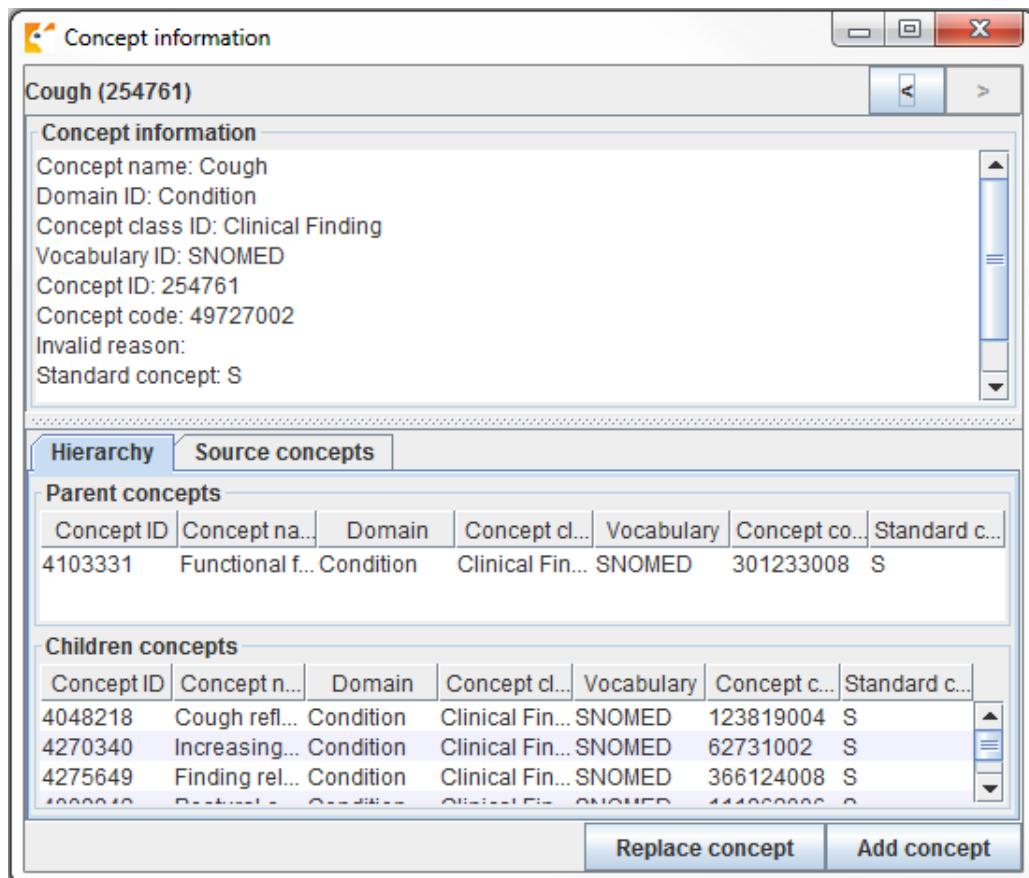


Figure 6.9: Usagi コンセプト情報パネル

- ・場合によっては、CONCEPT_ID = 0 にマッピングしても問題ありませんが、一部のコードは適切なマッピングが見つからず、また、一部のコードは適切なマッピングがないだけかもしれません。
- ・コンセプトの文脈、特にその親と子を考慮することが重要です。

Usagi で作成されたマッピングのエクスポート

USAGI 内でマッピングしたら、それをエクスポートしてボキャブラリ SOURCE_TO_CONCEPT_MAP テーブルに追加するのが、次に進むための最良の方法です。

マッピングをエクスポートするには、File (ファイル) -> Export source_to_concept_map (ソースからコンセプトへのマッピングをエクスポート) を選択します。どの SOURCE_VOCABULARY_ID を使用するかを尋ねるポップアップが表示されるので、短い ID (識別子) を入力ください。Usagi はこの ID を SOURCE_VOCABULARY_ID として使用し、SOURCE_TO_CONCEPT_MAP テーブルで特定のマッピングを識別できるようにします。

SOURCE_VOCABULARY_ID を選択後、出力した CSV に名前を付けて保存します。出力した CSV の構造は SOURCE_TO_CONCEPT_MAP テーブルの構造と同じです。このマッピングは、ボキャブラリの SOURCE_TO_CONCEPT_MAP テーブルに追加できます。また、上記の手順で定義した SOURCE_VOCABULARY_ID を定義する VOCABULARY テーブルに单一の行を追加することも意味があります。最後に、「Approved (承認)」ステータスのマッピングのみが CSV ファイルにエクスポートされることに留意ください。マッピングをエクスポートするには、USAGI でマッピングを完了する必要があります。

Usagi で作成されたマッピングの更新

多くの場合、マッピングは一度だけの作業ではありません。データが更新されると、新しいソースコードが追加され、ボキャブラリが定期的に更新されるため、マッピングの更新が必要になる場合があります。

ソース・コードのセットが更新された場合は、次の手順で更新できます。

1. 新しいソースコードファイルをインポートします。
2. File (ファイル) -> Apply previous mapping (以前のマッピングを適用する) を選択します。
3. 古いマッピングから引き継いだ承認済みのマッピングを継承していないコードを特定し、それらを通常通りマッピングします。

ボキャブラリが更新された場合は、以下の手順に従います：

1. Athena から新しいボキャブラリファイルをダウンロードします。
2. Usagi インデックスを再構築します (Help (ヘルプ) -> Rebuild index (インデックスを再構築する))。
3. マッピング・ファイルを開きます。

- 新しいバージョンのボキャブラリで標準コンセプトでなくなったコンセプトにマッピングされるコードを特定し、より適切なターゲットコンセプトを見つけます。

6.4 ステップ 3: ETL の実装

デザインとコードマッピングを完了すると、ETL プロセスをソフトウェアで実装することができます。ETL の設計段階では、ソースデータと CDM に詳しい人が共同で作業することをお勧めしました。同様に、ETL を実装する際には、大量のデータの取り扱い経験があり、ETL の実装経験がある人が作業を行うことが望ましいでしょう。これは、グループ外の人と協力することや、実装するために技術コンサルタントを雇うことを意味するかもしれません。また、これは一度きりの費用ではないことにも留意する必要があります。今後も ETL の維持と運用に、少なくともある程度の時間を割くことのできる担当者やチームを確保しておくことが望ましいでしょう（詳細は第 6.7 部を参照ください）。

実装の具体的な内容は施設ごとに異なり、インフラストラクチャ、データベースの規模、ETL の複雑さ、利用可能な技術専門知識など多くの要因に依存します。そのため、OHDSI コミュニティは ETL の最適な実装方法について正式な推奨は行っていません。シンプルな SQL ビルダー、SAS、C#、Java、Kettle を使用するグループもいます。それぞれに長所と短所があり、技術に精通している人がいなければどれも使えません。

ETL の例をいくつか挙げます（複雑さの順に記載）：

- ETL-Synthea - Synthea データベースを変換するために書かれた SQL ビルダー
 - <https://github.com/OHDSI/etl-synthea>
- ETL-CDMBuilder - 複数のデータベースを変換するためにデザインされた.NET アプリケーション
 - <https://github.com/OHDSI/etl-cdmbuilder>
- ETL-LambdaBuilder - AWS lambda 機能を使用するビルダー
 - <https://github.com/OHDSI/etl-lambdabuilder>

複数回の独立した試みの後、ユーザー「フレンドリーな」究極の”ETL ツールの開発を断念しました。このようなツールは ETL の 80% にはうまく機能しますが、残りの 20% については、ソースデータベース固有の低レベルのコードを記述する必要があります。

技術担当者が実装を開始する準備ができたら、ETL デザイン文書を彼らと共有するべきです。ドキュメントには開発を開始するための十分な情報が含まれているはずですが、開発プロセス中に開発者が ETL 設計者に質問できるようにしておくことが重要です。設計者にとっては明確な論理も、データや CDM に不慣れな実装者にはわかりにくいこともあります。実装フェーズはチーム全体の作業として維持するべきです。すべてのロジックが正しく実行されたという点で両グループが合意するまで、実装者と設計者がそれぞれ CDM の作成とテス

トを行うプロセスを経ることが、適切な方法であると考えられます。

6.5 ステップ 4: 品質管理

抽出、変換、読込のプロセスでは品質管理は反復的なプロセスとなります。典型的なパターンは、ロジックの記述-> ロジックの実装-> ロジックのテスト-> ロジックの修正・記述です。CDM をテストする方法はいくつもありますが、以下は長年にわたる ETL 実装を通じてコミュニティ全体で開発された推奨手順です。

- ETL 設計文書、コンピュータコード、およびコードマッピングのレビューどんな人でも間違いを犯す可能性があるため、常に少なくとももう 1 人の人間が、実施された内容を確認すべきです。
 - コンピュータコードにおける最大の課題は、ネイティブデータのソースコードが標準コンセプトにどのようにマッピングされるかに起因します。特に日付特有のコード（NDC など）の場合、マッピングが厄介になることがあります。どの領域でもマッピングが行われる場合は、正しいソースボキャブラリが適切なコンセプト ID に変換されているかを必ず再確認してください。
- ソースデータとターゲットデータのサンプルに関する情報を手動で比較します。
 - 理想的には、多数のユニークレコードを持つ人物のデータを 1 件ずつ確認すると役立つでしょう。CDM のデータが合意されたロジックに基づいて期待される形になっていない場合、1 人の人物のデータを追跡することで問題が浮き彫りになる可能性があります。
- ソースデータとターゲットデータの全体的なカウントを比較します。
 - 特定の問題にどのように対処するかによって、カウントに期待される多少の差異が生じるかもしれません。たとえば、一部のコラボレーターは、NULL 性別を持つ人々を削除することを選択しています。なぜなら、そのような人々は分析には含まれないからです。また、CDM におけるビジットは、ネイティブデータにおけるビジットや受診とは異なる方法で構築されている場合もあります。したがって、ソースデータと CDM データの全体的なカウントを比較する際には、これらの相違を考慮し、予想しておくことが重要です。
- ソースデータで既済の研究を CDM バージョンで再現します。
 - これはソースデータと CDM バージョンとの間の主な相違点を理解するのに適した方法ですが、多少時間がかかります。
- ETL で対処すべきソースデータのパターンを再現するユニットテストを作成します。例えば、ETL で性別情報が欠落している患者を削除するよう指定されている場合、性別が未設定の人物のユニットテストを作成し、ビルダーがそれをどのように処理するかを評価します。
 - ユニットテストは、ETL 変換の品質と精度を評価する際に非常に便利です。通常、変換元のデータ構造を模倣したより小規模なデータセットを作成します。このデータセット内の各個人またはレコードは、ETL 文書に記載されている特定のロジックをテストする必要が

あります。この方法を使用すると、問題を簡単に追跡し、エラーが発生したロジックを特定することができます。また、小規模であるため、コンピュータコードを非常に迅速に実行でき、より迅速な反復とエラーの特定が可能になります。

以上が ETL の観点から品質管理にアプローチするハイレベルの方法です。OHDSI コミュニティ内で進行中のデータ品質への取り組みの詳細については、第 15 章を参照ください。

6.6 ETL の規約と THEMIS

データを CDM に変換するグループが増えるにつれ、特定の状況で ETL がどのように対処すべきかを明確にする必要があることが明らかになりました。例えば、出生年が欠けている個人レコードの場合、ETL はどうすべきでしょうか？CDM の目標はヘルスケアデータを標準化することですが、各グループが特定のデータシナリオを異なる方法で処理すると、ネットワーク全体でデータを体系的に使用することが難しくなります。

OHDSI コミュニティは、一貫性を向上させるために慣行を文書化し始めました。OHDSI コミュニティが合意したこれらの定義された慣行は、CDM Wiki で参照できます⁶。各 CDM テーブルには、ETL を設計する際に参照できる独自の慣行セットがあります。たとえば、出生年の月日が欠けている個人は許容されますが、出生年が欠けている場合、その個人は除外する必要があります。ETL を設計する際には、コミュニティと一貫性のある設計上の決定を行うためにこれらの慣行を参照ください。

すべてのデータシナリオを文書化し、発生した場合に何をするかをドキュメント化することは不可能ですが、一般的なシナリオを文書化しようとしている OHDSI のワークグループがあります。THEMIS⁷は、コミュニティ内で慣行を収集し、それを明確にし、コミュニティと共有し、最終的な慣行を CDM Wiki に文書化するメンバーで構成されています。THEMIS は、古代ギリシャの神々を司る女神で、秩序、公正、法、自然法、慣習を司る存在であり、このグループの任務にふさわしいと考えられました。ETL を実行する際に、どのように処理すべきか判断に迷うシナリオがあった場合、THEMIS はそのシナリオについて OHDSI フォーラムに質問を投げかけることを推奨しています⁸。質問があるということは、おそらくコミュニティ内の他のメンバーも同じ疑問を抱いている可能性が高いでしょう。THEMIS はこれらの議論、ワークグループの会議、対面式のディスカッションなどをを利用して、他に文書化する必要がある慣行についても情報を収集します。

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

⁷<https://github.com/OHDSI/Themis>

⁸<http://forums.ohdsi.org/>

6.7 CDM および ETL のメンテナンス

ETL をデザインし、マッピングを作成し、ETL を実装し、品質管理措置を構築することは多大な労力を要します。残念ながら、この労力はそこで終わりではありません。最初の CDM が構築されると、ETL メンテナンスのサイクルが継続的に行われます。メンテナンスが必要となる一般的な要因は以下の通りです：ソースデータの変更、ETL のバグ、新しい OMOP ボキャブラリのリリース、CDM 自体の変更や更新などです。これらが発生すると、ETL ドキュメント、ETL を実行するソフトウェアプログラミング、テストケースや品質管理などの更新が必要になる場合があります。

医療データソースは常に変化し続けることがよくあります。新しいデータが利用可能になる場合もあります（例：データに新しい列が追加されるなど）。これまで存在しなかった患者のシナリオが突然現れるかもしれません（例：生まれる前に死亡記録がある新生児患者）。データの理解が改善される可能性があります（例：入院中の子の出産に関する記録の一部が、請求処理の方法により外来患者として記録されている）。すべてのソースデータの変更が ETL 処理の変更を引き起こすわけではありませんが、最低限、ETL 処理を中断する変更には対処する必要があります。

バグが見つかった場合、それに対処する必要があります。ただし、すべてのバグが同じ重要性を持っているわけではないことを念頭に置くことが重要です。例えば、COST テーブルでは cost 列が整数に丸められていたとしましょう（例：ソースデータに \$3.82 があったのに、CDM で \$4.00 になっている）。データを使用して主に患者の薬剤曝露やコンディションの特性評価を行う研究者が多い場合、このようなバグは重要ではなく、将来的に対処すればよいでしょう。しかし、データを使用する主要な研究者に医療経済学者も含まれていた場合、これは直ちに対処する必要がある重大なバグとなります。

OMOP ボキャブラリもまた、ソースデータと同様に常に変化しています。実際、ボキャブラリは 1 ヶ月に何度も更新されることがあります。各 CDM は特定のボキャブラリバージョンで実行されており、新しい改善されたボキャブラリで実行すると、標準化ボキャブラリへのソースコードのマッピング方法に変更が生じる可能性があります。多くの場合、ボキャブラリ間の相違は軽微なものであるため、新しいボキャブラリがリリースされるたびに新しい CDM を構築する必要はありません。しかし、年に 1~2 回は新しいボキャブラリを採用し、CDM を再処理することが推奨されます。ボキャブラリの新バージョンで変更が生じた場合、ETL コード自体を更新する必要が生じることはまれです。

CDM または ETL のメンテナンスが必要となる最後の要因は、共通データモデル自体が更新される場合です。コミュニティが成長し、新たなデータ要件が見つかった場合、CDM に追加データを保存する必要が生じる可能性があります。これは、以前は CDM に保存されていなかったデータが新しい CDM バージョンに保存される可能性があることを意味します。既存の CDM 構造の変更は頻繁ではありませんが、可能性はあります。例えば、CDM が元の DATE フィールドから DATETIME フィールドを採用したことで、ETL 処理にエラーが発生する

可能性があります。CDM バージョンは頻繁にはリリースされず、サイトは移行するタイミングを選択できます。

6.8 ETL に関する最終的な考察

ETL プロセスが異なる理由は数多くありますが、その主な理由のひとつは、私たちがすべてユニークなソースデータを処理しているため、「すべてに適合する」ソリューションを作成するのが難しいという事実です。しかし、長年の経験から、私たちは次のような教訓を得ました。

- 80/20 のルール。ソースコードを手動でコンセプトセットにマッピングするのにあまり時間をかけないようにしてください。理想的には、データの大部分をカバーするソースコードをマッピングします。これだけで、まずスタートを切ることができます。残りのコードについては、ユースケースに基づいて、今後対応することができます。
- 研究の品質に見合わないデータが失われるなどを恐れる必要はありません。これらのレコードは、いずれにしても分析を開始する前に破棄されるものです。代わりに、ETL プロセス中にそれらを削除するだけなのです。
- CDM はメンテナンスが必要です。ETL が完了したからといって、二度と触らないということではありません。生データが変更されるかもしれませんし、コードにバグがあるかもしれません。新しいボキャブラリや CDM の更新があるかもしれません。これらの変更に対応するためのリソースを確保し、ETL が常に最新の状態に保たれるようにしましょう。
- OHDSI CDM の開始、データベースの変換、分析ツールの実行のサポートが必要な場合は、実装者フォーラム⁹をご覧ください。

6.9 まとめ



- ETL にアプローチするための一般的に合意されたプロセスが存在します
 - * データ専門家と CDM 専門家が協力して ETL を設計する
 - * 医療知識を持つ人がコードマッピングを作成する
 - * 技術者が ETL を実装する
 - * すべての関係者が品質管理に関与する
- OHDSI コミュニティはこれらのステップを促進するためにツールを開発しており、これらは自由に利用できます
- 参考にできる多くの ETL 例や合意された慣行があります

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

6.10 演習

演習 6.1. ETL プロセスのステップを正しい順序に並べてください：

- A) データ専門家と CDM 専門家が協力して ETL を設計する
- B) 技術者が ETL を実装する
- C) 医療知識を持つ人がコードマッピングを作成する
- D) すべての関係者が品質管理に関与する

演習 6.2. 選択した OHDSI リソースを使用して、表 6.3 に示す PERSON レコードに関する 4 つの問題点を指摘してください（表はスペースのため省略されています）：

Table 6.3: PERSON テーブル

列	値
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	提供されていない

演習 6.3. VISIT_OCCURRENCE レコードを生成してみましょう。以下は Synthea のために書かれたロジック例です：PATIENT、START、END のデータを昇順で並べ替えます。その後、PERSON_ID ごとに、前の行の END と次の行の START の間に 1 日以内の時間がある場合、請求の行を統合します。統合された入院の請求は 1 つの入院ビジットと見なされ、設定されます：

- MIN(START) を VISIT_START_DATE として設定
- MAX(END) を VISIT_END_DATE として設定
- “IP” を PLACE_OF_SERVICE_SOURCE_VALUE として設定

ソースデータに図 6.10 に示されるようなビジットのセットがある場合、CDM で生成される VISIT_OCCURRENCE レコードはどのようになると予想しますか？

提案された回答は付録 E.3 を参照ください。

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: 例のソースデータ。

第 III 部

データ解析

第 7 章

データ解析の使用例

著者: David Madigan

OHDSI 共同研究は、通常、請求データベースや電子カルテデータベースなどの形式で、実世界のヘルスケアデータから信頼性の高いエビデンスを生成することに重点を置いています。OHDSI が重点的に取り組むユースケースは、主に以下の 3 つのカテゴリーに分類されます。

- 特性評価
- 集団レベルの推定
- 患者レベルの予測

以下でこれらについて詳しく説明します。すべてのユースケースにおいて、生成されるエビデンスはデータの限界を継承します；これらの限界については、エビデンスの質に関する本の（第 14 章- 第18章）で詳しく説明しています。

7.1 特性評価

特性評価は次のような質問に答えようとします

かれらに何が起こったのか？

データを用いて、コホートやデータベース全体の集団の特性、医療行為、経時的な変化に関する問い合わせることができます。

データが答えを提供できる問い合わせには次のような例があります：

- 新たに心房細動と診断された患者のうち、何人がワルファリンの処方を受けたのか？
- 人口股関節置換術を受けた患者の平均年齢は？
- 65 歳以上の患者の肺炎の発生率は？

典型的な特性評価は以下のように定式化されます：

- 何人の患者が…？
- どのくらいの頻度で…？
- 患者の何パーセントが…？
- 検査値の分布はどのようにになっているか…？
- の患者の HbA1c 値は…？
- の患者の検査値は…？
- の患者の曝露期間の中央値は…？
- 経時的な傾向は？
- これらの患者が使用している他の薬剤は何か？
- 併用療法は？
- の症例が十分にあるか？
- X を研究は可能か？
- の人口統計は？
- のリスク要因は？(特定のリスク要因を識別する場合、予測ではなく推定)
- の予測因子は？

そして期待されるアプローチは以下の通りです：

- カウントまたはパーセンテージ
- 平均
- 記述統計
- 発生率
- 有病率
- コホート
- ルールベースの表現型
- 薬剤利用
- 疾患の自然経過
- 服薬アドヒアランス
- 併存疾患のプロファイル
- 治療経路
- 治療方針

7.2 集団レベルの推定

限定的ではありますが、データは医療介入の効果に関する因果推論を裏付けることができ、次の問い合わせに答えます

因果効果とは何か？

私たちは因果効果を理解することで、行動の結果を理解したいと考えています。例えば、ある治療法を受けると決めた場合、将来にわたって私たちの身に何が起こるのかがどう変わるのでしょうか？

データは、次のような問い合わせに対する答えを提供することができます：

- 新たに心房細動と診断された患者において、治療開始後最初の 1 年間に、ワルファリンはダビガトランよりも重大な出血を引き起こすか？

- ・メトホルミンの下痢に対する因果効果は年齢によって異なるか？

典型的な集団レベルの効果推定の問い合わせは次のように定式化されます：

- ・…の効果は？
- ・介入を行った場合、どうなるのか？
- ・どちらの治療がより効果的か？
- ・Yに対するXのリスクは？
- ・のイベント発生までの時間は？

そして、期待されるアウトプットは以下の通りです：

- ・相対リスク
- ・ハザード比
- ・オッズ比
- ・平均治療効果
- ・因果効果
- ・関連
- ・相関
- ・安全性監視
- ・比較効果

7.3 患者レベルの予測

データベースに収集された患者の医療履歴に基づいて、将来の健康イベントに関する患者レベルの予測を行い、次の問い合わせ答えます

私には何が起こるのか？

データは、以下のような質問に対する答えを提供することができます：

- ・新たに重度うつ病と診断された特定の患者について、診断後1年以内に自殺を図る確率はどの程度か？
- ・新たに心房細動と診断され、ワルファリンによる治療開始後1年以内に虚血性脳卒中を発症する確率どの程度か？

典型的な患者レベルの予測に関する問い合わせは次のように定式化されます：

- ・この患者が…になる可能性はどの程度か？
- ・誰が…の候補となるのか？

そして、期待されるアウトプットは以下の通りです：

- ・個人の確率
- ・予測モデル
- ・高リスク/低リスクグループ
- ・確率的な表現型

集団レベルの推定と患者レベルの予測はある程度重複することがあります。例えば、予測の重要なユースケースとしては、特定の患者に薬剤Aが処方された

場合の治療結果を予測し、また薬剤 B が処方された場合の同じ治療結果を予測することが挙げられます。現実には、これらの薬剤のうちの 1 つだけが処方された（例えば薬剤 A）と仮定すると、薬剤 A による治療の結果が実際に起るかどうかを確認することができます。薬剤 B は処方されていないため、薬剤 B による治療の結果は予測可能ですが、「反事實」であり、実際には観察されません。予測作業はそれぞれ患者レベルの予測に該当します。しかし、2 つの結果の差（または比率）は単位レベルの因果効果であり、代わりに因果効果推定法を用いて推定すべきです。



人々は予測モデルを因果モデルとして誤って解釈する傾向があります。しかし、予測モデルは相関関係のみを示すことができ、因果関係を示すことはできません。例えば、糖尿病は心筋梗塞（MI）の強いリスク要因であるため、糖尿病治療薬の使用は心筋梗塞の強い予測因子であるかもしれません。しかし、糖尿病治療薬の使用を中止すれば心筋梗塞を予防できるというわけではありません！

7.4 高血圧症におけるユースケース

あなたは、高血圧症の第一選択治療として急性心筋梗塞や血管性浮腫に対する ACE 阻害薬単独療法とサイアザイド系利尿薬単剤療法の効果を研究することに関心のある研究者です。OHDSI の文献に基づいて、集団レベルの効果推定値を求める問い合わせすることになると理解していますが、まず、この特定の治療に関する特性評価を行うため、準備をする必要があります。

7.4.1 特性評価に関する問い合わせ

急性心筋梗塞は高血圧症患者に起こりうる心血管系の合併症であり、高血圧症に対する有効な治療によってリスクを軽減できるはずです。血管性浮腫は稀ですが重篤になる可能性のある ACE 阻害薬の既知の副作用です。あなたは、対象とする曝露（ACE 阻害薬の新規使用者およびサイアザイド系利尿薬の新規使用者）のコホートを作成することから始めます（第 10 章を参照）。曝露集団のベースライン特性（人口統計学的特性、併存疾患、併用薬など）を要約するため、特性評価（第 11 章を参照）の解析を実行します。この曝露集団における特定のアウトカムの発生率を推定するために、別の特性評価を実行します。ここでは、「ACE 阻害薬およびサイアザイド系利尿薬に曝露された期間に 1) 急性心筋梗塞および 2) 血管性浮腫がどのくらいの頻度で発生するか？」という問い合わせします。これらの特性評価により、集団レベルの効果推定研究を実施できるかどうかの実行可能性を評価し、2 つの治療グループが比較可能かどうかを評価し、患者がどちらの治療を選択したかを予測する「リスク因子」を特定することができます。

7.4.2 集団レベルの推定問題

集団レベルの効果推定研究（第12章を参照）は、急性心筋梗塞と血管性浮腫のアウトカムに対するACE阻害薬対サイアザイドの使用の相対リスクを推定します。ここでは、研究診断とネガティブコントロールにより、平均治療効果の信頼性の高い推定値を生成できるかどうかをさらに評価します。

7.4.3 患者レベルの予測問題

曝露の因果効果とは独立して、アウトカムのリスクが最も高い患者を特定しようとすることにも興味があります。これは患者レベルの予測問題です（第13章を参照）。ここでは、ACE阻害薬の新規ユーザーの中で、治療開始後1年以内に急性心筋梗塞を発症するリスクが最も高い患者を評価する予測モデルを開発します。このモデルにより、ACE阻害薬を初めて処方された患者について、その病歴から観察されたイベントに基づき、今後1年間に急性心筋梗塞を発症する確率を予測することができます。

7.5 観察研究の限界

OHDSIデータベースでは答えを出せない重要な医療問題は数多くあります。以下はその例です。：

- ・ プラセボと比較した介入の因果効果。治療と非治療の比較は可能であっても、プラセボ治療との比較による因果効果を考慮することはできない場合があります。
- ・ 市販薬に関するもの。
- ・ 多くのアウトカムやその他の変数は、ほとんど記録されていないか、まばらにしか記録されていません。これには、死亡率、行動アウトカム、ライフスタイル、社会経済的地位が含まれます。
- ・ 患者は体調が悪いときにしか医療システムを利用しない傾向があるため、治療の有益性を測定することが困難です。

7.5.1 誤ったデータ

OHDSIデータベースに記録された臨床データは、臨床の現実と乖離している可能性があります。例えば、患者が心筋梗塞を経験していないにもかかわらず、患者記録に心筋梗塞のコードが含まれている可能性があります。同様に、検査値が誤っていたり、プロシージャー（処置）の誤ったコードがデータベースに表示されている可能性もあります。第15章や第16章では、これらの問題について説明しており、ベストプラクティスでは、このような問題を可能な限り特定し、修正することを目指しています。ただし、誤ったデータは必然的にある程度は残存し、その後の分析の妥当性を損なう可能性はあります。広範な文献が、データエラーを考慮した統計的推論の調整に焦点を当てており、例えば、Fuller (2009) を参照ください。

7.5.2 欠損データ

OHDSI データベースにおける欠損は微妙な課題を呈します。データベースに記録されるべき健康イベント（例：処方、検査値など）が記録されていない場合、それは「欠損」となります。統計学の文献では、「完全にランダムに欠損」、「ランダムに欠損」、「非ランダムに欠損」など欠損のタイプを区別し、複雑性を増す手法によってこれらのタイプに対処しようとしています。Perkins et al. (2017) はこのトピックに関する有用な入門書を提供しています。

7.6 まとめ



- 観察研究では、3つの大きなユースケースのカテゴリーを区別します。
- 特性評価は「彼らに何が起こったか？」という問い合わせようとしています。
- 集団レベルの推定は「因果効果は何か？」という問い合わせようとしています。
- 患者レベルの予測は「私には何が起こるのか？」という問い合わせようとしています。
- 予測モデルは因果モデルではありません。強い予測因子に介入してもアウトカムに影響を与えると考える理由はありません。
- 観察医療データでは答えられない問い合わせもあります。

7.7 演習

演習 7.1. これらの質問はどのユースケースのカテゴリーに属しますか？

1. 最近非ステロイド性抗炎症薬 (NSAID) を投与された患者における消化管 (GI) 出血の発生率を計算する。
2. ベースラインの特性に基づいて、特定の患者が今後 1 年間に GI 出血を経験する確率を計算する。
3. セレコキシブと比較してジクロフェナクによる GI 出血のリスク増加を推定する。

演習 7.2. ジクロフェナクによる GI 出血のリスクがプラセボ（偽薬）に比べてどの程度高まるかを推定したい。医療観察データを使用して、この推定を行うことは可能だろうか？

推奨される解答は、付録 E.4を参照ください。

第 8 章

OHDSI 分析ツール

著者: Martijn Schuemie & Frank DeFalco

OHDSI は、患者レベルの観察データに対するさまざまなデータ分析のユースケースをサポートするための幅広いオープンソースツールを提供しています。これらのツールの共通点は、すべて共通データモデル（CDM）を使用して 1 つ以上のデータベースと相互にやりとりできることです。さらに、これらのツールはさまざまなユースケースに対して分析を標準化します。ゼロから始める必要はなく、標準テンプレートに入力することで分析を実装できます。これにより、分析が容易になり、再現性と透明性が向上します。例として、発生率を計算する方法は無数にあるように思われますが、OHDSI ツールではいくつかの選択肢で指定でき、同じ選択肢を選んだ人は同じ方法で発生率を計算します。

本章では、最初に分析を実装するさまざまな方法を説明し、分析で採用できる戦略について説明します。次に、さまざまな OHDSI ツールとそれらがどのようにさまざまなユースケースにどのように適合するかを検討します。

8.1 分析の実装

図 8.1 は、CDM を使用してデータベースに対して研究を実装するために選択できるさまざまな方法を示しています。

研究を実装するための主なアプローチは 3 つあります。最初の方法は、OHDSI が提供するツールを一切使用しないカスタムコードを作成することです。R、SAS、またはその他の言語で新規の分析を作成することができます。これにより最大の柔軟性が得られ、特定の分析がツールでサポートされていない場合は唯一の選択肢となるかもしれません。しかし、この方法は高度な技術、時間、労力を必要とし、分析が複雑になるほどコードのエラーを避けることが難しくなります。

2 番目の方法は、R で分析を開発し、OHDSI Methods Library のパッケージ

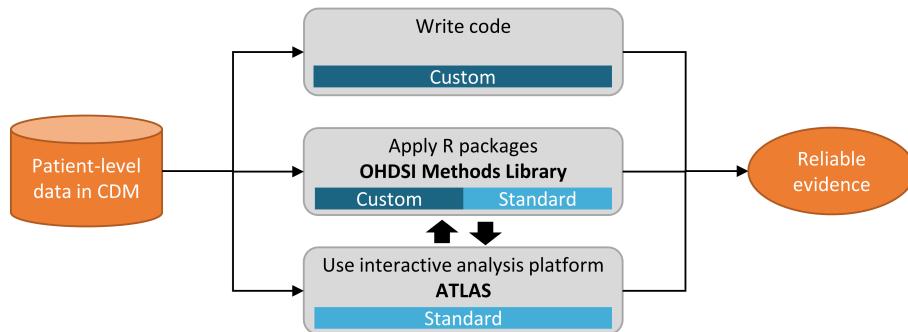


Figure 8.1: CDM のデータに対する分析を実装するさまざまな方法

を利用する方法です。少なくとも、SqlRenderとDatabaseConnectorのパッケージを使用できます。これらのパッケージについては、第 9 章で詳しく説明していますが、PostgreSQL、SQL Server、Oracle などのさまざまなデータベースプラットフォーム上で同じコードを実行することができます。また、CohortMethodやPatientLevelPredictionなどのパッケージでは、CDM に対する高度な分析のための R 関数が提供されており、コードから呼び出すことができます。これには依然として高度な専門知識が必要ですが、検証済の Methods Library のコンポーネントを再利用することで、完全にカスタムコードを使用する場合よりも効率的に作業を進めることができます。エラーが発生する可能性も低くなります。

3 番目の方法は、プログラマーでなくても幅広く分析を効率的に実行できるウェブベースのツール、ATLASを使用することです。ATLAS は Methods Libraries を使用しますが、分析をデザインするための単純なグラフィカルインターフェイスを提供し、多くの場合、分析を実行するために必要な R コードを生成します。ただし、Methods Library で利用可能なすべてのオプションをサポートしているわけではありません。大半の研究は ATLAS を通じて行うことができると予想されますが、2 番目の方法が提供する柔軟性を必要とする研究もあります。

ATLAS と Methods Library は独立したものではありません。ATLAS で呼び出される複雑な分析の一部は、Methods Library のパッケージへの呼び出しを通じて実行されます。同様に、Methods Library で使用されるコホートは、多くの場合、ATLAS でデザインされています。

8.2 分析戦略

CDM に対する分析を実装するための戦略に加え、例えばカスタムコーディングや Methods Library で提供される標準分析コードの利用など、エビデンスを生成するための分析技術を使用する複数の戦略もあります。図 8.2 は、OHDSI で採用されている 3 つの戦略を示しています。

最初の戦略では、各分析を個別の研究として扱います。分析はプロトコルで事

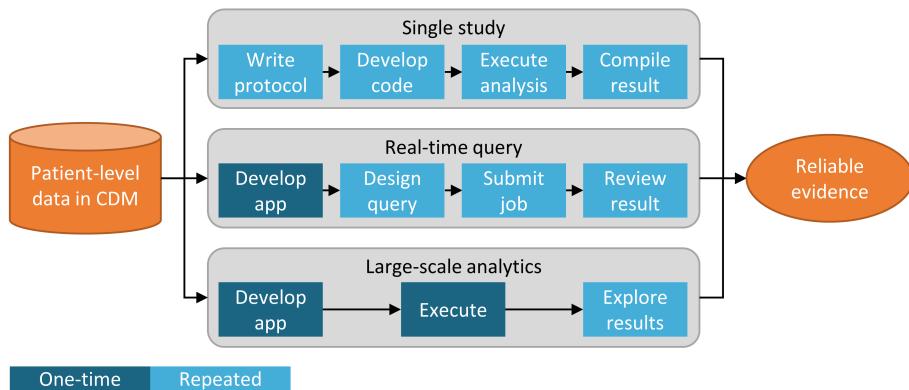


Figure 8.2: (臨床の) 問いに対するエビデンスを生成するための戦略

前に規定し、コードとして実装し、データに対して実行し、その後、結果をまとめ解釈します。各質問ごとに、すべてのステップを繰り返す必要があります。このような分析の一例は、levetiracetam と phenytoin を比較した際の血管性浮腫のリスクに関する OHDSI 研究です (Duke et al., 2017)。ここでは、まずプロトコルが作成され、OHDSI Methods Library を使用した分析コードが開発され、OHDSI ネットワーク全体で実行され、結果がまとめられて学術誌に公表されました。

第二の戦略では、リアルタイムまたはほぼリアルタイムで特定のクラスの問い合わせに答えられるアプリケーションを開発します。アプリケーションが開発されると、ユーザーはクエリをインタラクティブに定義し、それを送信して結果を表示できます。この戦略の一例は、ATLAS のコホート定義および生成ツールです。このツールは、ユーザーが複雑さの異なるコホート定義を指定し、データベースに対してその定義を実行して、さまざまな包含基準と除外基準を満たす人数を確認することができます。

第三の戦略では、同様に問い合わせに焦点を当てますが、その問い合わせのクラス内のすべてのエビデンスを網羅的に生成しようとします。ユーザーは、さまざまなインターフェースを通じて必要に応じてエビデンスを探索できます。一例は、うつ病治療の効果に関する OHDSI 研究です (Schuemie et al., 2018b)。この研究では、すべてのうつ病治療が、4 つの大規模な観察研究データベースで関心のあるアウトカムの大規模なセットに対して比較されました。17,718 の実証的にキャリブレーションされたハザード比と広範な研究診断を含む結果の全セットは、インタラクティブなウェブアプリ¹で利用できます。

8.3 ATLAS

ATLAS は、OHDSI コミュニティが開発した、標準化された患者レベルの観察データを CDM 形式で分析する設計と実行を支援する、無料で公開されているウ

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

エブベースのツールです。ATLAS は、OHDSI WebAPI と組み合わせてウェブアプリケーションとして展開され、通常は Apache Tomcat 上でホストされます。リアルタイム分析を行うには、CDM 内の患者レベルデータへのアクセスが必要であるため、通常は組織のファイアウォールのバックにインストールされます。ただし、パブリックな ATLAS⁴⁰ も存在し、この ATLAS インスタンスは少数の小規模なシミュレーションデータセットにしかアクセスできませんが、テストやトレーニングなど、多くの目的に使用できます。パブリックな ATLAS インスタンスを使用して効果の推定や予測研究を完全に定義し、研究を実行するための R コードを自動的に生成することも可能です。そのコードは、ATLAS と WebAPI をインストールすることなく、利用可能な CDM がインストールされている環境であればどこでも実行できます。

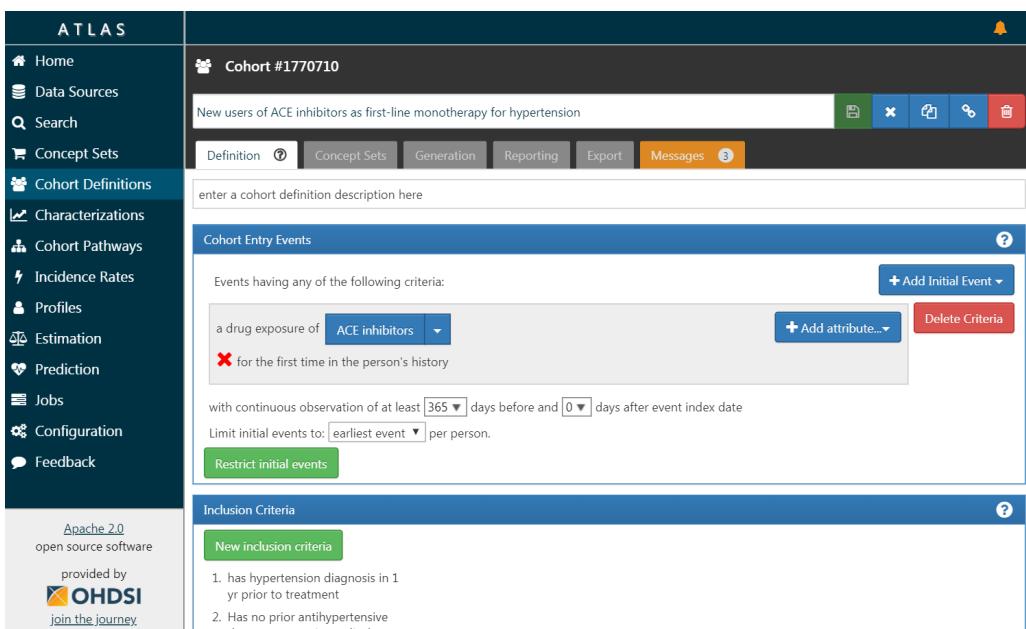


Figure 8.3: ATLAS ユーザインターフェース

図 8.3 に ATLAS のスクリーンショットを示します。左側には ATLAS の様々な機能を示すナビゲーションバーがあります。

データソース データソースは、ATLAS プラットフォームに構成された各データソースの記述的で標準化されたレポートをレビューする機能を提供します。この機能は大規模分析戦略を用いており、すべての記述は事前に計算されています。データソースについては、第11章で説明します。

ボキャブラリ検索 ATLAS は OMOP 標準化ボキャブラリを検索し、これらのボキャブラリにどのようなコンセプトが存在するのか、そしてそのコンセプトをどう適用するかを理解するための機能を提供します。この機能について、第5章で議論されています。

コンセプトセット コンセプトセットは、標準化された分析全体で使用するコ

ンセプトのセットを識別するために使用できる論理式のコレクションを作成する機能を提供します。コンセプトセットは単純なコードや値のリストよりも高度な機能を提供します。コンセプトセットは、標準化ボキャブラリからの複数のコンセプトと、関連するコンセプトの包含や除外を指定するための論理インジケータを組み合わせて構成されます。ボキャブラリの検索、コンセプトのセットの特定、そしてコンセプトセットを解決するために使用する論理の指定は、分析プランで使用されることが多い難解な医療用語を定義するための強力なメカニズムとなります。これらのコンセプトセットは ATLAS 内に保存され、その後の解析の一部としてコホート定義や解析仕様に使用できます。

コホート定義 コホート定義は、一定期間内に 1 つまたは複数の基準を満たす人のセットを構築する機能であり、これらのコホートはその後のすべての分析の入力の基礎として使用されます。この機能については、第10章で説明します。

特性評価 特性評価は、定義された 1 つまたは複数のコホートを調査し、これらの患者集団の特性を要約するための分析機能です。この機能はリアルタイムクエリ戦略を用いており、第11章で説明しています。

コホート経路 コホート経路は、1 つまたは複数の集団内で発生する臨床イベントのシーケンスを観察できる分析ツールです。この機能については、第11章で説明されています。

発生率 発生率は、対象集団内のアウトカムの発生率を推定するためのツールです。この機能については、第11章で説明されています。

プロファイル プロファイルは、個々の患者の縦断的観察データを調査し、特定の個人に起こっている状況を要約するためのツールです。この機能はリアルタイムクエリ戦略を使用します。

集団レベル推定 推定は、比較コホートデザインを使用して集団レベルの効果推定研究を定義し、1 つまたは複数の対象コホートと比較コホート間の比較の一連の結果を調査することができます。この機能はコーディングが不要で、リアルタイムクエリ戦略を実装していると言えます。この機能については第12章で説明されています。

患者レベルの予測 予測昨日は機械学習アルゴリズムを適用して、患者レベルの予測分析を行い、特定のターゲット曝露内でアウトカムを予測する機能です。この機能もリアルタイムクエリ戦略を実装しており、コーディングが不要です。第13章で説明されています。

ジョブ ジョブメニュー項目を選択して、WebAPI を通じて実行されているプロセスの状態を確認できます。ジョブは、コホートの生成やコホートの特性評価のレポートの計算など、長時間実行されるプロセスであることが多いです。

構成 構成メニュー項目を選択して、ソース構成セクションに構成されたデータソースを確認できます。

フィードバック フィードバックリンクをクリックすると、ATLAS の課題ログにアクセスし、新しい問題のログを記録したり、既存の問題を検索したりすることができます。新しい機能や拡張機能のアイデアがある場合は、開発コミュニティに伝える場所としても利用できます。

8.3.1 セキュリティ

ATLAS と WebAPI は、プラットフォーム全体で機能やデータソースへのアクセスを制御するための細かいセキュリティモデルを提供します。セキュリティシステムは Apache Shiro ライブラリを活用して構築されています。セキュリティシステムの詳細は、オンラインの WebAPI セキュリティ wiki² で確認できます。

8.3.2 ドキュメント

ATLAS のドキュメントは、ATLAS GitHub リポジトリの wiki³ でオンラインで確認できます³。この wiki には、さまざまなアプリケーション機能に関する情報や、オンラインビデオチュートリアルへのリンクが含まれています。

8.3.3 インストール方法

ATLAS のインストールは、OHDSI WebAPI と組み合わせて行います。各コンポーネントのインストールガイドは、ATLAS GitHub リポジトリのセットアップガイド⁴と WebAPI GitHub リポジトリのインストールガイド⁵でオンラインで参照できます。

8.4 Methods Library

OHDSI Methods library⁶ は、図 8.4 に示されているオープンソースの R パッケージのコレクションです。

これらのパッケージは、CDM 内のデータから始まり、推定値やそれを裏付ける統計、図表を生成する完全な観察研究を実施するための R 関数を提供しています。これらのパッケージは CDM 内の観察データと直接やりとりし、第9章で説明されているような完全にカスタマイズされた分析にクロスプラットフォームの互換性を提供するために用いることも、集団特性の評価（第11章）、集団レベルの効果推定（第12章）、患者レベルの予測（第13章）のための高度な標準化分析を提供することもできます。Methods library⁶ は、透明性、再現性、異なるコンテキストでのメソッドの操作特性の測定値、そのメソッドによって生成

²<https://github.com/OHDSI/WebAPI/wiki/Security-Configuration>

³<https://github.com/OHDSI/ATLAS/wiki>

⁴<https://github.com/OHDSI/Atlas/wiki/Atlas-Setup-Guide>

⁵<https://github.com/OHDSI/WebAPI/wiki/WebAPI-Installation-Guide>

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

Figure 8.4: OHDSI Methods Library のパッケージ

される推定値やその後の経験的キャリブレーションなど、過去や現在の研究から学んだベストプラクティスをサポートしています。

Methods library はすでに多くの公表された臨床研究 (Boland et al., 2017; Duke et al., 2017; Ramcharan 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) で使用されており、方法論の研究にも利用されています (Schuemie et al., 2014, 2016; Reps et al., 2018; Tian et al., 2018; Schuemie et al., 2018a,b; Reps et al., 2019)。Methods library 内のメソッドの実装の妥当性については第 17 章で説明されています。

8.4.1 大規模分析サポート

すべてのパッケージで組み込まれている重要な機能の一つは、多くの分析を効率的に実行できることです。例えば、集団レベルの推定を行う場合、Cohort-Method パッケージは多数の曝露とアウトカムに対して効果量の推定を行うことを可能にし、さまざまな分析設定を使用して、必要な中間データセットや最終データセットを計算するための最適な方法を自動的に選択します。共変量の抽出や、一つのターゲット・コンパレータペアに対して複数のアウトカムで使用される傾向スコアモデルの適合など、再利用可能なステップは一度だけ実行されます。可能な場合は、計算リソースを最大限に活用するために計算は並列処理われます。

この計算効率により、大規模な分析が可能になり、多くの質問に一度に回答することができます。また、コントロール仮説（ネガティブコントロールなど）を含めることで、当社の手法の運用特性を測定し、第18で説明されているように、経験則に基づくキャリブレーションを行うことも不可欠です。

8.4.2 ビッグデータ対応

Methods library は、非常に大規模なデータベースに対しても大量のデータを含む計算を実行できるようにデザインされています。これは次の三つの方法で実現されます：

1. 大部分のデータ操作はデータベースサーバー上で実行されます。分析は通常、データベース内の全データのごく一部しか必要としないため、Methods library は SqlRender や DatabaseConnector パッケージを介して関連データの前処理や抽出をする高度な操作をサーバー上で実行できるようにします。
2. 大量のローカルデータオブジェクトはメモリ効率の良い方法で保存されます。ローカルマシンにダウンロードされるデータについては、Methods library は ff パッケージを使用して大規模データオブジェクトを保存、処理します。これにより、メモリに収まらないほど大きなデータでも処理することが可能です。
3. 必要に応じて高性能コンピューティングが適用されます。例えば、Cyclops パッケージは、Methods library 全体で使用される非常に効率的な

回帰エンジンを実装しており、これにより通常は適合できない大規模な回帰（多くの変数、大量の観測値）を実行することができます。

8.4.3 ドキュメント

R はパッケージを文書化するための標準的な方法を提供しています。各パッケージには、パッケージに含まれるすべての関数とデータセットを文書化したパッケージマニュアルがあります。すべてのパッケージマニュアルは、Methods Library のウェブサイト⁶、パッケージの GitHub リポジトリ、CRAN で利用できます。さらに、R の内部からパッケージマニュアルを参照するにはクエスチョンマークを使用します。例えば DatabaseConnector パッケージを読み込んだ後、コマンド?connect を入力すると「connect」関数に関するドキュメントが表示されます。

パッケージマニュアルに加えて、多くのパッケージはビネットが提供されています。ビネットは、特定のタスクを実行するためにパッケージをどのように使用するかを説明した詳細なドキュメントです。例えば、一つのビネット⁷では、CohortMethod パッケージを使用して複数の分析を効率的に実行する方法が説明されています。ビネットは Methods Library のウェブサイト、パッケージの GitHub リポジトリ、CRAN で入手可能なパッケージは CRAN でも見つけることができます。

8.4.4 システム要件

システム要件を検討する際には、二つのコンピューティング環境が関連してきます：データベースサーバーと分析ワークステーションです。

データベースサーバーは CDM 形式の観察医療データを保持する必要があります。Methods library は、従来のデータベースシステム (PostgreSQL、Microsoft SQL Server、Oracle)、パラレルデータウェアハウス (Microsoft APS、IBM Netezza、Amazon Redshift)、に加えビッグデータプラットフォーム (Impala 経由での Hadoop、Google BigQuery) など、幅広いデータベース管理システムをサポートしています。

分析ワークステーションは、Methods library がインストールされ実行される場所です。これがローカルマシン（例えば、ノートパソコン）か、RStudio Server が実行されるリモートサーバーかに関わらず、R がインストールされている必要があります。可能であれば RStudio も一緒にインストールすることをお勧めします。また、Methods Library では Java がインストールされている必要があります。分析ワークステーションはデータベースサーバーに接続できる必要があり、具体的には、両者の間にファイアウォールがある場合は、ワークステーションでデータベースサーバーのアクセスポートを開いている必要があります。一部の分析は計算集中的であるため、複数のプロセッサコアと十分なメモ

⁶<https://ohdsi.github.io/MethodsLibrary>

⁷<https://ohdsi.github.io/CohortMethod/articles/MultipleAnalyses.html>

リを持つことが分析の高速化につながります。少なくとも 4 コアと 16 ギガバイトのメモリを推奨します。

8.4.5 インストール方法

OHDSI R パッケージを実行するために必要な環境をインストールするための手順は次の通りです。インストールする必要があるものは 4 つあります：

1. R は統計的コンピューティング環境です。基本的なユーザインターフェースとして主にコマンドラインインターフェースを提供します。
2. Rtools は、Windows で R パッケージをソースからビルドする際に必要なプログラム一式です。
3. RStudio は、R を使いやすくする IDE（統合開発環境）です。コードエディタ、デバッグ、およびビジュアルツールが含まれています。素晴らしい R 体験を得るために、これを使用ください。
4. Java は、OHDSI R パッケージの一部のコンポーネント、例えばデータベースへの接続に必要なコンポーネントを実行するために必要なコンピューティング環境です。

以下では、Windows 環境でのそれぞれのインストール方法を説明します。



Windows では、R と Java はどちらも 32 ビットと 64 ビットのアーキテクチャがあります。R を両方のアーキテクチャにインストールする場合、Java も両方のアーキテクチャにインストールしなければなりません。R の 64 ビットのみをインストールすることをお勧めします。

R のインストール

1. <https://cran.r-project.org/> で、図 8.5 に示されるように「Download R for Windows」、「base」の順にクリックし、ダウンロードしてください。



Figure 8.5: CRAN からの R のダウンロード

2. ダウンロードが完了したら、インストーラを実行します。2 つの例外を除いて、すべてデフォルトのオプションを使用してください：まず、プログラムファイルにはインストールしない方が良いでしょう。代わりに、図 8.6 のように、C ドライブのサブフォルダとして R を作成します。次に、R

と Java のアーキテクチャの違いによる問題を回避するため、図 8.7 のように 32 ビットアーキテクチャを無効にします。

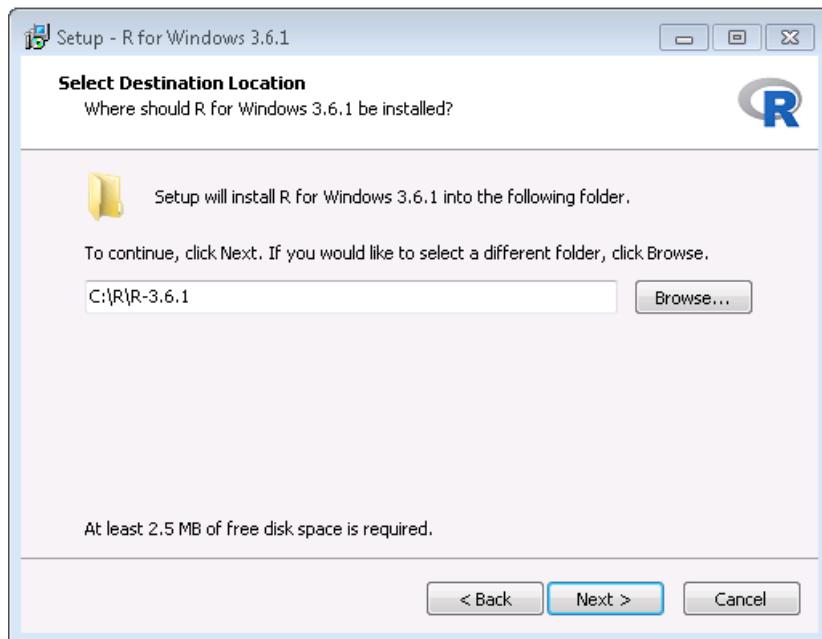


Figure 8.6: R フォルダの設定

完了すると、スタートメニューから R を選択できるようになります。

Rtools のインストール

1. <https://cran.r-project.org/> にアクセスし、「Download R for Windows」をクリックし、次に「Rtools」をクリックして、最新版の Rtools をダウンロードします。
2. ダウンロードが完了後、インストーラを実行します。すべてデフォルトのオプションを選択します。

RStudio のインストール

1. <https://www.rstudio.com/> にアクセスし、「Download RStudio」または RStudio の下の「ダウンロード」ボタンをクリックし、無料版を選択し、図 8.8 に示されるように Windows 用のインストーラをダウンロードします。
2. ダウンロードが完了後、インストーラを実行します。すべてデフォルトのオプションを選択してください。

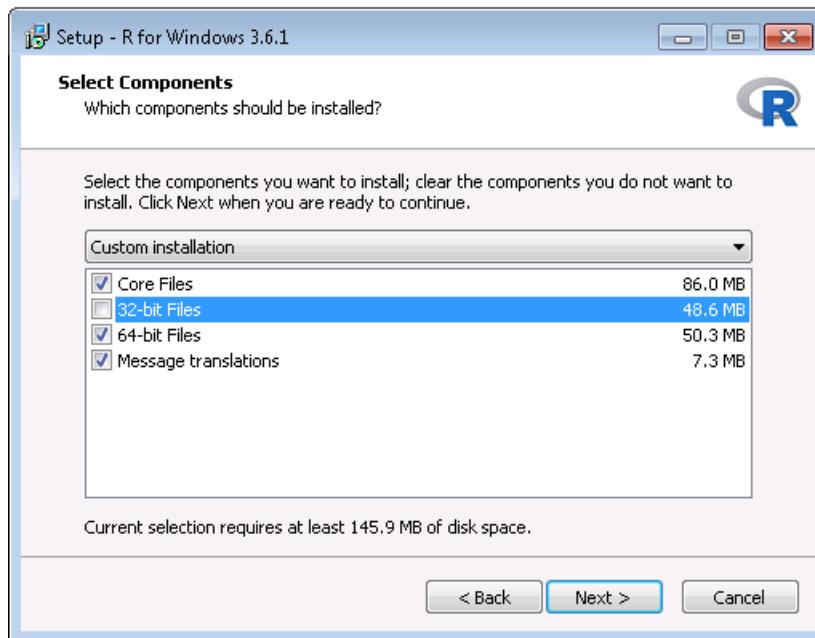


Figure 8.7: 32 ビットバージョンの R を無効化

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 のダウンロード

Java のインストール

1. <https://java.com/en/download/manual.jsp> にアクセスし、図 8.9 に示されるように、Windows64 ビット版のインストーラを選択します。32 ビット版の R もインストールしている場合には、Java も 32 ビット版をインストールしなければなりません。

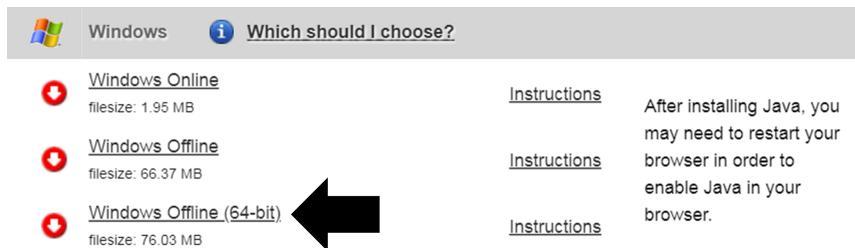


Figure 8.9: Java のダウンロード

2. ダウンロード後、インストーラを実行します。

インストールの確認

これで準備は整ったはずですが、念のため確認しておきましょう。R を起動し、下記のようにタイプしてください。

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

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

この関数は Java を使用するので、すべてがうまくいけば、R と Java の両方が正しくインストールされていることがわかります！

もう一つのテストは、ソースパッケージがビルドできるかどうかを確認することです。以下の R コードを実行して、OHDSI GitHub リポジトリから CohortMethod パッケージをインストールします：

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

8.5 展開戦略

ATLAS や Methods Library を含む OHDSI ツールスタック全体を組織内で展開することは、非常に困難な作業です。依存関係がある多くのコンポーネントを考慮し、設定を行う必要があります。このため、二つの取り組みが、スタック全体を一つのパッケージとしてインストールできる統合展開戦略を開発しました。一部の仮想化技術を使用して、これを実現します。それは、Broadsea および Amazon Web Services (AWS) です。

8.5.1 Broadsea

Broadsea⁸は Docker コンテナ技術⁹を使用しています。OHDSI ツールは依存関係とともに、Docker イメージと呼ばれる単一のポータブルなバイナリファイルにパッケージ化されています。このイメージは Docker エンジンサービス上で実行でき、すべてのソフトウェアがインストールされるとすぐに実行可能な仮想マシンが作成されます。Docker エンジンは Microsoft Windows、MacOS、Linux などのほとんどのオペレーティングシステムで利用可能です。Broadsea Docker イメージには、Methods library や ATLAS を含む主な OHDSI ツールが含まれています。

8.5.2 Amazon AWS

Amazon は、AWS クラウドコンピューティング環境でボタンをクリックするだけでインスタンス化できる二つの環境を用意しています：OHDSI-in-a-Box¹⁰ と OHDSIonAWS¹¹です。

OHDSI-in-a-Box は特に学習環境として作成されたものであり、OHDSI コミュニティが提供するほとんどのチュートリアルで使用されています。これには多くの OHDSI ツール、サンプルデータセット、RStudio、その他のサポートソフトウェアが低成本の単一の Windows 仮想マシンに含まれています。PostgreSQL データベースは、CDM の保存と、ATLAS からの中間結果の保存の両方に使用されます。OMOP CDM データマッピングと ETL ツールも、OHDSI-in-a-Box に含まれています。OHDSI-in-a-Box のアーキテクチャは、図 8.10 に示されています。

OHDSIonAWS は、企業向け、マルチユーザー対応、拡張性や耐障害性に優れた OHDSI 環境のためのリファレンスアーキテクチャであり、組織がデータ分析を行う際に使用することができます。複数のサンプルデータセットが含まれており、組織の実際のヘルスケアデータを自動的にロードすることも可能です。データは Amazon Redshift データベースプラットフォームに配置され、OHDSI ツールによってサポートされます。ATLAS の中間結果は PostgreSQL データベ

⁸<https://github.com/OHDSI/Broadsea>

⁹<https://www.docker.com/>

¹⁰<https://github.com/OHDSI/OHDSI-in-a-Box>

¹¹<https://github.com/OHDSI/OHDSIonAWS>

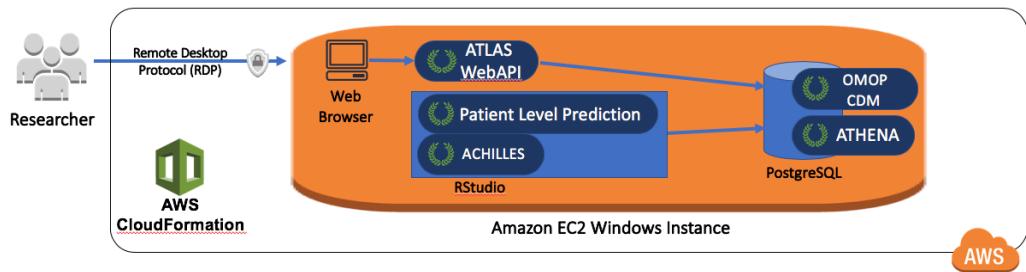


Figure 8.10: OHDSI-in-a-Box の Amazon Web Services アーキテクチャ

ースに保存されます。ユーザーはフロントエンドで、ウェブインターフェース（RStudio Server を活用）を通じて ATLAS や RStudio にアクセスできます。RStudio には OHDSI Methods Library がすでにインストールされており、データベースへの接続に使用できます。OHDSIonAWS の自動展開はオープンソースであり、組織の管理ツールやベストプラクティスを含めるようにカスタマイズできます。OHDSIonAWS のアーキテクチャは図8.11に示されています。

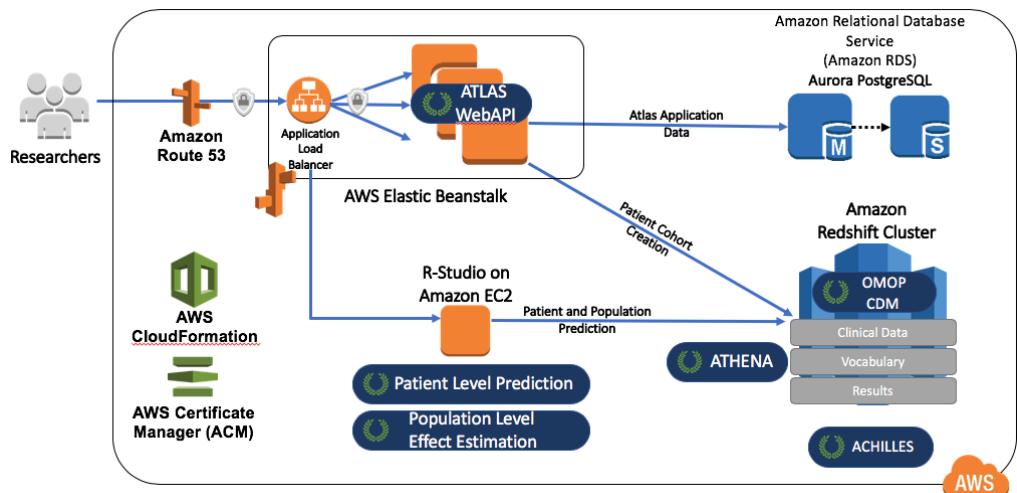


Figure 8.11: OHDSIonAWS の Amazon Web Services アーキテクチャ

8.6 まとめ



- CDM 内のデータに対して分析を行うには
 - * カスタムコードを作成する
 - * OHDSI Methods Library の R パッケージを使用したコードを作成する
 - * インタラクティブな分析プラットフォーム ATLAS を使用する

- OHDSI ツールはさまざまな分析戦略を用いています
 - * 単一研究
 - * リアルタイムクエリ
 - * 大規模アナリティクス
- OHDSI アナリティクスツールのほとんどは、以下に組み込まれています
 - * インタラクティブな分析プラットフォーム ATLAS
 - * OHDSI Methods Library の R パッケージ
- OHDSI ツールの展開を容易にするいくつかの戦略があります。

第 9 章

SQL と R

著者: Martijn Schuemie & Peter Rijnbeek

共通データモデル (CDM) はリレーションナルデータベースモデルです（すべてのデータはフィールドを持つテーブルのレコードとして表されます）。そのため、データは通常、PostgreSQL、Oracle、Microsoft SQL Server などのソフトウェアプラットフォームを使用してリレーションナルデータベースに保存されます。ATLAS や Methods Library などのさまざまな OHDSI ツールは、バックグラウンドでデータベースにクエリを出すことで動作しますが、適切なアクセス権があれば、私たちも自身も直接データベースにクエリを出すことができます。その主な理由は、現在のツールではサポートされていない分析を行うためです。ただし、OHDSI ツールは多くの場合、ユーザーがデータを適切に分析できるよう、ガイドするように設計されているため、データベースを直接クエリすると、間違いを犯すリスクも高くなります。直接のクエリでは、そのようなガイドは提供されていません。

リレーションナルデータベースをクエリする標準的な言語は SQL (Structured Query Language) で、クエリやデータ変更に使用できます。SQL の基本コマンドは確かに標準化されており、ソフトウェアプラットフォーム間で同じ意味を持ちますが、各プラットフォームには独自の「方言」があり、微妙な違いがあります。例えば、SQL Server 上の PERSON テーブルの最初の 10 行を取得するには、次のように入力します。:

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

一方、PostgreSQL では同じクエリは次のようにになります：

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

OHDSI では、プラットフォーム固有の表現に依存しないことを望んでいます。すなわち、すべての OHDSI データベースで同じ SQL 言語を使用したいと考え

ています。このため、OHDSI は SqlRender パッケージを開発しました。これは、ある標準の表現から後述するサポート対象の表現に翻訳できる R パッケージです。この標準表現である - OHDSI SQL - は主に SQL Server SQL 表現のサブセットです。本章で例示する SQL 文はすべて OHDSI SQL を使用します。

各データベースプラットフォームには、SQL を使用したデータベースのクエリのための独自のソフトウェアツールも付属しています。OHDSI では、多くのデータベースプラットフォームに接続できる R パッケージ、DatabaseConnector パッケージを開発しました。DatabaseConnector も本章の後半で説明します。

そのため、CDM に準拠したデータベースに対して OHDSI ツールを使用せずにクエリを実行できますが、推奨されるパスは DatabaseConnector と SqlRender パッケージを使用することです。これにより、あるサイトで開発されたクエリが他のサイトでも修正することなく使用できるようになります。R 自体も、データベースから抽出されたデータをさらに分析する機能を提供しており、統計分析の実行や（インタラクティブな）プロットの生成などが可能です。

本章では、読者が SQL の基本的な理解をしていることを前提としています。まず、SqlRender と DatabaseConnector の使用方法を確認します。これらのパッケージを使用しない場合は、このセクションをスキップいただいて構いません。セクション 9.3 では、CDM にクエリを出すための SQL（この場合 OHDSI SQL）を使用する方法を説明します。次のセクションでは、CDM にクエリする際に OHDSI 標準化ボキャブラリを使用する方法を説明します。CDM に対する一般的に使用されるクエリのコレクションであり、一般に公開されている QueryLibrary に焦点を当てます。本章の最後では、発生率を推定する研究例を取り上げ、SqlRender と DatabaseConnector を使用してこの研究を実施します。

9.1 SqlRender

SqlRender パッケージは CRAN (Comprehensive R Archive Network) で入手可能であり、以下のコマンドでインストールできます：

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

SqlRender は、従来のデータベースシステム (PostgreSQL、Microsoft SQL Server、SQLite、Oracle) や並列データウェアハウス (Microsoft APS、IBM Netezza、Amazon Redshift) に加え、ビッグデータプラットフォーム (Hadoop から Impala、Google BigQuery) など、幅広い技術プラットフォームをサポートしています。R パッケージには、パッケージマニュアルと、全機能を紹介するビネットが付属しています。ここでは、主な機能の一部を紹介します。

9.1.1 SQL のパラメータ設定

パッケージの機能のひとつは、SQL のパラメータ化をサポートすることです。しばしば、いくつかのパラメータに基づいて、SQL の小さなバリエーションを生成する必要があります。SqlRender は、SQL コード内にシンプルなマークアップ構文を提供し、パラメータ化を可能にします。パラメータ値に基づく SQL のレンダリングは、`render()` 関数を使用して行います。

パラメータ値の置換

◎ 文字を使用して、レンダリング時に実際のパラメータ値と置換する必要があるパラメータ名を示します。以下の例では、SQL 内で `a` という変数が SQL で言及されています。`render` 関数の呼び出しでは、このパラメータの値が定義されています：

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

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

ほとんどのデータベース管理システムが提供するパラメータ化とは異なり、テーブル名やフィールド名を値と同様に簡単にパラメータ化できることに注目ください。：

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

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

パラメータ値は、数値、文字列、ブーリアン変数、ベクトル（カンマ区切りのリストに変換される）とすることができます。：

```
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

時には、1つまたは複数のパラメータの値に基づいてコードブロックをオンまたはオフにする必要があります。これは、`{Condition} ? {if true} : {if false}` 構文を使用して行います。`condition` が `true` または 1 の場合、`if true` ブロックが使用され、それ以外の場合は `if false` ブロックが（存在する場合）表示されます。

```
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"
```

簡単な比較もサポートされています：

```
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 演算子もサポートされています：

```
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 他の SQL 表現への置換

SqlRender パッケージのもう一つの機能は、OHDSI SQL から他の SQL 表現へ変換することです。例えば：

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

```
## [1] "SELECT * FROM person LIMIT 10;"
## attr("sqlDialect")
## [1] "postgresql"
```

targetDialect パラメータには次の値が設定可能です：“oracle”，“postgresql”，“pdw”，“redshift”，“impala”，“netezza”，“bigquery”，“sqlite”，“sql server”。



SQL 関数や構文を適切に変換できる範囲には限界があります。その理由は、パッケージには限られた数の変換ルールしか実装されていないことと、一部の SQL 機能にはすべての方言に相当するものが無いことが挙げられます。これが、OHDSI SQL が独自の新しい SQL 方言として開発された主な理由です。しかし、可能な限り、車輪の再発明を避けるために SQL Server の構文に従うようにしています。

最大限の努力を尽くしても、サポートされているすべてのプラットフォーム上でエラーなく実行される OHDSI SQL を記載するには、考慮すべき点がいくつあります。以下では、これらの考慮事項について詳しく説明します。

Translate がサポートする関数と構造

これらの SQL Server 関数はテスト済であり、各表現への正確な変換が確認されています：

Table 9.1: “translate (翻訳) によりサポートされる関数と構造

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

* Oracle では特別な権限が必要です。SQLite では同等のものはありません。

同様に、多くの SQL 構文構造がサポートされています。以下は、正確に翻訳されることが確認されている非網羅的なリストです：

```
-- 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;
```

文字列の連結

文字列の連結は、SQL Server が他の言語よりも特異ではない領域の 1 つです。SQL Server では、`SELECT first_name + ' ' + last_name AS full_name` `FROM table` と書きますが、これは PostgreSQL と Oracle では `SELECT first_name || ' ' || last_name AS full_name` `FROM table` でなければなりません。SqlRender は、連結される値が文字列であるかどうかを推測しようとします。上記の例では、明示的な文字列（シングルクォーテーションで囲まれたスペース）があるため、変換は正しく行われます。しかし、クエリが `SELECT first_name + last_name AS full_name` `FROM table` であった場合、SqlRender は 2 つのフィールドが文字列であることを知る手がかりがなく、プラス記号を正しく残さないでしょう。値が文字列であることのもう一つの手がかりは、明示的な VARCHAR へのキャストです。そのため、`SELECT last_name + CAST(age AS VARCHAR(3)) AS full_name` `FROM table` も正しく翻訳されます。曖昧さを避けるために、2 つ以上の文字列を連結する場合は、`CONCAT()` 関数を使用するのが最善の方法です。

テーブルエイリアスと AS キーワード

多くの SQL 表現ではテーブルエイリアスを定義する際に AS キーワードを使用できますが、キーワードなしでも問題なく動作します。例えば、以下の SQL 文は SQL Server、PostgreSQL、Redshift などでは問題なく動作します：

```
-- 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;
```

しかし、Oracle では AS キーワードを使用するとエラーが発生します。上記の

例では、最初のクエリは失敗します。そのため、テーブルにエイリアスを付ける際には AS キーワードを使用しないことを推奨します。(注 : SqlRender では Oracle が AS の使用を許可していないテーブルエイリアスと Oracle が AS の使用を要求しているフィールドエイリアスを区別できないため、この問題に対応するのは難しいです。)

テンポテーブル

テンポテーブルは中間結果を保存するのに非常に有用であり、正しく使用するとクエリのパフォーマンスを大幅に向上させることができます。ほとんどのデータベースプラットフォームでは、テンポラリテーブルには非常に優れた特性があります：現在のユーザーのみに参照でき、セッションが終了すると自動的に削除され、書き込みアクセス権がなくても作成できます。残念ながら、Oracle ではテンポラリテーブルは基本的に恒久的なテーブルであり、唯一の違いはテーブル内のデータが現在のユーザーのみに参照されるという点です。このため、Oracle では SqlRender が以下の方法でテンポラリテーブルをエミュレートしようとします。

1. 異なるユーザーからのテーブルが競合しないように、テーブル名にランダムな文字列を追加します。
2. テンポラリテーブルが作成されるスキーマをユーザーが指定できるようにします。

例えば：

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

```
## Warning: The 'oracleTempSchema' argument is deprecated. Use 'tempEmulationSchema'  
## This warning is displayed once every 8 hours.
```

```
## [1] "SELECT * FROM temp_schema.h5xk09snchildren ;"  
## attr(,"sqlDialect")  
## [1] "oracle"
```

ユーザーは temp_schema に書き込み権限を持っている必要があります。

また、Oracle ではテーブル名の長さが 30 文字に制限されているため、テンポラリテーブル名は最大 22 文字以内である必要があります。セッション ID を追加すると名前が長くなりすぎるためです。

さらに、Oracle ではテンポラリテーブルは自動的に削除されないため、使用後に明示的にすべてのテンポラリテーブルを TRUNCATE および DROP して、孤立したテーブルが Oracle の一時スキーマに蓄積しないようにする必要があります。

暗黙の型変換

SQL Server が他の言語よりも特異である数少ない点の 1 つは、暗黙の型変換が許可されていることです。例えば、次のコードは SQL Server で動作します：

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

txt が VARCHAR フィールドで、それを整数と比較しているとしても、SQL Server は比較を可能にするために、2 つのうちの 1 つを自動的に正しい型に変換します。これに対して、PostgreSQL などの他の方言では、VARCHAR と INT を比較しようとするとエラーが発生します。したがって、キャストは常に明示的に行う必要があります。上記の例では、最後のステートメントを以下のいずれかに置き換える必要があります。:

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

または

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

文字列比較における大文字・小文字の区別

SQL Server など的一部の DBMS プラットフォームは常に大文字と小文字を区別せずに文字列比較を行いますが、PostgreSQL などの他のプラットフォームは常に大文字と小文字を区別します。そのため、常に大文字・小文字を区別する比較を行うことを前提とし、大文字・小文字の区別が不明な場合は明示的大文字・小文字の区別をしないようにすることを推奨します。例えば、次のように：

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

代わりに以下のように記述することが推奨されます：

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

スキーマとデータベース

SQL Server では、テーブルはスキーマ内にあり、スキーマはデータベース内にあります。例えば、`cdm_data.dbo.person` は `cdm_data` データベース内の `dbo` スキーマ内の `person` テーブルを指します。他の言語でも同様の階層が存在しますが、使用方法が大きく異なります。SQL Server では、データベースごとに通常 1 つのスキーマ（`dbo` と呼ばれることが多い）が存在し、ユーザーは異なるデータベース内のデータを簡単に使用できます。他のプラットフォーム、例えば PostgreSQL では、単一セッションでデータベースをまたいだデータの使用はできませんが、データベース内に多くのスキーマが存在することがよくあります。PostgreSQL では、SQL Server のデータベースに相当するものがスキーマであると言えます。そのため、SQL Server のデータベースとスキーマを 1 つのパラメータに結合することを推奨します。通常、これを `@databaseSchema` と呼びます。例えば、パラメータ化された SQL では次のようにになります：

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

SQL Server では、値にデータベース名とスキーマ名の両方を含めることができます：`databaseSchema = "cdm_data.dbo"`。他のプラットフォームでは、同じコードを使用し、パラメータ値としてスキーマのみを指定します：`databaseSchema = "cdm_data"`。

この方法が失敗する唯一の状況は `USE` コマンドを使用した場合です。`USE cdm_data.dbo;` エラーが発生します。したがって、常にデータベース/スキーマを指定してテーブルの場所を示すようにし、`USE` コマンドの使用を避けることを推奨します。

パラメータ化された SQL のデバッグ

パラメータ化された SQL のデバッグは少し複雑になることがあります。レンダリングされた SQL のみがデータベースサーバーでテストできますが、コードの変更はパラメータ化された（レンダリング前の）SQL で行う必要があります。

ソースの SQL をインタラクティブに編集し、レンダリングおよび翻訳された SQL を生成するための Shiny アプリが `SqlRender` パッケージに含まれています。このアプリは次の方法で起動できます：

```
launchSqlRenderDeveloper()
```

これにより、図 9.1 に示すように、アプリがデフォルトのブラウザで開きます。アプリはウェブ上でも公開されています¹。

このアプリでは、OHDSI SQL を入力し、ターゲットの方言を選択し、SQL に表示されるパラメータの値を入力すると、翻訳が自動的に下部に表示されます。

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

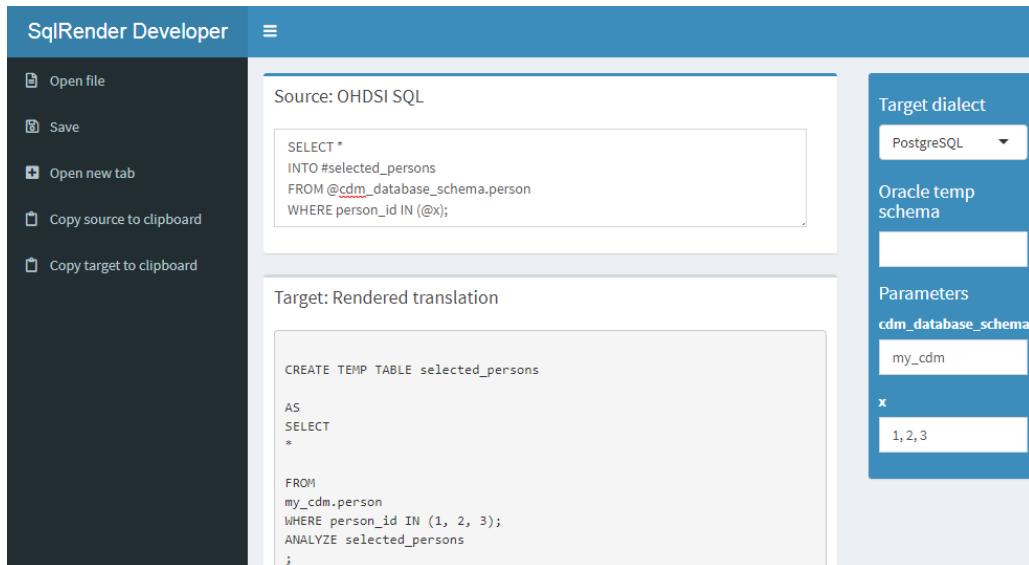


Figure 9.1: The SqlDeveloper Shiny app.

9.2 DatabaseConnector

DatabaseConnectorは、Java の JDBC ドライバを使用してさまざまなデータベースプラットフォームに接続するための R パッケージです。DatabaseConnector パッケージは CRAN (Comprehensive R Archive Network) で入手可能で、次のようにインストールできます：

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

DatabaseConnector は、従来のデータベースシステム (PostgreSQL, Microsoft SQL Server, SQLite、および Oracle)、並列データウェアハウス (Microsoft APS、IBM Netezza、Amazon)、ならびにビッグデータプラットフォーム (Hadoop を介した Impala、および Google BigQuery) など、広範な技術プラットフォームをサポートしています。このパッケージにはすでにほとんどのドライバが含まれていますが、ライセンス上の理由から BigQuery、Netezza、Impala のドライバは含まれておらず、ユーザーが入手する必要があります。これらのドライバのダウンロード方法については、?jdbcDrivers を参照ください。ダウンロード後、connect、dbConnect、createConnectionDetails 関数の pathToDriver 引数を使用できます。

9.2.1 接続の作成

データベースに接続するには、データベースプラットフォーム、サーバーの位置、ユーザー名、パスワードなど、多くの詳細を指定する必要があります。connect 関数を呼び出し、これらの詳細を直接指定することができます：

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

Connecting using PostgreSQL driver

各プラットフォームに必要な詳細情報については、?connect を参照ください。
接続を閉じたことを必ず確認ください：

```
disconnect(conn)
```

サーバー名を指定する代わりに、JDBC 接続文字列を提供することも可能です。
さらに便利な場合は、こちらを使用することもできます。:

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

Connecting using PostgreSQL driver

場合によっては、接続の詳細を先に指定し、接続を後にしたい場合もあるでしょう。例えば、接続が関数内で確立される場合、詳細を引数として渡す必要がある場合に有用です。この目的には、createConnectionDetails 関数を使用できます：

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

Connecting using PostgreSQL driver

9.2.2 クエリの実行

データベースにクエリを実行するための主な関数は、querySql と executeSql です。querySql はデータがデータベースから返されることを想定しており、一度に 1 つの SQL 文のみを処理できます。一方、executeSql はデータが返されることを想定せず、複数の SQL 文を 1 つの SQL 文字列で受け入れます。

いくつかの例を挙げます：

```
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;")
```

どちらの関数も広範なエラーレポートを提供します：サーバーによってエラーが発生した場合、エラーメッセージと問題のある SQL 文がテキストファイルに書き込まれ、デバッグが容易になります。また、executeSql 関数はデフォルトで進行状況バーを表示し、実行された SQL 文の割合を示します。それらの属性が不要な場合は、lowLevelQuerySql と lowLevelExecuteSql 関数がパッケージに用意されています。

9.2.3 ffd़f オブジェクトを使用したクエリの実行

データベースから取得するデータがメモリに収まりきらないほど大きい場合もあります。セクション8.4.2で述べたように、そのような場合には ff パッケージを使用して R データオブジェクトをファイルに保存し、メモリ上にあるかのように使用することができます。DatabaseConnector はデータを直接 ffd़f オブジェクトにダウンロードすることができます：

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

x は現在 ffd़f オブジェクトです。

9.2.4 同じ SQL を用いて異なるプラットフォームにクエリを実行する

SqlRender パッケージの render と translate 関数を最初に呼び出す便利な関数があります：renderTranslateExecuteSql、renderTranslateQuerySql、renderTranslateQuerySql.ffd़f。例えば：

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

SQL Server 固有の「TOP 10」構文は、PostgreSQL では「LIMIT 10」などに変換され、SQL パラメーター @schema は提供された値「cdm_synpuf」に置き換えられます。

9.2.5 テーブルの挿入

データをデータベースに挿入するには `executeSql` 関数を使用して SQL ステートメントを送信することも可能ですが、最適化により `insertTable` 関数を使用する方がより便利で高速です：

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

この例では、`mtcars` データフレームをサーバー上の「`mtcars`」というテーブルにアップロードします。このテーブルは自動的に作成されます。

9.3 CDM へのクエリ

以下の例では、OHDSI SQL を使用して CDM に準拠したデータベースにクエリを実行します。これらのクエリでは、CDM のデータベーススキーマを示すために `@cdm` を使用します。

まず、データベースに何人の人がいるかをクエリで取得してみましょう：

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

PERSON_COUNT
26299001

あるいは、観察期間の平均的な長なさに興味があるのかもしれません：

```
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

テーブルを結合して追加の統計を生成することができます。結合は通常、テーブル内の特定のフィールドに同じ値があることを要求することによって、複数のテーブルのフィールドを結合します。例えば、ここでは、両方のテーブルの `PERSON_ID` フィールドで、`PERSON` テーブルと `OBSERVATION_PERIOD` テーブルを結合しています。つまり、結合の結果は、2つのテーブルのすべてのフィールドを持つ新しいテーブルのようなセットですが、すべての行において、2

つのテーブルの PERSON_ID フィールドは同じ値でなければなりません。例えば、OBSERVATION_PERIOD テーブルの OBSERVATION_PERIOD_END_DATE フィールドと、PERSON テーブルの year_of_birth フィールドを組み合わせて使用することで、観察終了時の最大年齢を計算することができます。：

```
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

観察開始時の年齢分布を決定するには、はるかに複雑なクエリが必要です。このクエリでは、まず PERSON を OBSERVATION_PERIOD テーブルに結合して観察開始時の年齢を計算します。また、この結合されたセットの順序を年齢に基づいて計算し、それを order_nr として保存します。この結合の結果を複数回使用したい場合には、共通テーブル式 (CTE) として定義し (WITH ... AS を使用)、“ages” と呼びます。これにより、ages を既存のテーブルであるかのように参照することができます。ages の行数を数えて “n” を生成し、各分位数に対して、order_nr が分数の n 倍より小さい最小年齢を求めます。例えば、中央値を求めるには \$order_nr < .50 * n の最小年齢を使用します。最小年齢と最大年齢は別々に計算されます：

```
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

より複雑な計算は、SQL の代わりに R を使用して行うこともできます。例えば、同じ結果を得るために、次の R コードを使用することができます：

```

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

```

ここでは、サーバー上で年齢を計算し、すべての年齢をダウンロードし、年齢分布を計算します。しかし、これにはデータベースサーバーから数百万行ものデータをダウンロードする必要があり、効率的ではありません。計算を SQL で行うか R で行うかは、ケースバイケースで判断する必要があります。

クエリでは、CDM 内のソース値を使用することができます。例えば、最も頻度の高いコンディションのソースコードのトップ 10 を取得するには、以下を用います：

```

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

ここでは、CONDITION_OCCURRENCE テーブル内の CONDITION_SOURCE_VALUE フィールドの値でレコードをグループ化し、各グループのレコード数をカウントしました。CONDITION_SOURCE_VALUE とそのカウントを取得し、カウントで逆順で並べ替えていきます。

9.4 クエリ実行時にボキャブラリを使用する

多くの操作では、ボキャブラリが有用です。ボキャブラリテーブルは CDM の一部であり、SQL クエリを使用して利用できます。ここでは、ボキャブラリに対するクエリを CDM に対するクエリと組み合わせる方法を示します。CDM の多くのフィールドにはコンセプト ID が含まれていますが、これらは CONCEPT テーブルを使用して解決できます。例えば、データベース内的人数を性別で階層化してカウントしたい場合、GENDER_CONCEPT_ID フィールドをコンセプト名に解決すると便利です：

```
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

SUBJECT_COUNT	CONCEPT_NAME

ボキャブラリの非常に強力な機能の一つは、その階層構造です。よくあるクエリは、特定のコンセプトと そのすべての下位層を探すものです。例えば、イブプロフェンという成分を含む処方件数を数えるとします：

```
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

QueryLibrary は、CDM 用の一般に使用される SQL クエリのライブラリです。これはオンラインアプリケーション²として提供されており、図9.2に示すように、R パッケージとしても利用できます³。

このライブラリの目的は、新しいユーザーが CDM のクエリ方法を学習するのを支援することです。ライブラリ内のクエリは、OHDSI コミュニティによって審査され、承認されています。クエリライブラリは主にトレーニング目的で使用されますが、経験豊富なユーザーにとっても貴重なリソースとなります。

QueryLibrary は、SqlRender を利用して、選択した SQL 方言でクエリを実行します。ユーザーは CDM のデータベーススキーマ、ボキャブラリデータベーススキーマ（別々のものがある場合）、Oracle テンポラリスキーマ（必要な場合）を指定することもでき、これらの設定でクエリが自動的にレンダリングされます。

²<http://data.ohdsi.org/QueryLibrary>

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

Select a query

Group	Name
["drug exposure"]	All drug
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='rxnev'

```

Figure 9.2: クエリライブラリ：CDMに対するSQLクエリのライブラリ。

9.6 簡単な研究のデザイン

9.6.1 問題の定義

血管性浮腫は、ACE 阻害薬（ACEi）のよく知られた副作用です。Slater et al. (1988)によると、ACEi 治療開始後 1 週間の血管性浮腫の発症率は 3,000 人中 1 例/週と推定されています。ここでは、この結果を再現し、年齢と性別によって層別化します。単純化するため、ACEi の一つである（リシノプリル）に焦点を当てます。したがって、次の問い合わせに答えます：

リシノプリル投与開始後の最初の 1 週間での血管性浮腫の発生率は、年齢と性別で層別化するとどの程度でしょうか？

9.6.2 曝露

曝露をリシノプリルへの初回の曝露として定義します。初回とは、以前にリシノプリルへの曝露がないことを意味します。初回の曝露の前に 365 日間の連続した観察期間が必要となります。

9.6.3 アウトカム

血管性浮腫は、入院中または救急室ビジット中に血管性浮腫の診断コードが記録された場合と定義します。

9.6.4 リスク期間

治療開始後の最初の 1 週間の発症率を計算します。患者が 1 週間にわたって継続的に曝露されたかどうかは問いません。

9.7 SQL と R を使用した研究の実施

OHDSI ツールの慣例に縛られることはありませんが、同じ原則に従うことは有益です。この場合、OHDSI ツールが動作するのと同様に、SQL を用いてコホートテーブルを作成します。COHORT テーブルは CDM に定義されており、使用する事前定義されたフィールドセットもあります。まず、書き込み権限のあるデータベーススキーマに COHORT テーブルを作成する必要がありますが、これは CDM 形式でデータを保持しているスキーマとは異なるスキーマである可能性が高いです。

```
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)
```

ここでは、データベーススキーマとテーブル名をパラメータ化しています。異なる環境に簡単に適応させることができます。その結果、データベースサーバー上に空のテーブルが作成されます。

9.7.1 曝露コホート

次に、曝露コホートを作成し、COHORT テーブルに挿入します：

```
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 -- リシノプリル
    ) 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)
```

ここでは、CDM の標準テーブルである DRUG_ERAS テーブルを使用します。このテーブルは DRUG_EXPOSURE テーブルから自動的に派生するものです。DRUG_ERAS テーブルには成分レベルでの継続的な曝露期間が含まれるため、リ

シノプリルを検索すると、自動的にリシノプリルを含む薬剤への曝露がすべて特定されます。次に、OBSERVATION_PERIOD テーブルに結合し、1 人当たりの最初の薬物曝露を取り出します。1 人の患者が複数の観察期間を持つ可能性があるため、薬物曝露を含む期間のみに結合する必要があります。また、OBSERVATION_PERIOD_START_DATE と COHORT_START_DATE の間には、少なくとも 365 日の間隔が必要となります。

9.7.2 アウトカムコホート

最後に、アウトカムコホートを作成する必要があります：

```
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 -- 血管浮腫
) 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) -- 入院または ER;
"
renderTranslateExecuteSql(conn, sql,
    cohort_db_schema = cohortDbSchema,
    cohort_table = cohortTable,
    cdm_db_schema = cdmDbSchema)
```

ここでは、CONDITION_OCCURRENCE テーブルを CONCEPT_ANCESTOR テーブルと結合して、血管性浮腫またはその下位層に含まれるすべての出現を見つけます。同じ日に複数の診断がある場合、それは同じ発生である可能性が高いため、各日 1 件のレコードのみを取得するように DISTINCT を使用します。

次に、診断が入院または ER で行われたことを確認するために、これらの発生を VISIT_OCCURRENCE テーブルと結合します。

9.7.3 発症率の計算

コホートが設定されたので、年齢と性別で層別化された発症率を計算できます：

```
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 -- 曝露
)
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 -- 結果
  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)

```

まず、CTE 「tar」 を作成し、適切なリスク時間を伴うすべての曝露を含めます。OBSERVATION_PERIOD_END_DATE でリスク期間が切り捨てられることに留意ください。また、10 年ごとの年齢階層を計算し、性別を特定します。CTE を使用する利点は、クエリ中に同じ中間結果セットを複数回使用できることです。このユースケースでは、リスク期間の合計およびリスク期間中に発生する血管性浮腫のイベント数を数えるために使用します。

`snakeCaseToCamelCase = TRUE` を用いるのは、SQL ではフィールド名に `snake_case` を使用する傾向がある（SQL は大文字と小文字を区別しないため）のに対し、R では `camelCase` を使用する傾向がある（R は大文字・小文字を区別するため）からです。`results` データフレームの列名は `camelCase` になります。

ggplot2 パッケージを使用すると、結果を簡単にプロットできます：

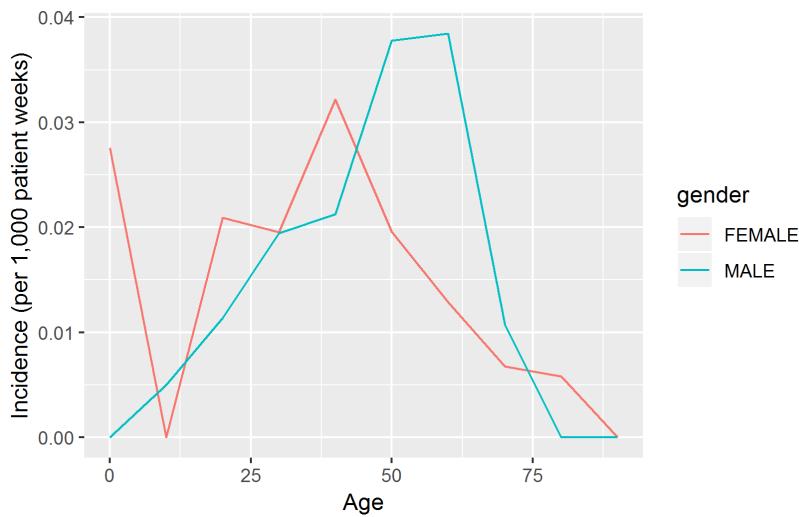
```

# 発症率 (IR) を算出
results$ir <- 1000 * results$events / results$days / 7

# 年齢スケールを修正
results$age <- results$age * 10

library(ggplot2)
ggplot(results, aes(x = age, y = ir, group = gender, color = gender)) +
  geom_line() +
  xlab("年齢") +
  ylab("発症率 (1,000 患者/週)")

```



9.7.4 クリーンアップ

作成したテーブルをクリーンアップし、忘れずに接続を閉じます：

```
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 互換性

OHDSI SQL と DatabaseConnector と SqlRender を組み合わせて使用するため、ここで紹介したコードは OHDSI がサポートする任意のデータベースプラットフォームで実行できます。

デモンストレーション用に、手作業で SQL を使用してコホートを作成することにしましたが、ATLAS でコホート定義を構築し、ATLAS で生成された SQL を使用してコホートをインスタンス化する方が便利です。ATLAS も OHDSI SQL を生成するため、SqlRender と DatabaseConnector と簡単に併用することができます。

9.8 まとめ



- SQL (Structured Query Language) は、共通データモデル (CDM) に準拠したデータベースを含む、データベースに照会するための標準言語です。
- 異なるデータベースプラットフォームは異なる SQL 表現を持っており、照会するためには異なるツールが必要です。
- SqlRender と DatabaseConnectorR パッケージは、CDM 内のデータを照会するための統一された方法を提供し、同じ分析コードを修正することなく異なる環境で実行できるようにします。
- R と SQL を併用することで、OHDSI ツールではサポートされていないカスタム分析を実装できます。
- QueryLibrary は、CDM 用の再利用可能な SQL クエリのコレクションを提供します。

9.9 演習

前提条件

これらの演習では、セクション 8.4.5 に記載されているように、R、R-Studio、Java がインストールされていることを前提とします。また、SqlRender、DatabaseConnector、および Eunomia パッケージも必要です。以下の手順でインストールできます。

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

Eunomia パッケージは、CDM 内でローカル R セッション内で動作するシミュレートされたデータセットを提供します。接続の詳細は以下の方法で取得できます。

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

CDM データベースのスキーマは「main」です。

演習 9.1. SQL と R を使用して、データベース内に何人いるかを計算します。

演習 9.2. SQL と R を使用して、セレコキシブの処方を少なくとも 1 回受けたことがある人の人数を計算します。

演習 9.3. SQL と R を使用して、セレコキシブの服用中に消化管出血と診断された人の人数を計算します。(ヒント: 消化管出血のコンセプト ID は 192671 です)

推奨される解答は付録 E.5 を参照ください。

第 10 章

コホートの定義

著者: Kristin Kostka

観察型健康データ（リアルワールドデータとも呼ばれる）は、患者の健康状態や医療の提供に関連するデータで、さまざまな情報源から日常的に収集されたデータです。そのため、OHDSI データスチュワード (OHDSI の共同研究者のうち、各サイトのデータを CDM で維持している人) は、電子的健康記録 (EHR)、医療保険請求データ、製品や疾患のレジストリ、在宅環境下を含む患者が生成したデータ、モバイル機器など健康状態に関する情報を提供できるその他の情報源など、複数の情報源からデータを取得することができます。これらのデータは研究目的で収集されたものではないため、対象としたい臨床データ要素を明確に捉えられていない可能性があります。

例えば、医療保険請求データベースは、ある症状（例：血管性浮腫）に対して提供されたすべての医療を捕捉し、関連費用が適切に払い戻されることを目的として設計されており、実際の症状に関する情報はこの目的の一部としてのみ捕捉されます。このような観察データを研究目的で使用したい場合、データに記録されているものを使用して本当に興味のあることを推測するロジックを作成する必要があります。つまり、臨床イベントがどのように発現するかを定義してコホートを作成する必要があるのです。たとえば、保険請求データベースで血管性浮腫イベントを特定したい場合、過去の血管性浮腫の発生に対する経過観察を単に説明する請求とは区別するために、血管性浮腫の診断コードが救急外来の設定で記録されていることを必要とするロジックを定義することが考えられます。同様の考慮事項は、EHR に記録された日常診療中に収集されたデータにも適用される可能性があります。データが二次的な目的で使用されるため、各データベースが本来何を目的として設計されたかを認識する必要があります。研究をデザインするたびに、さまざまな医療環境においてコホートがどのように存在するのか、細かな点を十分に考慮する必要があります。

本章では、コホート定義の作成と共有とは何か、コホートを開発する方法、ATLAS または SQL を使用して独自のコホートを構築する方法を説明します。

10.1 コホートとは？

OHDSI 研究では、コホートを、ある一定期間に 1 つ以上の適格基準を満たす人々の集合体と定義しています。この用語は表現型という用語とよく置き換えられます。コホートは、OHDSI 分析ツールやネットワーク研究全体で研究課題を実行するための主要な構成要素として使用されます。例えば、ACE 阻害薬の開始による血管性浮腫のリスクを予測することを目的とした研究では、2 つのコホートを定義します：アウトカムコホート（血管性浮腫）と対象コホート（ACE 阻害薬の使用を開始する人々）。OHDSI におけるコホートの重要な特徴は、通常、研究内の他のコホートから独立して定義されるため、再利用が可能であることです。たとえば、血管性浮腫のコホートは対象集団以外も含め、その集団全体のすべての血管性浮腫イベントを特定します。分析時に必要に応じてこれらの二つのコホートの共通部分を抽出します。この利点は、同じ血管性浮腫のコホート定義が、たとえば ACE 阻害薬と他の曝露を比較する推定研究など、他の分析でも使用できるということです。コホート定義は、研究課題に応じて異なることがあります。



コホートは、ある一定期間に 1 つ以上の適格基準を満たす人々の集合体です。

OHDSI で使用されているコホートの定義は、この分野の他の人々が使用するものとは異なるかもしれないことを認識することが重要です。例えば、多くの査読付きの科学論文では、コホートが特定の臨床コードのコードセット（例：ICD-9/ICD-10、NDC、HCPCS など）に類似しているとされています。コードセットはコホートを組み立てる際の重要な要素ですが、コホートはコードセットによって定義されるわけではありません。コホートは、条件を満たすコードセットの使用方法に関して具体的なロジックが必要となります（例：これは ICD-9/ICD-10 コードの初回の発生か？それとも発生すべてか？）。明確に定義されたコホートでは、患者がコホートに組み入れられる方法とコホートから離脱する方法が指定されています。

OHDSI のコホート定義を利用するためのユニークなニュアンスには以下があります。

- 一人の人が複数のコホートに属する可能性があります
- 一人の患者が複数の異なる期間に同じコホートに属する可能性があります。
- 一人の患者が同じ期間内に同じコホートに複数回属することはありません。
- コホートにメンバーがゼロまたは複数含まれる場合があります。

コホートを構築するための主なアプローチは二つあります：

1. ルールベースのコホート定義は、患者がコホートにいる時期を明示的なルールで説明します。これらのルールの定義は、通常、コホートをデザインする人の専門分野の知識に大きく依存し、関心のある治療分野の知識を

活用してコホートに含める基準となるルールを構築します。

2. 確率ベースのコホート定義は、患者がコホートに属する患者の確率（0から100%の間の確率）を計算する確率モデルを使用します。この確率は、ある閾値を用いてイエス・ノーの分類に変換するか、または研究デザインによってはそのまま使用できます。確率的モデルは、予測に役立つ関連する患者特性を自動的に特定するために、いくつかのサンプルデータを使用して機械学習（例えばロジスティック回帰）で通常は訓練されます。

次のセクションでは、これらのアプローチについて詳しく説明します。

10.2 ルールベースのコホート定義

ルールベースのコホート定義は、特定の期間（例：「過去6ヶ月以内にその状態を発症した人」につながるまたは、複数の明確な適格基準（例：「血管性浮腫のある人」）を明示することから始まります）。

これらの基準を構成する際に使用する標準的な構成要素は以下の通りです：

- ドメイン：データが格納されている CDM ドメイン（例：「処置（プロセッサー）の発生」、「薬剤曝露」）は、臨床情報の種類とその CDM テーブル内に表現可能なコンセプトを定義します。ドメインについては、セクション4.2.4で詳しく説明されています。
- コンセプトセット：対象とする臨床実態を包含する一つ以上の標準コンセプトを定義するデータに依存しない表現です。これらのコンセプトセットは、異なる観察健康データ間で相互運用可能であり、ボキャブラリ内の標準用語にマッピングされる臨床実態を表します。コンセプトセットについては、セクション10.3で詳しく説明されています。
- ドメイン固有の属性：関心のある臨床実態に関連する追加の属性（例：DRUG_EXPOSURE の DAYS_SUPPLY や MEASUREMENT の VALUE_AS_NUMBER や RANGE_HIGH）。
- 時間的なロジック：適格基準とイベントの関係が評価される時間間隔（例：指定された状態は曝露開始前の365日間もしくは曝露開始日に発生する必要があります）。

コホート定義を構築する際、コホート属性を表すビルディングブロックのようにドメインを考えると便利です（図10.1参照）。各ドメインで許容される内容について疑問がある場合は、共通データモデルの章（チャプター4）を参照ください。

コホート定義作成時に自問すべきいくつかの質問があります：

- コホート組入れの時間を定義する初期イベントは何か？
- 初期イベントに適用される適格基準は何か？
- コホート離脱を定義するものは何か？



Figure 10.1: コホート定義のビルディングブロック

コホート組入れイベント：コホート組入れイベント（初期イベント）は、人々がコホートに参加する時点、つまりコホートインデックス日を定義します。コホート組入れイベントは、薬剤曝露、コンディション、処置、測定値、受診期間など、CDMで記録された任意のイベントがあり得ます。初期イベントは、データが保存されているCDMドメイン（例：PROCEDURE_OCCURRENCE、DRUG_EXPOSUREなど）、臨床活動を特定するために構築されたコンセプトセット（例：SNOMEDコードによるコンディション、RxNormコードによる薬剤）、その他の特定の属性（例：発生時の年齢、最初の診断/処置/その他、開始日と終了日の指定、受診期間または基準の指定、処方日数など）によって定義されます。組入れイベントを持つ人々のセットは初期イベントコホートと呼ばれます。

適格基準：適格基準は、初期イベントコホートに適用され、さらに入々のセットを制限します。各適格基準は、データが保存されているCDMドメイン、臨床活動を表すコンセプトセット、ドメイン固有の属性（例：処方日数、受診期間など）、コホートインデックス日に関連する時間的なロジックによって定義されます。各適格基準は、初期イベントコホートから離脱する人への影響を評価するために使用されます。適格コホートは、すべての適格基準を満たす初期イベントコホート内のすべての人々として定義されます。

コホート離脱基準：コホート離脱イベントは、ある人がコホートメンバーとしての資格を失うことを示します。コホート離脱は、観察期間の終了、最初の組み入れイベントからの固定時間の間隔、一連の関連する観察の最後のイベント（例：持続的な薬剤曝露）、または観察期間の他の打ち切りによって定義される場合があります。コホート離脱戦略は、ある人が異なる時間間隔で複数回コホートに属することができるかどうかに影響を与えます。



OHDSI ツールでは、適格基準と除外基準の区別はありません。すべての基準は適格基準として形式化されます。例えば、「高血圧の既往ある人を除外する」という除外基準は、「以前の高血圧の発生が 0 回の人を含む」という選択基準として策定することができます。

10.3 コンセプトセット

コンセプトセットは、さまざまな分析で再利用可能なコンポーネントとして使用できるコンセプトのリストを表す表現です。これは、観察研究でよく使用されるコードリストの標準化されたコンピュータ実行可能な同等物と考えることができます。コンセプトセットの表現は、次の属性を持つコンセプトのリストで構成されます：

- ・除外：このコンセプト（および選択されている場合はその下位層に含まれるもの）をコンセプトセットから除外します。
- ・下位層に含まれる：このコンセプトだけでなく、その下位層に含まれるものも考慮します。
- ・マッピング済み：標準化されていないコンセプトの検索を許可します。

例えば、コンセプトセットの表現は、図に示されるように 2 つのコンセプトを含むことができます（表 10.1）。ここでは、コンセプト 4329847（「心筋梗塞」）とそのすべての下位層に含まれるコンセプトを含みますが、コンセプト 314666（「陳旧性心筋梗塞」）とそのすべての下位層を除外します。

Table 10.1: コンセプトセットの表現の例

コンセプト ID	コンセプト名	除外	下位層	マッピング対象
4329847	心筋梗塞	いいえ	はい	いいえ
314666	陳旧性心筋梗塞	はい	はい	いいえ

図に示すように（図10.2）、これは「心筋梗塞」およびその下位層に含まれるものと「陳旧性心筋梗塞」およびその下位層に含まれるものは除外します。全体で、このコンセプトセットの表現は、ほぼ 100 の標準コンセプトを意味します。これらの標準コンセプトは、さまざまなデータベースに表示されるかもしれない何百ものソースコード（例：ICD-9 および ICD-10 コード）を反映します。

10.4 確率的コホート定義

ルールベースのコホート定義は、コホート定義を組み立てるための一般的な方法です。しかし、研究コホートを作成するために必要な専門家の合意を取りま

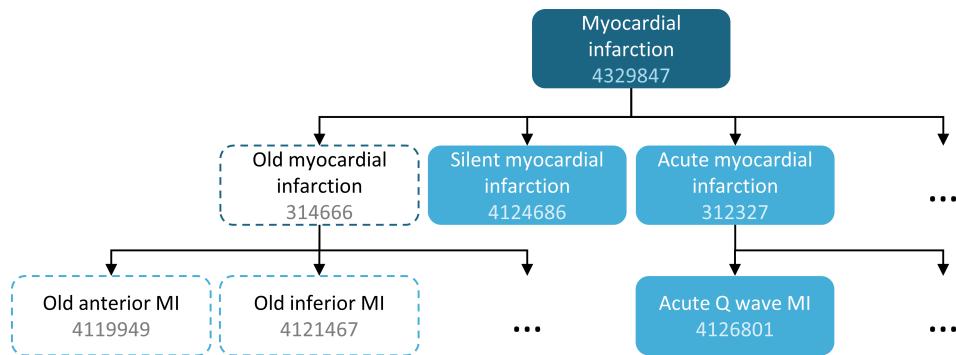


Figure 10.2: 「心筋梗塞」（下位層を含む）を含むが、「陳旧性心筋梗塞」（下位層を含む）を除外するコンセプトセット

とめるには非常に時間がかかります。確率的コホートの設計は、コホート属性の選択を迅速化する、機械駆動型の代替手法です。このアプローチでは、教師あり機械学習により、コホート構成に寄与する属性を、ラベル付けされた一連の事例（症例）から学習する表現型アルゴリズムが使用されます。このアルゴリズムは、表現型の定義的特徴をより正確に把握し、表現型の基準を変更した場合に研究全体の正確性にどのようなトレードオフが生じるかを把握するためには使用することができます。

このアプローチを CDM データに適用した例として、APHRODITE (Automated PHenotype Routine for Observational Definition, Identification, Training and Evaluation) R パッケージ¹があります。このパッケージは、不完全にラベル付けされたデータから学習する能力を組み合わせたコホート構築フレームワークを提供します (Banda et al., 2017)。

10.5 コホート定義の妥当性

コホートを構築する際、次のどちらが重要かを考慮すべきです：適格患者をすべて見つけることが重要か？それとも 確信を持てる患者のみを組み入れることが重要か？

コホートの構築戦略は、専門家の合意が疾患をどのように定義するかという臨床的な厳格性に依存します。つまり、適切なコホート設計は答えを求めている問いに依存することです。すべてを組み入れるコホート定義を選択するか、OHDSI サイト全体で共有できる最小公約数を用いるのか、またはその両者の妥協案を選ぶのか。最終的には、研究者の判断により、対象コホートの適切な研究に必要な厳格性の閾値が決まります。

本章の冒頭で述べたように、コホート定義は記録されたデータから観察したいことを推測しようとする試みです。この試みがどの程度うまくいったかという

¹<https://github.com/OHDSI/Aphrodite>

疑問が生じます。一般に、ルールに基づくコホート定義や確率的アルゴリズムの検証は、提案されたコホートを「ゴールドスタンダード」の参照形式（例：ケースの手動チャートレビュー）と比較するテストと考えることができます。詳細は第 16（「臨床的妥当性」）で詳しく説明しています。

10.5.1 OHDSI ゴールドスタンダードフェノタイプライブラリ

既存のコホート定義とアルゴリズムのインベントリーと全体的な評価を支援するために、OHDSI ゴールドスタンダードフェノタイプライブラリ (GSPL) ワークグループが設立されました。GSPL ワークグループの目的は、ルールベースと確率的手法から、コミュニティが支援するフェノタイプライブラリを開発することです。GSPL により、OHDSI コミュニティのメンバーは、コミュニティによって妥当性が確認されたコホート定義を研究やその他の活動で検索、評価、利用できるようになります。これらの「ゴールドスタンダード」定義は特定の設計や評価基準を満たすエントリが格納されたライブラリに保存されます。GSPL に関する追加情報は OHDSI ワークグループページを参照ください²。このワークグループの研究には、前のセクションで議論された APHRODITE (Banda et al., 2017) や PheValuator ツール (Swerdel et al., 2019) の他、OHDSI ネットワーク全体での電子カルテとゲノミクスの eMERGE Phenotype Library を共有するための取り組みなどが含まれます (Hripcsak et al., 2019)。表現型のキュレーションに関心がある場合は、このワークグループへの貢献を検討ください。

10.6 高血圧のコホート定義

コホート定義をルールベースのアプローチでまとめることによって、コホートスキルの練習を始めます。この例では、高血圧の第一選択治療として ACE 阻害薬の単剤療法を開始した患者を見つけたいと考えます。

このコンテキストを念頭に、コホートを構築します。この演習を通して、標準的な脱落チャートに似た方法でコホートを構築します。図 10.3 は、このコホートをどのように構築するかの論理的なフレームワークを示しています。

コホートは ATLAS のユーザーインターフェースで作成することも、CDM に対して直接クエリを書くこともできます。本章では、これらの両方について簡単に説明します。

10.7 ATLAS を用いたコホートの実装

まず ATLAS で始めるには、 Cohort Definitions モジュールをクリックします。モジュールを読み込み、「New cohort」をクリックします。次の画面では空のコホート定義が表示されます。図10.4に示す内容が画面に表示されます。

²<https://www.ohdsi.org/web/wiki/doku.php?id=projects:workgroups:gold-library-wg>

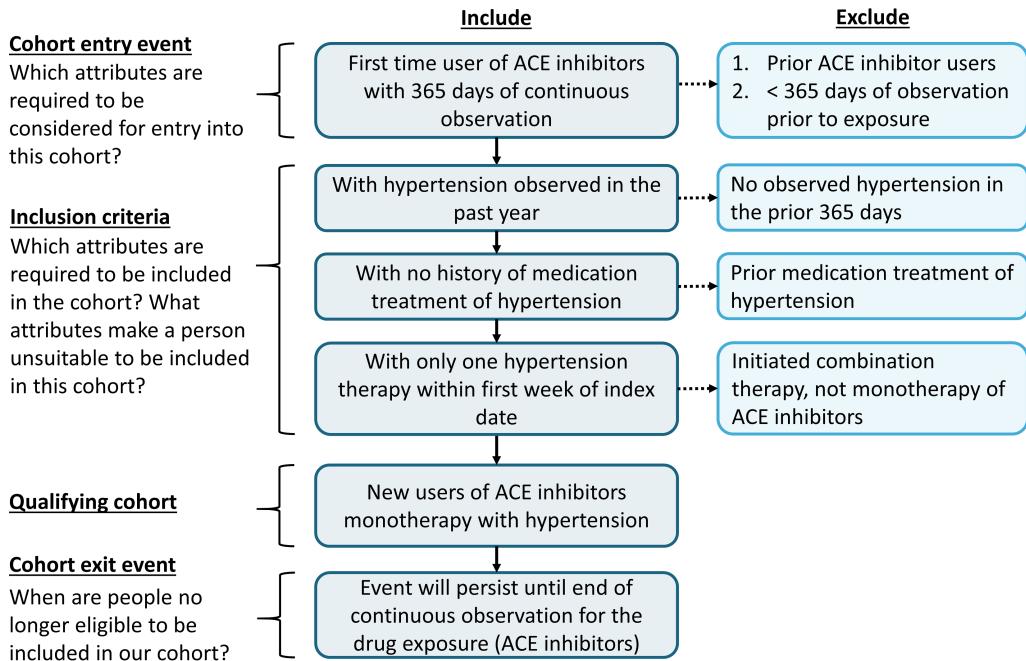


Figure 10.3: 目標とするコホートの論理図

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.

Inclusion Criteria

New inclusion criteria

Limit qualifying events to: earliest event per person.

Figure 10.4: 新しいコホート定義

まず最初に、「New Cohort Definition」からコホートの名前を固有の名前に変更することをお勧めします。「高血圧に対する第一選択単剤療法としてのACE阻害薬の新規ユーザー」のような名前を付けることができます。



ATLAS は二つのコホートに全く同じ名前を付けることはできません。他の ATLAS コホートで既に使われている名前を選択した場合、ATLAS はpopup アップエラーメッセージを表示します。

名前を入力後、をクリックしてコホートを保存します。

10.7.1 初期イベント基準

では、初期コホートイベントの定義に進みます。“Add initial event (初期イベントを追加)”をクリックします。どのドメインに基づいて基準を設定するかを選ばなければなりません。「どのドメインが初期コホートイベントかどうかをどのように知るのか?」という疑問を抱くかも知れません。これを解決しましょう。

The screenshot shows the ATLAS Cohort Definition interface. 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".

Cohort Entry Events

Events having any of the following criteria:

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

Limit initial events to: per person.

Inclusion Criteria

New inclusion criteria

Limit qualifying events to: per person.

Cohort Exit

On the right side, there is a sidebar with a list of event types:

- Add Initial Event
- Add Condition Era
- Add Condition Occurrence
- Add Death
- Add Device Exposure
- Add Dose Era
- Add Drug Era
- Add Drug Exposure

Figure 10.5: 初期イベントの追加

図10.5に示されているように、ATLAS は各基準の説明を提供しています。もし CONDITION_OCCURRENCE に基づいた基準を構築している場合、特定の診断を持つ患者を探しているということになります。DRUG_EXPOSURE に基づいた基準を構築している場合、特定の薬剤または薬剤クラスを持つ患者を探しているということになります。高血圧に対する第一選択治療として ACE 阻害薬 単剤療法を開始する患者を見つけるため、DRUG_EXPOSURE 基準を選択しま

す。「しかし、高血圧としての診断も重要では？」と思うかも知れません。その通りです。高血圧は構築する別の基準です。しかし、コホートの開始日は ACE 阻害薬治療の開始によって定義されるため、それが初期イベントです。高血圧の診断は、追加の適格基準と呼ばれるものです。この基準を構築した後で再度この問題に戻ります。“Add Drug Exposure (薬剤曝露を追加)” をクリックします。

画面は選択した基準を表示するように更新されますが、まだ終了ではありません。図10.6を参照すると、ATLAS はどの薬剤を探しているのか、認識していません。ATLAS に ACE 阻害薬に関するコンセプトセットを伝える必要があります。

The screenshot shows the 'Cohort Entry Events' interface. At the top, there's a blue header bar with the title and a question mark icon. Below it is a search bar with placeholder text 'Events having any of the following criteria:' and a 'Add Initial Event' button. The main area contains a search input 'a drug exposure of Any Drug' with dropdown arrows, an 'Add attribute...' button, and a 'Delete Criteria' button. Below this, there are two input fields: 'with continuous observation of at least [0] days before and [0] days after event index date' and 'Limit initial events to: earliest event per person'. At the bottom left is a green 'Restrict initial events' button.

Figure 10.6: 薬剤曝露の定義

10.7.2 コンセプトセットの定義

コンセプトセットを定義するためには、をクリックして、ACE 阻害薬を定義するためのコンセプトセットを取得するダイアログボックスを開きます。

シナリオ 1: コンセプトセットを構築していない場合

基準に適用するコンセプトセットをまだ作成していない場合は、先にそれを行う必要があります。コホート定義内で“Concept set (コンセプトセット)”タブに移動し、“New Concept Set (新規コンセプトセット)”をクリックしてコンセプトセットを作成することができます。“Unnamed Concept Set (命名されていないコンセプトセット)”から任意の名前に変更する必要があります。そこから、**Search** モジュールを使用して ACE 阻害薬を表す臨床コンセプトを検索できます（図10.7を参照）。

使用したいボキャブラリを見つけたら、をクリックし、そのコンセプトを選択します。図10.7の左矢印を使用してコホート定義に戻ります対象とする臨床コンセプトを検索するためのボキャブラリのナビゲーション奉納は、第5章（標準化ボキャブラリ）を参照ください。

図10.8はコンセプトセット表現を示しています。対象とするすべての ACE 阻害薬成分を選択し、その下位層すべてを含め、これらの成分を含むすべての薬

The screenshot shows the ATLAS search interface. At the top, there's a navigation bar with a back arrow, the text 'EXAMPLE: new users of ACE inhibitors as first-line mono-therapy for hypertension', and a right arrow pointing to 'ACE Inhibitors'. Below this is a search bar with a magnifying glass icon and the text 'Search' and 'Import'. The main area has a search input field containing 'ace inhibitors' and a blue search button. There are also 'Column visibility', 'Copy', and 'CSV' buttons, along with a 'Show 15 entries' dropdown. A 'Filter:' input field is present with a placeholder 'Previous [1] Next'. On the left, there's a sidebar with a tree view under 'Vocabulary' and 'Class'. The 'Vocabulary' section shows 'ATC (6)', 'Multilex (1)', 'VA Class (1)', and 'LOINC (1)'. The 'Class' section shows 'ATC 4th (4)'. The main table lists 9 entries, each with columns for Id, Code, Name, Class, RC, DRC, Domain, and Vocabulary. The first entry is 'ACE inhibitors, plain' (Id: 21601784, Code: C09AA). The last entry is 'ACE INHIBITORS, COMBINATIONS' (Id: 21601801, Code: C09B).

	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: 語彙の検索 - ACE 阻害薬

剤を含めています。” “Included concepts (包含されるコンセプト)” をクリックして、この表現に含まれている 21,536 のコンセプトすべてを確認することができ、“Included Source Codes (包含されるソースコード)” をクリックすると、様々なコーディングシステムに含まれるすべてのソースコードを探索することができます。

シナリオ 2: すでにコンセプトセットを構築している場合

すでにコンセプトセットを作成して ATLAS に保存している場合、“Import Concept Set (コンセプトセットをインポート)” をクリックします。ダイアログボックスが開き、ATLAS のコンセプトセットリポジトリからコンセプトを検索するように促されます（図10.9を参照）。図の例ではユーザーは ATLAS に保存されているコンセプトセットを取得しています。右側の検索に “ace inhibitors” と入力し、コンセプトセットのリストがマッチングする名前のコンセプトのみに絞り込まれます。そこからユーザーはコンセプトセットの行をクリックして選択します。（注：コンセプトセットを選択するとダイアログボックスは消えます。） “Any Drug” ボックスが選択したコンセプトセットの名前に更新されると、この操作が成功したことがわかります。

10.7.3 初期イベントの追加基準

コンセプトセットを添付したら、作業終了ではありません。問い合わせでは、新規ユーザーまたは ACE 阻害薬に初めて曝露したユーザーを探しています。これは、患者の記録における ACE 阻害薬の最初の曝露に相当します。これを指定するには、“+Add attribute (属性を追加)” をクリックします。次に “Add first exposure criteria (最初の曝露の基準を追加)” を選択します。作成する基準の他の属性を指定できることに注意ください。発症時の年齢、発症日、性別、または薬剤に関連するその他の属性を指定できます。選択可能な基準は、各ドメ

Concept Set Expression Included Concepts 21536 Included Source Codes Export Import

Name: ACE Inhibitors

Show 25 entries Search: Previous 1 Next

Showing 1 to 15 of 15 entries

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

Classification Non-Standard Standard

Figure 10.8: ACE 阻害薬を含むコンセプトセット

Import Concept Set From Repository...

New Concept Set

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) Previous 1 Next

Figure 10.9: ATLAS リポジトリからのコンセプトセットのインポート

インで異なります。選択したら、ウィンドウは自動的に閉じます。この追加属性は最初の基準と同じボックスに表示されます（図10.10を参照）。



現在の ATLAS のデザインは一部の人を混乱させるかもしれません。見た目通り、**X**は「No」を意味するものではありません。これは、基準を削除できる機能です。もし**X**をクリックすると、この基準は削除されます。したがって、基準を有効に保つには**X**を残しておく必要があります。

これで最初の適格イベントが構築できました。最初に観察された薬剤曝露を確実に捕捉するため、見落しがないことを知るために、ルックバックウィンドウを追加する必要があります。観察期間の短い患者は把握していない曝露を別の場所で受けている可能性もあります。これを制御することはできませんが、インデックス日付の前に患者がデータに存在していなければならぬ最低期間を規定することはできます。連続観察のドロップダウンを調整することで、これを実行できます。また、ボックスをクリックして、これらのウィンドウに値を入力することもできます。最初のイベントの前に、365 日間の連続観察が必要になります。観察期間は、図10.10に示すように、365 日間の連続観察を追加して更新されます。このルックバック期間は、研究チームの裁量で決定します。他のコホートでは異なる選択肢を選ぶこともできます。これは、最初の記録を確実に取得するために、患者を観察する最小期間を可能な限り長く確保するためのものです。この基準は過去の履歴に関するものであり、インデックスイベント後の期間は考慮しません。したがって、インデックスイベント後の期間は 0 日とします。適格イベントは、ACE 阻害薬の初回使用です。したがって、初回イベントは、各人における「最も早いイベント」に限定されます。

The screenshot shows the 'Cohort Entry Events' dialog box. At the top, it says 'Events having any of the following criteria:' with a blue '+' button for 'Add Initial Event'. Below this, there is a dropdown menu showing 'a drug exposure of ACE inhibitors' and a red 'X' button next to the text 'for the first time in the person's history'. To the right of this are blue '+ Add attribute...' and red 'Delete Criteria' buttons. Further down, it says 'with continuous observation of at least 365 days before and 0 days after event index date'. Below that, it says 'Limit initial events to: earliest event per person.' and has a green 'Restrict initial events' button.

Figure 10.10: インデックス日付前に必要な継続的観察を設定

このロジックがどのように組み合わさるかをさらに説明するため、患者のタイムラインを組み立てることを考えてみましょう。

図10.11では、各線はコホートに参加資格がある可能性のある単一の患者を表しています。塗りつぶされた星は、患者が指定された基準を満たすタイミングを表しています。追加の基準が適用されると、いくつかの星がより薄い色になります。これは、これらの患者が基準を満たす他の記録を持っていることを意

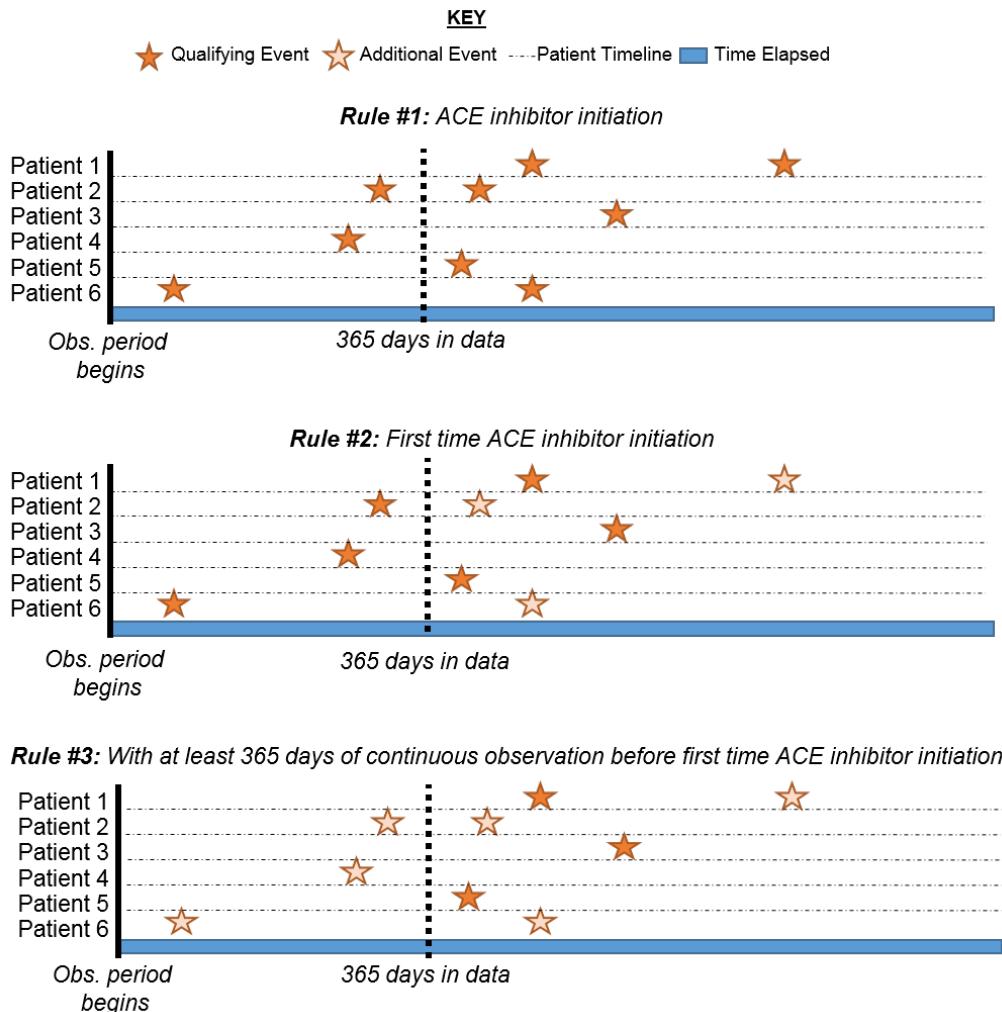


Figure 10.11: 基準の適用による患者の適格性の説明

味しますが、その基準を満たす前に他の記録があることを意味します。最終的な基準に達するまでに、初めて ACE 阻害薬を使用し、その前の 365 日間の連続する観察を持つ患者の累積ビューが表示されます。論理的には、最初のイベントに限定することは冗長ですが、すべての選択で明確なロジックを維持することに有益です。独自のコホートを構築するときには、OHDSI フォーラムの研究者セクションに参加して、コホート論理の構築についてセカンドオピニオンを求める 것도できます。

10.7.4 適格基準

コホート組入れイベントを指定すると、追加の適格イベントを “Restrict initial events (初期イベントの制限)” または “New inclusion criteria (新規の適格基準)” のいずれかに追加することができます。これら二つのオプション間の基本的な違いは、ATLAS が提供する中間情報の内容です。“Restrict initial events” を選択して “Cohort Entry Event (コホート組入れイベント)” ボックスに追加基準を加えると、ATLAS でカウントを生成するときに、これらすべての基準を満たす人の数だけが返されます。“New inclusion criteria” に基準を追加すると、追加の適格基準を適用することによって失う患者数を示す脱落チャートが表示されます。各ルールがコホート定義の全体的な成功に与える影響を理解するために、“Inclusion Criteria (適格基準)” セクションを最大限に活用することを強く推奨します。特定の適格基準がコホートに入る人の数を大幅に制限する可能性があることに気が付くかもしれません。この段階では、コホートをより大きくするために、この基準を緩和する選択をするかもしれません。最終的には、このコホートを組み立てる専門家の合意に従います。

コホートのメンバーシップに関するロジックをさらに追加するために “New inclusion criteria” をクリックします。このセクションの機能は、前述のコホート基準の構築方法と同じです。最初の追加基準は次のとおりです：ACE 阻害薬の最初の開始日から 365 日後から 0 日以内に少なくとも 1 回の高血圧症が発生した患者のみ。新しい適格基準を追加するには “New inclusion criteria” をクリックします。基準に名前を付け、必要に応じて探している内容についての簡単な説明を付け加えることができます。これは定義している基準を覚えるためのもので、コホートの整合性に影響を与えるものではありません。

新しい条件に注釈を付けたら、“+Add criteria to group (グループへ基準を追加)” ボタンをクリックして、このルールの実際の基準を構築します。このボタンは “Add Initial Event” と同様に機能します。ただし、初期イベントを指定するわけではありません。複数の基準を追加できるため、“add criteria to group” と指定されています。たとえば、疾患を見つけるための方法が複数ある場合（例：CONDITION_OCCURRENCE のロジック、この条件の代理としての DRUG_EXPOSURE のロジック、この条件の代理としての MEASUREMENT のロジック）があります。これらは別々のドメインであり、異なる基準を必要としますが、この状態を求める 1 つの条件にまとめることができます。この場合、高血圧症の診断を見つけたいので、“Add condition occurrence (コンディションの出現を追加)” を追加します。このレコードにコンセプトセットを

添付することで、最初のイベントと同様の手順を踏みます。また、イベントがインデックス日（最初の ACE 阻害薬の使用）の 365 日前から 0 日後までの間に開始したことを指定します。ここで、図 ?(fig:ATLASIC1) と照らし合わせてロジックを確認ください。

Inclusion Criteria

New inclusion criteria

has hypertension diagnosis in 1 yr prior to treatment

Copy Delete

1. has hypertension diagnosis in 1 yr prior to treatment

enter an inclusion rule description

having all of the following criteria:

+ Add criteria to group...

with at least 1 using all occurrences of:
a condition occurrence of Hypertensive disorder + Add attribute...
where event starts between 365 days Before and 0 days
After index start date add additional constraint
 restrict to the same visit occurrence
 allow events from outside observation period

Delete Criteria

Limit qualifying events to: earliest event per person.

Figure 10.12: 追加の選択基準 1

次に、患者を検索するための別の条件を追加します：インデックス開始日の前日から当日までの全期間において、高血圧治療薬の使用歴がまったくない（ACE 阻害薬の使用開始以前に高血圧治療薬を使用していない）。このプロセスは、以前と同様に “New inclusion criteria (新規の選択基準)” ボタンをクリックし、この条件に注釈を追加し、“+Add criteria to group (グループに条件を追加)” をクリックして開始します。これは DRUG_EXPOSURE なので、“Add Drug Exposure (薬剤曝露を追加)” をクリックし、高血圧治療薬のコンセプトセットを添付し、インデックス日から遡るすべての日とインデックス日 0 日（または図で示されているように「1日前」も同様）を指定します。選択した発生回数が正確に 0 であることを確認ください。ここで、図 10.13 で、ロジックを確認します。

「発生なし」が「正確に 0 回出現」としてコード化される理由がわからないかもしれません。これは、ATLAS が知識を消費する方法のニュアンスです。ATLAS は適格条件のみを処理します。特定の属性が存在しないことを指定する場合は、論理演算子を使用する必要があります。例えば、「正確に 0 回」などです。ATLAS の適格基準で使用できる論理演算子については、徐々に理解が深まるでしょう。

最後に、患者を絞り込むために、もう一つ別の条件を追加します：インデックス開始日の 0 日前から 7 日後までの間に高血圧治療薬が 1 回だけ処方されており、かつ、高血圧治療薬（ACE 阻害薬）を 1 種類しか処方されていない患者。このプロセスは、前回と同様に “New inclusion criteria (新しい選択基準)” ボタンをクリックし、この条件に注釈を追加してから” +Add criteria to group (グループに条件を追加) ”をクリックして開始します。これは DRUG_ERA なので、“Add Drug Era (薬剤曝露期間を追加) ”をクリックし、高血圧治療薬のコンセ

The screenshot shows the ATLAS Inclusion Criteria interface. A new inclusion criterion is being added:

- New inclusion criteria:** Has no prior antihypertensive drug exposures in medical history.
- Inclusion rule description:** enter an inclusion rule description
- Criteria:** having all of the following criteria:
 - with exactly 0 occurrences of: a drug exposure of Hypertension drugs
 - 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
- Buttons:** Copy, Delete, + Add criteria to group, Delete Criteria

Limit qualifying events to: earliest event per person.

Figure 10.13: 追加の選択基準 2

プトセットを添付し、インデックス日の前 0 日および後 7 日を指定します。次に、図 10.14 と照らし合わせてロジックを確認します。

The screenshot shows the ATLAS Inclusion Criteria interface. A new inclusion criterion is being added:

- New inclusion criteria:** Is only taking ACE as monotherapy, with no concomitant combination treatments.
- Inclusion rule description:** enter an inclusion rule description
- Criteria:** having all of the following criteria:
 - with exactly 1 occurrences of: a drug era of Hypertension drugs
 - where event starts between 0 days Before and 7 days After
 - index start date add additional constraint
 - allow events from outside observation period
- Buttons:** Copy, Delete, + Add criteria to group, Delete Criteria

Limit qualifying events to: earliest event per person.

Figure 10.14: 追加の選択基準 3

10.7.5 コホート離脱基準

これで、すべての適格基準が追加されました。次に、コホート離脱基準を指定する必要があります。「このコホートに含まれる対象ではなくなるのはどのような場合か？」と自問することになります。このコホートでは、薬剤曝露を受けた新規ユーザーを追跡しています。薬剤曝露に関連する継続的な観察期間を調べたいと考えています。そのため、コホート離脱基準は、継続的な薬剤曝露の全般について適用されるように指定します。もしその後、薬剤への曝露が中断された場合は、その時点で患者はコホートから離脱します。薬剤への曝露が中断された間の患者の状態が不明であるため、このような措置を取っています。また、薬剤への曝露間の許容可能なギャップを特定する持続ウィンドウの基準

を設定することもできます。このケースでは、持続曝露の時期を推定する際に、曝露レコードの間の最大許容期間は 30 日間であると、この研究を主導する専門家が結論づけました。

なぜギャップが許容されるのでしょうか？データセットによっては、受療の一部しか観察できないことがあります。特に薬剤曝露は、一定期間をカバーする処方箋の調剤を表している可能性があります。そのため、調剤単位が 1 日を超える場合、患者は論理的には依然として最初の薬剤曝露にアクセスできる可能性があることを考慮し、薬剤曝露間の一定の時間差を許容します。

これを設定するには、イベントは”end of a continuous drug exposure (連続した薬剤曝露の終了)”で継続するを選択します。次に、持続期間を”allow for a maximum of 30 days (最大 30 日間)“に設定し、「ACE 阻害剤」のコンセプトセットを追加します。図 10.15 と照らし合わせてロジックを確認しましょう。

The screenshot shows the 'Cohort Exit' configuration interface. At the top, it says 'Event Persistence:' with a dropdown set to 'end of a continuous drug exposure'. Below that, 'Continuous Exposure Persistence:' is described as specifying a concept set that contains one or more drugs, derived from all drug exposure events within the specified persistence window. Under 'Concept set containing the drug(s) of interest:', 'ACE Inhibitors' is selected. A note below specifies a persistence window of 30 days and a surveillance window of 0 days. In the 'Censoring Events:' section, it says 'Exit Cohort based on the following criteria:' followed by a button '+ Add Censoring Event'.

Figure 10.15: コホート離脱基準

このコホートの場合、他に打ち切りイベントはありません。しかし、打ち切りを指定する必要があるようなコホートを作成することもあるかもしれません。その場合には、コホート定義に他の属性を追加したのと同様の手順で作業を進めます。これで、コホートの作成が完了しました。ボタンをクリックしてください。おめでとうございます！コホートの作成は、OHDSI ツールでリサーチ・クエスチョンに答えるための最も重要な構成要素です。“Export (出力)” タブを使用して、SQL コードまたは ATLAS に読み込むための JSON ファイルの形式により、他の共同研究者とコホート定義を共有することができます。

10.8 SQL を使用したコホートの実装

ここでは、SQL と R を使用して同じコホートを作成する方法について説明します。第9章で説明したように、OHDSI は SqlRender と DatabaseConnector と

いう 2 つの R パッケージを提供しており、これらを組み合わせることで、多様なデータベースプラットフォームに対して自動的に変換・実行可能な SQL コードを記述できます。

分かりやすくするために、SQL をいくつかのチャンクに分割し、各チャンクが次のチャンクで使用される一時テーブルを生成するようにします。これは最も計算効率の良い方法ではないかもしれません、非常に長い一つのステートメントよりも読みやすくなります。

10.8.1 データベースへの接続

最初に R に対してサーバーへの接続方法を指示する必要があります。ここでは DatabaseConnector パッケージを使用し、createConnectionDetails という名前の関数が提供用意されています。?createConnectionDetails ‘と入力すると、各データベース管理システム (DBMS) に必要な具体的な設定を行うことができます例えは、以下のコードで PostgreSQL データベースに接続することができます：

```
library(CohortMethod)
connDetails <- createConnectionDetails(dbms = "postgresql",
                                         server = "localhost/ohdsi",
                                         user = "joe",
                                         password = "supersecret")

cdmDbSchema <- "my_cdm_data"
cohortDbSchema <- "scratch"
cohortTable <- "my_cohorts"
```

最後の 3 行で、cdmDbSchema、cohortDbSchema、および cohortTable の変数を定義しています。これらは後で、CDM 形式のデータが格納されている場所と、対象となるコホートを作成する場所を R に伝えるために使用します。Microsoft SQL Server の場合、データベースのスキーマはデータベースとスキーマの両方を指定する必要があることに注意ください。例えば、cdmDbSchema <- "my_cdm_data.dbo" となります。

10.8.2 コンセプトの指定

可読性を高めるために、必要なコンセプト ID を R で定義し、それらを SQL に渡します：

```
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 初回使用の発見

まず、各患者の ACE 阻害薬の初回使用を見つけます：

```
conn <- connect(connDetails)

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 テーブルを CONCEPT_ANCESTOR テーブルと結合することで、ACE 阻害薬を含むすべての薬剤を見つけていていることに注意ください。

10.8.4 365 日の事前観察が必要

次に、OBSERVATION_PERIOD テーブルと結合して 365 日間の連続した事前の観察を要求します：

```
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 高血圧の前診断が必要

365 日以内の高血圧の診断が必要です：

```
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)
```

過去に複数の高血圧診断がある場合でも、重複するコホート組入れを作成しないように SELECT DISTINCT を使用していることに注意ください。

10.8.6 治療の無前使用が必要

高血圧症の治療歴がないことを求めます：

```
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 (@call_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 JOIN を使用し、DRUG_EXPOSURE テーブルからの person_id が NULL の場合のみ行を許可することに注意ください。これは、一致するレコードが見つからなかったことを意味します。

10.8.7 単剤療法

コホート組入れの最初の 7 日間に高血圧症治療への曝露が一回のみである必要があります：

```

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 コホート離脱

コホートの終了日を除いて、これでコホートは完全に指定されました。コホートは曝露が停止した時点で終了と定義され、次の曝露との間に最大で 30 日間のギャップを許容します。これは、最初の薬剤曝露だけでなく、それに続く ACE 阻害薬の曝露も考慮する必要があることを意味します。連続する曝露を統合するための SQL は非常に複雑になることがあります。幸い、連続する曝露を効率的に作成する標準コードが定義されています（このコードはクリス・ノールに

よって作成されたもので、OHDSI 内では「the magic」と呼ばれることがよくあります)。まず、統合したいすべての曝露を含む一時テーブルを作成します:

```
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)
```

次に、連続する曝露を統合するための標準コードを実行します:

```
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)

```

このコードは、その後のすべての曝露をマージし、max_gap 引数で定義された曝露間のギャップを許容します。結果として得られた薬剤曝露の期間は、#exposure_era と呼ばれる一時テーブルに書き込まれます。

次に、ACE 阻害薬の曝露期間を元のコホートに結合し、期間終了日をコホートの終了日として使用します：

```
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)
```

ここで、先に定義したスキーマとテーブルに最終的なコホートを格納します。同じテーブルに格納する可能性のある他のコホートと区別するために、コホート定義 ID として「1」を割り当てます。

10.8.9 クリーンアップ

最後に、作成した一時テーブルをすべてクリーンアップし、データベースサーバーから切断することを推奨します：

```
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 要約



- コホートとは、一定期間に 1 つ以上の適格基準を満たす人の集合体を指します。
- コホート定義とは、特定のコホートを識別するために使用されるロジックの説明です。
- コホートは、対象とする曝露やアウトカムを定義するために、OHDSI 分析ツール全体で使用（再利用）されます。
- コホートを構築するには、2 つの主要なアプローチがあり、ルールベースと確率論的なアプローチです。
- ルールベースのコホート定義は、ATLAS または SQL を使用して作成できます。

10.10 演習

前提条件

最初の演習には、ATLAS インスタンスへのアクセスが必要です。以下のインスタンス <http://atlas-demo.ohdsi.org> またはアクセス可能な他のインスタンスを使用できます。

演習 10.1. 以下の条件に従って ATLAS でコホート定義を作成してください。:

- ジクロフェナクの新規ユーザー
- 16 歳以上
- 曝露前に少なくとも 365 日の継続的な観察期間があること
- 以前に（非ステロイド性抗炎症薬（NSAID）への曝露がないこと
- 以前に癌の診断がないこと
- コホートからの離脱は、曝露の中止（30 日間のギャップを許容）と定義すること

前提条件

2 番目の演習では、R、R-Studio、Java がインストールされていることを前提とします。セクション 8.4.5 で説明されている。また、SqlRender、DatabaseConnector、Eunomia パッケージが必要です。これらは、以下の方法でインストールできます :

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

Eunomia パッケージは、ローカルの R セッション内で実行される CDM 内のシミュレートされたデータセットを提供します。接続の詳細は以下の方法で取得できます：

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

CDM データベーススキーマは「main」です。

演習 10.2. 以下の基準に従って、SQL および R を使用して、既存の COHORT テーブルに急性心筋梗塞（AMI）のコホートを作成してください：

- 心筋梗塞の診断の発生（コンセプト 4329847 「心筋梗塞」およびそのすべての下位層に含まれるもの、コンセプト 314666 「陳旧性心筋梗塞」およびその下位層に含まれるもの）。
- 入院または救急外来受診期間（コンセプト 9201、9203、262；それぞれ「入院ビジット」、「救急外来ビジット」、「救急外来および入院ビジット」）。

提案された解答は、付録 E.6 を参照ください。

第 11 章

特性評価

著者: Anthony Sena & Daniel Prieto-Alhambra

観察医療データベースは、さまざまな特性に基づく集団の差異を理解するための貴重なリソースとなります。記述統計を用いて集団の特性を把握することは、健康と疾患の決定要因に関する仮説を生成するための重要な第一歩です。本章では特性評価の方法について説明します：

- ・データベースの特性評価：データベース全体のデータプロファイルを全体的に理解するための、トップレベルの要約統計のセットを提供します。
- ・コホート特性評価：集団をその累積的な医療履歴に基づいて記述します。
- ・治療経路：特定の期間に受けた一連の介入を説明します。
- ・発生率：リスク期間における集団のアウトカムの発生率を測定する。

データベースレベルの特性評価を除き、これら的方法は「インデックス日」と呼ばれるイベントに対して集団を記述することを目的としています。この対象集団は、第 10 章で説明されているようにコホートとして定義されます。コホートは対象集団内の各人のインデックス日を定義します。インデックス日をアンカー（基点）として、インデックス日以前の時間をベースライン期間と定義し、インデックス日以降のすべての時間をポストインデックス期間と呼びます。

特性評価のユースケースには、疾患の自然経過、治療の利用状況、品質向上などが含まれます。本章では特性評価の方法を説明します。高血圧症患者の集団を例に、ATLAS と R を使用してこれらの特性評価のタスクを実行する方法を示します。

11.1 データベースレベルの特性評価

関心集団についての特性評価の問い合わせに答える前に、使用するデータベースの特性をまず理解する必要があります。データベースレベルの特性評価では、データベースレベルの特性評価では、時間的傾向と分布の観点からデータベースの

全体像を説明しようとします。データベースの定量的評価には、通常、以下のような質問が含まれます。：

- ・このデータベースには全体で何人が含まれていますか？
- ・年齢分布は？
- ・このデータベースで観察されている期間は？
- ・時間の経過とともに {治療、コンディション、処置など} が記録・処方された人の割合は？

これらのデータベースレベルの記述統計は、研究者がデータベースに欠けている可能性のあるデータを理解するのにも役立ちます。第 15 章では、データ品質についてさらに詳しく説明します。

11.2 コホート特性評価

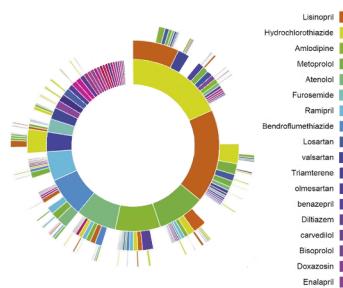
コホート特性評価は、コホート内の人々のベースラインとポストインデックスの特徴を記述します。OHDSI は、個人の履歴に存在するすべてのコンディション、薬剤、デバイスへの曝露、処置、その他の臨床観察の記述統計によって特徴を分析します。また、インデックス日時点でのコホートメンバーの社会人口統計学的属性も要約します。このアプローチは対象集団の完全な要約を提供します。重要なのは、これによりデータのばらつきを考慮しながらコホートを十分に探索でき、潜在的に欠損値となりうる値も特定できるということです。

コホートの特性評価の方法は、特定の治療を受けている患者の適応症や禁忌の有病率を推定する個人レベルの薬剤使用実態研究 (DUS) にも使用できます。このコホート特性評価の普及は、観察研究における推奨されるベストプラクティスであり、Strengthening the Reporting of Observation Studies in Epidemiology (STROBE) ガイドラインで詳述されています (von Elm et al., 2008)。

11.3 治療経路

集団の特性を評価するもう一つの方法は、インデックス後の期間における治療シーケンスを記述することが挙げられます。たとえば、Hripcsak et al. (2016) は、OHDSI の共通データ標準を利用して、2 型糖尿病、高血圧症、抑うつ症に対する治療経路を特性づける記述統計を作成しました。この分析アプローチを標準化することにより、Hripcsak 氏らは、同じ分析を OHDSI ネットワーク全体で実行して、これらの対象集団の特徴を記述することができました。

経路分析は、特定の疾患を診断された人が最初の薬剤処方/供給を受けた治療 (イベント) を要約することを目的としています。この研究では、治療はそれぞれ 2 型糖尿病、高血圧症および抑うつ症の診断後に記述されました。その後、各人のイベントは集計され、各疾患、各データベースに対して要約統計として視覚化されました。



例として、図 11.1 は高血圧症治療を開始する患者集団を表しています。中央にある最初の円は、第一選択治療に基づいた人々の割合を示しています。この例では、ヒドロクロロチアジドがこの集団で最も一般的な最初の治療法です。ヒドロクロロチアジドの部分から伸びるボックスは、コホート内の患者に対して記録された 2 番目および 3 番目の治療法を示しています。

経路分析は、集団における治療利用に関する重要なエビデンスを提供します。この分析から、最初の治療法として最も一般的に利用される第一選択治療、治療を中止する人の割合、治療を変更する人、または治療を強化する人の割合を記述することができます。経路分析を使用して、Hripcsak et al. (2016) はメトホルミンが糖尿病治療に対して最も一般的に処方されている薬剤であることを確認し、米国内分泌学会の糖尿病治療アルゴリズムの第一選択推奨が一般的に採用されていることを確認されました。さらに、糖尿病患者の 10%、高血圧症患者の 24%、抑うつ症者の 11% が、いずれのデータソースにおいても共通しない治療経路をたどっていたことが明らかになりました。

従来の DUS（薬剤使用実態研究）用語では、治療経路分析は、指定された集団における一つまたは複数の薬剤の使用の普及率などの集団レベルの DUS 推定値や継続性やさまざまな治療法間の切り替えの測定などの個人レベルの DUS を含みます。

11.4 発生率

発生率および発生割合は、時間の経過とともに集団における新たなアウトカムの発生を評価するための公衆衛生の統計です。図 11.2 は、単一の人に対する発生率の計算要素を示すことを目的としています：

図 11.2 では、人がデータで観察される期間が観察開始と終了時間によって示されています。次に、個人がいくつかの適格基準を満たしてコホートに組入れられる時点と離脱する時点があります。リスク時間ウィンドウは、アウトカムの発生を理解しようとする期間を示しています。アウトカムがリスク期間内に発生した場合、それをアウトカムの発生としてカウントします。

発生率を計算するための 2 つの尺度があります：

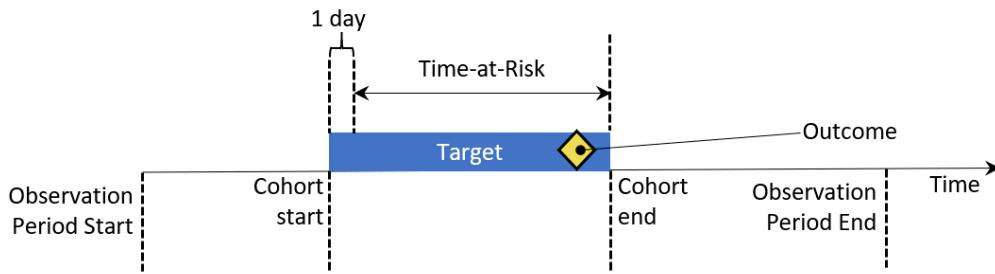


Figure 11.2: 発生率計算要素の人単位のビュー。この例では、リスク時間はコホート開始の翌日に始まり、コホート終了時に終了します。

$$= \frac{\#}{\#}$$

発生割合は、リスク期間中に集団内で発生した新規のアウトカムの割合を提供します。別の言い方をすると、これは定義された期間内に対象集団内でアウトカムを得た割合を示します。

$$= \frac{\#}{\#}$$

発生率は、集団の累積的なリスク期間内に新規のアウトカムの数を測定する指標です。リスク期間中にある人がアウトカムを経験した場合、その人のリスク期間への寄与はアウトカムの発生時点で停止します。累積的なリスク期間は人年と呼ばれ、日、月、または年単位で表現されます。

治療に対して計算される場合、発生割合および発生率は、特定の治療の使用における集団レベルの DUS の古典的な尺度です。

11.5 高血圧症患者の特性評価

世界保健機関（WHO）の高血圧症に関するグローバル概要 (Who, 2013)によると、高血圧症の早期発見、適切な治療、良好な管理には、健康と経済上の両面で大きな利益がもたらされるとしています。WHO の概要是、高血圧症についての概観を提供し、各国における疾病負担の特性を評価しています。WHO は、地理的地域、社会経済的階級、性別ごとに高血圧症の記述統計を提供しています。

観察研究のデータソースは、WHO が行ったように高血圧症患者集団の特性を評価する方法を提供します。本章の後のセクションでは、ATLAS と R を使用してデータベースを探査し、高血圧症患者集団を研究するための構成を理解する方法を探ります。その後、これらのツールを使用して、高血圧症患者集団の自然経過と治療パターンを記述します。

11.6 ATLAS におけるデータベースの特性評価

ここでは、ACHILLES で作成されたデータベースの特性評価統計を調査するために、ATLAS のデータソースモジュールを使用する方法を示します。まず、ATLAS の左側のバーにある **Data Sources** をクリックして開始します。ATLAS に表示される最初のドロップダウンリストで、調査するデータベースを選択します。次に、データベースの下のドロップダウンを使用してレポートの探索を始めます。これを行うには、レポートドロップダウンリストから「Condition Occurrence」を選択し、データベースに存在するすべての症状のツリーマップを表示します：

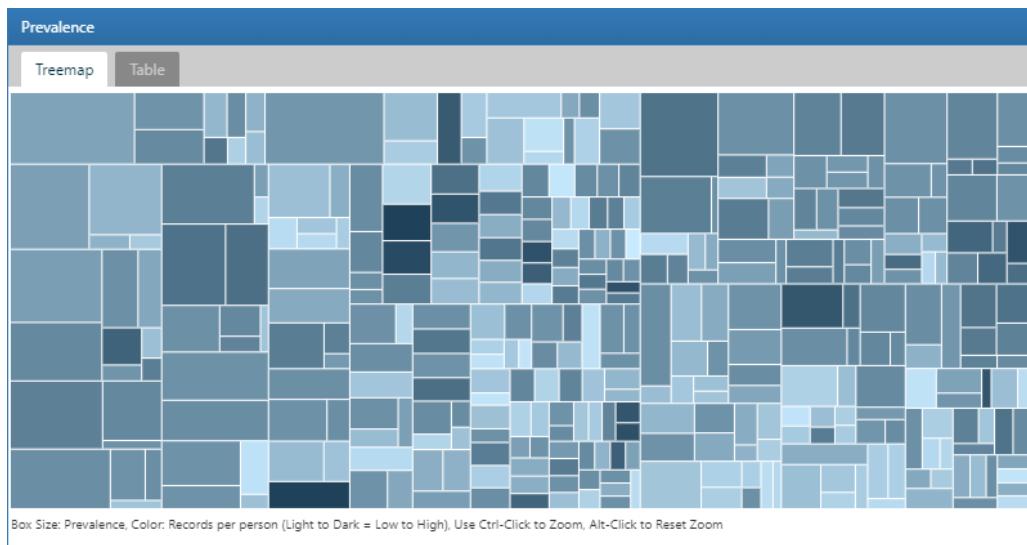


Figure 11.3: ATLAS データソース: コンディション出現のツリーマップ

特定の関心のあるコンディションを検索するには、テーブルタブをクリックして、データベース内のすべてのコンディションのリストを表示し、人数、有病率、個人ごとのレコード数を確認します。上部のフィルターボックスを使用して、コンセプト名に “hypertension (高血圧)” を含む項目に基づいてリストをフィルタリングできます：

特定のコンディションの詳細なドリルダウンレポートを表示するには、行をクリックします。ここでは、“essential hypertension (本態性高血圧)” を選択し、選択されたコンディションの経時的および性別ごとの傾向、月ごとの有病率、記録された病型、診断の初回発生時の年齢の分布を確認します：

高血圧症のコンセプトの有無と経時的な傾向についてデータベースの特性を確認した後、高血圧症患者の治療に使用される薬剤を調査することもできます。これを行うには、同じ手順に従い、RxNorm の成分に要約された薬剤の特性を確認するため、“Drug Era (薬剤曝露期間)” レポートを使用します。データベースの特性を探索して関心のある項目を確認し、高血圧症患者を特徴化するためのコホートの構築を進める準備を整えます。

Prevalence				
Concept	Name	Person Count	Prevalence	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 データソース: コンセプト名に” hypertension (高血圧) ”が含まれるコンディション

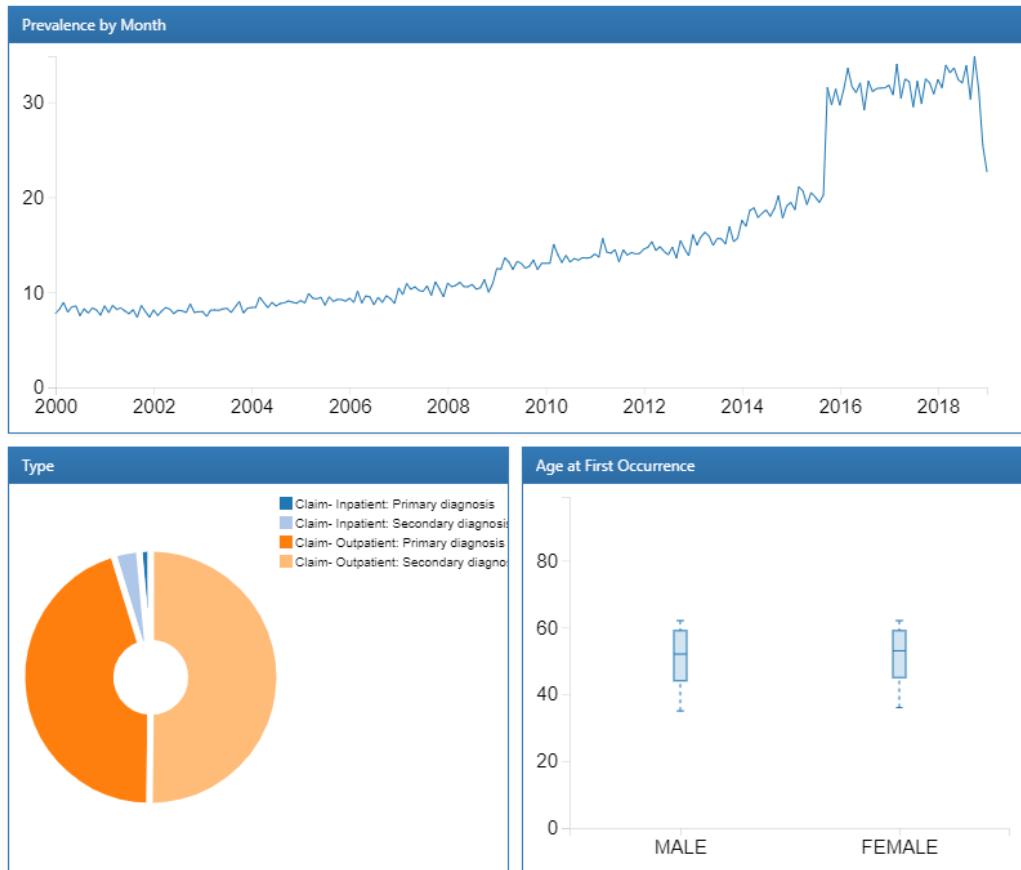


Figure 11.5: ATLAS データソース: 本態性高血圧ドリルダウンレポート

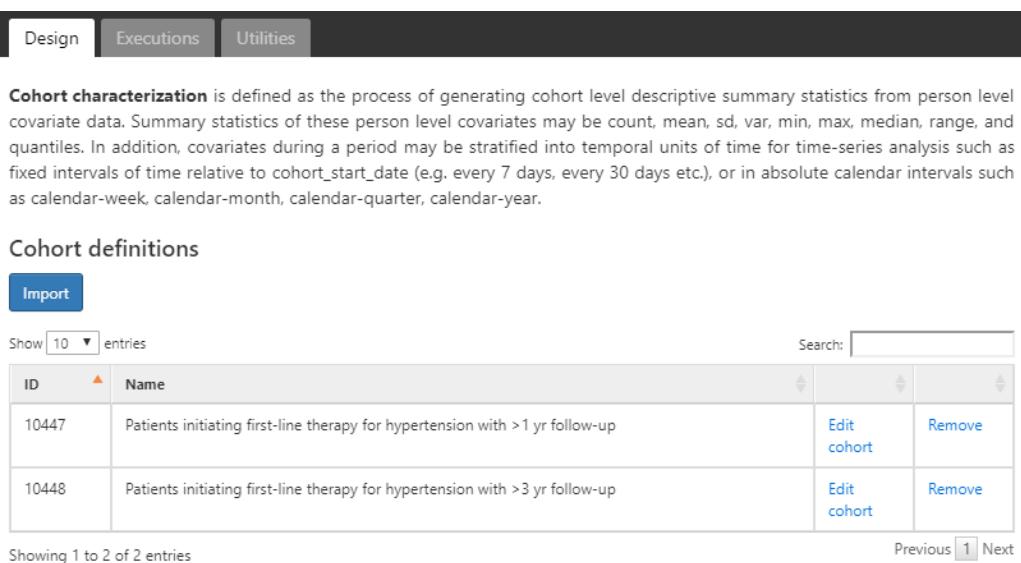
11.7 ATLAS におけるコホート特性分析

ここでは、ATLAS を使用して複数のコホートの大規模な特性評価を行う方法を示します。左側のバーにある  Characterizations をクリックし、新しい特性評価を作成します。分析に名前を付け、 ボタンを使用して保存します。

11.7.1 デザイン

特性評価には、少なくとも 1 つのコホートと少なくとも 1 つの特性が必要です。この例では、2 つのコホートを使用します。最初のコホートでは、その前の 1 年間に少なくとも 1 つの高血圧症診断を受けた人をインデックス日と定義します。また、このコホートに属する人が治療開始後少なくとも 1 年間の観察期間があることも必要です（付録 B.6）。2 つ目のコホートは、最初のコホートと同様ですが、1 年間の代わりに少なくとも 3 年間の観察期間を必要とします（付録 B.7）。

コホート定義



The screenshot shows the 'Design' tab selected in the top navigation bar. Below it, a detailed description of cohort characterization is provided:

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.

Below this, the 'Cohort definitions' section is shown. It includes an 'Import' button and a table listing two cohorts:

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

At the bottom, it says 'Showing 1 to 2 of 2 entries' and has 'Previous' and 'Next' buttons.

Figure 11.6: 特性設計タブ - コホート定義の選択

コホートは既に ATLAS で作成されていると仮定しています（第 10 章を参照）。 をクリックし、図 11.6 に示すようにコホートを選択します。次に、これらのコホートを特性評価するために使用する特性を定義します。

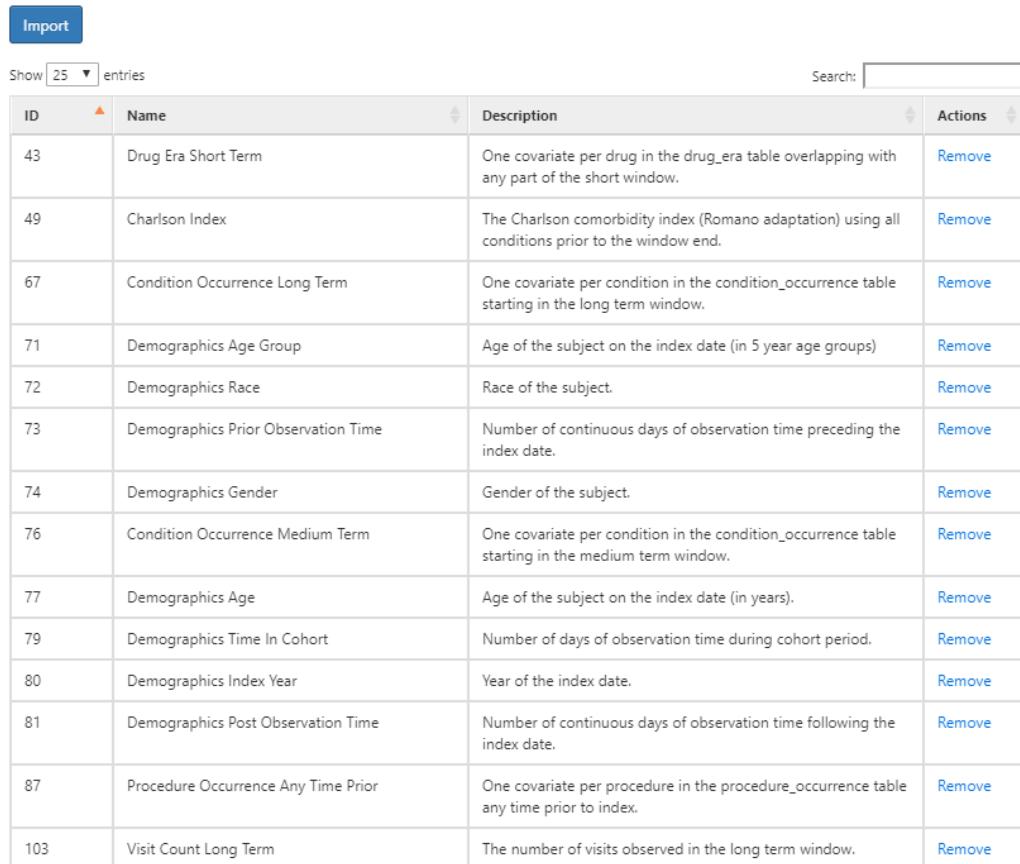
特徴量の選択

ATLAS には OMOP CDM でモデル化された臨床ドメイン全体で特性評価を行うため、約 100 の事前設定された特徴量分析が備わっています。これらのそれ

それの事前に規定された特微量分析は、各ターゲットコホートに対して臨床観察を集約および要約する機能を提供します。これらの計算は、コホートのベースラインおよびポストインデックス特性を説明するために、潜在的に数千の特微量を提供します。この元で、ATLAS は、各コホートの特性評価を行うために、OHDSI の FeatureExtraction R パッケージを利用しています。次のセクションでは、FeatureExtraction と R の使用について詳しく説明します。

Import をクリックして、特性評価をするための特微量を選択します。以下は、これらのコホートを特性評価するために使用する特微量のリストです：

Feature analyses



The screenshot shows a table titled "Feature analyses" with a "Import" button at the top left. The table has columns for "ID", "Name", "Description", and "Actions". There are 25 entries listed, each with a "Remove" link. The entries include various demographic and clinical measures such as Drug Era Short Term, Charlson Index, Condition Occurrence Long Term, Demographics Age Group, and Visit Count Long Term.

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: 特性設計タブ - 特微量の選択

上の図は、選択された特微量のリストと、各特微量が各コホートについて何を特性評価するかの説明を示しています。“Demographics (人口動態的特性)”で始まる特微量は、コホート開始日における各人の人口統計情報を計算します。ドメイン名（例：ビジット、処置、コンディション、薬剤など）で始まる特微量は、そのドメインにおけるすべての記録された観測値を特徴づけます。各ドメインの特微量には、コホート開始前の時間ウィンドウとして、以下の 4 つの選択肢があります：

- Any time prior (任意の期間)：コホート開始前のすべての利用可能な期間で、その人の観察期間に該当するものを使用。
- Long term (長期)：コホート開始日を含む最大 365 日前まで。
- Medium term (中期)：コホート開始日を含む最大 180 日前まで。
- Short term (短期)：コホート開始日を含む最大 30 日前まで。

サブグループ分析

性別に基づいて異なる特性を作成したい場合、「サブグループ分析」セクションを用いて、新たにサブグループを定義し特性評価ができます。

サブグループを作成するには、サブグループのメンバーシップの条件をクリックして追加します。この手順は、コホート登録を識別するために使用する条件と類似しています。この例では、コホート内の女性を識別するための条件を定義します。：

Figure 11.8: 特性評価の設計 - 女性サブグループ分析



ATLAS のサブグループ分析は階層とは異なります。階層は相互に排他的ですが、サブグループは選択された基準に基づいて同じ人を含む場合があります。

11.7.2 実行

特性評価のデザインが完了したら、環境内の 1 つ以上のデータベースに対してこのデザインを実行できます。実行タブに移動し、Generate ボタンをクリックしてデータベースで分析を開始します：

分析が完了したら、“All Executions (すべてを実行)” ボタンをクリックしてレポートを表示し、実行リストから “View Reports (レポートを見る)” を選択

The screenshot shows the ATLAS software interface with the 'Executions' tab selected. There are two main sections for 'SYNPUF 1K' and 'SYNPUF 5%'. Each section includes a 'Generate' button, a 'View latest result' button, and a 'All executions (3)' link.

Execution Type	Count	Generate	View latest result	All executions (3)
SYNPUF 1K	1			
SYNPUF 5%	1			

Figure 11.9: 特性評価設計の実行 - CDM ソース選択

します。あるいは、“View latest result (最新の結果を見る)”をクリックして、最後に実行されたアウトカムを表示することもできます。

11.7.3 結果

CONDITION / Condition Occurrence Long Term / stratified by Female																	
			Covariate		Explore		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 ▼	
								Female		Female		Female					
			Count	Pct	Count	Pct		Count	Pct	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 Next

Figure 11.10: 特性アウトカム - 過去 1 年間の疾患発生

結果は、デザインで選択した各コホートについて、さまざまな特徴量を一覧表示します。図 11.10 では、コホート開始日の前の 365 日間に存在するすべての条件の概要が提供されています。各共変量には、コホートごとおよび定義した女性サブグループごとのカウントと割合が表示されています。

検索ボックスを使用してアウトカムをフィルタリングし、「不整脈」の既往を持つ人の割合を確認することで、集団でどのような心血管関連診断が観察されるかを理解しようとした。 「Explore」リンクをクリックして新しいウィンドウを開き、單一コホートのコンセプトに関する詳細を表示することができます(図 11.11 参照)。

コホートのすべての条件コンセプトを特性評価したため、“explore (探索する)” オプションを使用して、選択されたコンセプト(この場合は不整脈)のすべての上位層と下位層に含まれるコンセプトを表示します。この探索により、高血圧症のある人に現れる可能性のある他の心疾患を探査するためのコンセプト

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	Count	Pct	Count
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: 特性アウトカム - 単一コンセプトの探索

トの階層をナビゲートすることができます。要約表示と同様に、カウントとパーセンテージが表示されます。

この特性結果を用いて、高血圧症治療に禁忌のある条件（例：血管性浮腫）を見つけることもできます。これを行うには、上記と同じ手順を実行しますが、今回は “edema (浮腫)” を検索します（図 11.12 を参照）。

CONDITION / Condition Occurrence Long Term / stratified by Female

Covariate		Explore	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	Concept ID	Count	Pct	Female		Count	Pct	Female				
					Count	Pct			Count	Pct	Count		
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: 特性評価の結果 - 禁忌条件の探索

再度、特徴量を “explore (探索する)” を使用して、高血圧症集団における浮腫の特性を調べ、血管性浮腫の有病率を確認します：



Figure 11.13: 特性アウトカム - 禁忌条件の詳細を探索

ここでは、降圧薬を開始する前の 1 年間に血管性浮腫の既往歴がこの集団の一部にあることが確認されました。

DEMOGRAPHICS / Demographics Age

The screenshot shows a comparison of patient ages across different cohorts. The top bar says 'DEMOGRAPHICS / Demographics Age'. Below it, buttons for 'Export', 'Export comparison', 'Show 10 entries', and a search bar. The main table compares 'Patients initiating first-line therapy for hypertension with > 1 yr follow-up' and 'Patients initiating first-line therapy for hypertension with > 3 yr follow-up'. It includes columns for 'Strata' (Female, All stratas), 'Count', 'Avg', 'Std Dev', 'Median', and 'Std diff'. The data shows that the average age is higher for the > 3 yr cohort compared to the > 1 yr cohort, with a standard difference of approximately -0.0291 for females and -0.0232 for all strata. Navigation buttons at the bottom include 'Showing 1 to 2 of 2 entries', 'Previous 1 Next', and 'Search: '.

Figure 11.14: 各コホートとサブグループの年齢特性アウトカム

ドメイン共変量は、コホート開始前の時間枠にコードの記録が存在したかどうかを示す二元指標を用いて計算されますが、一部の変数はコホート開始時の年齢のように連續変数として提供されます。上の例では、人数、平均年齢、中央値、標準偏差などで表現された、2 つのコホートの年齢を示しています。

11.7.4 カスタム特徴量の定義

プリセットの特徴量に加えて、ATLAS はユーザー定義のカスタム特徴量を用いることもできます。これを行うには、左側のメニューの Characterization (特性評価) をクリックし、Feature Analysis (特徴量分析) タブをクリックして、New Feature Analysis (新規の特徴量分析) ボタンをクリックします。カスタム特性に名前を付け、 ボタンを使用して保存します。

この例では、ACE 阻害剤の服用歴が各コホート開始後にある、コホート内の人数を特定するカスタム特徴量を定義します：

上で定義した基準は、コホート開始日に適用されることを前提としています。

Figure 11.15: ATLAS でのカスタム特微量定義

基準を定義し保存したら、前のセクションで作成した特微量の評価にこの基準を適用できます。特微量の評価のデザインを開き、Feature Analysis (特微量分析) セクションに移動します。**Import** ボタンをクリックし、メニューから新しいカスタム特微量を選択します。これで、特性評価デザインのリストに表示されます。前述のように、このデザインをデータベースに対して実行して、このカスタム特微量による特性評価を実行することができます：

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	Count	Pct	Female		Count	Pct	Female			
					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 [2]

Figure 11.16: カスタム機能の結果表示

11.8 R でのコホートの特性評価

R を使用してコホートをの特性を評価することもできます。このセクションでは、OHDSI R パッケージである FeatureExtraction を使用して、高血圧症コホートのベースライン特徴量を生成する方法について説明します。FeatureExtraction は、3 つの方法で共変量を構築する機能を提供します。

- デフォルトの共変量セットを選択する
- 事前に指定された分析セットから選択する
- カスタム分析セットを作成する

FeatureExtraction は、個人レベルの特徴量と集約特徴の 2 つの異なる方法で共変量を作成します。個人レベルの特徴量は機械学習アプリケーションで有用です。本セクションでは、対象とするコホートを説明するベースライン共変量を生成するのに有用な集約統計量の使用に焦点を当てます。さらに、事前に指定された分析とカスタム分析という 2 つの方法で共変量を構築する方法に焦点を当て、デフォルトのセットを使用する方法は読者への課題として残します。

11.8.1 コホートのインスタンス化

最初に、特性を評価するためにコホートをインスタンス化する必要があります。コホートのインスタンス化は、第 10 章で説明されています。この例では、高血圧症に対して一次治療を開始し、1 年間のフォローアップを行う人を使用します（付録 B.6）。付録 B の他のコホートの特性評価は、読者への練習問題として残しておきます。ここでは、コホートが scratch.my_cohorts というテーブルでインスタンス化され、コホート定義 ID が 1 であると仮定します。

11.8.2 データ抽出

まず、R にサーバーへの接続方法を指示する必要があります。FeatureExtraction は DatabaseConnector パッケージを使用し、createConnectionDetails という関数を提供します。さまざまなデータベース管理システム（DBMS）に必要な特定の設定については、?createConnectionDetails を参照ください。例えば、次のコードで PostgreSQL データベースに接続できます：

```
library(FeatureExtraction)
connDetails <- createConnectionDetails(dbms = "postgresql",
                                         server = "localhost/ohdsi",
                                         user = "joe",
                                         password = "supersecret")

cdmDbSchema <- "my_cdm_data"
cohortsDbSchema <- "scratch"
cohortsDbTable <- "my_cohorts"
cdmVersion <- "5"
```

最後の 4 行は、`cdmDbSchema`、`cohortsDbSchema`、`cohortsDbTable` 変数、および CDM バージョンを定義します。これらはの変数は、CDM 形式のデータがどこにあるか、対象とするコホートがどこに作成されたか、どの CDM バージョンが使用されているか、を R に通知します。Microsoft SQL Server の場合、データベーススキーマはデータベースとスキーマの両方を指定する必要があることに注意ください。したがって、例えば `cdmDbSchema <- "my_cdm_data.dbo"` となります。

11.8.3 事前に指定された分析の使用

`createCovariateSettings` 関数は、ユーザーが定義済みの多くの共変量から選択できるようにします。利用可能なオプションの概要については、`?createCovariateSettings` を入力して参照ください。例えば、以下のようになります：

```
settings <- createCovariateSettings(
  useDemographicsGender = TRUE,
  useDemographicsAgeGroup = TRUE,
  useConditionOccurrenceAnyTimePrior = TRUE)
```

これにより、性別、年齢（5 歳との年齢グループ）、およびコホート開始日までの期間（開始日を含む）の `condition_occurrence` テーブルで観測された各コンセプトのバイナリ共変量が作成されます。

多くの事前に指定された分析は、短期、中期、長期の時間枠を参照しています。デフォルトでは、これらの枠は次のように定義されています：

- ・長期：コホート開始日を含む 365 日前まで
- ・中期：コホート開始日を含む 180 日前まで
- ・短期：コホート開始日を含む 30 日前まで

ただし、ユーザーはこれらの値を変更できます。例を以下に示します：

```
settings <- createCovariateSettings(useConditionEraLongTerm = TRUE,
                                      useConditionEraShortTerm = TRUE,
                                      useDrugEraLongTerm = TRUE,
                                      useDrugEraShortTerm = TRUE,
                                      longTermStartDays = -180,
                                      shortTermStartDays = -14,
                                      endDays = -1)
```

これは、長期ウィンドウをコホート開始日の 180 日前から当日まで（当日を含まず）と再定義し、短期ウィンドウをコホート開始日の 14 日前から当日まで（当日を含まず）と再定義します。また、共変量を構築する際に使用すべき、または使用すべきでないコンセプト ID を指定することもできます。：

```
settings <- createCovariateSettings(useConditionEraLongTerm = TRUE,
                                      useConditionEraShortTerm = TRUE,
                                      useDrugEraLongTerm = TRUE,
                                      useDrugEraShortTerm = TRUE,
                                      longTermStartDays = -180,
                                      shortTermStartDays = -14,
                                      endDays = -1,
                                      excludedCovariateConceptIds = 1124300,
                                      addDescendantsToExclude = TRUE,
                                      aggregated = TRUE)
```



上記すべての例について、「aggregated = TRUE」の使用は、FeatureExtraction に要約統計量を提供することを指示します。このフラグを除外すると、コホート内の各人の共変量が計算されます。

11.8.4 集約共変量の作成

次のコードブロックは、コホートの集計統計を生成します：

```
covariateSettings <- createDefaultCovariateSettings()

covariateData2 <-getDbCovariateData(
  connectionDetails = connectionDetails,
  cdmDatabaseSchema = cdmDatabaseSchema,
  cohortDatabaseSchema = resultsDatabaseSchema,
  cohortTable = "cohorts_of_interest",
  cohortId = 1,
  covariateSettings = covariateSettings,
  aggregated = TRUE)

summary(covariateData2)
```

出力は次のようになります：

```
## CovariateData Object Summary
##
## Number of Covariates: 41330
## Number of Non-Zero Covariate Values: 41330
```

11.8.5 出力形式

集計された covariateData オブジェクトの主なコンポーネントは、二値および連続の共変量に対してそれぞれ covariates と covariatesContinuous です：

```
covariateData2$covariates
covariateData2$covariatesContinuous
```

11.8.6 カスタム共変量

FeatureExtraction は、カスタム共変量を定義および利用する機能も提供します。これらの詳細は高度なトピックであり、ユーザードキュメントに記載されています：<http://ohdsi.github.io/FeatureExtraction/>

11.9 ATLAS におけるコホート経路分析

経路分析の目標は、1つまたは複数の対象とするコホート内で治療がどのように順序づけられているかを理解することです。適用される方法は Hripcsak et al. (2016) によって報告されたデザインに基づいています。これらの方法は一般化され、ATLAS の Cohort Pathways という機能に組み込まれました。

コホート経路の目的は、1つまたは複数の対象とするコホートのコホート開始日以降のイベントを要約する分析機能を提供することです。そのために、対象となる集団の臨床イベントを特定するためのイベントコホートと呼ばれるコホートセットを作成します。これが対象とするコホート内的人物に対してどのように見えるかを例として示します。

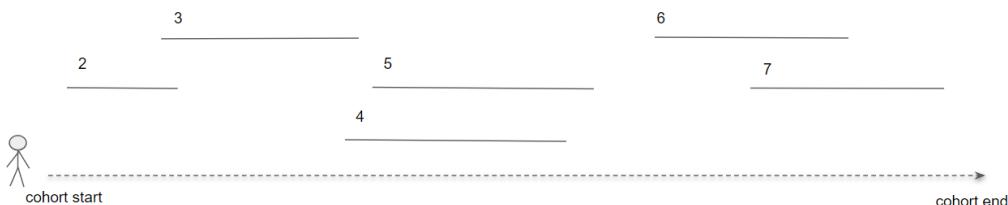


Figure 11.17: 単一の人物におけるパスウェイ分析の文脈

図 11.17 では、その人物が開始日と終了日が定義された対象コホートに属していることを示しています。その後、番号付きの線分は、その人物が特定の期間、イベントコホートで特定され他期間を示しています。イベントコホートは、CDM に表された任意の臨床イベントを記述することができるため、単一のドメインまたはコンセプトに対してパスウェイを作成する必要はありません。

まず、ATLAS の左側のバーで **Cohort Pathways** をクリックして、新しいコホートパスウェイスタディを作成します。分かりやすい名前を入力し、保存ボタンを押します。

11.9.1 デザイン

まず、高血圧症の第一選択療法を開始するコホートと、1年および3年間のフォローアップ（付録B.6、B.7）を継続して使用します。ボタンを使用して、2つのコホートをインポートします。

The screenshot shows the 'Design' tab selected in the top navigation bar. A descriptive text box states: '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.' Below this, a section titled 'Target Cohorts' indicates that each cohort will be analyzed for pathways through event cohorts. An 'Import' button is visible. The main area displays a table of target cohorts:

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

Below the table, it says 'Showing 1 to 2 of 2 entries'. Navigation buttons for 'Previous' and 'Next' are at the bottom right.

Figure 11.18: 対象コホートを選択したパスウェイ分析

次に、対象となる各第一選択の降圧薬のイベントコホートを作成して、イベントコホートを定義します。まず、ACE阻害薬使用者のコホートを作成し、コホートの終了日を継続曝露の終了日と定義します。同様に他の8つの降圧薬のコホートも作成します。これらの定義は付録B.8 – B.16に記載されていることを確認ください。完了したら、**Import**ボタンをクリックして、これらの定義を経路デザインのイベントコホートセクションにインポートします。

完了すると、デザインは上記のようになります。次に、いくつかの追加の分析設定を決定する必要があります：

- 組み合わせウィンドウ：この設定では、イベント間の重複がイベントの組み合わせと見なされる日数の期間を定義できます。たとえば、2つのイベントコホート（イベントコホート1およびイベントコホート2）で表される2つの薬剤が組み合わせウィンドウ内で重複する場合、パスウェイアルゴリズムはそれらを「イベントコホート1 + イベントコホート2」として組み合わせます。
- 最小セル数：この人数に満たないイベントコホートは、プライバシー保護のため、出力から削除されます。
- 最大経路長：分析の対象となる一連のイベントの最大数を指します。

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: <input type="text"/>	
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: 初回第一選択降圧治療を開始するためのイベントコホート

11.9.2 実行

パスウェイ分析のデザインが完了すると、環境内の 1 つ以上のデータベースに対してこのデザインを実行できます。これは、ATLAS でのコホートの特性評価で説明したのと同じ方法で機能します。完了したら、分析の結果を確認できます。

11.9.3 結果の表示

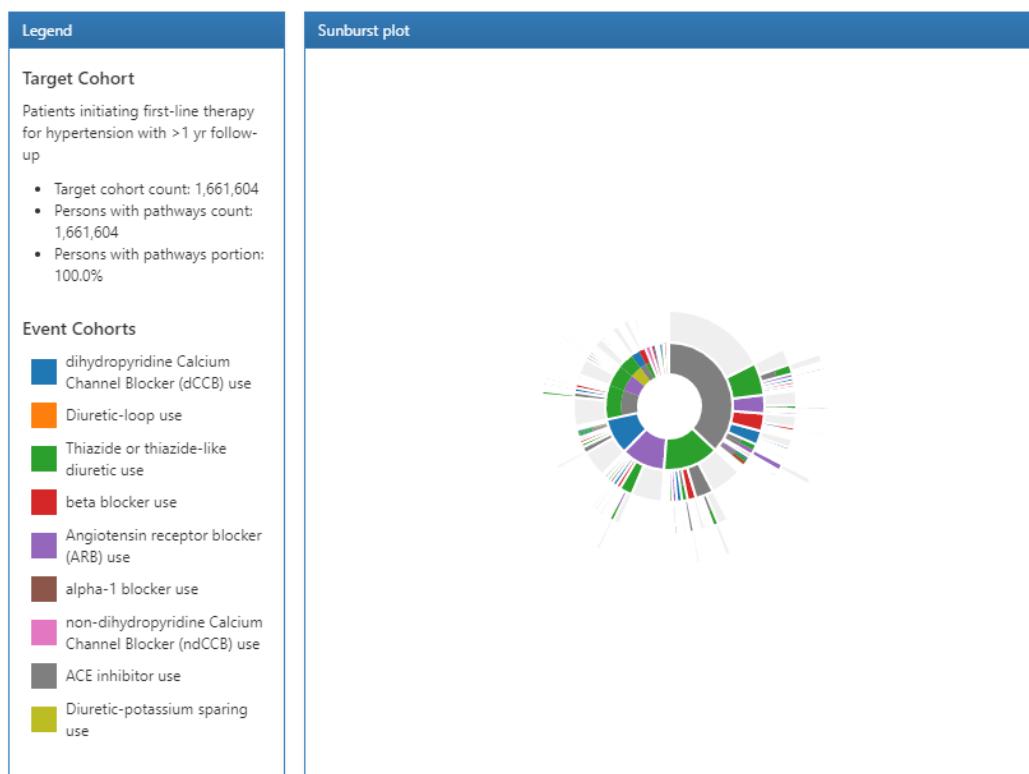


Figure 11.20: 経路結果の凡例とサンバースト図

パスウェイ分析の結果は 3 つのセクションに分かれています：凡例セクションでは、対象コホートの総人数と、パスウェイ分析で 1 つ以上のイベントがあった人数が表示されます。その下には、サンバーストプロットの中央セクションに表示される各コホートの色分けが表示されます。

サンバースト図は、時間の経過に伴うさまざまなイベント経路を視覚的に表現したものです。図の中心はコホートへの組入れを表しており、最初の色分けは各イベントコホートにいる人の割合を示しています。例では、円の中心は第一選択薬による治療を開始した高血圧症患者を表しています。次に、サンバースト図の最初のリングは、イベントコホートによって定義された第一選択薬の種類（すなわち、ACE 阻害薬、アンジオテンシン受容体拮抗薬など）。2 番目のリングセットは、人々にとって 2 番目のイベントコホートを表しています。特定

のイベントシーケンスでは、データで2番目のイベントコホートが観察されない場合があり、その割合はリングの灰色の部分で表されます。

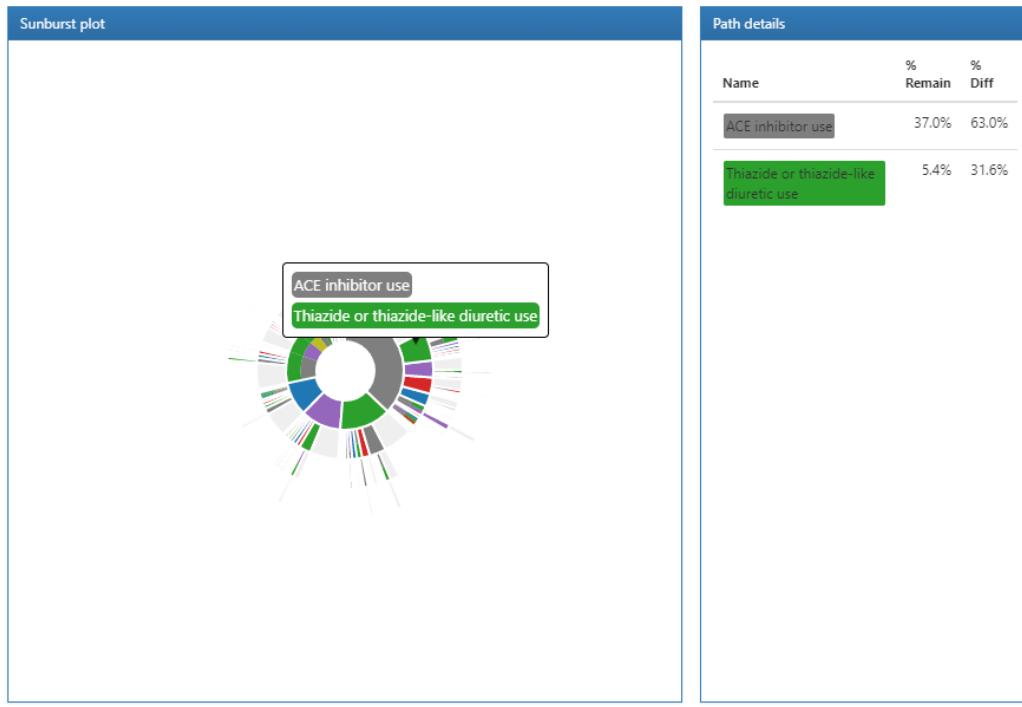


Figure 11.21: 経路の詳細を表示するパスウェイアウトカム

サンバーストプロットのセクションをクリックすると、右側に経路の詳細が表示されます。ここでは、対象コホートにおける大多数の人が ACE 阻害薬による第一選択療法を開始し、そのグループからさらに少数の人がサイアザイド系またはサイアザイド系利尿薬による治療を開始していることが分かります。

11.10 ATLAS における発生率分析

発生率の算出では、以下の内容を記述しますク期間中に、対象コホートに属する人の中で、アウトカムコホートを経験した人。ここでは、ACE 阻害薬 (ACEi)、サイアザイド系およびサイアザイド系利尿薬 (THZ) の新規使用者における血管性浮腫および急性心筋梗塞のアウトカムを特徴づける発生率の分析をデザインします。対象者が薬剤に曝露された TAR 期間中のこれらのアウトカムを評価します。さらに、アンジオテンシン受容体拮抗薬 (ARB) への曝露による転帰を追加し、対象コホート (ACEi および THZ) への曝露期間中の ARB の新規使用の発生率を測定します。このアウトカム定義により、対象集団における ARB の利用状況を把握することができます。

まず、ATLAS の左側のバーにある **Incidence Rates** をクリックし、新規の発生率分析を作成します。分かりやすい名前を入力し、保存ボタン をクリック

クします。

11.10.1 デザイン

本例で使用されるコホートは、既に ATLAS に作成されていると仮定します（第 10 章で説明）。付録には、対象コホート（付録 B.2、B.5）およびアウトカムコホート（付録 B.4、B.3、B.9）の完全な定義が記載されています。

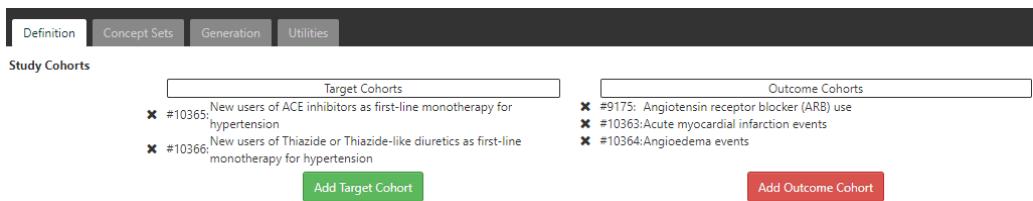


Figure 11.22: 対象およびアウトカム定義の発生率

定義タブで、New users of ACE inhibitors (ACE 阻害薬の新規ユーザー) コホートと New users of Thiazide or Thiazide-like diuretics (サイアザイドまたはサイアザイド様利尿薬の新規ユーザー) コホートをクリックして選択します。ダイアログを閉じて、これらのコホートが追加されたことを確認します。次に、アウトカムコホートを追加するためにクリックし、ダイアログボックスから acute myocardial infarction events (急性心筋梗塞イベント)、angioedema events、および Angiotensin receptor blocker (ARB) use (アンジオテンシン受容体拮抗薬 (ARB) の新規ユーザー) のアウトカムコホートを選択します。再びウィンドウを閉じて、これらのコホートがアウトカムコホートセクションに追加されたことを確認します。

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: 対象およびアウトカム定義の発生率

次に、分析のリスク期間を定義します。上に示すように、リスク期間はコホートの開始日と終了日を基準として定義されます。ここでは、対象コホートの開始日の翌日をリスク期間の開始として定義します。次に、リスク期間終了日をコホートの終了日に終了するよう定義します。この場合、ACEi および THZ コホートの定義には、薬剤曝露が終了する時点をコホート終了日としています。

ATLAS では、分析仕様の一部として対象コホートを層別化する方法も提供しています：

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: 女性における発生率の層別定義

これを行うには、[New Stratify Criteria] ボタンをクリックし、第 11 章で説明されている手順に従います。設計が完了したので、一つまたは複数のデータベースに対して設計を実行します。

11.10.2 実行

[生成] タブをクリックし、 ボタンをクリックして、分析を実行するデータベースの一覧を表示します：

Figure 11.25: 発生率分析実行

一つ以上のデータベースを選択し、「Generation」ボタンをクリックして、指定された対象コホートとアウトカムのすべての組み合わせの分析を開始します。

11.10.3 結果の表示

「Generation」タブでは、画面の上部でターゲットおよびアウトカムを選択してアウトカムを表示することができます。そのすぐ下には、分析で使用された各データベースの発生率のサマリーが表示されます。

それらのドロップダウンリストから ACEi 使用者のターゲットコホートと急性

心筋梗塞 (AMI) を選択します。  ボタンをクリックして発生率分析の結果を表示します:

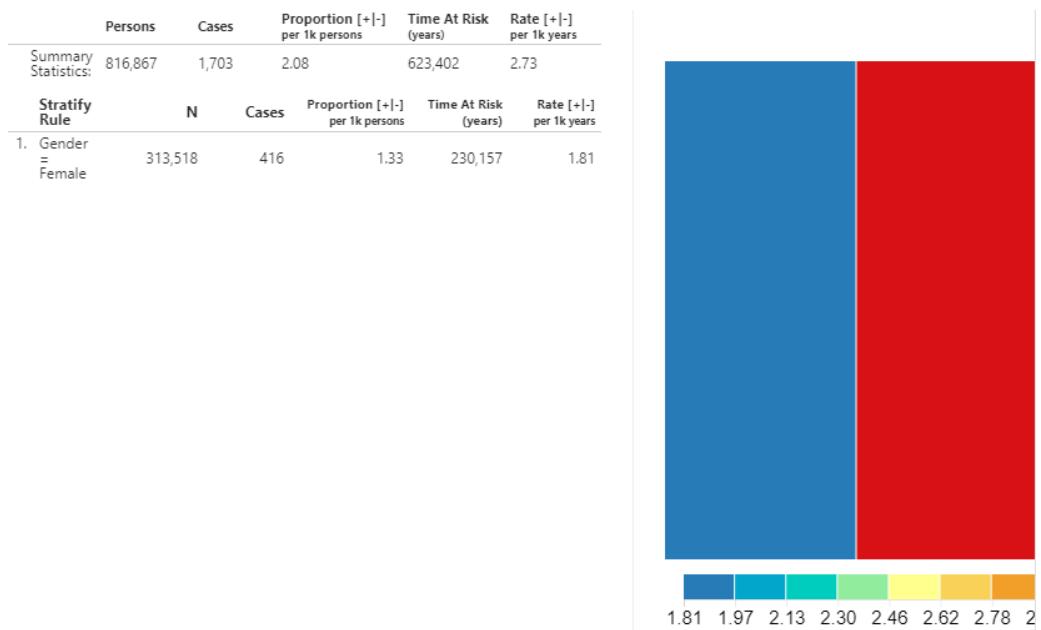


Figure 11.26: 発生率分析の出力 - AMI のアウトカムを持つ新規 ACEi 使用者

データベースの要約は、TAR 期間中に観察されたコホート内の総人数と総症例数を示します。割合は 1000 人当たりの症例数を示しています。対象コホートのリスク期間は年単位で計算されます。発生率は 1000 人年当たりの症例数として表されます。

設計で定義した層の発生率メトリクスも見ることができます。上記のメトリクスは各層についても計算されます。さらに、ツリーマップの視覚化は、それぞれの層が表す割合をボックスエリアとして視覚的に表示します。色は、下部の目盛りで示されるスケールに沿って発生率を示しています。

ACEi 集団の中で ARB 新規使用の発生率を確認するために、同じ情報を収集することができます。画面上部のドロップダウンでアウトカムを ARB 使用に変更し、 ボタンをクリックして詳細を確認します。

示されているように、算出されたメトリクスは同じですが、解釈は異なります。なぜなら、入力 (ARB 使用) が健康アウトカムではなく薬剤使用量の推定値を参照しているためです。

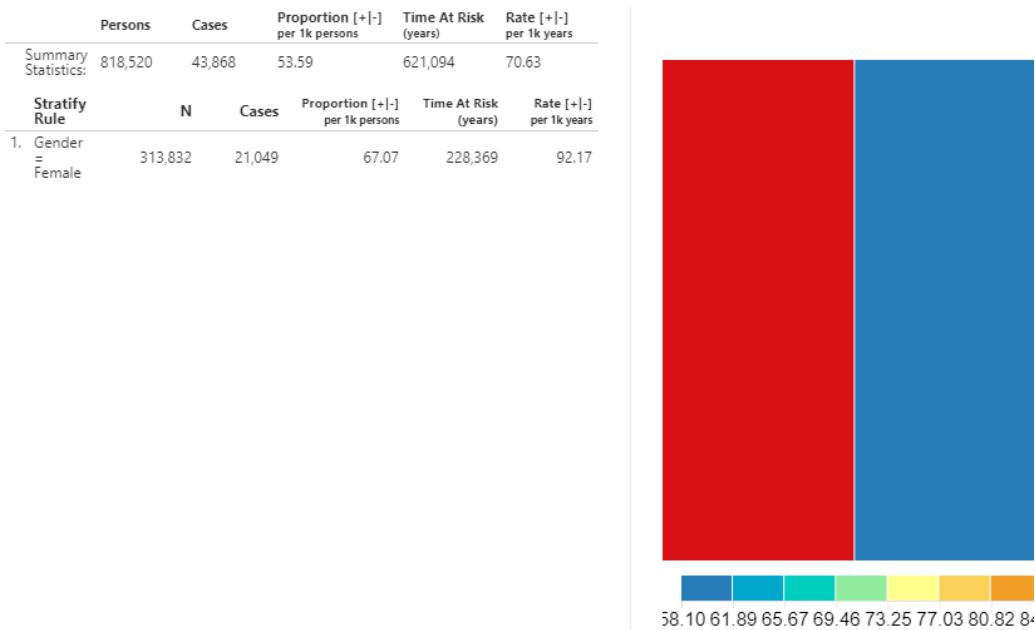


Figure 11.27: 発生率 - ACEi 曝露中に ARB 処理を受けている新規 ACEi 使用者

11.11 まとめ



- OHDSI は、データベース全体または対象とするコホートの特性を評価するためのツールを提供しています。
- コホートの特徴付けは、インデックス日（ベースライン）前およびインデックス日後（ポストインデックス）の期間に対象とするコホートを記述します。
- ATLAS の特徴付けモジュールと OHDSI Methods Library は、複数の時間枠の基準特性を算出する機能を提供します。
- ATLAS の経路および発生率モジュールは、ポストインデックス期間中の記述統計を提供します。

11.12 演習

前提条件

これらの演習には、ATLAS インスタンスへのアクセスが必要です。<http://atlas-demo.ohdsi.org> のインスタンスや、アクセス可能なその他のインスタンスを使用できます。

演習 11.1. セレコキシブが実世界でどのように使用されているかを理解したいと思います。まず、このデータベースがこの薬についてどのようなデータを持っているかを理解したいと思います。ATLAS データソースモジュールを使用して、セレコキシブに関する情報を検索します。

演習 11.2. セレコキシブの使用者の疾患の自然経過について、より深く理解したいと思います。365 日間のウォッシュアウト期間を使用して、セレコキシブの新規使用者の単純なコホートを作成し（作成方法の詳細については、(第 10 章を参照してください)）、ATLAS を使用して、併存疾患と薬剤曝露を示すこのコホートの特性を作成します。

演習 11.3. セレコキシブ処方開始後に消化管出血 (GI 出血) がどのくらいの頻度で発生するのかに興味があります。192671 (“消化管出血”）またはその下位層に含まれるいずれかのコンセプトの発生として単純に定義される GI 出血イベントのコホートを作成します。前の演習で定義した曝露コホートを使用して、セレコキシブ開始後のこれらの GI 出血イベントの発生率を計算してください。

推奨される解答は付録E.7 を参照ください。

第 12 章

集団レベルの推定

著者: Martijn Schuemie, David Madigan, Marc Suchard & Patrick Ryan

保険請求データや電子的健康記録などの観察的な医療データは、治療の効果に関するリアルワールドのエビデンスを生成する機会を提供し、患者の生活を有意に改善することができます。本章では、特定の健康アウトカムに対する曝露（例えば、薬剤曝露や処置（プロシージャー）などの医療介入）の平均的な因果効果の推定を指す集団レベルの効果推定に焦点を当てます。以下では、2つの異なる推定タスクを検討します。：

- ・直接効果推定: アウトカムのリスクに対する曝露の効果を、曝露なしと比較して推定する。
- ・比較効果推定: アウトカムのリスクに対する曝露（ターゲット曝露）の効果を、別の曝露（比較対照の曝露）と比較して推定する。

いずれの場合でも、患者レベルの因果効果は事実のアウトカム、すなわち曝露を受けた患者に何が起こったかと、反事実のアウトカム、すなわち曝露がなかった場合（直接）や異なる曝露があった場合（比較）に何が起こったかを対比させます。各患者は事実のアウトカムのみを明らかにするため（因果推論の基本問題）、さまざまな効果推定デザインは異なる分析デバイスを用いて反事実のアウトカムを明らかにします。

集団レベルの効果推定のユースケースには、治療選択、安全性監視、および比較効果が含まれます。方法論は、特定の仮説を一度に1つずつテストすること（例：「シグナル評価」）や、複数の仮説を一度に探索すること（例：「シグナル検出」）ができます。いずれの場合も、目的は同じです：因果効果の高品質な推定を生成することです。

本章ではまず、OHDSI Methods LibraryとしてRパッケージで実装されているさまざまな集団レベルの推定研究デザインについて説明します。次に、具体的な推定研究の設計を詳細に説明し、ATLASおよびRを使用してデザインを実装する手順ガイドを提供します。最後に、研究から生成されるさまざまな出力、包括的な研究診断と効果量の推定について確認します。

12.1 コホートメソッドの設計

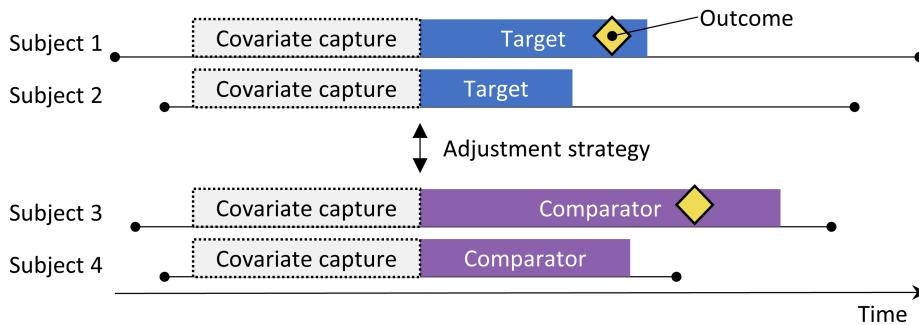


Figure 12.1: 新規ユーザーコホートデザイン。ターゲット治療を開始した対象は比較対照治療を開始した対象と比較されます。2つの治療グループ間の違いを調整するために、傾向スコアによる層化、マッチング、重み付け、あるいはベースライン特性をアウトカムモデルに追加するなど、いくつかの調整戦略が使用されます。傾向スコアモデルまたはアウトカムモデルに含まれる特性は治療開始前に取得されます。

コホートメソッドはランダム化臨床試験を模倣することを試みます (Hernan and Robins, 2016)。ある治療（ターゲット）を開始した対象は別の治療（比較対照）を開始した対象と比較され、治療開始後の特定の期間、例えば治療を継続する期間にわたって追跡されます。コホート研究において私たちが答えるべき問いは、表 12.1 にハイライトされた 5 つの選択を行うことで具体化されます。

Table 12.1: 比較コホートデザインの主要なデザイン選択

選択	説明
ターゲットコホート	対象とする治療を代表するコホート
比較対照コホート	比較対照の治療を代表するコホート
アウトカムコホート	対象とするアウトカムを代表するコホート
リスク期間	どの時点で（通常はターゲットおよび比較対照コホートの開始および終了日）アウトカムのリスクを考慮するか
モデル	ターゲットと比較対照の間の違いを調整しながら効果を推定するために使用されるモデル

モデルの選択には、他の要素の中でも、アウトカムモデルの種類が含まれます。例えば、ロジスティック回帰を使用することができ、これはアウトカムが発生したかどうかを評価し、オッズ比を生成します。ロジスティック回帰はリスク期間がターゲットと比較対照で同じ長さであるか、または関係がないと仮定します。あるいは、一定の発生率を仮定するポアソン回帰を選択することもでき

ます。多くの場合、対象と比較対照の間で比例ハザードを仮定し、最初のアウトカムまでの時間を考慮してハザード比を推定するコックス回帰が使用されます。



新規ユーザーコホートメソッドは本質的に比較効果推定の方法であり、治療を比較対照と比較します。この方法を使用して治療対未治療を比較するのは難しいです。なぜなら、未曝露群と曝露群が比較可能となる群を定義するのが難しいからです。このデザインを直接的な効果推定に使用したい場合は、対象とする曝露に対する比較対照として、同じ適応症の治療を選択するのが望ましいです。ただし、必ずしもそのような比較対照が利用可能であるとは限りません。

重要な懸念事項は、ターゲット治療を受ける患者が比較対照治療を受ける患者と系統的に異なる可能性があることです。例えば、ターゲットコホートが平均60歳であり、比較対照コホートが平均40歳であるとします。年齢に関連する健康アウトカム（例：脳卒中）に関してターゲットと比較対照を比較する場合、コホート間で顕著な違いが見られるかもしれません。無知な研究者は、ターゲット治療と比較対照に比べて脳卒中の間に因果関係があると結論づけるかもしれません。もっと平凡な、あるいはありふれた結論として、ターゲット患者が脳卒中を経験したことが、比較対照を受けていたらそうならなかつたであろうと結論づけるかもしれません。この結論は完全に間違っている可能性があります！おそらくこれらのターゲット患者は、ただ年を取っているだけで脳卒中が発生したかもしれません。治療を受けていたとしても同様であった可能性があります。この文脈では、年齢は「交絡因子」です。観察研究で交絡因子に対処する一つのメカニズムは傾向スコアを介することです。

12.1.1 傾向スコア

ランダム化試験では、（仮想的な）コイン投げが患者を各グループに割り当てます。したがって、デザインによって、患者がターゲット治療を受ける確率は患者の特性（例：年齢）とは無関係です。コインは患者を知りませんし、何よりも、曝露を受ける患者の確率は確実に分かっています。その結果、試験の患者数が増えるにつれて、両方のグループの患者はどのような患者特性においても系統的に異なることは基本的にありえません。この保証されたバランスは、試験で測定された特性（例：年齢）と試験で特定されなかった特性（例：患者の遺伝的要因）にも適用されます。

ある患者に対する傾向スコア（PS）は、その患者が比較対照群に対して対象治療を受ける確率です（Rosenbaum and Rubin, 1983）。バランスの取れたランダム化試験では、傾向スコアはすべての患者で0.5です。傾向スコアで補正された観察研究では、治療開始時とその前のデータに基づいて患者が対象治療を受ける確率を推定します（実際に受けた治療に関係なく）。これは単純な予測モデリングの応用です。ロジスティック回帰などのモデルを適合させ、患者が対象治療を受けるかどうかを予測し、各対象者の予測確率（PS）を生成するた

めにこのモデルを使用します。標準的なランダム化試験とは異なり、各患者は異なる確率で対象治療を受けることになります。PS は、PS が似たターゲット対象者と比較対照の対象者をマッチングする、PS に基づいて研究集団を層化する、PS から導き出された治療重み付けの逆確率 (IPTW) を使うなど、いくつかの方法で使用できます。マッチングの場合、各対象者に対して一人の比較対照者だけを選択することも、一人以上の比較対照者を許容することもできます。これは可変比率マッチングと呼ばれる手法です (Rassen et al., 2012)。

例えば、1 対 1 の PS マッチングを使用し、ヤンが対象治療を受ける事前確率が 0.4 で、実際に対象治療を受けたとします。もし、対象治療を受ける事前確率が 0.4 で、実際には比較治療を受けた患者（ジュンと名付けます）を見つけることができれば、少なくとも測定された交絡因子に関しては、ヤンとジュンの結果の比較はミニ無作為化試験のようなものとなります。この比較により、無作為化で得られたものと同等のヤンとジュンの因果コントラストの推定値が得られます。推定は以下のように行われます。対象治療を受けた患者ごとに、比較対象治療を受けたが対象治療を受ける以前の確率が同じであった 1 人以上の適合患者を見つけます。対象患者のアウトカムと、これらの適合グループ内の比較対象患者のアウトカムを比較します。

傾向スコアは、測定された交絡因子を制御します。実際、測定された特性を考慮して治療割り当てが「無視できる」場合、傾向スコアは因果効果の偏りのない推定値を導きます。「無視できる」とは、本質的には、未測定の交絡因子が存在せず、測定された交絡因子が適切に調整されていることを意味します。残念ながら、これは検証可能な仮定ではありません。この問題に関するさらなる議論については、第 18 章で説明します。

12.1.2 変数選択

以前は、PS は手動で選択された特性に基づいて計算されていましたが、OHDSI ツールはそのような実践をサポートする一方で、特定の曝露やアウトカムに基づいて選択されていない、より多くの汎用特性を使用することを好みます (Tian et al., 2018)。これらの特性には、人口統計学的特性に加え、治療開始前および当日に観察された診断、薬剤曝露、測定値、医療処置が含まれます。通常、モデルには 10,000 から 100,000 の固有の特性が含まれ、これらを大規模な正則化回帰 (Suchard et al., 2013) を使用して適合させ、Cyclops パッケージで実装します。本質的には、どの特性が治療割り当ての予測に関連するかをデータに決定させ、モデルに含めます。



通常、治療開始日の特性は治療の原因となる診断などの多くの関連データがその日に記録されているため、共変量捕捉のウィンドウに含まれるべきです。この日には、対象および比較対照の治療自体も記録されていますが、これらは傾向スコアモデルに含まれるべきではありません。なぜなら、私たちはまさにこれらを予測しようとしているからです。したがって、対象と比較対照治療は共変量セットから明示的に除外する必要があります。

「正しい」因果構造を特定する際に臨床的専門知識に依存しないデータ主導型の共変量選択アプローチは、いわゆる操作変数や共変変数を誤って含めるリスクがあり、その結果、分散が増加し、潜在的にバイアスがもたらされる可能性があるという意見もあります (Hernan et al., 2002)。しかし、このような懸念は現実のシナリオでは大きな影響を与える可能性は低いでしょう (Schneeweiss, 2018)。さらに、医学においては真の因果構造が判明することはほとんどなく、異なる研究者が特定の研究課題に対して「正しい」共変量を特定するように求められると、それぞれの研究者は必ず異なるリストを作成し、そのプロセスを再現不能にします。最も重要なのは、傾向スコアモデルの確認、すべての共変量のバランス評価、およびネガティブコントロールの組み込みなどの診断によって、コライダーや操作変数に関連するほとんどの問題を特定できることです。

12.1.3 カリパー

傾向スコアは 0 から 1 の範囲で連続的に変化するため、厳密なマッチングはほとんど不可能です。その代わり、マッチングプロセスでは、「カリパー」として知られるある程度の許容範囲内で、対象患者の傾向スコアに一致する患者を見つけます Austin (2011) に従い、ロジットスケールにおける標準偏差の 0.2 倍をデフォルトのキャピラとして使用します。

12.1.4 オーバーラップ：選好スコア

傾向スコア方法は一致する患者が存在することを必要とします。このため、主要な診断は二つのグループの傾向スコアの分布を示します。解釈を容易にするために、OHDSI ツールは「選好スコア」と呼ばれる傾向スコアの変換をプロットします@walker_2013。選好スコアは二つの治療の「市場占有率」を調整します。例えば、10% の患者が対象治療を受け (90% の患者が比較対照治療を受ける)、選好スコアが 0.5 の患者は、対象治療を受ける確率が 10% です。数学的には、選好スコアは

$$\ln \left(\frac{F}{1-F} \right) = \ln \left(\frac{S}{1-S} \right) - \ln \left(\frac{P}{1-P} \right)$$

ここで F は選好スコア、 S は傾向スコア、 P は対象治療を受ける患者の割合です。

Walker et al. (2013) は「経験的均衡」の概念を述べています。彼らは、少なくとも半数の曝露が選好スコアの 0.3 から 0.7 の間にある場合、これらの曝露ペアを経験的均衡にあると見なします。

12.1.5 バランス

優れた実践では常に PS 調整がバランスの取れた患者群を生成するかどうかをチェックします。図 12.19 はバランスをチェックするための標準的な OHDSI 出力を示しています。各患者特性について、曝露グループ間の平均の標準化差

Table 12.2: 自己対照コホートデザインの主要なデザイン選択肢。

選択	説明
対象コホート	治療を表すコホート
アウトカムコホート	対象とするアウトカムを表すコホート
リスク時間	アウトカムのリスクをどのタイミング（通常対象コホートの開始および終了日が基準）で考慮するか？
対照時間	対照時間として使用される期間

を PS 調整前後でプロットします。いくつかのガイドラインは、PS 調整後の標準化差の上限を 0.1 とすることを推奨しています (Rubin, 2001)。

12.2 自己対照コホートデザイン

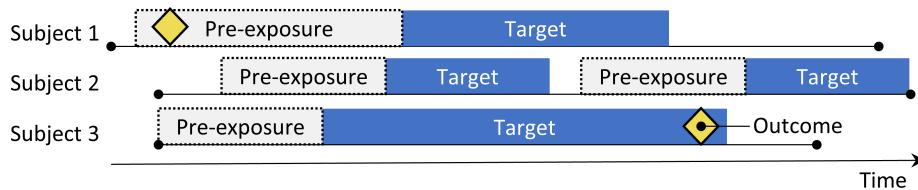


Figure 12.2: 自己対照コホートデザイン。対象への曝露中のアウトカムの発生率を曝露前の期間中の発生率と比較します。

自己対照コホート (SCC) デザイン (Ryan et al., 2013a) は曝露中のアウトカムの発生率を、曝露直前の期間におけるアウトカムの発生率と比較します。表 12.2 に示す 4 つの選択肢が、自己対照コホートの問い合わせを定義します。

曝露群を構成する同じ対象者が対照群としても使用されるため、対象者間の差異を調整する必要はありません。ただし、この方法は、異なる期間間におけるアウトカムのベースラインリスクの差異など、その他の違いには脆弱です。

12.3 症例対照デザイン

症例対照研究 (Vandenbroucke and Pearce, 2012) は、「特定の疾患のアウトカムを持つ人が、その疾患を持たない人よりも特定の因子により曝露される頻度が高いかどうか」を検討します。このため、中心となる考え方は、対象とするアウトカムを経験した対象者（「症例」）を、対象とするアウトカムを経験していない対象者（「対照」）と比較することです。表 12.3 の選択肢が、症例対照の問い合わせを定義しています。

通常、症例を年齢や性別などの特性で一致させて対照を選択し、症例と対照を比較しやすくなります。もう 1 つの広く行われている方法は、対象とする曝露の



Figure 12.3: 症例対照デザイン。アウトカムを持つ対象者（「症例」）は、アウトカムを持たない対象者（「対照」）との曝露状況の観点から比較されます。通常、症例と対照は年齢や性別などの様々な特性で一致するようにします。

Table 12.3: 症例対照デザインの主要なデザイン選択オプション

選択	説明
アウトカムコホート	症例（対象とするアウトカム）を表すコホート
対照コホート	対照を表すコホート。通常、選択ロジックを使用してアウトカムコホートから自動的に導出される
対象コホート	治療を表すコホート
ネスティングコホート	任意で症例および対照が抽出されるサブポピュレーションを定義するコホートを指定
リスク時間	曝露状況をどのタイミング（通常、インデックス日が基準）で考慮するか？

いずれかの適応症と診断されたすべての人々など、特定のサブグループに分けて分析を行うことです。

12.4 ケース・クロスオーバーデザイン

ケース・クロスオーバー (Macleure, 1991) デザインは、アウトカムのタイミングでの曝露率が、アウトカムよりも前の事前に決められた日数での曝露率と異なるかどうかを評価します。これは、アウトカムが発生した日に特別な理由があるかどうかを判断しようとするものです。表 12.4 は、ケース・クロスオーバーの質問を定義するオプションを示します。

症例は自分自身が対照として機能します。自己対照デザインであるため、個人間の差異による交絡に対して頑健であるはずです。ただし、アウトカムの日付が常に対照の日付よりも後であるため、曝露の全体的な頻度が時間とともに増加する（または減少する）場合には、この方法がポジティブにバイアスを受ける可能性があります。この問題に対処するために、ケース・タイム・コントロールデザイン (Suisissa, 1995) が開発され、例えば年齢や性別で一致させた対照をケース・クロスオーバーデザインに追加して、曝露のトレンドを調整します。

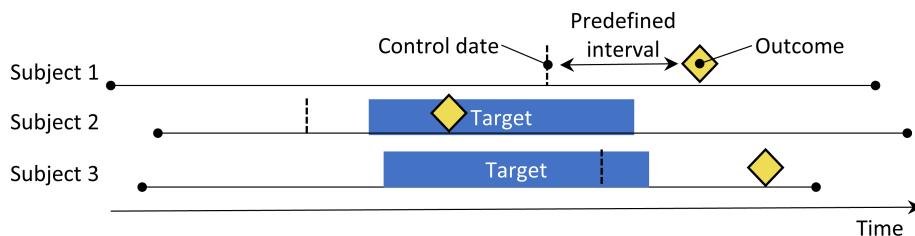


Figure 12.4: ケース・クロスオーバーデザイン。アウトカムの周りの時間を、アウトカムの日付より前の事前に決められた間隔のコントロール日と比較します。

Table 12.4: ケース・クロスオーバーデザインの主要なデザインオプション

選択	説明
アウトカムコホート	症例（対象とするアウトカム）を表すコホート
対象コホート	治療を表すコホート
リスク時間	曝露状況をどのタイミング（通常インデックス日が基準）で考慮するか
対照時間	対照時間として使用される期間

12.5 自己対照症例シリーズデザイン

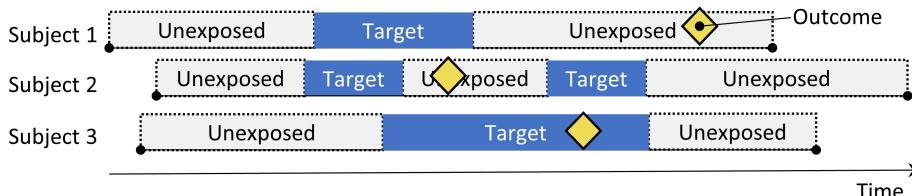


Figure 12.5: 自己対照症例シリーズデザイン。曝露期間中のアウトカム発生率と非曝露期間中のアウトカム発生率を比較する。

自己対照症例シリーズ (SCCS) デザイン (Farrington, 1995; Whitaker et al., 2006) は、曝露期間中のアウトカム発生率を、曝露前、曝露中、曝露後のすべての非曝露期間中の発生率と比較します。これは、個人に依存したポアソン回帰であり、「患者がアウトカムを有する場合、曝露期間中の方が非曝露期間中よりもアウトカムが発生しやすいか？」という問い合わせに対する答えを導きだそうとします。表 12.5 のオプションは SCCS の問い合わせを定義しています。

他の自己対照デザインと同様に、SCCS は個人間の差異による交絡に対して頑健ですが、時間変動による影響による交絡には脆弱です。これらを考慮するためのいくつかの調整が可能であり、たとえば年齢や季節を含めることができます。SCCS の特別なバリエントでは、対象とする曝露だけでなく、データベースに

Table 12.5: 自己対照症例シリーズデザインの主なデザインオプション

選択	説明
対象コホート	治療を代表するコホート
アウトカムコホート	対象とするアウトカムを代表するコホート
リスク期間	どの時点（多くの場合、対象コホートの開始日または終了日と関連のある時点）でアウトカムのリスクを考慮するか？
モデル	時間変動する交絡因子の調整を含む効果の推定モデル

記録された他の薬剤すべての曝露を含める (Simpson et al., 2013) ことで、モデルに数千の追加変数が追加されます。この場合、対象とする曝露以外のすべての曝露の係数に、正則化ハイパーパラメータをクロスバリデーションで選択する L1 正則化が適用されます。

SCCS の重要な仮定の一つは、観察期間の終了がアウトカムの日付とは独立していることです。一部のアウトカム、特に心筋梗塞などの致命的なアウトカムにおいては、この仮定が当てはまらないことがあります。SCCS の拡張版が開発されており、このような依存関係を修正できるものがあります (Farrington et al., 2011)。

12.6 高血圧症研究のデザイン

12.6.1 問題の定義

ACE 阻害薬 (ACEi) は、高血圧症や虚血性心疾患を持つ患者、特にうっ血性心不全、糖尿病、慢性腎臓病などの併存疾患を持つ患者によく使用されます (Zaman et al., 2002)。血管性浮腫は、唇、舌、口、喉頭、咽頭、または眼窩周囲の腫れとして現れる、深刻で時には命に関わる有害事象であり、これらの薬剤の使用と関連付けられています (Sabroe and Black, 1997)。しかし、これらの薬剤使用に関する血管性浮腫の絶対および相対リスクについての情報は限られています。既存のエビデンスは、主に特定のコホート（例えば、主に男性の退役軍人やメディケイド受給者）に基づいたものであり、他の集団に一般化できない可能性があります。また、イベント数が少ない研究に基づくものであり、不安定なリスク推定しかえられません (Powers et al., 2012)。いくつかの観察研究は、ACEi を β 遮断薬と比較して血管性浮腫のリスクを評価しています (Magid et al., 2010; Toh et al., 2012) が、 β 遮断薬はもはや高血圧症の第一選択治療法としては推奨されていません (Whelton et al., 2018)。代替治療法として有効なのは、サイアザイド系およびサイアザイド様利尿薬 (THZ) が考えられます。これらは高血圧症や急性心筋梗塞 (AMI) などの関連リスクを管理する上で同等に有効であり、血管性浮腫のリスクを増加させない可能性があります。

以下では、観察医療データに集団レベル推定フレームワークを適用し、次の比較推定に関する疑問に対処する方法を示します：

ACE 阻害薬の新規使用者とサイアザイド系およびサイアザイド様利尿薬の新規使用者を比較した場合の血管性浮腫のリスクはどれくらいですか？

ACE 阻害薬の新規使用者とサイアザイド系およびサイアザイド様利尿薬の新規使用者を比較した場合の急性心筋梗塞のリスクはどれくらいですか？

これらは比較効果推定の問題であるため、セクション @ref(CohortMethod) で説明されたコホート方法を適用します。

12.6.2 対象および比較対照

高血圧症に対する最初の治療が、ACEi または THZ クラスのいずれかの有効成分による単剤療法であった患者を、新規患者と見なします。治療開始後 7 日以内に他の抗高血圧薬を開始しない患者を単剤療法と定義します。患者は最初の曝露前に少なくとも 1 年間の継続的な観察期間および治療開始前 1 年以内に高血圧症と診断された記録が必要です。

12.6.3 アウトカム

血管性浮腫は、入院または救急外来（ER）訪問中の血管性浮腫のコンセプトに該当するコンディションが発生した場合と定義し、その 7 日前までには血管性浮腫の診断が記録されていないことを必要とします。AMI は、入院または ER 訪問中の AMI コンディションコンセプトの発生として定義し、180 日前までに AMI 診断が記録されていないことを必要とします。

12.6.4 リスク期間

リスク期間を治療開始の翌日から開始し、曝露が終了するまでと定義し、後続の薬剤曝露の間に 30 日間のギャップを許容します。

12.6.5 モデル

デフォルトの共変量セットを使用して PS モデルを適合させます。このセットには、人口統計、病状、薬剤、処置、測定値、観察、いくつかの併存疾患スコアが含まれます。ACEi と THZ を共変量から除外します。変数比率マッチングを行い、マッチングセットに条件付けてコックス回帰を行います。

12.6.6 研究要約

Table 12.6: 比較コホート研究の主なデザインオプション

選択肢	値
対象コホート	高血圧症の第一選択単剤療法としてのACE阻害薬の新規使用者。
比較コホート	高血圧症の第一選択単剤療法としてのサイアザイド系およびサイアザイド様利尿薬の新規使用者。
アウトカムコホート	血管性浮腫または急性心筋梗塞。
リスク期間	治療開始の翌日から開始し、曝露が終了するまで。
モデル	変数比率マッチングを用いたコックス比例ハザードモデル。

12.6.7 コントロールクエスチョン

研究デザインが真実に沿った推定を生成するかどうかを評価するために、真の効果量が既知であるコントロールクエスチョンのセットを追加で含めます。コントロールクエスチョンは、ハザード比が 1 であるネガティブコントロールと、ハザード比が 1 より大きいことが既知であるポジティブコントロールに分けることができます。いくつかの理由により、実際のネガティブコントロールを使用し、これらのネガティブコントロールに基づいてポジティブコントロールを合成します。コントロールクエスチョンの定義と使用方法については、第 18 章で説明しています。

12.7 ATLAS を使用した研究の実施

ここでは、ATLAS の推定機能を使用してこの研究を実施する方法を示します。ATLAS の左側のバーで  Estimation をクリックし、新しい推定研究を作成します。研究に簡単に分かりやすい名前を付けてください。研究デザインは  ボタンをクリックして保存できます。

推定デザイン機能には、比較、分析設定、評価設定の 3 つのセクションがあります。複数の比較と複数の分析設定を指定でき、ATLAS はそれらのすべての組み合わせを個別の分析として実行します。ここでは、それぞれのセクションについて説明します。

12.7.1 比較コホート設定

研究には 1 つ以上の比較を含めることができます。「比較を追加」をクリックすると、新しいダイアログが開きます。 をクリックしてターゲットおよび比較コホートを選択します。「Add Outcome」をクリックして 2 つのアウトカムコホートを追加できます。コホートがすでに ATLAS で作成されていると仮定して

います（第 10 章を参照）。ターゲット（付録 B.2）、比較（付録 B.5）、アウトカム（付録 @ref(Angioedema)、付録 B.3）コホートの完全な定義は付録に記載されています。完了すると、ダイアログは図 12.6 のようになります。

ID	Name	Edit cohort	Remove
1770712	Angioedema outcome	Edit cohort	Remove
1770713	Acute myocardial infarction outcome	Edit cohort	Remove

Figure 12.6: 比較ダイアログ

対象と比較コホートのペアに対して複数のアウトカムを選択できることに注意ください。各アウトカムは独立したものとして扱われ、別々の分析結果が得られます。

ネガティブコントロールアウトカム

ネガティブコントロールのアウトカムは、対象または比較対照によって引き起こされていないと考えられるアウトカムであり、真のハザード比が 1 であると仮定されます。理想的には各アウトカムコホートの適切なコホート定義が必要ですが、通常は各ネガティブコントロールのアウトカムごとに 1 つのコンセプトセットと、それらをアウトカムコホートに変換するための標準的なロジックしか持ちません。ここではコンセプトセットが第 18 章で説明されているとおり、すでに作成されていると仮定し、それを選択するだけです。ネガティブコントロールのコンセプトセットには、ネガティブコントロールごとに 1 つのコンセプトが含まれ、その下位層に含まれるものは含めるべきではありません。図 12.7 は、この研究に使用されたネガティブコントロールのコンセプトセットを示しています。

含めるコンセプト

コンセプトを選択する際、生成したい共変量を指定できます。たとえば、傾向スコアモデルで使用するためです。ここで共変量を指定すると、それ以外の共

Concept Set Expression		Included Concepts 75	Included Source Codes	Explore Evidence	Export	Compare	
Showing 1 to 25 of 75 entries						Search: <input type="text"/>	
Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	Exclude	Descendants	Mapped
72748	74779009	Strain of rotator cuff capsule	Condition	Standard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
73241	197210001	Anal and rectal polyp	Condition	Standard	<input checked="" 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: ネガティブコントロールのコンセプトセット

変量（指定したもの以外）は除外されます。通常、ベースラインのすべての共変量を含め、正則化回帰モデルがすべての共変量をバランスさせるモデルを構築します。特定の共変量を指定する唯一の理由は、手動で共変量を選択した既存の研究を再現する場合です。これらの項目は、この比較セクションまたは分析セクションで指定できます。その理由は、特定の比較に関連する場合（たとえば、比較における既知の交絡因子）、または特定の共変量選択戦略を評価する場合など、分析に関連する場合があるからです。

除外するコンセプト

含めるコンセプトを指定する代わりに、除外するコンセプトを指定することもできます。このフィールドにコンセプトセットを送信すると、送信したコンセプトを除くすべての共変量を使用します。デフォルトの共変量セット（治療開始日のすべての薬剤および処置を含む）を使用する場合、対象の治療と比較治療、およびそれらに直接関連するコンセプトを除外する必要があります。たとえば、対象とする曝露が注射薬である場合、薬剤だけでなく、プロペンシティモデルからその投与手技も除外する必要があります。この例では、除外したい共変量は ACEi と THZ です。図 12.8 は、これらのコンセプトを含むコンセプトセットを示しています（その下位層も含まれます）。

ネガティブコントロールと除外する共変量を選択した後、比較ダイアログの下半分は図 12.9 のようになります。

12.7.2 効果推定の分析設定

比較ダイアログを閉じた後、「Add Analysis Settings」をクリックできます。「Analysis Name」とラベル付けされたボックスには、今後、簡単に検索・参照できるよう固有の名前を入力します。たとえば、「傾向スコアマッチング」という名前を設定することもできます。

Concept Set #1798551										
Concepts to exclude for ACEi and THZ										
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										
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 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 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: 除外するコンセプトを定義するコンセプトセット

Negative control concept set:

Negative controls for ACEi and THZ		
<p>Covariate selection</p> <p>Concepts to include when constructing the covariates to be used in this study. (Leave blank if you want to include every concept).*</p> <input type="text"/> <p>* Concepts defined here are combined with those defined in the Analysis settings section.</p>		
<p>Concepts to exclude when constructing the covariates to be used in this study.*</p> <p>Concepts to exclude for ACEi and THZ</p> <p>* Concepts defined here are combined with those defined in the Analysis settings section.</p>		

Figure 12.9: ネガティブコントロールおよび除外するコンセプトセットを示す比較ウィンドウ

研究対象集団

分析の対象となる対象者の集合である研究対象集団を指定するには、さまざまなオプションがあります。オプションの多くは、コホート定義ツールで対象および比較コホートを設計する際に利用可能なオプションと重複しています。Estimation のオプションを使用する理由の 1 つは再利用性です。ターゲット、比較、アウトカムコホートを完全に独立して定義し、後でそれらの間に依存関係を追加できます。例えば、治療開始前にアウトカムを持っていた人を除外したい場合、対象および比較コホートの定義でこれを行なうことができますが、すべてのアウトカムごとに別のコホートを作成する必要があります。代わりに、分析設定で事前のアウトカムを持つ人々を除外することができ、これで興味のある 2 つのアウトカム（およびネガティブコントロールのアウトカム）に対して、対象および比較コホートを再利用できます。

研究開始日と終了日を使用して、特定の期間に分析を制限できます。研究終了日はリスクウィンドウを切り詰めることになり、研究終了日以降のアウトカムは考慮されません。研究開始日を選択する理由の 1 つは、研究対象の薬剤の 1 つが新しく、以前の期間には存在しなかったことが考えられます。自動で調整するには、「両方の曝露がデータ内に存在する期間に分析を制限しますか？」の質問に「はい」と回答します。研究の開始日と終了日を調整するもう一つの理由は、医療行為が時とともに変化した（例えば、薬の警告による）場合で、特定の方法で医療行為が行われた期間のみに興味がある場合です。

オプション “Should only the first exposure per subject be included? (各対象者の初回の曝露のみを含まれるべきか)” を使用すると、患者ごとの最初の曝露に限定することができます。多くの場合、この例のようにコホート定義ですでに行われています。同様に、「The minimum required continuous observation time prior to index date for a person to be included in the cohort (コホートに含める対象者のインデックス日付以前の最小限必要な連続観察期間)」というオプションは、コホート定義すでに設定されていることが多いので、ここでは 0 のままにしておきます。インデックス日より前に観察時間がある(OBSERVATION_PERIOD テーブルで定義されているように) ことは、傾向スコアを計算するのに十分な患者に関する情報があること、また患者が真の新規ユーザーであり、したがって以前に曝露されていないことを担保するためにもよく使われます。

“Remove subjects that are in both target and comparator cohort? (対象コホートと比較群コホートの両方に含まれる対象を除外しますか)” は、“If a subject is in multiple cohorts, should be censored time-at-risk when the new time-at-risk starts to prevent overlap? (対象が複数のコホートに含まれる場合、新しいリスク評価期間が開始された際に、重複を避けるためにリスク評価期間を打ち切りますか)” というオプションと併せて、対象が対象コホートと比較コホートの両方に存在する場合にどのように取り扱うかを定義します。最初の設定には 3 つの選択肢があります：

- “Keep All (すべて保持)” は、両方のコホートに対象を保持することを意味します。このオプションでは、対象者とアウトカムがダブルカウントさ

れる可能性があります。

- “Keep First (最初を保持)” は、最初に発生したコホートに対象者を残すことを意味します。
- “Remove All (すべて除外)” は、すべてのコホートから対象者を除外することを意味します。

もし “Keep All” または “Keep First” のオプションが選択された場合、ある対象が両方のコホートにいる時間打ち切りたいと思うかもしれません。これを図 12.10 に示します。デフォルトでは、リスク期間はコホートの開始日と終了日を基準に定義されます。この例では、リスク期間はコホート組入れの 1 日後に始まり、コホート終了までです。2 つのコホートが重なる可能性があるリスク期間を打ち切らない場合、この重複したリスク期間に発生したアウトカムは、特に、すべてを保持することを選択した場合、(ここに示されているように) 重複した時間に出現したアウトカムは 2 回カウントされるため、問題となります。打ち切りを選択した場合、最初のコホートのリスク期間は、2 番目のコホートのリスク時間の開始時に終了します。

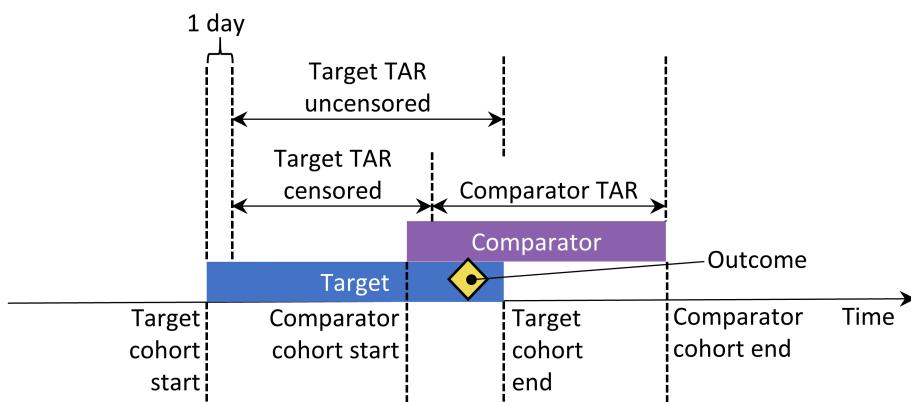


Figure 12.10: リスク時間 (Time-at-risk (TAR)) が薬剤曝露開始日から薬剤曝露終了時までと仮定した場合の 2 つのコホートに含まれる対象のリスク期間

アウトカムの 2 回目の出現は 1 回目の継続であることが多いため、リスクウインドウが始まる前にアウトカムが出現した対象を除外することを選択できます。例えば、ある人が心不全を発症した場合、2 回目の心不全の出現はよくあることでしょう、なぜなら 1 回目と 2 回目の心不全の間でその心不全は完全に治癒することがなかったということになるからです。一方、いくつかのアウトカムは間欠的なもので、上気道感染症のように、患者が複数の独立したアウトカムを持つことが予想されます。以前にそのアウトカムを経験した人を除外する場合、以前にアウトカムが出現したことを見定する際に何日前まで遡るかを選択できます。

、図 12.11 に、この研究例における選択を示します。対象コホートと比較対照コホートの定義は、すでに初回曝露に限定しており、治療開始前の観察時間が必要なので、ここではこれらの基準は適用しません。

 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)

Should only the first exposure per subject be included?

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?

If a subject is in multiple cohorts, should time-at-risk be censored when the new time-at-risk start to prevent overlap?

Remove subjects that have the outcome prior to the risk window start?

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: 研究対象集団の設定

共変量の設定

ここでは構築する共変量を指定します。これらの共変量は通常、傾向スコアモデルで使用されますが、アウトカムモデル（この場合は Cox 比例ハザードモデル）にも含めることもできます。click to view details (詳細を見るにはここをクリック) をクリックすると、どの共変量の組み合わせを使用するか、選択することが出来ます。しかし、人口統計学的要素、コンディションすべて、薬剤、処置、測定値などを含むデフォルト

include (組入れ) および/または exclude (除外) するコンセプトを指定することで、共変量のセットを変更できます。これらの設定は、セクション 12.7.1 にある比較のための設定と同じです。これらの設定が 2 つの場所にある理由は、これらの設定が特定の比較に関連している場合があるためです。特定の比較に対して特定の分析設定を使用して分析を実行する場合、OHDSI ツールはこれらのセットの共通部分を使用します。

図 12.12 は、この研究で選択した内容を示しています。図 12.9 に定義するように、比較の設定で除外するコンセプトに下位層を追加することを選択していることに注意ください。

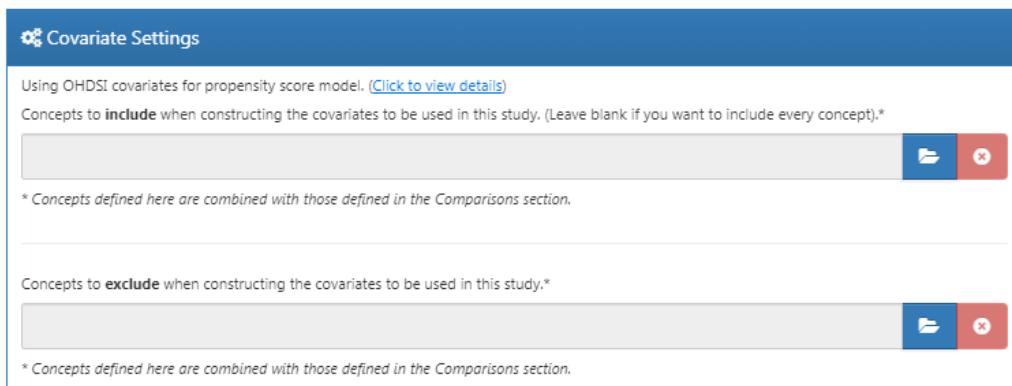


Figure 12.12: 共変量の設定

リスク期間

リスク期間 (Time-at-risk) は、対象コホートや比較コホートにおける開始日と終了日を基準に定義されます。例では、対象集団の開始日を治療開始日とし、終了日を曝露が停止した日（少なくとも 30 日間）としました。リスクにさらされている期間の開始日を、対象集団の開始日の翌日、つまり治療開始日の翌日としました。コホート開始日よりも後の時点をリスク期間開始日とする理由は、生物学的にはその薬剤が原因である可能性が低いと考える場合、治療開始日に発生したアウトカム事象を除外したい場合があるからです。

リスク期間の終了日は、コホートの終了日、つまり曝露が停止した時点としました。例えば、治療終了直後の事象が依然として曝露に起因すると考えられる

場合は、終了日を遅く設定することもできます。極端な例では、コホート終了日の後、長い日数を経て（例えば 99999 日）リスク期間の終了日に設定することもできます。このようなデザインは intent-to-treat デザインと呼ばれることもあります。

リスク時間がゼロの患者は何の情報も提供しないので、最小リスク日数は通常 1 日に設定されます。副作用の潜伏期間がわかっている場合は、より意味のある発生割合を得るために、この日数を増やすことができます。また、比較するランダム化試験に近いコホートを作成するために使用することもできます（例えば、ランダム化試験のすべての患者が少なくとも N 日間観察された）。



コホート研究を計画する際の鉄則は、バイアスが含まれる可能性を排除するため、コホート開始日以降の情報を研究集団の定義に使用しないことです。例えば、全対象者に少なくとも 1 年間のリスク期間を要求した場合、解析対象は、治療に十分耐えられる人に限定することになります。そのため、この設定は細心の注意を払って行う必要があります。

The screenshot shows the 'Time At Risk' configuration screen. It has a blue header bar with the title 'Time At Risk'. Below it is a white form area with the following fields:

- '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 number of days at risk?
1 ▾'

Figure 12.13: リスク期間の設定

傾向スコアによる調整

傾向スコア値が極端な人を除外して、研究対象集団をトリミングすることができます。上位と下位の何パーセントを除外するか、または選好スコアが指定した範囲から外れる対象を除外するかを選択できます。コホートのトリミングは、一般的に推奨されません。なぜなら、観察を破棄する必要があり、統計的パワーが低下するからです。IPTW を使用する場合など、一部のケースではトリミングが望ましい場合があります。

トリミングに加えて、またはトリミングの代わりに、傾向スコアで層別化またはマッチングを選択することができます。層別化する場合は、層数と、対象集団、比較対象集団、研究対象集団全体のいずれに基づいて層を選択するかを指定する必要があります。マッチングの際には、比較対照群から対象群の各人にマッチさせる最大人数を指定する必要があります。典型的な値は、1 対 1 のマッチングの場合は 1、変数比率マッチングの場合は大きな数（例えば 100）になります。また、キャリパー、すなわちマッチングする傾向スコア間の最大許

容差を指定する必要があります。キャリパーは差のキャリパー・スケールで定義できます：

- ・傾向スコア尺度：傾向スコアそのもの
- ・標準化尺度：傾向スコア分布の標準偏差による
- ・標準化ロジット尺度：傾向スコアをより正規分布に近づけた後のプロスペクティブ・スコア分布の標準偏差。

疑問がある場合は、デフォルト値を使用するか、このトピックに関する@austin_2011 の研究を参照ください。

大規模な傾向スコアモデルの適合は計算コストがかかることがあるので、モデルの適合に使用するデータをデータのサンプルだけに制限することが望ましい場合があります。デフォルトでは、対象コホートと比較対照コホートの最大サイズは 250,000 に設定されています。ほとんどの研究では、この上限に達することはありません。また、データが多ければ多いほど、より良いモデルになることもあります。データのサンプルはモデルのフィットに適合させることはできますが、そのモデルは集団全体の傾向スコアを計算するために使用されることに注意してください。

Test each covariate for correlation with the target assignment? (各共変量とターゲットの割付の相関を検定しますか。) 共変量が異常に高い相関（正または負）を持つ場合、エラーが発生します。これにより、完全に分離していることが判明するまで、傾向モデルの計算が長時間行われることを避けることができます。非常に高い単変量相関が見つかった場合は、その共変量を検証し、相関が高い理由と削除すべきかどうかを判断できます。

Use regularization when fitting the model? (モデルを適合する際に正則化しますか。) 標準的な手順では、傾向モデルに多くの共変量（通常 10,000 以上）を含めます。このようなモデルを適合させるには何らかの正則化が必要です。少數の厳選された共変量のみが含まれる場合は、正則化なしでモデルを適合させることも可能です。

図 12.14 は、この研究での選択を示しています。最大マッチング人数を 100 人に設定することで、可変比率マッチングを選択していることに注意してください。

アウトカムモデルの設定

最初に、対象コホートと比較コホート間のアウトカムの相対リスクを推定するために使用する統計モデルを指定する必要があります。セクション 12.1 で簡単に述べたように、Cox、Poisson、ロジスティック回帰から選択できます。この例では、打ち切りの可能性がある最初のイベントまでの時間を考慮する Cox 比例ハザード・モデルを選択します。次に、回帰を層で条件付けるかどうかを指定する必要があります。条件づけを理解する 1 つの方法は、各層で別々の推定値が生成され、そして層で結合されることを考えることです。1 対 1 のマッチ

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** ▼

This screenshot shows the 'Propensity Score Adjustment' configuration page. It includes sections for trimming cohorts, performing matching or stratification, setting caliper parameters, specifying the number of people included in the propensity score model, testing covariates for correlation, and using regularization. At the bottom, there are tabs for 'Control Settings' and 'Prior'.

Figure 12.14: 傾向スコアによる調整の設定

ングでは、これは不要で、むしろ検出力を失うことになります。層別マッチングや可変比率マッチングでは必要です。

また、共変量をアウトカムモデルに追加して分析を調整することもできます。これは傾向モデルを使うことに加えて、または代わりに行うことができます。しかし、傾向モデルに適合させるのに十分なデータが通常は存在し、両方の治療グループに多くの人が含まれる一方で、結果モデルに適合させるのに十分なデータは通常はほとんど存在せず、結果を持つ人はわずかしかいません。そのため、結果モデルはできるだけシンプルに保ち、追加の共変量を含めないことをお勧めします。

傾向スコアで層別化またはマッチングする代わりに、逆確率重み付け (IPTW) を用いることもできます。

アウトカムモデルにすべての共変量を含めることを選択した場合、共変量が多ければ、モデルを適合させるとときに正則化を使用することが理にかなっているかもしれません。不偏推定を可能にするために、治療変数には正則化は適用されないことに注意してください。

図 12.15 は、この研究での選択を示しています。可変比率マッチングを用いているため、回帰モデルでは層別条件付けをしなければなりません（マッチしたセットによって）。

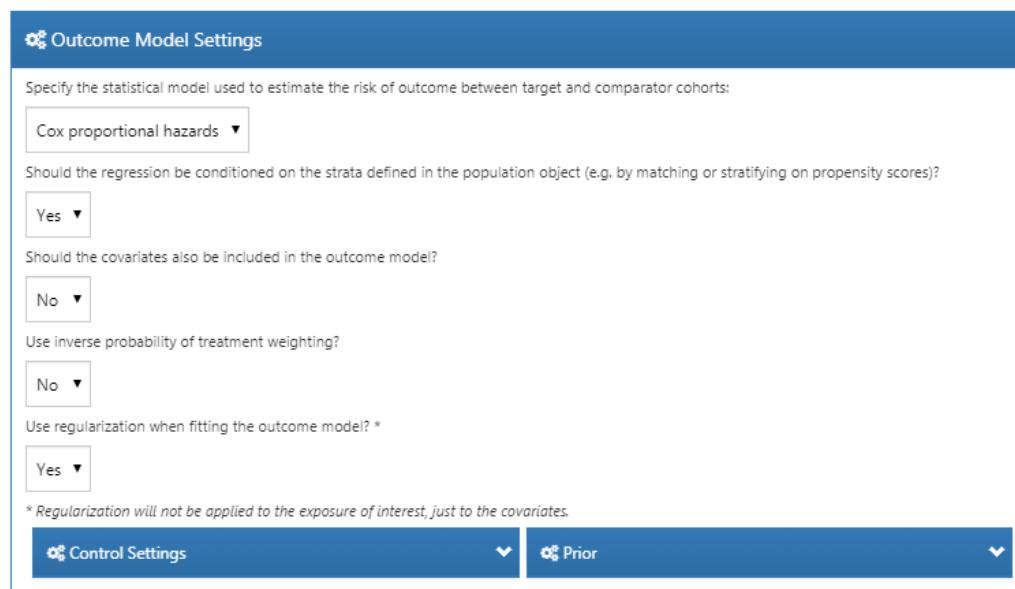


Figure 12.15: アウトカムモデルの設定

12.7.3 評価の設定

第 18 章にあるように、ネガティブコントロールとポジティブコントロールを検討し、操作特性を評価し、経験的キャリブレーションを行う必要があります。

ネガティブコントロールのアウトカムコホートの定義

セクション 12.7.1 では、ネガティブコントロールのアウトカムを表すコンセプトセットを選択しました。しかし、分析でアウトカムとして使用するために、コンセプトをコホートに変換するロジックが必要です。ATLAS は 3 つの選択肢を持つ標準ロジックを提供します。最初の選択肢は、コンセプトのすべての出現を使用するか、最初の出現のみを使用するかです。2 番目の選択肢は、下位層のコンセプトの出現を考慮するかどうかを決定します。例えば、下位層の “*ingrown nail of foot* (足の陷入爪)” の出現も、上位層の “*ingrown nail* (陷入爪)” の出現として数えることができます。3 番目の選択肢は、コンセプトを探すときにどのドメインを考慮するかを指定します。

Negative Control Outcome Cohort Definition

This expression will define the criteria for inclusion and duration of time for cohorts intended for use as negative control outcomes. The type of occurrence of the event when selecting from the domain.

First occurrence ▾

When true, descendant concepts for the negative control outcome concept IDs will be used to detect the outcome and roll up the occurrence to the concept ID.

Yes ▾

What domains should be considered to detect negative control outcomes? (Hold control to select multiple domains)

Condition
Drug
Device
Measurement
Observation
Procedure
Visit

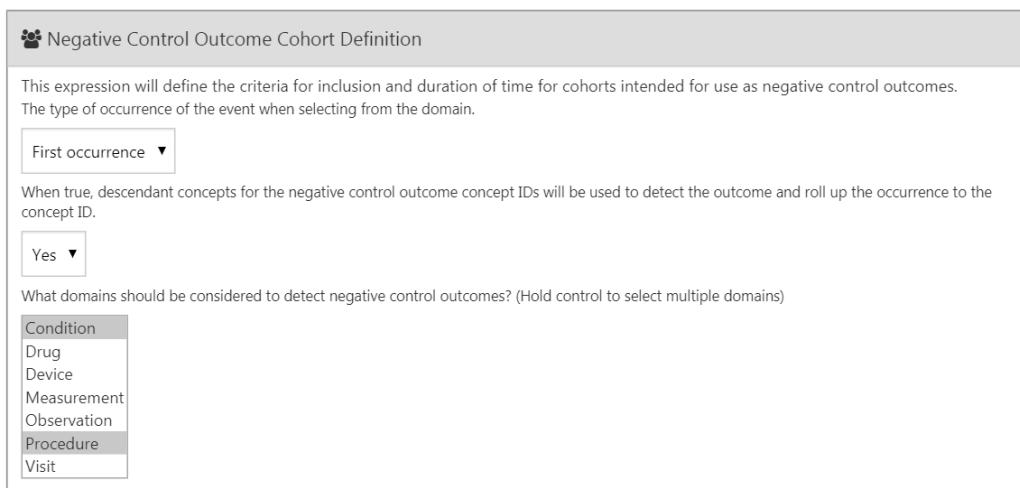


Figure 12.16: ネガティブコントロールのアウトカムコホートの定義の設定

ポジティブコントロールの合成

ネガティブコントロールに加えて、因果関係があると思われる曝露-アウトカムのペアで、効果量が既知であるポジティブコントロールも含めることができます。様々な理由から、実際のポジティブコントロールには問題があるため、代わりに、第 18 章で説明したように、ネガティブコントロールから得られる合成ポジティブコントロールを用いることがあります。ポジティブコントロールの合成を行うかどうかを選択できます。もし「はい」であれば、モデル・タイプを選択しなければなりませんが、現在の選択は「Poisson」と「survival」です。集団レベルの推定の研究では生存 (Cox) モデルを使用するので、「survival」を選択します。ポジティブコントロール合成のためのリスク期間モデルを推定の設定と同じになるように定義し、同じく、曝露前に最低限必要な連続した観察期間、最初の曝露のみを含めるべきか、最初のアウトカムのみを含めるべきか、および過去にアウトカムを持つ人を除外するの選択肢を模倣します。図 12.15 にポジティブコントロール合成の設定を示します。

⚙ Positive Control Synthesis

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: ポジティブコントロールのアウトカムの定義の設定

12.7.4 研究パッケージの実行

これで研究の定義が完了したので、実行可能な R パッケージとしてエクスポートできます。このパッケージは、CDM にデータを持つ施設で研究を実行するために必要なすべての内容を含みます。これには、対象群、比較群、アウトカムのコホートをインスタンス化するために使用できるコホート定義、ネガティブコントロールのコンセプトセット、ネガティブコントロールのアウトカムコホートを作成するロジック、さらに分析を実行する R コードが含まれます。パッケージを生成する前に、必ず研究を保存し、Utilities (ユーティリティ) タブをクリックしてください。ここで、実行される一連の分析をレビューできます。前述したように、比較と分析設定のすべての組み合わせは、別々の分析になります。この例では、2つの解析を指定しています：AMI に対する ACEi 対 THZ、血管性浮腫に対する ACEi 対 THZ、両者とも傾向スコアマッチングを使用しています。

パッケージの名前を指定し、“Download (ダウンロード)” をクリックして zip ファイルをダウンロードします。zip ファイルには R パッケージが含まれており、R パッケージに通常必要なフォルダ構成になっています (Wickham, 2015)。このパッケージを使用するには、R Studio の使用をお勧めします。R Studio をローカルで実行している場合は、ファイルを解凍し、.Rproj ファイルをダブルクリックして R Studio で開きます。R スタジオを R スタジオサーバーで実行している場合は、 Upload をクリックしてファイルをアップロードし、解凍した後、.Rproj ファイルをクリックしてプロジェクトを開きます。

R Studio でプロジェクトを開いたら、README ファイルを開き、指示に従ってください。すべてのファイルのパスを、システム上の既存のパスに変更してください。

研究の実行時に表示される一般的なエラーメッセージは、“High correlation between covariate(s) and treatment detected.” (共変量と治療の間に高い相関が検出されました。) です。これは傾向モデルのフィッティングの際に、いくつかの共変量が曝露と高い相関があることが観察されたことを示します。エラーメッセージに記載されている共変量を確認し、適切に共変量セットから除外してください (セクション 12.1.2 参照)。

12.8 R を使用した研究の実施

ATLAS を使用して研究を実行する R コードを記述する代わりに、R コードを自分自身で書くこともできます。これを行う理由の一つは、R が ATLAS で公開されているものよりもはるかに柔軟性を提供するからです。例えば、カスタム共変量や線形アウトカムモデルを使用したい場合は、カスタム R コードを作成し、OHDSI R パッケージが提供する機能と組み合わせる必要があります。

例として、CohortMethod パッケージを使用して研究を実行します。CohortMethod は、CDM に含まれるデータベースから必要なデータを抽出し、プロペンシティモデルのための多数の共変量を利用できます。以下の例では、最初に

アウトカムとして血管性浮腫のみを考慮します。セクション 12.8.6では、これを拡張して AMI とネガティブコントロールのアウトカムを含める方法を説明します。

12.8.1 コホートのインスタンス化

最初に対象コホートおよびアウトカムコホートをインスタンス化する必要があります。コホートのインスタンス化は、セクション(10)で説明しています。付録にはターゲット（付録(B.2)）、比較（付録(B.5)）、およびアウトカム（付録(B.4)）コホートの完全な定義が示されています。ACEi、THZ、および血管性浮腫コホートが、それぞれコホート定義 ID 1、2、3 である scratch.my_cohorts という表にインスタンス化されていると仮定します。

12.8.2 データ抽出

最初に、R にサーバーへの接続方法を教える必要があります。CohortMethodはDatabaseConnectorパッケージを使用しており、createConnectionDetails という関数を提供しています。さまざまなデータベース管理システム (DBMS) に必要な特定の設定については、?createConnectionDetails と入力してください。たとえば、PostgreSQL データベースに接続するには、以下のコードを使用します。:

```
library(CohortMethod)
connDetails <- createConnectionDetails(dbms = "postgresql",
                                         server = "localhost/ohdsi",
                                         user = "joe",
                                         password = "supersecret")

cdmDbSchema <- "my_cdm_data"
cohortDbSchema <- "scratch"
cohortTable <- "my_cohorts"
cdmVersion <- "5"
```

最後の 4 行は cdmDbSchema、cohortDbSchema、cohortTable 変数と CDM バージョンを定義しています。これらは後ほど R に CDM 形式のデータがどこに格納されているか、対象となるあるコホートがどこに作成されたか、そして使用されている CDM のバージョンを伝えるために使用します。Microsoft SQL Server の場合、データベーススキーマはデータベースとスキーマの両方を指定する必要があるため、たとえば cdmDbSchema <- "my_cdm_data.dbo" のようになります。

次に、CohortMethod にコホートを抽出し、共変量を構築し、分析に必要なすべてのデータを抽出するよう指示できます：

```
# ターゲットおよび比較対照薬剤の成分のコンセプト
aceI <- c(1335471, 1340128, 1341927, 1363749, 1308216, 1310756, 1373225,
          1331235, 1334456, 1342439)
thz <- c(1395058, 974166, 978555, 907013)

# 構築すべき共変量のタイプを定義
cs <- createDefaultCovariateSettings(excludedCovariateConceptIds = c(aceI,
                                                                     thz),
                                         addDescendantsToExclude = TRUE)

# データをロード
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 オブジェクト
##
## 治療コンセプトID : 1
## 比較対照コンセプトID : 2
## アウトカムコンセプトID(s) : 3
```

多くのパラメーターがありますが、すべてCohortMethod マニュアルに記載されています。createDefaultCovariateSettings 関数はFeatureExtractionパッケージで説明されています。簡単に言えば、コホートを含むテーブルを指定し、そのテーブル内で対象、比較対照、アウトカムを識別するコホート定義 ID を指定します。デフォルトの共変量セットが構築される指示を行い、インデックス日前日までに見つかったすべてのコンディション、薬剤曝露、処置に関する共変量を含むようにします。セクション(12.1)で述べたように、共変量のセットから対象と比較対照の治療を除外する必要があり、ここでは、2つのクラスのすべての成分を一覧表示し、FeatureExtraction にこれらの成分を含む下位層のすべての薬剤を除外するように指示します。

コホート、アウトカム、共変量に関するすべてのデータはサーバーから抽出され、`cohortMethodData` オブジェクトに保存されます。このオブジェクトは `ff` パッケージを使用して情報を保存するため、データが大きくても R がメモリ不足にならないようにします（セクション (8.4.2) で述べた通りです）。

抽出したデータの詳細を確認するために、汎用 `summary()` 関数を使用できます：

```
summary(cmData)
```

```
## CohortMethodDataオブジェクトの要約
##
## 治療コンセプトID : 1
## 比較対照コンセプトID : 2
## アウトカムコンセプトID(s) : 3
##
## 治療を受けた人数 : 67166
## 比較対照の人数 : 35333
##
## アウトカウント :
##           イベント数      人数
## 3            980        891
##
## 共変量 :
## 共変量の数 : 58349
## ゼロでない共変量値の数 : 24484665
```

`cohortMethodData` ファイルの作成にはかなりの計算時間がかかる可能性がありますので、今後のセッションのために保存しておくのが良いでしょう。`cohortMethodData` は `ff` を使用するため、R の通常の保存関数は使用できません。代わりに、`saveCohortMethodData()` 関数を使用します：

```
saveCohortMethodData(cmData, "AceiVsThzForAngioedema")
```

今後のセッションでデータをロードするには、`loadCohortMethodData()` 関数を使用できます。

新規ユーザーの定義

通常、新規ユーザーは薬剤（対象または比較対象）の初回使用として定義され、通常、ウォッシュアウト期間（初回使用前の最小日数）を使用して、それが本当に初回使用である可能性を高めます。CohortMethod パッケージを使用する場合、新規使用に必要な要件を 3 つの方法で適用できます。：

1. コホートを定義する場合。

2. コホートを `getDbCohortMethodData` 関数を使用して読み込む際、`firstExposureOnly`、`removeDuplicateSubjects`、`restrictToCommonPeriod`、および `washoutPeriod` 引数を使用。
3. `createStudyPopulation` 関数を使用して研究集団を定義する際（下記参照）。

オプション 1 の利点は、入力コホートがすでに `CohortMethod` パッケージ外で完全に定義されているため、外部コホート特性化ツールをこの分析で使用するのと同じコホートに使用できることです。オプション 2 および 3 の利点は、`DRUG_ERAS` テーブルを直接使用できるなど、自身で初回使用に制限する手間を省くことです。オプション 2 は 3 よりも効率的であるため、最初の使用に必要なデータを取得するだけで済みますが、オプション 3 は効率が低いものの、元のコホートと研究対象集団とを比較することができます。

12.8.3 研究集団の定義

通常、曝露コホートと結果コホートは互いに独立して定義されます。効果量の推定値を算出したい場合、これらのコホートをさらに制限し、まとめておく必要があります。例えば、曝露前に結果が判明している被験者を除外し、定義されたりスク期間内の結果のみを残すなどです。この目的には、`createStudyPopulation` 関数を使用できます。：

```
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")
```

`firstExposureOnly` と `removeDuplicateSubjects` を `FALSE` に設定し、`washoutPeriod` を 0 に設定しているのは、コホート定義内でこれらの基準をすでに適用しているためです。使用するアウトカム ID を指定し、リスク期間の開始日より前にアウトカムがある対象者を削除するように指示します。リスク期間はコホート開始日の翌日から始まり (`riskWindowStart = 1` および `startAnchor = "cohort start"`)、リスク期間はコホート定義で定義された曝露終了時に終了します (`riskWindowEnd = 0` および `endAnchor = "cohort end"`)。リスク期間は自動的に観察終了時または研究終了日に切り捨てられます。リスクの時間がない対象者も削除します。研究集団に残っている人数を確認するには、`getAttritionTable` 関数を使用できます：

```
getAttritionTable(studyPop)
```

##	説明	ターゲット人数	比較群人数	...
## 1	元のコホート	67212	35379	...
## 2	両コホートの重複削除	67166	35333	...
## 3	前のアウトカムなし	67061	35238	...
## 4	リスク期間が1日以上有り	66780	35086	...

12.8.4 傾向スコア

`getDbcCohortMethodData()` で構築された共変量を使用してプロペンシティモデルを適合させ、各人に対して傾向スコア (PS) を計算します：

```
ps <- createPs(cohortMethodData = cmData, population = studyPop)
```

`createPs` 関数は Cyclops パッケージを使用して大規模な正則化ロジスティック回帰を適合します。プロペンシティモデルを適合するために、Cyclops は事前分布の分散を指定するハイパーパラメータ値を知る必要があります。デフォルトでは、Cyclops はクロスバリデーションを使用して最適なハイパーパラメータを推定します。ただし、これには非常に長い時間がかかる場合があります。`createPs` 関数の事前および制御パラメータを使用して、Cyclops の動作を指定することができます。これには、クロスバリデーションを高速化するための複数の CPU の使用などが含まれます。

ここでは、変数比のマッチングを使用して PS を使用します：

```
matchedPop <- matchOnPs(population = ps, caliper = 0.2,  
                        caliperScale = "standardized logit", maxRatio = 100)
```

10.2.5 二十九六二三

アウトカムモデルは、どの変数がアウトカムと関連しているかを説明するモデルです。厳密な仮定の下では、治療変数の係数は因果効果として解釈できます。ここではマッチングに基づいた Cox 比例ハザードモデルを適合します：

```
## モデルタイプ : cox
## 階層化 : TRUE
## 共変量の使用 : FALSE
## 治療重量の逆確率 : FALSE
## ステータス : OK
##
##          推定値 下限95% 上限95% logRr   seLogRr
## 治療      4.3203   2.4531   8.0771  1.4633   0.304
```

12.8.6 複数の分析の実行

一般に、ネガティブコントロールを含む多くのアウトカムに対して複数の分析を行いたい場合がよくあります。CohortMethodは、そのような研究を効率的に実行するための関数を提供します。これは複数の分析の実行に関するパッケージのビネットで詳細に説明されています。要約すると、対象となるアウトカムとネガティブコントロールのコホートが既に作成されていると仮定し、分析したいすべての対象・比較対照・アウトカムの組み合わせを指定できます：

```
# 関心のあるアウトカム :
ois <- c(3, 4) # Angioedema, AMI

# ネガティブコントロール :
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)
```

次に、先ほどの例で説明した様々な関数を呼び出す際に、どのような引数を使うべきかを指定します：

```
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)

```

次に、これらを 1 つの分析設定オブジェクトに結合し、一意の分析 ID といくつかの説明を提供します。1 つ以上の分析設定オブジェクトをリストにまとめることができます：

```

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)

```

これで、すべての比較と分析設定を含む研究を実行することができます：

```

result <- runCmAnalyses(connectionDetails = connectionDetails,
                         cdmDatabaseSchema = cdmDatabaseSchema,
                         exposureDatabaseSchema = cohortDbSchema,
                         exposureTable = cohortTable,
                         outcomeDatabaseSchema = cohortDbSchema,
                         outcomeTable = cohortTable,
                         cdmVersion = cdmVersion,
                         outputFolder = outputFolder,
                         cmAnalysisList = cmAnalysisList,
                         targetComparatorOutcomesList = tcosList)

```

result オブジェクトには、作成されたすべてのアーティファクトへの参照が含まれます。例えば、AMI のアウトカムモデルを取得することができます：

```

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
##
##          推定値 下限95% 上限95% logRr   seLogRr
## 治療    1.1338    0.5921    2.1765 0.1256    0.332

```

また、1つのコマンドですべてのアウトカムに対する効果量として推定値を取得することもできます：

```

summ <- summarizeAnalyses(result, outputFolder = outputFolder)
head(summ)

```

	解析ID	ターゲットID	比較群ID	アウトカムID	リスク比 ...
## 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 研究の結果

推定値は、いくつかの仮定が満たされている場合にのみ有効です。これが満たされているかどうかを評価するために、幅広い診断ツールを使用します。これらは ATLAS によって生成された R パッケージが生成したアウトカムで利用可能であり、特定の R 関数を使用して隨時生成することもできます。

12.9.1 傾向スコアとモデル

まず、対象コホートと比較対象コホートがある程度比較可能かどうかを評価する必要があります。そのために、傾向モデルの Area Under the Receiver Operator Curve (AUC) 統計量を計算できます。AUC が 1 の場合、治療の割り当てはベースライン共変量に基づいて治療割り当てが完全に予測可能であり、したがって、2つのグループは比較不可能であることを示します。computePsAuc 関数を使用して AUC を計算できます。この例では 0.79 です。plotPs 関数を使用して、図 12.18 に示すような選好スコア分布も生成できます。この図から、多くの人にとって受けた治療が予測可能だったことがわかりますが、重複も多く、調整して比較可能なグループを選択することができます。

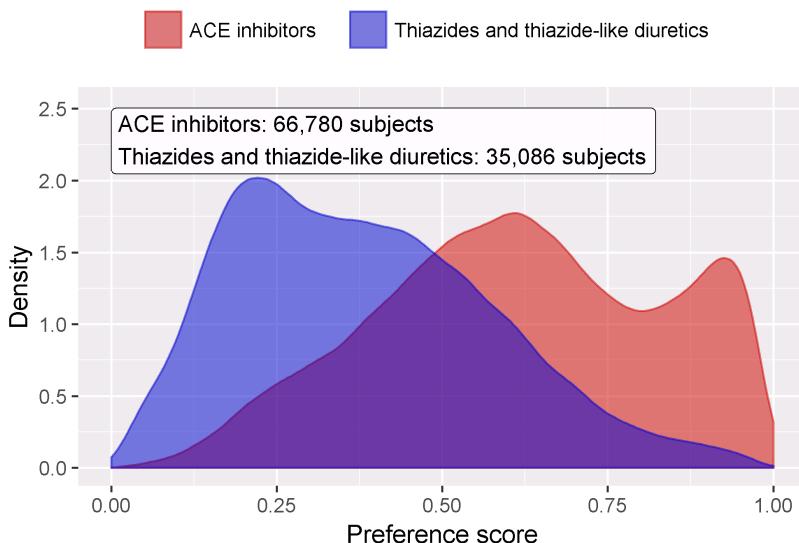


Figure 12.18: 選好スコアの分布

一般に、傾向モデル自体も検査することが望ましく、特にモデルが非常に予測的な場合はそのようにすべきです。そうすれば、どの変数が最も予測的であるかがわかるかもしれません。表 12.7 は、この傾向モデルにおける主要な予測因子を示しています。変数があまりにも予測的である場合、CohortMethod パッケージは情報的なエラーを発生させますが、すでに完全に予測可能であることがわかっているモデルを適合させようとはしません。

Table 12.7: ACEi と THZ の傾向モデルにおける上位 10 の予測因子。正の値は、共変量を持つ対象が治療を受ける可能性が高いことを意味します。「(Intercept)」は、このロジスティック回帰モデルの切片を示します。

ベータ	共変量
-1.42	基準日から-30 日から 0 日までの期間の疾患エラ: 浮腫
-1.11	基準日から 0 日から 0 日までの期間の薬剤エラ: 塩化カリウム
0.68	年齢グループ: 05-09
0.64	基準日から-365 日から 0 日までの期間のメジャーメント: レニン
0.63	基準日から-30 日から 0 日までの期間の疾患エラ: 莖麻疹
0.57	基準日から-30 日から 0 日までの期間の疾患エラ: タンパク尿
0.55	基準日から-365 日から 0 日までの期間の薬剤エラ: インスリン及び類似体
-0.54	人種: 黒人またはアフリカ系アメリカ人
0.52	(切片)
0.50	性別: 男性



変数が非常に予測的であると判明した場合、2つの可能な結論があります。変数が明らかに曝露の一部であると判明し、モデルを適合させる前に削除する必要があるか、または2つの集団が本当に比較不可能であり、解析を中止しなければならないという結論に達します。

12.9.2 共変量のバランス

PS を使用する目的は、2つのグループを比較可能にすることです（少なくとも比較可能なグループを選択すること）。これが達成されたかどうかを確認する必要があります。例えば、調整後のベースライン共変量が実際にバランスしているかどうかを確認することです。`computeCovariateBalance` および `plotCovariateBalanceScatterPlot` 関数を使用して図 12.19 を生成できます。経験則として、傾向スコア調整後には、共変量間の平均値の標準化差の絶対値が 0.1 より大きくなってはならないというものがあります。ここでは、マッチング前にはかなりの不均衡があったものの、マッチング後にはこの基準を満たしていることがわかります。

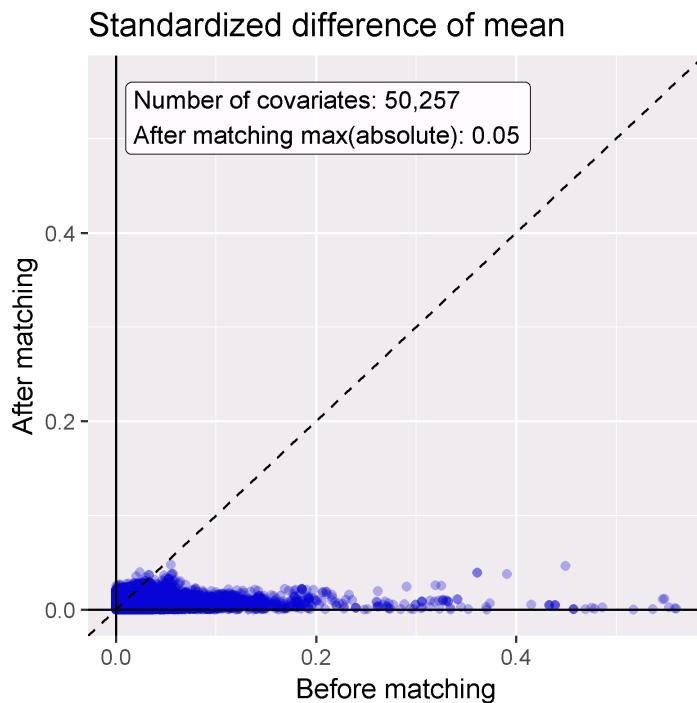


Figure 12.19: 共変量バランスの図。傾向スコアマッチング前およびマッチング後の平均の絶対標準化差を示す。各ドットは共変量を表します。

12.9.3 フォローアップとパワー

アウトカムモデルを適合させる前に、特定の効果量を検出するのに十分な検出力があるかどうかを知りたい場合があります。様々な適格基準および除外基準(例えば、事前のアウトカムなし)による脱落、マッチングおよび/またはトリミングによる脱落を考慮に入れるため、研究対象集団が完全に定義された時点で、これらの検出力の計算を行うことが理にかなっています。drawAttritionDiagram 関数を使用して、研究対象者の脱落を把握することができます。図 12.20 を参照ください。

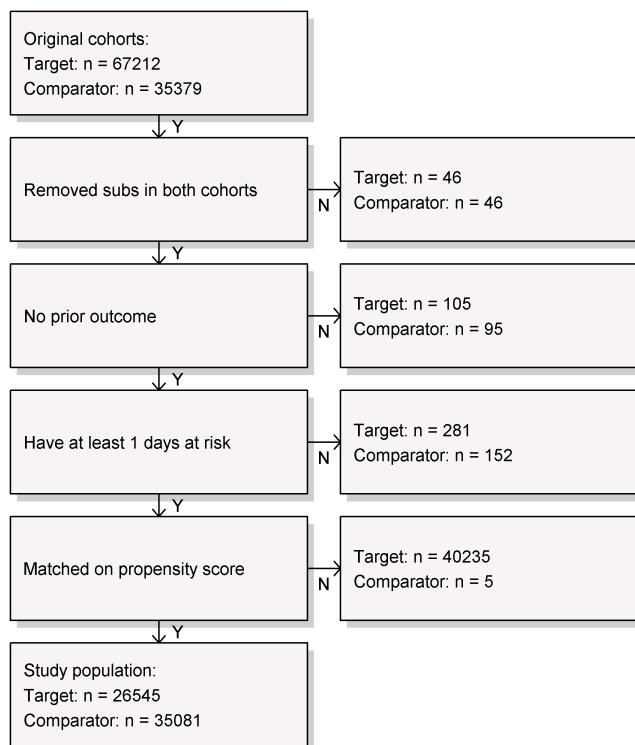


Figure 12.20: 脱落図。上部に示されているカウントは目標および比較対象コホートの定義を満たしているものです。下部に示されているカウントは、アウトカムモデルに入るものです。この場合、Cox 回帰です。

レトロスペクティブ研究ではサンプルサイズは固定されており(データはすでに収集されている)、真の効果サイズは不明であるため、期待される効果サイズに基づいて検出力を計算することに意味がありません。代わりに、CohortMethod パッケージは、最小検出相対リスク(MDRR)を計算するための computeMdrr 関数を提供します。この研究例における MDRR は 1.69 です。

追跡可能なフォローアップの量をよりよく理解するために、フォローアップ期間の分布を調査することもできます。追跡期間をリスクにさらされ

る期間と定義し、アウトカムが発生するまでの期間として検討できます。getFollowUpDistribution 関数は簡単な概要を提供でき、図 12.21 に示されるように、両コホートのフォローアップ期間が比較可能であることがわかります。

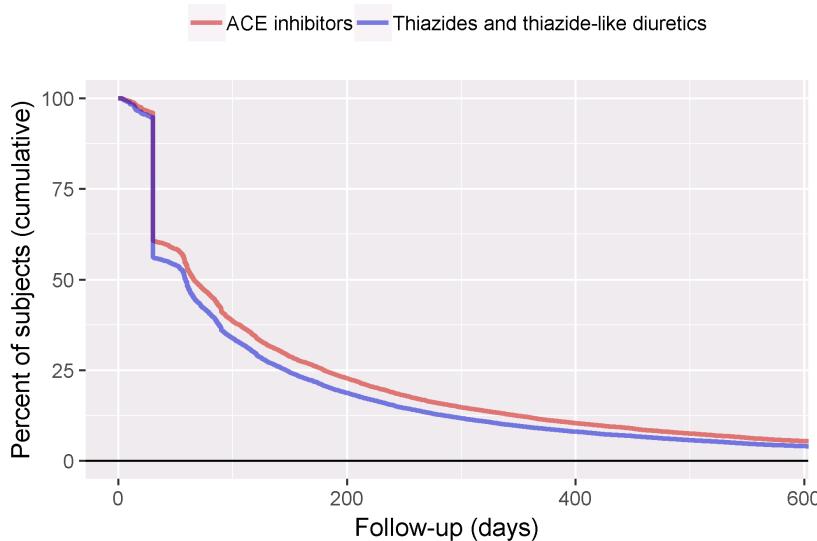


Figure 12.21: 対象および比較対照コホートのフォローアップ時間の分布

12.9.4 カプラン・マイヤー

最後に、カプラン・マイヤー・プロットを確認し、両コホートの時間経過による生存率を示します。plotKaplanMeier 関数を使用して 12.22 を作成し、ハザードの比例性の仮定が保持されているかどうかなどを確認できます。カプラン・マイヤー・プロットは PS による層別化や重み付けを自動的に調整します。この場合、変比率マッチングが使用されるため、比較対象グループの生存曲線は、ターゲットグループが比較対照に曝露されていた場合に曲線がどのように見えたであろうかを模倣するように調整されます。

12.9.5 効果量の推定

血管性浮腫に対するハザード比は 4.32 (95%信頼区間 : 2.45 - 8.08) でした。これは、ACEi が THZ と比較して血管性浮腫のリスクを増加させる可能性があることを示しています。同様に、AMI に対するハザード比は 1.13 (95%信頼区間 : 0.59 - 2.18) を観察し、AMI に対してはほとんどまたは全く影響がないことを示唆しています。前述の診断では、疑う理由がありません。しかし、最終的には、このエビデンスの質とそれを信頼するかどうかは、第 14 章で説明されている研究診断ではカバーされていない多くの要因に依存します。

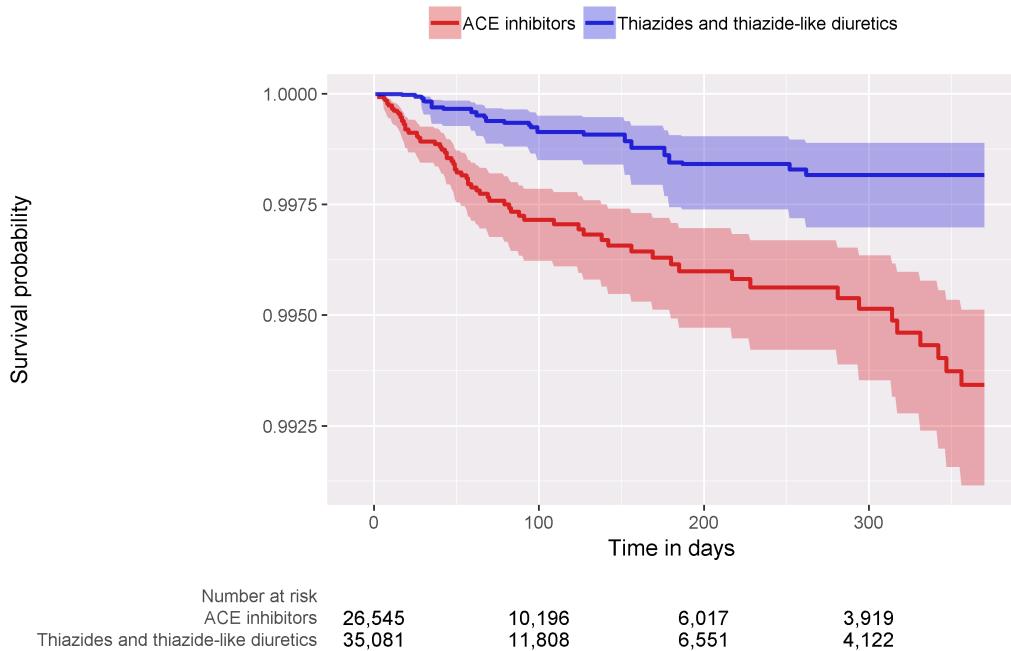


Figure 12.22: カプラン・マイヤー・プロット

12.10 まとめ



- 集団レベルの推定は、観察データから因果効果を推測することを目的としています。
- 反事実とは、対象者が別の曝露または何も曝露を受けなかった場合に何が起こったかということですが、それは観察できません。
- 異なるデザインは、異なる方法で反事実を構築することを目的としています。
- OHDSI Methods Library に実装されているさまざまなデザインは、適切な反事実を作成するための仮定が満たされているかどうかを評価するための診断を提供します。

12.11 演習

前提条件

これらの演習を行うためには、R、R-Studio、およびJavaがセクション8.4.5で説明されているようにインストールされていることを前提とします。また、SqlRender、DatabaseConnector、Eunomia、およびCohortMethodパッケージも必要です。これらは次のコマンドでインストールできます：

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

Eunomiaパッケージは、ローカルのRセッション内で実行されるCDM内のシミュレートされたデータセットを提供します。接続の詳細は次のコマンドで取得できます：

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

CDMデータベースのスキーマは「main」です。また、これらの演習ではいくつかのコホートも使用します。EunomiaパッケージのcreateCohorts関数を使用して、これらをCOHORTテーブル内に作成できます：

```
Eunomia::createCohorts(connectionDetails)
```

問題定義

セレコキシブの新規使用者とジクロフェナクの新規使用者における消化管(GI)出血のリスクは？

セレコキシブ新規使用者コホートのCOHORT_DEFINITION_IDは1です。ジクロフェナク新規使用者コホートのCOHORT_DEFINITION_IDは2です。GI出血コホートのCOHORT_DEFINITION_IDは3です。セレコキシブとジクロフェナクの成分コンセプトIDは、それぞれ1118084と1124300です。リスク期間は治療開始の日から始まり、観察終了時に終了します（いわゆる治療意図分析）。

演習12.1. CohortMethod Rパッケージを使用して、デフォルトの共変量セットを使用し、CDMからCohortMethodDataを抽出します。CohortMethodDataのサマリーを作成します。

演習12.2. createStudyPopulation関数を使用して、180日間のウォッシュアウト期間を必要とし、アウトカムの既往のある人を除外し、両方のコホートに出現する人は除外して、研究対象集団を作成します。脱落した人は何人ですか？

演習 12.3. 調整を行わずにコックス比例ハザードモデルを適合させます。これを行うと何が問題になるでしょうか？

演習 12.4. 傾向スコアモデルを適合させます。2つの群は比較可能ですか？

演習 12.5. 5つの層を用いて PS 層別化を行います。共変量のバランスは達成されましたか？

演習 12.6. PS 層を使用してコックス比例ハザードモデルを適合させます。そのアウトカムが未調整モデルと異なる理由は何ですか？

解答例は付録 E.8 を参照ください。

第 13 章

患者レベルの予測

著者: Peter Rijnbeek & Jenna Reps

臨床判断は、臨床医が患者の入手可能な病歴と現在の臨床ガイドラインに基づいて診断や治療方針を推測しなければならない複雑な作業です。この意思決定プロセスをサポートするために臨床予測モデルが開発され、幅広い専門分野の臨床現場で使用されています。これらのモデルは、人口統計学的情報、病歴、治療歴などの患者特性の組み合わせに基づいて診断結果や予後を予測します。

臨床予測モデルに関する論文の数は、過去 10 年間で大幅に増加しています。現在使用されているモデルのほとんどは、小規模なデータセットを使用して推定され、患者特性の小規模なセットのみを考慮しています。このサンプルサイズの小ささ、つまり統計的なパワーの弱さにより、データ分析者は強いモデリング仮定を立てざるを得なくなります。患者特性の限定的なセットの選択は、手元にある専門知識に強く影響されます。これは、患者が豊富なデジタル履歴を残している現代医療の現実とは大きく対照的です。医療従事者がそのすべてを完全に把握することは不可能です。現在、医療は膨大な量の患者固有の情報を電子的健康記録（EHR）に保存しています。これには、診断、投薬、臨床検査結果などの構造化データと、臨床記録に含まれる非構造化データが含まれます。患者の完全な EHR から得られる大量のデータを活用することで、予測精度がどの程度向上するのかは不明です。

大規模データセット分析のための機械学習の進歩により、この種のデータに患者レベルの予測を適用することへの関心が高まっています。しかし、患者レベルの予測に関する多くの公開された取り組みは、モデル開発ガイドラインに従っておらず、広範な外部検証を実施していないか、またはモデルの詳細が不十分であるため、独立した研究者がモデルを再現し、外部検証を行う能力が制限されています。そのため、モデルの予測性能を公平に評価することが難しくなり、臨床現場でモデルが適切に使用される可能性が低くなります。基準を改善するために、予測モデルの開発と報告におけるベストプラクティスのガイドラインを詳細に説明した論文がいくつか発表されています。例えば、多変量予測

モデルの透明性報告に関する声明（TRIPOD）¹では、予測モデルの開発と検証の報告に関する明確な推奨事項が提示されており、透明性に関する懸念の一部に対処しています。

OHDSIにより、大規模かつ患者固有の予測モデリングが現実のものとなりました。共通データモデル（CDM）により、前例のない規模で一貫性のある透明性の高い分析が可能になりました。CDMに標準化されたデータベースのネットワークが拡大していることで、世界規模でさまざまな医療環境におけるモデルの外部検証が可能になっています。これにより、医療の質の向上を最も必要としている患者の大きなコミュニティに即座に貢献できる機会が得られると私たちは考えています。このようなモデルは、真に個別化された医療の実現につながり、患者の予後を大幅に改善できる可能性があります。本章では、OHDSIの患者レベル予測のための標準化されたフレームワーク（Reps et al., 2018）について説明し、開発と検証のための確立されたベストプラクティスを実装したPatientLevelPrediction R パッケージについて議論します。まず、患者レベル予測の開発と評価に必要な理論を説明し、実装された機械学習アルゴリズムの概要を説明します。次に、予測問題の例を挙げ、ATLAS またはカスタム R コードを使用した定義と実装の手順を説明します。最後に、研究結果の普及に Shiny アプリケーションを使用する方法について説明します。

13.1 予測課題

図 13.1 ここで取り上げる予測問題を示しています。リスクのある集団において、定義された時点 ($t = 0$) で、リスク期間中に何らかの結果を経験する患者を予測することを目的としています。予測は、その時点以前の観察期間における患者の情報のみを使用して行われます。

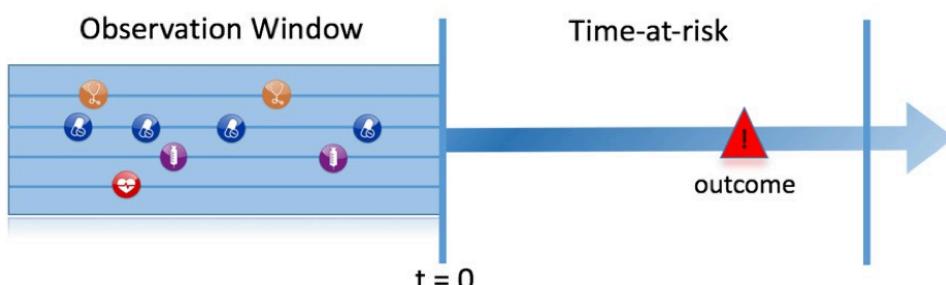


Figure 13.1: 予測課題

表 13.1 に示すように、予測課題を定義するには、対象コホートによって $t=0$ を定義し、アウトカムコホートによって予測したいアウトカムを定義し、リスク期間も定義する必要があります。標準的な予測問題は次のように定義されます：

¹<https://www.equator-network.org/reporting-guidelines/tripod-statement/>

[対象コホートの定義] の中で、リスク期間内に [アウトカムコホートの定義] を持つのは誰ですか？

さらに、開発したいモデルのデザインオプションを検討し、内部および外部検証を行うための観察データセットを決定する必要があります。

Table 13.1: 予測デザインにおける主要なデザインオプション

選択肢	説明
対象コホート	予測したい対象者のコホートをどのように定義しますか？
アウトカムコホート	予測したいアウトカムをどのように定義しますか？
リスク期間	$t=0$ に対してどの時間枠で予測を行いますか？
モデル	どのアルゴリズムを使用し、どの潜在的な予測変数を含めますか？

この概念的フレームワークは、あらゆる種類の予測課題に適用できます。例えば：

- 疾病の発症と進行
 - 構造: [病気] と新たに診断された患者のうち、[診断から見た時間枠内] に [別の疾患または合併症] を発症するのは誰ですか？
 - 例: 心房細動と新たに診断された患者のうち、次の 3 年以内に虚血性脳卒中を発症するのは誰ですか？
- 治療選択
 - 構造: [対象疾患] と診断された患者で、[治療 1] または [治療 2] で治療された患者うち、[治療 1]* で治療された患者は？
 - 例: ワルファリンまたはリバロキサバンを服用した心房細動患者のうち、ワルファリンを服用した患者は誰ですか？(傾向スコアモデルの場合など)
- 治療反応
 - 構造: [治療] の新規使用者のうち、[時間枠内] に [何らかの効果] を経験するのは誰ですか？
 - 例: メトホルミンを開始した糖尿病患者のうち、3 年間メトホルミンを継続するのは誰ですか？
- 治療安全性
 - 構造: [治療] の新規使用者の中で、[時間枠内] に [有害事象] を経験するのは誰ですか？
 - 例: ワルファリンの新規使用者の中で、1 年以内に消化管出血を起こすのは誰ですか？
- 治療遵守

- 構造: [治療] の新規使用者の中で、[時間枠] で [遵守指標] を達成するのは誰ですか？
- 例: メトホルミンを開始した糖尿病患者のうち、1 年以内に 80% 以上の遵守率を達成するのは誰ですか？

13.2 データ抽出

予測モデルを作成する際には、機械学習の一種であると呼ばれるプロセスを使用します。これは、ラベル付けされたサンプルセットに基づいて、共変量とアウトカムの状態の間の関係を推測するものです。したがって、対象コホートに属する人について、CDM から共変量を抽出する方法が必要であり、またアウトカムのラベルを取得する必要があります。

共変量（“予測因子”、“特徴量”、“独立変数”とも呼ばれる）は、患者の特徴を説明します。共変量には、年齢、性別、特定のコンディションの有無、患者記録内の曝露コードなどが含まれます。共変量は一般に、FeatureExtractionパッケージを使用して構築され、詳細は第 11 章で説明しています。予測には、その人が対象コホートに入る日付（本書ではこれをインデックス日と呼びます）の前および当日のデータのみ使用できます。

また、リスク期間中の全ての患者のアウトカムステータス（“ラベル”または“クラス”とも呼ばれる）も取得する必要があります。リスク期間中にアウトカムが発生した場合、アウトカムステータスは「陽性」と定義されます。

13.2.1 データ抽出の例

表 13.2 は、2 つのコホートが含まれた COHORT テーブルの例を示しています。コホート定義 ID が 1 のコホートは対象コホート（例：「最近、心房細動と診断された人」）です。コホート定義 ID が 2 は、アウトカムコホートを定義します（例：「脳卒中」）。

Table 13.2: 例示的な COHORT テーブル。簡潔のために COHORT_END_DATE は省略しています。

COHORT_DEFINITION_ID	SUBJECT_ID	COHORT_START_DATE
1	1	2000-06-01
1	2	2001-06-01
2	2	2001-07-01

表 13.3 は、例示的な CONDITION_OCCURRENCE テーブルを示しています。Concept ID 320128 は「本態性高血圧」に該当します。

Table 13.3: 例示的な CONDITION_OCCURRENCE テーブル。簡潔のため、3 つの列のみ表示しています。

PERSON_ID	CONDITION_CONCEPT_ID	CONDITION_START_DATE
1	320128	2000-10-01
2	320128	2001-05-01

この例示的なデータに基づき、リスク期間をインデックス日（対象コホートの開始日）から 1 年間と仮定すると、共変量とアウトカムステータスを構築できます。「前年の本態性高血圧」を示す共変量は、個人 ID 1 に対して 0（非該当）（アウトカムがインデックス日の後に発生）と、個人 ID 2 に対して 1（該当）となります。同様に、アウトカムステータスは個人 ID 1 に対して 0（この人はアウトカムコホートに該当しない）、個人 ID 2 に対して 1（インデックス日から 1 年以内にアウトカムが発生）となります。

13.2.2 負の値と欠損

観察医療データは、値が負であるか欠損であるかを反映することはまれです。先の例では、ID 1 の人物にはインデックス日以前に本態性高血圧症の発生がなかったことを単純に観察しました。これは、その時点ではその状態が存在していないかった（負）ためであるか、あるいは記録されていなかった（欠損）ためである可能性があります。機械学習アルゴリズムは負と欠損を区別できず、利用可能なデータにおける予測値を単純に評価することに留意することが重要です。

13.3 モデルの適合

予測モデルの適合を行う際には、ラベル付けされた例から、共変量と観測されたアウトカムの状態の関係を学習しようとしています。例えば、共変量が 2 つしかない場合、収縮期血圧と拡張期血圧だとすると、各患者を 2 次元空間のプロットとして表現できます（図 13.2 を参照）。この図では、データポイントの形が患者のアウトカム状態（例：脳卒中）に対応しています。

教師あり学習モデルは、2 つのアウトカムクラスを最適に分離する決定境界を見つけようとします。異なる教師あり学習手法は異なる決定境界をもたらし、決定境界の複雑性に影響を与えるハイパーパラメータが存在することがよくあります。

図 13.2 では、3 つの異なる決定境界線を見ることができます。この境界線は、新しいデータポイントの結果の状態を推測するために使用されます。新しいデータポイントが影の部分に該当する場合、モデルは「結果あり」と予測し、それ以外は「結果なし」と予測します。理想的には、決定境界線は 2 つのクラスを完全に分割すべきです。しかし、あまりにも複雑なモデルはデータに「過

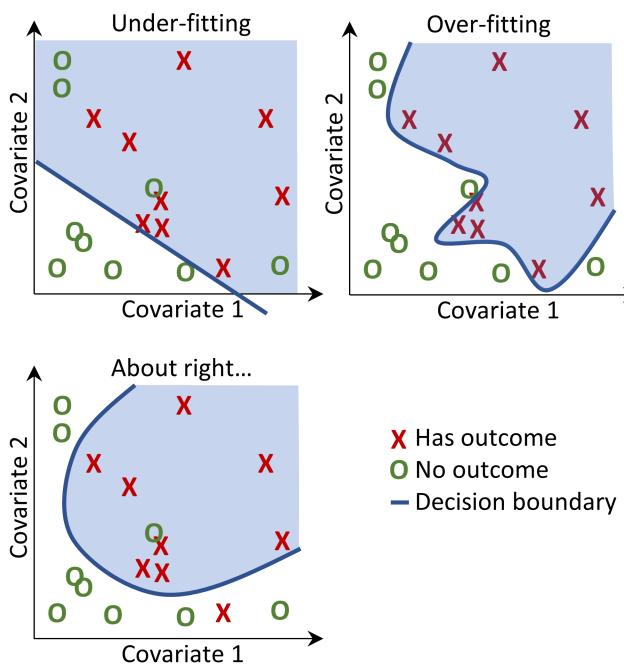


Figure 13.2: 決定境界

「剩適合」するリスクがあります。これは、未確認データに対するモデルの汎用性に悪影響を及ぼす可能性があります。例えば、データにノイズが含まれている場合、つまりラベル付けが誤っていたり、データポイントの位置が不正確であったりする場合には、そのノイズにモデルを適合させることは望ましくありません。そのため、トレーニングデータでは完全に識別しないが、「実際の」複雑さを捉える決定境界を定義することが望ましい場合があります。正則化などの手法は、複雑さを最小限に抑えながらモデルのパフォーマンスを最大化することを目的としています。

各教師あり学習アルゴリズムは、決定境界を学習する方法が異なり、どのアルゴリズムがデータに最適であるかは一概には言えません。「No Free Lunch」定理が示すように、すべての予測問題において常に他のアルゴリズムよりも優れたアルゴリズムは存在しません。そのため、患者レベルの予測モデルを開発する際には、さまざまなハイパーパラメータ設定で複数の教師あり学習アルゴリズムを試すことをお勧めします。

この後に示すアルゴリズムは以下から取得可能です：PatientLevelPrediction パッケージ

13.3.1 正則化ロジスティック回帰

LASSO（最小絶対値収縮選択オペレーター）ロジスティック回帰は一般化線形モデルに属し、変数の線形結合が学習され、最終的にロジスティック関数がそ

の線形結合を 0 から 1 の間の値に写像します。LASSO 正則化では、モデルをトレーニングする際に、目的関数にモデルの複雑さに基づくコストが追加されます。このコストは係数の線形結合の絶対値の合計です。モデルは、このコストを最小化することで自動的に特徴量の選択を行います。大規模な正則化ロジスティック回帰を行うために、Cyclops（ロジスティック、ポアソン、生存時間分析のための循環座標降下法）パッケージを使用しています。

Table 13.4: 正則化ロジスティック回帰のハイパーパラメータ

パラメータ	説明	典型的な値
初期分散	事前分布の初期分散	0.1

分散はクロスバリデーションでのサンプル外の尤度を最大化して最適化されるため、初期分散はアウトカムとして得られるモデルの性能にはほとんど影響しません。ただし、最適値からあまりに離れた初期分散を選択すると、適合時間が長くなる可能性があります。

13.3.2 勾配ブースティングマシン

勾配ブースティングマシンはブースティングアンサンブル手法であり、この枠組みでは複数の決定ツリーを組み合わせます。ブースティングは、繰り返し決定ツリーを追加し、前の決定ツリーで誤分類されたデータポイントにより多くの重みをコスト関数に追加して次のツリーをトレーニングします。高効率の勾配ブースティングフレームワークを実装した、CRAN で利用可能な xgboost R パッケージを使用しています。

Table 13.5: 勾配ブースティングマシンのハイパーパラメータ

パラメータ	説明	典型的な値
earlyStopRound	改善がない場合の停止ラウンド数	25
learningRate	ブースティングの学習率	0.005, 0.01, 0.1
maxDepth	ツリーの最大深さ	4, 6, 17
minRows	ノード内の最小データポイント数	2
ntrees	ツリーの数	100, 1000

13.3.3 ランダムフォレスト

ランダムフォレストは複数の決定ツリーを組み合わせるバギング法の集合手法です。バギング法の背景にある考え方とは、弱い分類器を使用し、それを強力な分類器に組み合わせることで、過剰適合の可能性を低減することです。ランダムフォレストは、複数の決定ツリーをトレーニングすることでこれを実現しま

ですが、各ツリーでは変数のサブセットのみを使用し、ツリーごとに変数のサブセットが異なります。当社のパッケージでは、Python のランダムフォレストの実装である `sklearn` を使用しています。

Table 13.6: ランダムフォレストのハイパーパラメータ

パラメータ	説明	典型的な値
maxDepth	ツリーの最大深さ	4, 10, 17
mtries	各ツリーの特徴量数	-1 = 総特徴量数の平方根, 5, 20
ntrees	ツリーの数	500

13.3.4 K-近傍法

K-近傍法 (K-nearest neighbors; KNN) は、ある距離尺度を使用して、新しいラベルのないデータポイントに最も近いラベル付きデータポイントを K 個見つけるアルゴリズムです。新しいデータポイントの予測は、K 個の最も近いラベル付きデータポイントの中で最も一般的なクラスとなります。KNN には共有に関する制限があり、新しいデータの予測を行うにはラベル付きデータが必要となるため、このデータをデータサイト間で共有できないことがよくあります。OHDSI で開発された大規模 KNN 分類器である `BigKnn` パッケージを組み込んでいます。

Table 13.7: K-近傍法のハイパーパラメータ

パラメータ	説明	典型的な値
k	近傍数	1000

13.3.5 ナイーブベイズ

ナイーブベイズアルゴリズムは、クラス変数の値を考慮した際の特徴量の各ペア間の条件付き独立性という単純な仮定を適用したベイズの定理です。データがクラスに属する可能性とクラスの事前分布に基づいて事後分布が取得されます。ナイーブベイズにはハイパーパラメータはありません。

13.3.6 AdaBoost

AdaBoost はブースティングのアンサンブル手法です。ブースティングは分類器を繰り返し追加することで機能しますが、次の分類器をトレーニングする際には、コスト関数において、先行する分類器によって誤分類されたデータポイントにより大きな重みを追加します。Python では、`sklearn` の `AdaboostClassifier` 実装を使用します。

Table 13.8: AdaBoost のハイパーパラメータ

パラメータ	説明	典型的な値
nEstimators	ブースティングを終了する最大推定器数	4
learningRate	learningRate によって各分類器の寄与を縮小する学習率。learningRate と nEstimators にはトレードオフの関係がある。	1

13.3.7 決定ツリー

決定ツリーは、貪欲法で選択された個々のテストを使用して変数空間を分割する分類器です。クラスを分離する際に、最も高い情報量を持つ分割を見つけることを目的としています。決定ツリーは、多数の分割（ツリーの深さ）を有効にすることで簡単に過学習状態になるため、多くの場合、正則化（例えば、枝刈りやモデルの複雑性を制限するハイパーパラメータの指定）が必要です。Python では、sklearn の DecisionTreeClassifier 実装を使用します。

Table 13.9: 決定ツリーのハイパーパラメータ

パラメータ	説明	典型的な値
classWeight	“Balance” または “None”	None
maxDepth	木の最大深さ	10
minImpuritySplit	ツリーの成長中に早期停止するための閾値。ノードの不純物が閾値を上回る場合は分割され、そうでない場合はリーフとなります	10^-7
minSamplesLeaf	各リーフの最小サンプル数	10
minSamplesSplit	各分割の最小サンプル数	2

13.3.8 多層パーセプトロン

多層パーセプトロンは、非線形関数を使用して入力を重み付けする複数のノード層を含むニューラルネットワークです。最初の層は入力層、最後の層は出力層であり、その間にあるのが隠れ層です。ニューラルネットワークは一般にバックプロパゲーションを用いて訓練されます。これは、訓練入力がネットワー

クを通じて順方向に伝搬され、出力が生成されることを意味します。出力とアウトカムの状態の間の誤差が計算され、この誤差がネットワークを通じて逆方向に伝搬され、線形関数の重みが更新されます。

Table 13.10: 多層パーセプトロン用のハイパー・パラメータ

パラメータ	説明	典型的な値
alpha	L2 正則化	0.00001
size	隠れノードの数	4

13.3.9 深層学習

ディープネット、畳み込みニューラルネットワーク、再帰型ニューラルネットワークなどのディープラーニングは、多層パーセプトロンと類似していますが、予測に役立つ潜在的な表現を学習することを目的とした複数の隠れ層を持っています。PatientLevelPrediction パッケージの別のビネットで、これらのモデルとハイパー・パラメータの詳細について説明しています。

13.3.10 その他のアルゴリズム

患者レベルの予測フレームワークには他のアルゴリズムを追加できますが、これは本章の範囲外です。詳細は、PatientLevelPrediction パッケージの “Adding Custom Patient-Level Prediction Algorithms” ビネットを参照ください。

13.4 予測モデルの評価

13.4.1 評価の種類

予測モデルの評価は、モデルの予測と観測されたアウトカムの一一致度を測定することによって行うことができます。これには、アウトカムのステータスが既知であるデータが必要です。



評価には、モデル開発に使用されたデータセットとは異なるデータセットを使用しなければなりません。そうしないと、過剰適合したモデルを支持してしまう可能性があり（セクション 13.3 を参照）、新規患者に対して良好な結果が得られない可能性があります。

評価の種類には、以下のものがあります：

- 内部検証：同じデータベースから抽出された異なるデータセットを使用してモデルを開発および評価します。
- 外部検証：一つのデータベースでモデルを開発し、別のデータベースで評価します。

内部検証の方法には、次の 2 つがあります：

- ホールドアウトセットアプローチ：ラベル付けされたデータを 2 つの独立したセット（トレーニングセットとテストセット（ホールドアウトセット））に分割します。トレーニングセットはモデルの学習に使用され、テストセットはモデルの評価に使用されます。患者をランダムに訓練用とテスト用に単純に分割することもできますが、以下のような方法も選択できます。
 - 時間に基づいた分割（時間的検証）：例えば、特定の日付以前のデータで訓練を行い、その日付以降のデータで評価を行う。これにより、モデルが異なる期間に一般化できるかどうかを判断できる可能性があります。
 - 地理的位置に基づいた分割（空間的検証）。
- クロスバリデーション：データが限られている場合に有用です。データを等しいサイズに分割し、 n にプレ設定されたセットに分割します（例： $n = 10$ ）。各セットでは、そのセットデータ以外のすべてのデータを使用してモデルがトレーニングされ、保留セットの予測を生成するために使用されます。このように、すべてのデータが一度使用されてモデル構築アルゴリズムが評価されます。患者レベルの予測フレームワークでは、最適なハイパーパラメータを選択するためにクロスバリデーションを使用します。

外部検証は、別のデータベースのデータ（すなわち、開発された環境以外のデータ）でモデルのパフォーマンスを評価することを目的としています。モデルの移植性を評価するこの手法は重要です。なぜなら、モデルをトレーニングしたデータベースのみでなく、他のデータベースにも適用したいからです。異なるデータベースは、異なる患者集団、異なる医療システム、異なるデータ収集プロセスを表している可能性があります。多数のデータベースにおける予測モデルのクロスバリデーションは、モデルの受容と臨床現場への導入において重要なステップであると考えています。

13.4.2 性能指標

閾値測定

予測モデルは、リスク期間中にアウトカムが起こる患者のリスクに対応する各患者に 0 から 1 の間の値を割り当てます。値が 0 の場合はリスクが 0%、0.5 の場合は 50% のリスク、1 の場合は 100% のリスクを意味します。リスク期間にアウトカムが得られるリスクを患者に分類するために使用する閾値を最初に指定することで、一般的な測定基準である正確度、感度、特異度、陽性的中率を計算することができます。例えば、表 13.11 にあるように閾値を 0.5 と設定すると、患者 1、3、7、10 は閾値 0.5 以上の予測リスクを持つため、アウトカムを持つと予測されます。他のすべての患者は 0.5 未満の予測リスクを持つため、アウトカムを持たないと予測されます。

Table 13.11: 予測確率に対する閾値の利用例

患者 ID	予測リスク	0.5 閾値での 予測クラス	リスク期間中 にアウトカム を持つ	タイプ
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

患者が予測されたアウトカムを持ち、実際にアウトカムを持つ場合、それを真陽性 (TP) と呼びます。患者が予測されたアウトカムを持っているが実際にはアウトカムを持っていない場合、それを偽陽性 (FP) と呼びます。患者がアウトカムを持たないと予測され、実際にアウトカムを持っていない場合、それを真陰性 (TN) と呼びます。最後に、患者がアウトカムを持たないと予測され、実際にアウトカムを持っている場合、それを偽陰性 (FN) と呼びます。

以下の閾値ベースの指標を計算できます：

- 正解率: $(TP + TN)/(TP + TN + FP + FN)$
- 感度: $TP/(TP + FN)$
- 特異度: $TN/(TN + FP)$
- 陽性的中率: $TP/(TP + FP)$

これらの値は、閾値が下げられると増減する可能性があります。分類器の閾値を下げるとき、アウトカムの数を増やすことで分母を増やすことができます。以前の閾値が高すぎた場合、新しいアウトカムはすべて真陽性である可能性があり、これにより陽性的中率が増加します。以前の閾値が適切であったか低すぎた場合、さらなる閾値の低下は偽陽性をもたらすため、陽性的中率が減少する可能性があります。感度の場合、分母は分類器の閾値に依存しません ($TP + FN$ は一定です)。このため、分類器の閾値を下げることで真陽性のアウトカム数を増やし、感度を向上させる可能性があります。また、閾値を下げても感度が変わらない一方で、陽性的中率が変動することもあります。

識別力

識別力とは、リスク期間中にアウトカムを経験する患者に対して、より高いリスクを割り当てる能力のことです ROC (Receiver Operating Characteristics) 曲線は、すべての可能な閾値において、 $1 - \text{特異度}$ を x 軸に、感度を y 軸にプロ

ットして作成されます。ROC プロットの例は、本章の後のほうの図 13.17 で示されています。ROC 曲線下面積 (AUC) は、判別能の全体的な尺度であり、0.5 はリスクをランダムに割り当てるることを意味し、1 は完璧な判別を意味します。ほとんどの公表された予測モデルは、0.6~0.8 の AUC を取得しています。

AUC は、リスク期間中にアウトカムを経験する患者と経験しない患者との間で予測リスク分布がどれだけ異なるかを判断する方法を提供します。AUC が高い場合、分布はほとんど重ならないのに対し、重なりが多い場合は AUC が 0.5 に近くなります。図 13.3 に示されている通りです。

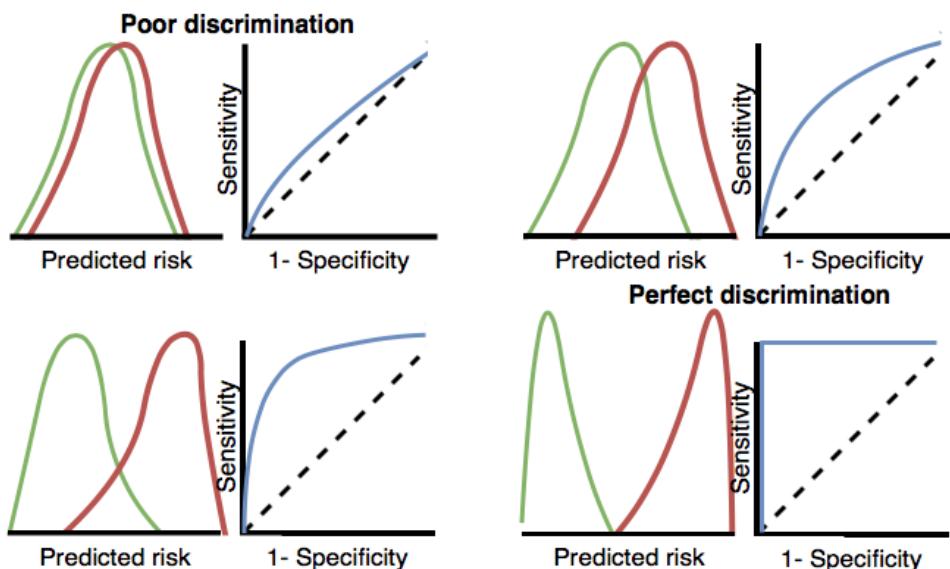


Figure 13.3: 識別力に関する ROC プロット。2 つのクラスの予測リスクの分布が類似している場合、ROC は対角線に近く、AUC は 0.5 に近くなります。

非常に稀なアウトカムに対しては、AUC が高くても実際には実用的でない場合があります。なぜなら、閾値を超えるすべての陽性の背後には多くの陰性が存在し、陽性的中率が低くなる可能性があるからです。アウトカムの重大性や介入のコスト（健康リスクまたは金銭的）によっては、高い偽陽性率は望ましくないかもしれません。そのため、稀なアウトカムに対しては、適合率-再現率曲線下面積 (AUPRC) と呼ばれる別の測定値が推奨されます。AUPRC は感度を x 軸（再現率としても知られる）に、陽性的中率（適合率としても知られる）を y 軸にプロットして生成される曲線の下の面積です。

キャリブレーション

キャリブレーションは、モデルが正しいリスクを割り当てる能力です。例えば、モデルが 100 人の患者に 10% のリスクを割り当てた場合、そのうち 10 人がリスク期間中にアウトカムを経験するべきです。同様に、モデルが 100 人の患者に 80% のリスクを割り当てた場合、そのうち 80 人がリスク期間中にアウト

カムを経験するべきです。キャリブレーションは、一般に予測リスクに基づいて患者を十分位に分割し、各グループで平均予測リスクとリスク期間中にアウトカムを経験した患者の割合を計算することによって測定されます。次に、これらの 10 点（予測リスクを y 軸、観測リスクを x 軸にプロット）をプロットし、それらが $x = y$ の線上に位置するかどうかを確認します。これがモデルが適切にキャリブレーションされていることを示します。キャリブレーションプロットの例は、本章の後半に図 13.18 で示されています。また、これらの点を使用して線形モデルをフィットし、切片（ゼロに近いはず）と傾き（1 に近いはず）を計算します。もし、傾きが 1 より大きい場合、モデルは実際のリスクよりも高いリスクを割り当てており、傾きが 1 より小さい場合、モデルは実際のリスクよりも低いリスクを割り当てていることを意味します。非線形関係をよりよく捕捉するために、Smooth Calibration Curves も実装しています。

13.5 患者レベル予測研究のデザイン

本セクションでは予測研究のデザイン方法について説明します。最初のステップは、予測問題を明確に定義することです。興味深いことに、多くの発表論文では予測問題の定義が不十分であり、例えば、インデックス日（対象コホートの開始日）がどのように定義されているのかが不明確です。予測問題の定義が不十分であると、他者による外部検証はおろか、臨床現場での実施も不可能になります。患者レベルの予測フレームワークでは、表 13.1 で定義された主要な選択肢を明示的に定義することを要求することで、予測問題の適切な仕様を強制しています。ここでは、「治療の安全性」タイプの予測問題を例に、このプロセスを説明します。

13.5.1 問題の定義

血管性浮腫は ACE 阻害薬のよく知られた副作用であり、ACE 阻害薬のラベルに記載されている血管性浮腫の発生率は 0.1% から 0.7% の範囲です (Byrd et al., 2006)。この副作用をモニタリングすることは重要です。なぜなら、血管性浮腫は稀であるものの、生命を脅かす可能性があり、呼吸停止や死亡に至ることがあるからです (Norman et al., 2013)。さらに、血管性浮腫が最初に認識されないと、その原因を特定するまでに広範かつ高額な検査が必要となる可能性があります (Norman et al., 2013; Thompson and Frable, 1993)。アフリカ系アメリカ人患者におけるリスクの増加以外に、ACE 阻害薬関連の血管性浮腫の発症に対する既知の素因は知られていません (Byrd et al., 2006)。ほとんどの反応は初めての治療の最初の週または 1 か月以内に発生し、最初の投与から数時間以内に発生することもあります (Cicardi et al., 2004)。しかし、一部の症例は治療開始から数年後に発生することもあります (O’ Mara and O’ Mara, 1996)。リスクのある人を特定する特定の診断テストはありません。もしリスクのある人を特定できれば、医師は例えば ACE 阻害薬を別の降圧薬に切り替えるなどの対応が可能になります。

患者レベル予測フレームワークを観察医療データに適用して、次の患者レベル

の予測問題に取り組みます：

初めて ACE 阻害薬を開始した患者のうち、1年以内に血管性浮腫を発症するのは誰か？

13.5.2 研究集団の定義

モデルを開発する最終的な研究対象集団は、対象コホートのサブセットとなることがあります。例えば、アウトカムに依存する基準を適用する場合や、対象コホートのサブグループで感度分析を行いたい場合などです。この場合、以下の問い合わせに答えなければなりません。：

- 対象コホートの開始前にどの程度の観察期間が必要ですか？この選択は、トレーニングデータで利用可能な患者の時間にも依存しますが、将来的にモデルを適用したいデータソースで利用可能な時間にも依存します。最小の観察期間が長ければ長いほど、各人の特徴量抽出に使用できるベースライン期間を得られますが、分析対象となる患者数は少なくなります。さらに、短いまたは長い観察期間を選択する臨床的な理由がある場合もあります。今回の例では、365日間をルックバック期間（ウォッシュアウト期間）として使用します。
- 患者が対象コホートに複数回組み入れられますか？対象コホートの定義では、例えば、異なる病気のエピソードや医療製品への曝露期間が異なる場合など、異なる期間に複数回、その集団に適格となる可能性があります。コホート定義は、患者が一度だけ参加できるように制限を適用するとは限りませんが、特定の患者レベルの予測問題の文脈では、最初の適格エピソードにコホートを限定したい場合もあります。例では、ACE 阻害薬の初回使用を基準としているため、患者は一度だけ対象コホートに参加できます。
- 以前にアウトカムを経験した人をコホートに含めることができますか？対象コホートに適格となる前にアウトカムを経験した人をコホートに含めるかどうかを決める必要があります。特定の患者レベルの予測問題によっては、初回のアウトカムの発生を予測したい場合があるため、以前にアウトカムを経験した患者はリスクがないため、ターゲットコホートから除外する必要があります。他の状況では、前回エピソードの予測を希望しているため、以前のアウトカムが将来のアウトカムの予測要因になる可能性もあります。私たちの予測例では、以前に血管性浮腫を持つ人を含めないことにします。
- 対象コホート開始日に対してアウトカムを予測する期間をどう定義しますか？この質問に答えるために、2つの決定を下す必要があります。まず、リスク期間の開始日を対象コホートの開始日またはそれ以降に設定するかどうかの議論があります。開始日を遅らせる理由には、記録が遅れて入力された結果を避けたい場合や、結果を予防する介入が理論上実施可能であった期間を空けておきたい場合などが考えられます。次に、リスク期間の終了日を設定して、対象コホートの開始日または終了日からの相対的な

日数オフセットをある程度指定することで、リスク期間を定義する必要があります。今回の問題では、対象コホートの開始日の1日後から365日後までの期間をリスク期間として予測します。

- 最小限のリスク期間は必要でしょうか？アウトカムが発生しなかったが、リスク期間終了前にデータベースを離脱した患者を含めるかどうかを決める必要があります。これらの患者は観察期間が終わった後にアウトカムを経験する可能性があります。この予測問題では、この質問に「はい」と答え、その理由で最小のリスク期間を必要とします。さらに、この制約がアウトカムを経験した人に適用されるかどうか、リスク期間の長短に関わらず、アウトカムを経験した人全員を対象とするのかを決定しなければなりません。アウトカムが死亡の場合、リスク期間が完了する前に打ち切られる可能性が高いでしょう。

13.5.3 モデル開発の設定

予測モデルを開発するには、どのアルゴリズムをトレーニングするか決定しなければなりません。特定の予測問題に対する最良のアルゴリズムの選択は経験的な問題であると捉え、すなわち、データに語らせ、最良のものを発見するためにさまざまなアプローチを試みることを好みます。そのため、私たちのフレームワークでは、セクション13.3に記載されているように多くのアルゴリズムを実装しており、他のアルゴリズムを追加することも可能です。この例では、シンプルにするために、Gradient Boosting Machines (GBM) を一つのアルゴリズムとして選択します。

さらに、モデルのトレーニングに使用する共変量も決定する必要があります。この例では、性別、年齢、コンディションすべて、薬剤および薬剤グループ、来院回数などを追加したいと思います。これらの臨床イベントは、インデックス日の1年前から、それ以前の任意の期間にわたって検索します。

13.5.4 モデル評価

最後に、モデルの評価方法を定義する必要があります。ここでは、単純化のため、内部検証を選択します。データセットをトレーニング用とテスト用に分割する方法、患者をこれら2つのセットに割り当てる方法を決定する必要があります。ここでは、典型的な75% - 25% 分割を使用します。非常に大規模なデータセットの場合は、トレーニング用にさらに多くのデータを使用できます。

13.5.5 研究概要

これで、表13.12に示されるように、研究を完全に定義しました。

Table 13.12: この研究の主なデザイン選択

選択	値
対象コホート	初めて ACE 阻害薬を開始した患者。 観察期間が 365 日未満、または血管性浮腫の既往がある患者は除外されます。
アウトカムコホート	血管性浮腫。
リスク期間	コホート開始日の 1 日後から 365 日後。少なくとも 364 日のリスク期間が必要。
モデル	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. データ分割: 75% トレーニング - 25% テスト、個人ごとにランダムに割り当てられます。

13.6 ATLAS での研究の実装

予測研究をデザインするインターフェースは、ATLAS メニューの左側にある  Prediction ボタンをクリックすることで開くことができます。新しい予測研究を作成しましょう。研究にわかりやすい名前をつけておくことを忘れないでください。研究のデザインはいつでも  ボタンをクリックして保存できます。

予測デザイン機能には、予測の問題設定、分析設定、実行設定、トレーニング設定の 4 つのセクションがあります。それぞれのセクションについて説明します。

13.6.1 予測の問題設定

ここでは、分析の対象となる母集団コホートとアウトカムコホートを選択します。予測モデルは、対象となる母集団コホートとアウトカムコホートのすべての組み合わせに対して作成されます。例えば、2 つの対象集団と 2 つのアウトカムを指定すると、4 つの予測問題が指定されたことになります。

対象となる母集団コホートを選択するには、事前に ATLAS で定義しておく必要があります。コホートのインスタンス化については、第 10 章で説明しています。この例で使用する対象（付録 B.1）およびアウトカム（付録 B.4）コホートの完全な定義は付録に掲載しています。対象集団をコホートに追加するに

は、「Add Target Cohort」ボタンをクリックします。アウトカムコホートの追加も同様に、「Add Outcome Cohort」ボタンをクリックすることで行います。完了すると、ダイアログが図 13.4 のようになります。

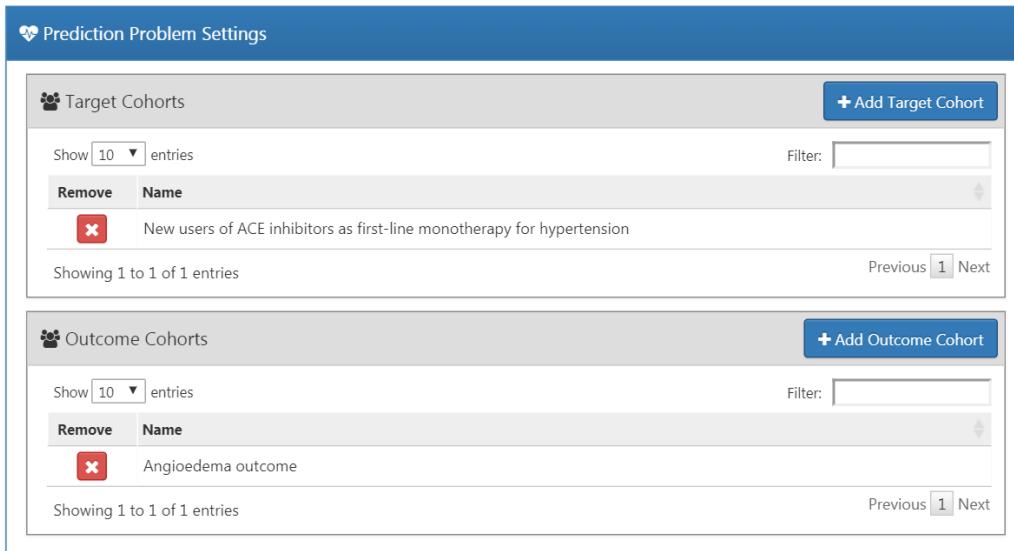


Figure 13.4: 予測問題の設定

13.6.2 分析設定

分析設定では、教師あり学習アルゴリズム、共変量と母集団の設定を選択できます。

モデル設定

モデル開発には、1つまたは複数の教師あり学習アルゴリズムを選択することができます。教師あり学習アルゴリズムを追加するには、「Add Model Settings」ボタンをクリックします。ATLAS インターフェースで現在サポートされているすべてのモデルを含むドロップダウンが表示されます。ドロップダウンメニューから名前をクリックして、調査に含めたい教師あり学習モデルを選択します。これにより、その特定のモデルのビューが表示され、ハイパーパラメータ値の選択が可能になります。複数の値が指定されている場合、グリッドサーチがすべての値の組み合わせに対して実行され、クロスヴァリデーションを使用して最適な組み合わせが選択されます。

今回の例では、勾配ブースティングマシンを選択し、図 13.5 に示すようにハイパーパラメータを設定します。

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: 勾配ブースティングマシンの設定

共変量設定

私たちは、CDM 形式の観察データから抽出できる標準共変量のセットを定義しました。共変量設定ビューでは、どの標準共変量を含めるかを選択できます。異なるタイプの共変量設定を定義でき、各モデルは指定された共変量設定ごとに個別に作成されます。

研究に共変量設定を追加するには、「Add Covariate Settings」をクリックします。これで共変量設定ビューが開きます。

共変量設定ビューの最初の部分は除外/包括オプションです。共変量は一般に任意のコンセプトに対して構築されますが、例えば対象コホート定義にリンクされている場合、特定のコンセプトを除外または含めることができます。特定のコンセプトのみを含めるには、ATLAS でコンセプトセットを作成し、“What concepts do you want to include in baseline covariates in the patient-level prediction model? (Leave blank if you want to include everything) (患者レベルの予測モデルにおけるベースライン共変量として、どのようなコンセプトを含めたいですか？(すべてを含めたい場合は空白のままにしてください)” の下で をクリックしてコンセプトセットを選択します。コンセプトセット内のコンセプトに下位層のコンセプトを自動的に追加するには、“Should descendant concepts be added to the list of included concepts? (含まれるコンセプトのリストに下位層コンセプトを追加すべきでしょうか?)” の質問に「yes」と答えます。同じプロセスを、共変量に対応する選択されたコンセプトを除外する “What concepts do you want to exclude in baseline covariates in the patient-level prediction model? (Leave blank if you want to include everything) (患者レベルの予測モデルにおけるベースライン共変量から除外したいコンセプトは何ですか？(すべてを含める場合は空白のままにしてください)” の質問にも繰り返します。最後のオプション “A comma delimited list of covariate IDs that should be restricted to (制限すべき共変数 ID のコンマ区切りリスト)” では、共変量 ID (コンセプト ID ではなく) をカンマ区切りで追加し、これらがモデルに含まれるようにすることができます。このオプションは上級ユーザ向けです。完了すると、適格基準設定と除外基準設定は図 13.6 のようになります。

次のセクションでは、時間に依存しない変数の選択ができます：

- ・ 性別：男性または女性の性別を示す二値変数
- ・ 年齢：年単位の連續変数
- ・ 年齢グループ：5 年ごとのバイナリ変数 (0-4、5-9、…、95+)
- ・ 人種：各人種に関するバイナリ変数で、1 は患者がその人種を記録していることを意味し、0 はそうでないことを意味します。
- ・ 民族：各民族性に関するバイナリ変数で、1 は患者がその民族性を記録していることを意味し、0 はそうでないことを意味します。
- ・ インデックス年：各コホート開始日の年を表すバイナリ変数で、1 は患者

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: 共変量の包括と除外設定

のコホート開始年、0 はそれ以外を表します。インデックス年を含めるこ
とは、しばしば意味をなさないことがあります。なぜなら、私たちは将来
にわたってモデルを適用したいからです。

- ・インデックス月：各コホート開始日の月を表すバイナリ変数で、1 は患者
のコホート開始日の月を表し、0 はそれ以外を表します。
- ・前観察期間：[予測には推奨されません] コホート開始日以前に患者がデ
ータベースに存在した日数を示す連続変数
- ・後観察期間：[予測には推奨されません] コホート開始日以降に患者がデ
ータベースに存在した日数を示す連続変数
- ・コホート時間：患者がコホートに属していた日数（コホート終了日－コ
ホート開始日）に対応する連続変数
- ・インデックス年と月：[予測には推奨されません] 各コホート開始日の年
と月の組み合わせを表すバイナリ変数。1 は患者のコホート開始日の年と
月であることを表し、0 はそれ以外を表します。

これが完了すると、このセクションは図 13.7 のようになるはずです。

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: 共変量の選択

標準の共変量は共変量に対して柔軟な 3 つの時間間隔を設定できます：

- ・終了日までの日数：コホート開始日からの終了日まで [デフォルトは 0]
- ・長期 [デフォルトはコホート開始前 365 日から終了日まで]

- 中期 [デフォルトはコホート開始前 180 日から終了日まで]
- 短期 [デフォルトはコホート開始前 30 日から終了日まで]

これが完了すると、このセクションは図 13.8 のようになるはずです。

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: 時間に依存する共変量

次のオプションは、期間テーブルから抽出される共変量です：

- コンディション：選択された各コンディションのコンセプト ID と時間間隔ごとに共変量を構築し、CONDITON_ERA テーブルにおいて、選択されたコホート開始日前の時間間隔の間に、患者にコンディション期間をもつ（すなわち、その時間間隔の間にそのコンディションが開始または終了するか、またはその時間間隔の前に開始し、その時間間隔の後に終了する）コンセプト ID がある場合、共変量の値は 1、そうでない場合は 0。
- コンディショングループ：選択されたコンディションのコンセプト ID と時間間隔について共変量を構築し、CONDITON_ERA テーブルにおいて、選択されたコホート開始日前の時間間隔の間に、患者にコンディション期間を持つコンセプト ID またはその下位層のコンセプト ID がある場合、共変量値は 1、そうでない場合は 0。
- 薬剤：選択された各薬剤コンセプト ID と時間間隔ごとに共変量を構築し、DRUG_ERA テーブルにおいて、選択されたコホート開始日前の時間間隔の間に、患者に薬剤（曝露）期間をもつコンセプト ID がある場合、共変量の値は 1、そうでない場合は 0。
- 薬剤グループ：選択された各薬剤コンセプト ID と時間間隔ごとに共変量を構築し、DRUG_ERA テーブルにおいて、選択されたコホート開始日前の時間間隔の間に、患者に薬剤（曝露）期間をもつコンセプト ID またはその下位層のコンセプト ID がある場合、共変量の値は 1、そうでない場合は 0。

重複する時間間隔の設定は、薬剤または症状がコホート開始日以前に開始し、終了がコホート開始日以後に続くものが含まれます。期間の開始オプションは時間間隔内に開始したものに限定します。

これが完了すると、このセクションは図 13.9 のようになるはずです。

次のオプションは、さまざまな時間間隔について、各ドメインのコンセプト ID に対応する共変量を選択します。：

- コンディション：選択されたコンディションコンセプト ID と時間間隔ごとに共変量を構築し、CONDITION_OCCURRENC テーブルのコホート開

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: 期間時間共変量.

始日前の指定された時間間隔に、患者がそのコンセプト ID を記録している場合、共変量の値は 1、そうでない場合は 0。

- 主たる入院コンディション (Condition Primary Inpatient) : condition_occurrence テーブルで入院患者の主たる診断として、CONDITION_OCCURRENCE テーブル中に観察されたコンディションごとのバイナリ共変量。
- 薬剤：選択された薬剤コンセプト ID と時間間隔ごとに共変量を構築し、DRUG_EXPOSURE テーブルのコホート開始日前の指定された時間間隔に、患者がコンセプト ID を記録している場合、共変量の値は 1、そうでない場合は 0。
- 処置（プロシージャー）：選択されたプロシージャー コンセプト ID と時間間隔ごとに共変量を構築し、PROCEDURE_OCCURRENCE テーブルのコホート開始日前の指定された時間間隔に、患者がコンセプト ID を記録している場合、共変量の値は 1、そうでない場合は 0。
- 測定（メジャーメント）：選択されたメジャーメントコンセプト ID と時間間隔ごとに共変量を構築し、MEASUREMENT テーブルのコホート開始日前の指定された時間間隔に、患者がコンセプト ID を記録している場合、共変量の値は 1、そうでない場合は 0。
- 測定値：測定値が伴う選択された測定値コンセプト ID と時間間隔ごとに共変量を構築し、MEASUREMENT テーブルのコホート開始日前の指定された時間間隔に、患者がコンセプト ID を記録している場合、共変量の値は 1、そうでない場合は 0。
- 測定値範囲グループ：測定値が正常範囲以下、範囲内、または正常範囲以上であるかを示すバイナリ共変量。
- 観察（オブザベーション）：選択された観察コンセプト ID と時間間隔ごとに共変量を構築し、OBSERVATION テーブルのコホート開始日前の指定された時間間隔に、患者がそのコンセプト ID を記録している場合、共変量の値は 1、そうでない場合は 0。
- デバイス：選択されたデバイスコンセプト ID と時間間隔ごとに共変量を構築し、DEVICE テーブルのコホート開始日前の指定された時間間隔に、

患者がそのコンセプト ID を記録している場合、共変量の値は 1、そうでない場合は 0。

- ・ ビジット回数：選択されたビジット回数と時間間隔ごとに共変量を構築し、その時間間隔に記録されたビジット回数を共変量値としてカウント。
- ・ ビジットコンセプト数：選択された各ビジット、ドメイン、時間間隔ごとに共変量を構築し、その時間間隔に記録されたレコード数を、各ドメインのビジットタイプと時間間隔ごとに共変量値としてカウント。

“distinct count (重複を除いたカウント)” オプションは、ドメインと時間間隔ごとに、異なるコンセプト ID の数をカウントします。

これらが完了すると、このセクションは下図 13.10 のようになっているはずです。

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: 時間制約共変量

最後のオプションは、一般に使われるリスクスコアを共変量として含めるかどうかです。設定が完了すると、リスクスコアの設定は図 13.11 のようになります。

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: リスクスコア共変量設定

研究対象集団の設定

対象集団の設定は、追加の適格基準を対象集団に適用できる場所であり、また、リスク期間もここで定義されます。研究に対象集団の設定を追加するには、“Add Population Settings (対象集団の追加)” ボタンをクリックします。これにより、対象集団の設定ビューが表示されます。

最初のオプションセットでは、リスク期間を指定することができます。これは、対象とするアウトカムが起こるかどうかを観察する時間間隔です。リスク期間中に患者にアウトカムが起きた場合は “Has outcome (アウトカムあり)” に分類し、そうでない場合は “No outcome (アウトカムなし)” に分類します。

“Define the time-at-risk window start, relative to target cohort entry: (ターゲットコホート組入れを基準に、リスク時間ウインドウの開始を定義します。) は、ターゲットコホートの開始または終了日を基準としたリスク期間の開始を定義します。同様に、” Define the time-at-risk window end: (対象コホートの開始または終了日を基準とした、リスク期間の終了を定義します。) は、リスク期間の終了を定義します。

“Minimum lookback period applied to target cohort (対象コホートに適用される最小のルックバック期間)” は、患者がコホート開始日より前の継続的に観察された日数の最低値である、最小のベースライン期間を指定します。デフォルトは 365 日です。最小のルックバック期間を長くすると、患者の全体像がより明確になりますが（より長い期間観察されているはずであるため）、開始日前の観察が最低日数に満たない患者は除外されます。

“Should subjects without time at risk be removed? (リスク期間がない対象は除外すべきですか。)” が “Yes (はい)” に設定されている場合、“Minimum time at risk: (最低リスク時間: :)” の値も必要となります。これにより、追跡不能となった人（すなわち、リスク期間中にデータベースから離脱した人）を除外することができます。例えば、リスク期間がコホート開始後 1 日からコホート開始後 365 日までであった場合、リスク期間は 364 日間 (365-1) となります。もし、その全期間にわたって観察された患者のみを含めたいのであれば、最小リスク時間を 364 に設定します。最初の 100 日間リスク期間に該当していればよいのであれば、最小リスク期間を 100 に設定します。この場合、リスク期間の開始はコホート開始から 1 日後なので、コホート開始日から少なくとも 101 日間データベースに存在する患者が対象となります。” Should subjects without time at risk be removed? (リスク期間がない対象は削除すべきですか。)” を “No (いいえ)” に設定すると、リスク期間中にデータベースから離脱した患者も含め、すべての患者が対象となります。

“Include people with outcomes who are not observed for the whole at risk period? (全リスク期間で観察されていないアウトカムを持つ人を含めますか。)” というオプションは、前のオプションに関連しています。” Yes (はい)” に設定すると、指定された最低リスク期間で観察されていない場合でも、リスク期間中にアウトカムを経験した人は常にコホートに保持されます。

“Should only the first exposure per subject be included? (対象ごとに最初の

曝露のみを含めるべきですか)” というオプションは、対象コホートに異なるコホート開始日で複数回含まれる患者がいる場合にのみ有用です。この状況で “Yes (はい)” を選択すると、分析では患者ごとに最も早い対象コホートの日付のみが保持されます。そうでない場合、患者はデータセットに複数回含まれる可能性があります。

“Remove patients who have observed the outcome prior to cohort entry? (コホート組入れの前にアウトカムが観察された患者を除外しますか)” を “Yes (はい)” に設定すると、リスク期間開始日より前にアウトカムを経験した患者を除外するため、そのモデルは以前にアウトカムを経験したことがない患者を対象とします。もし “No (いいえ)” が選択されると、患者は以前にアウトカムを持つ可能性があります。患者が以前にアウトカムを経験したことが、リスク期間中にアウトカムが起きることを非常に高い確率で予測することが多いです。

完了すると、対象集団の設定のダイアログは図 13.12 のようになります。

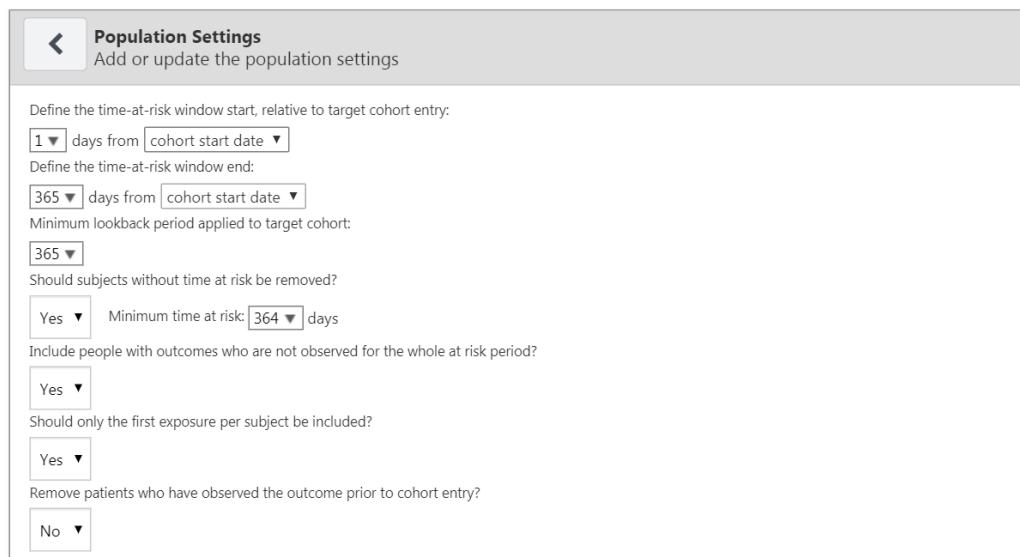


Figure 13.12: 対象集団の設定

これで分析の設定が終わり、ダイアログ全体が図 13.13 のようになります。

13.6.3 実行の設定

オプションは 3 つあります：

- Perform sampling (サンプリングの実行)：ここでサンプリングを実行するかどうかを選択します（デフォルトは “no (いいえ)”）。 “yes (はい)” を選択すると、別のオプションが表示されます：“How many patients to use for a subset? (サブセットに何人の患者を含めますか。)” で、サンプルの大きさを指定できます。サンプリングは、大規模な母集団（例えば

The screenshot shows the 'Analysis Settings' interface with three main sections:

- Model Settings**: Shows a single entry for GradientBoostingMachineSettings with the following JSON configuration:

```
{"ntrees": [5000], "nthread": 20, "maxDepth": [4, 7, 10], "minRows": [20], "learnRate": [0.001, 0.01, 0.1, 0.9], "seed": null}
```
- Covariate Settings**: Shows a single entry for DemographicsGender, DemographicsAgeGroup, DemographicsRace, DemographicsEthnicity, DemographicsIndexMonth, ConditionGroupEraLongTerm, and 12 more covariate settings.
- Population Settings**: Shows a single entry with the following configuration:

Remove	Risk Window Start	Risk Window End	Washout Period	Include All Outcomes	Remove Subjects With Prior Outcome	Minimum Time At Risk
X	1d from cohort start date	365d from cohort start date	365d	true	false	364d

Figure 13.13: 分析の設定

1000 万人の患者) のモデルが予測可能かどうかを判断する効率的な手段となり得ます。例えば、サンプルにおける AUC が 0.5 に近い場合、そのモデルは破棄されるかもしれません。

- “Minimum covariate occurrence: If a covariate occurs in a fraction of the target population less than this value, it will be removed: (最小共変量出現率 : もし共変量がこの値より小さい割合で対象集団に出現する場合、その共変量は除外されます:)” : ここでは共変量の出現率の最小値を選択します（デフォルトは 0.001）。共変量出現の最小閾値は、全体集団を代表しないまれなイベントを除外するために必要です。
- “Normalize covariate (共変量を正規化する)” : ここで共変量を正規化するかどうかを選択します（デフォルトは “yes (はい)”）。共変量の正規化は、通常、LASSO モデルをうまく実行するために必要です。

この例では、図 13.14 のように選択します。

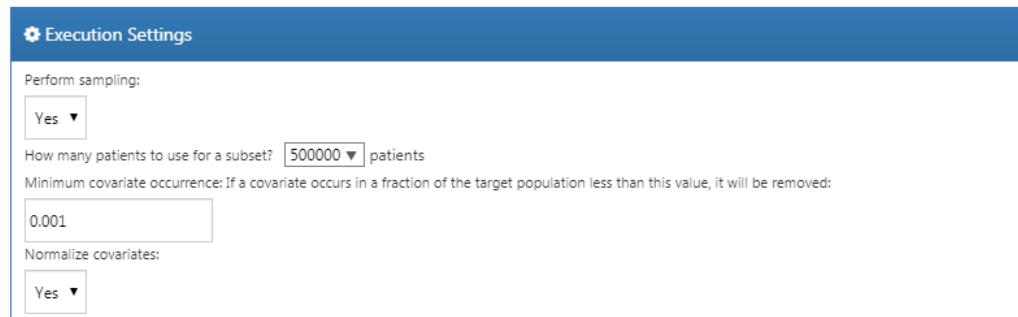


Figure 13.14: 実行の設定

13.6.4 トレーニングの設定

4 つのオプションがあります :

- “Specify how to split the test/train set (テストセットとトレーニングセットをどのように分けるかを指定します。)” : トレーニング/テストデータを人別（アウトカムで層別化）または時間別（古いデータをトレーニングに、新しいデータは評価に）に分割するかを選択します。
- “Percentage of the data to be used as the test set (0-100%) (テストセットとして使用するデータの割合 (0-100%))” : テストデータとして使用するデータの割合を選択します（デフォルトは 25%）。
- “The number of folds used in the cross validation (クロスバリデーションで使用するフォールド数)” : 最適なハイパーパラメータを選択するために使用するクロスバリデーションのフォールド数を選択します（デフォルトは 3）。

- “The seed used to split the test/train set when using a person type testSplit (optional): (人単位で testSplit を使う場合の、テストセットとトレーニングセットを分割するためのシード (オプション) :)” : 人単位でテストセットとトレーニングセットを分割する場合に、分割に使用するランダムシードを選択します。

この例では、図 13.15 のように選択します。

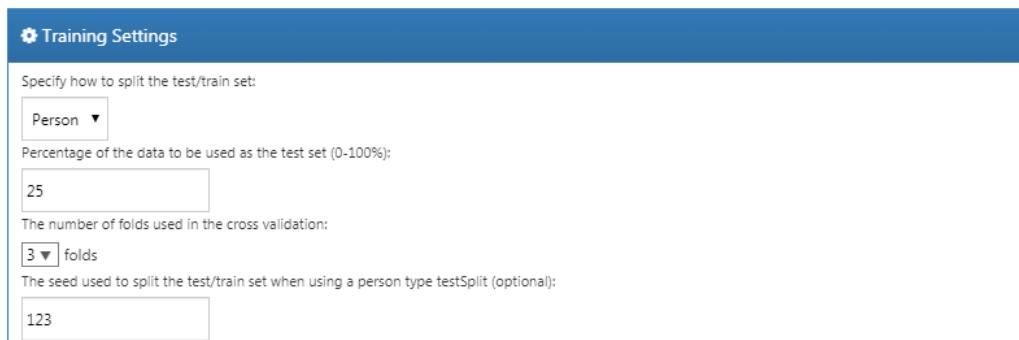


Figure 13.15: トレーニングの設定

13.6.5 研究のインポートとエクスポート

研究をエクスポートするには、“Utilities (ユーティリティ)” の下にある “Export (エクスポート)” タブをクリックします。ATLAS は、研究の名称、コホート定義、選択したモデル、共変量、設定など、研究実行に必要なすべてのデータを含むファイルに直接コピー＆ペーストできる JSON を作成します。

研究をインポートするには、“Utilities (ユーティリティ)” の下にある “Import (インポート)” タブをクリックします。患者レベルの予測研究の JSON の内容をこのウィンドウに貼り付け、他のタブボタンの下にある “Import (インポート)” ボタンをクリックします。これにより、その研究の以前の設定がすべて上書きされることに注意してください。通常は、新しい空の研究デザインを使用して実行します。

13.6.6 研究パッケージのダウンロード

“Utilities (ユーティリティ)” の下にある “Review & Download (レビューとダウンロード)” レビューとダウンロード” タブをクリックします。“Download Study Package (研究パッケージをダウンロード)” セクションで、R パッケージのわかりやすい名前を入力します。R で使えない文字は、ATLAS によって自動的にファイル名から削除されることに注意してください。 をクリックして、R パッケージをローカルフォルダにダウンロードします。

13.6.7 研究の実行

セクション 8.4.5 の説明のように、R パッケージを実行するには、R、RStudio、Java がインストールされている必要があります。また、R で下記のようにインストールできる PatientLevelPrediction パッケージも必要です：

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

機械学習アルゴリズムの中には、追加ソフトウェアのインストールが必要なものがあります。PatientLevelPrediction パッケージのインストール方法の詳細については、“Patient-Level Prediction Installation Guide” vignette を参照ください。

study R パッケージを使用するには、R Studio の使用をお勧めします。R Studio をローカルで実行している場合は、ATLAS で生成されたファイルを解凍し、.Rproj ファイルをダブルクリックして R Studio で開きます。R スタジオを R スタジオサーバーで実行している場合は、 Upload をクリックしてファイルをアップロードし、解凍した後、.Rproj ファイルをクリックしてプロジェクトを開きます。

R Studio でプロジェクトを開いたら、README ファイルを開き、指示に従ってください。すべてのファイルのパスを、システム上の既存のパスに変更してください。

13.7 R での研究実施

研究デザインを ATLAS で実装する代わりに、R で直接コードを記述して実施することもできます。ここでは、PatientLevelPrediction パッケージを利用します。このパッケージは、OMOP CDM に変換されたデータベースからデータを抽出し、モデルを構築し、評価することができます。

13.7.1 コホートのインスタンス化

まず、ターゲットコホートとアウトカムコホートをインスタンス化する必要があります。コホートのインスタンス化については、第 10 章で説明しています。付録にはターゲットコホート（付録 B.1）とアウトカムコホート（付録 B.4）の完全な定義があります。この例では、ACE 阻害薬コホートの ID が 1、血管浮腫コホートの ID が 2 であると仮定します。

13.7.2 データ抽出

まず、R にサーバへの接続方法を伝える必要があります。PatientLevelPrediction は、DatabaseConnector パッケージを使用します。このパッケージには

`createConnectionDetails` という関数があります。さまざまなデータベース管理システム (DBMS) の設定については、`?createConnectionDetails` と入力してください。例えば、次のように PostgreSQL データベースに接続できます：

```
library(PatientLevelPrediction)
connDetails <- createConnectionDetails(dbms = "postgresql",
                                         server = "localhost/ohdsi",
                                         user = "joe",
                                         password = "supersecret")

cdmDbSchema <- "my_cdm_data"
cohortsDbSchema <- "scratch"
cohortsDbTable <- "my_cohorts"
cdmVersion <- "5"
```

最後の 4 行は `cdmDbSchema`、`cohortsDbSchema`、`cohortsDbTable` 変数の定義と、CDM バージョンを指定しています。これらを使用して CDM フォーマットのデータが存在する場所、関心のあるコホートが作成された場所、および使用されている CDM バージョンを R に伝えます。Microsoft SQL Server の場合、データベーススキーマはデータベースとスキーマの両方を指定する必要があることに注意してください。例えば、`cdmDbSchema <- "my_cdm_data.dbo"` のように指定します。

まず、コホート作成が成功したかを確認するために、コホート組入れの数をカウントします：

```
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
```

`PatientLevelPrediction` に我々の分析に必要なすべてのデータを抽出するように指示します。共変量は `FeatureExtraction` パッケージを使用して抽出します。`FeatureExtraction` パッケージの詳細については、そのビネットを参照してください。今回の研究例では次の設定を使用しました：

```
covariateSettings <- createCovariateSettings(
  useDemographicsGender = TRUE,
  useDemographicsAge = TRUE,
  useConditionGroupEraLongTerm = TRUE,
  useConditionGroupEraAnyTimePrior = TRUE,
  useDrugGroupEraLongTerm = TRUE,
  useDrugGroupEraAnyTimePrior = TRUE,
  useVisitConceptCountLongTerm = TRUE,
  longTermStartDays = -365,
  endDays = -1)
```

データ抽出の最終ステップは、`getPlpData` 関数を実行し、接続の詳細、コホートが保存されているデータベーススキーマ、コホートと結果のコホート定義 ID、およびコホートインデックス日付の前にデータに含めるために観察されなければならない最低日数であるウォッシュアウト期間を入力し、最後に以前に構築した共変量設定を入力することです。

```
plpData <- getPlpData(connectionDetails = connDetails,
                        cdmDatabaseSchema = cdmDbSchema,
                        cohortDatabaseSchema = cohortsDbSchema,
                        cohortTable = cohortsDbSchema,
                        cohortId = 1,
                        covariateSettings = covariateSettings,
                        outcomeDatabaseSchema = cohortsDbSchema,
                        outcomeTable = cohortsDbSchema,
                        outcomeIds = 2,
                        sampleSize = 10000
  )
```

`getPlpData` 関数には多くの追加パラメータがあります。これらはすべて PatientLevelPrediction マニュアルに詳細に記載されています。生成された `plpData` オブジェクトは `ff` パッケージを使用し、大量のデータでも R のメモリ不足を避けるようにデザインされています。

`plpData` オブジェクトの生成にはかなりの計算時間がかかります。おそらく、今後のセッション用に保存しておくのが良いでしょう。`plpData` は `ff` を使用しているため、R の通常の保存機能を使用することはできません。代わりに、`savePlpData` 関数を使用します：

```
savePlpData(plpData, "angio_in_ace_data")
```

`loadPlpData()` 関数を使用して、今後のセッションでデータを読み込むことができます。

13.7.3 追加の適格基準

最終的な研究対象集団は、先に定義した 2 つのコホートに追加の制約を適用することによって得られます。例えば、最小リスク期間を設定することができます (`requireTimeAtRisk`, `minTimeAtRisk`) し、これがアウトカムを持つ患者にも適用されるかどうかを指定できます (`includeAllOutcomes`)。ここでは、ターゲットコホートの開始時点を基準としたリスク期間の開始と終了も指定します。例えば、リスクのあるコホート開始から 30 日後にリスク期間を開始し、1 年後に終了したい場合は、`riskWindowStart = 30`, `riskWindowEnd = 365` と設定します。場合によっては、リスク期間をコホート終了日に開始する必要があります。これは、`addExposureToStart = TRUE` を設定し、コホート（曝露）期間を開始日に加算することで実現できます。

以下の例では、研究用に定義したすべての設定を適用しています：

```
population <- createStudyPopulation(plpData = 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,
                                      verbosity = "DEBUG"
)
```

13.7.4 モデル開発

アルゴリズムのセット関数では、ユーザーは各ハイパーパラメータの有効な値のリストを指定することができます。ハイパーパラメータのすべての可能な組み合わせは、トレーニングセットのクロスバリデーションを使用したいわゆるグリッドサーチに含まれます。ユーザーが値を指定しない場合は、代わりにデフォルト値が使用されます。

例えば、次の設定を勾配ブースティングマシン (Gradient Boosting Machine) で使用するとします：`ntrees = c(100, 200)`, `maxDepth = 4`。このグリッドサーチは、`ntrees = 100` および `maxDepth = 4`、または `ntrees = 200` および `maxDepth = 4` の設定でデフォルトの他のハイパーパラメータ設定を含めて勾配ブースティングマシンアルゴリズムを適用します。クロスバリデーションの性能が最も高いハイパーパラメータが最終モデルに選ばれます。この課題では、いくつかのハイパーパラメータ値で勾配ブースティングマシンを構築することにしました：

```
gbmModel <- setGradientBoostingMachine(ntrees = 5000,
                                         maxDepth = c(4,7,10),
                                         learnRate = c(0.001,0.01,0.1,0.9))
```

`runPlp` 関数は集団、`plpData`、モデル設定を使用してモデルをトレーニングし評価します。データを 75%-25% に分割して患者レベルの予測パイプラインを実行するために `testSplit` (人/時間) および `testFraction` パラメータを使用できます：

```
gbmResults <- runPlp(population = population,
                      plpData = plpData,
                      modelSettings = gbmModel,
                      testSplit = 'person',
                      testFraction = 0.25,
                      nfold = 2,
                      splitSeed = 1234)
```

このパッケージは内部的に R の `xgboost` パッケージを使用して、75% のデータを用いて勾配ブースティングマシンモデルを適合させ、残りの 25% のデータでモデルを評価します。アウトカムデータ構造には、モデルやその性能などに関する情報が含まれます。

`runPlp` 関数には、`plpData`、`plpResults`、`plpPlots`、`evaluation` などのオブジェクトを保存するためのいくつかのパラメータがあり、デフォルトで `TRUE` に設定されています。

モデルを保存するには：

```
savePlpModel(gbmResults$model, dirPath = "model")
```

モデルを読み込むには：

```
plpModel <- loadPlpModel("model")
```

完全なアウトカム構造を保存することもできます：

```
savePlpResult(gbmResults, location = "gbmResults")
```

完全なアウトカム構造を読み込むには：

```
gbmResults <- loadPlpResult("gbmResults")
```

13.7.5 内部バリデーション

学習を実行すると、`runPlp` 関数は学習済みのモデルと、学習用/テスト用セットにおけるモデルの評価を返します。`viewPlp(runPlp = gbmResults)` を実行すると、結果をインタラクティブに表示できます。これにより、Shiny App が開き、フレームワークによって作成されたすべてのパフォーマンス指標（インタラクティブなプロットを含む）を表示できます（Shiny Application のセクションの図 13.16を参照）。

すべての評価プロットをフォルダーに生成して保存するには、次のコードを実行します：

```
plotPlp(gbmResults, "plots")
```

プロットの詳細については、セクション 13.4.2を参照ください。

13.7.6 外部バリデーション

常に外部バリデーションを行うことをお勧めします。すなわち、最終モデルを可能な限り多くの新しいデータセットに適用し、そのパフォーマンスを評価します。ここでは、データ抽出がすでに 2 番目のデータベース上で実行され、`newData` フォルダに格納されていると仮定します。以前にフィッティングしたモデルを `model` フォルダーから読み込みます：

```
# トレーニング済みモデルをロード
plpModel <- loadPlpModel("model")

# 新しい plpData をロードし、集団を作成
plpData <- loadPlpData("newData")

population <- createStudyPopulation(plpData = 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)
```

さらに簡単にできるように、必要なデータの抽出も行う外部検証を行うための externalValidatePlp 関数も提供しています。result <- runPlp(...) を実行したと仮定すると、モデルに必要なデータを抽出して、新しいデータで評価することができます。検証対象集団が ID 1 と 2 のテーブル mainschema.dob.cohort にあり、CDM データがスキーマ cdmschema.dob にあると仮定すると：

```
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  
)
```

モデルを検証する複数のデータベースがある場合、以下を実行できます：

```
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 結果の公表

13.8.1 モデル性能

予測モデルのパフォーマンスを調査するには、viewPlp 関数が最も簡単です。この関数には、結果オブジェクトを入力として指定する必要があります。R でモデルを開発している場合は、runPLp の結果を入力として使用できます。ATLAS で生成されたスタディパッケージを使用している場合は、モデルの 1 つを読み込む必要があります（この例では、Analysis_1 を読み込みます）。

```
plpResult <- loadPlpResult(file.path(outputFolder,
                                         'Analysis_1',
                                         'plpResult'))
```

ここで「Analysis_1」は先に特定した分析に対応しています。

次に、以下を実行して Shiny アプリケーションを起動します。

```
viewPlp(plpResult)
```

Shiny アプリケーションはテストセットとトレインセットの性能指標の要約から始まります（図 13.16 参照）。結果を見ると、トレーニングセットの AUC は 0.78 であり、テストセットではこれが 0.74 に低下しています。テストセットの AUC の方がより正確な指標です。全体的には、このモデルは ACE 阻害薬の新規ユーザーで結果がどうなるかを予測できそうですが、トレーニングセットのパフォーマンスがテストセットよりも高いことから、やや過適合しているようです。ROC プロットは図 13.17 に示されています。

図 13.18 のキャリブレーションプロットでは、一般的に観測されたリスクと予測されたリスクが一致していることが、点が対角線付近に位置していることから分かります。しかし、図 13.19 の人口統計学的キャリブレーションプロットでは、40 歳未満の患者については、青線（予測リスク）が赤線（観測リスク）と異なっていることから、モデルのキャリブレーションがうまくいっていないことが分かります。これは、40 歳未満の患者をターゲット集団から除外する必要があることを示しているのかもしれません（若い患者の観察されたリスクはほぼゼロだからです）。

最後に、適格基準に基づくラベル付きデータからの患者の脱落を示す attrition プロットがあります（図 13.20 参照）。このプロットは、リスク期間（1 年間の追跡調査）全体にわたって観察されなかつたために、対象集団の大部分を失ったことを示しています。興味深いことに、結果の患者の多くは、リスク期間全体を欠損していました。

The screenshot shows a Shiny application window titled "PatientLevelPrediction Explorer". The top navigation bar includes tabs for "Internal Validation" and "External Validation". Below the tabs, a horizontal menu bar contains "Evaluation Summary", "Characterization", "ROC", "Calibration", "Demographics", "Preference", "Box Plot", and "Settings". The main content area is titled "Evaluation Summary" and displays a table of 11 rows. The table has three columns: "Metric", "test", and "train". The "Metric" column lists various performance metrics, and the "test" and "train" columns show their respective values. At the bottom of the table, it says "Showing 1 to 11 of 11 entries". Navigation buttons for "Previous", "1", and "Next" are also present.

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 アプリケーションにおける評価統計の要約

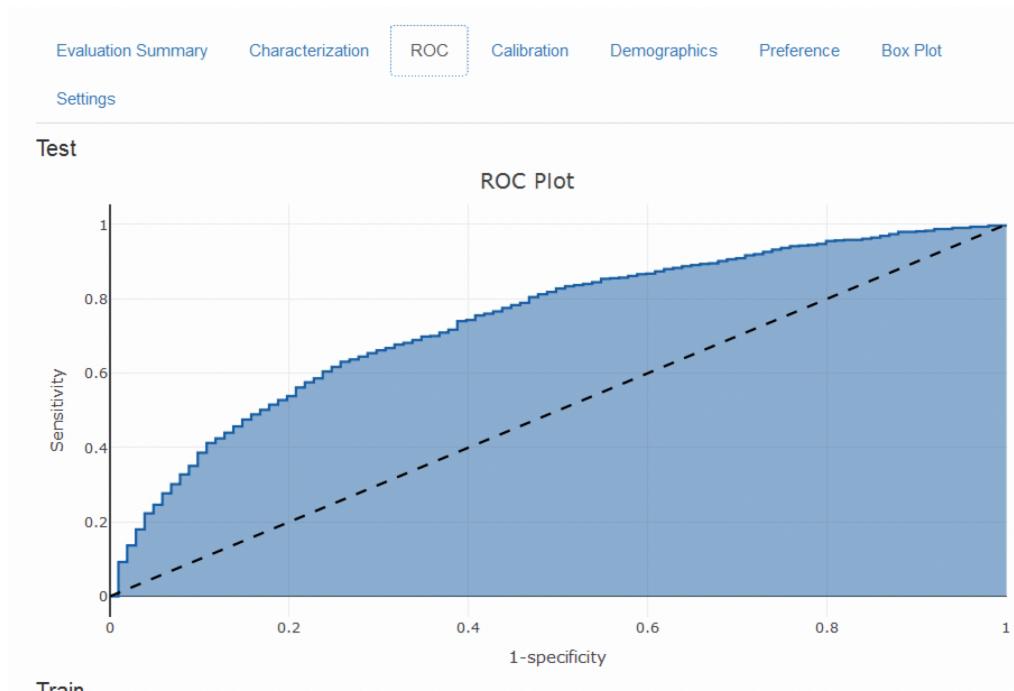


Figure 13.17: ROC プロット

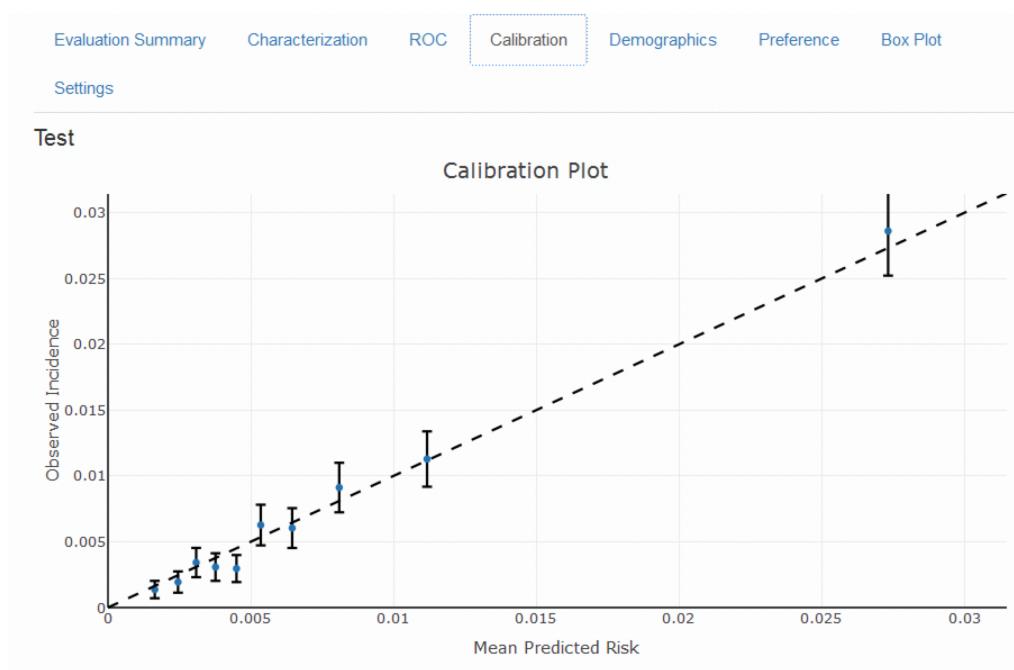


Figure 13.18: モデルのキャリブレーション

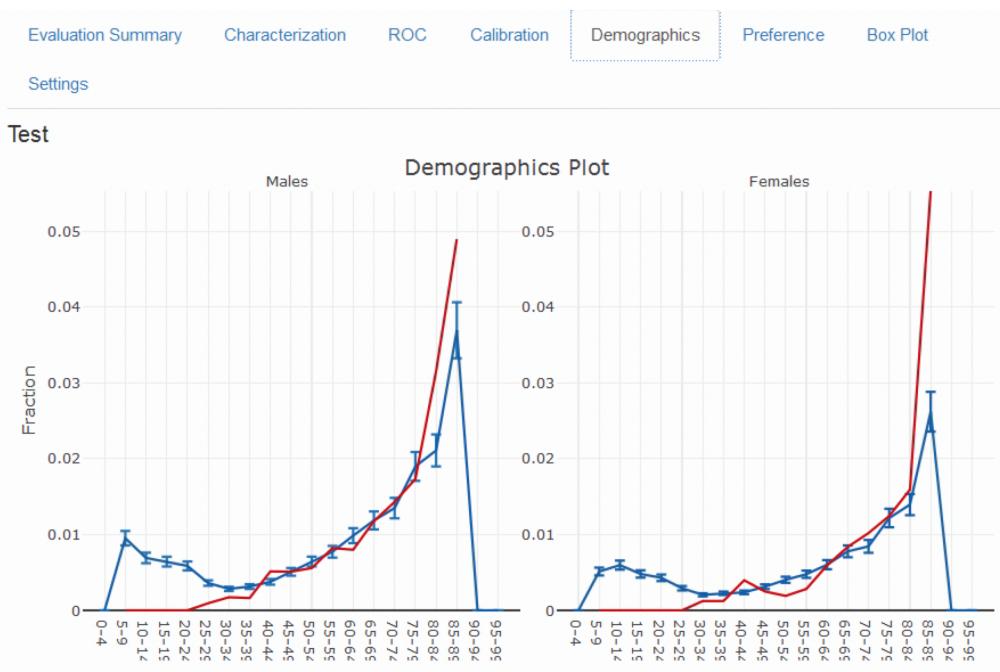


Figure 13.19: モデルの人口統計学的キャリブレーション

The table displays attrition information for four cohorts. The columns are "description", "targetCount", "uniquePeople", and "outcomes". The first three rows represent different cohorts, while the fourth row shows the total count of people at risk.

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: 予測問題における attrition プロット

13.8.2 モデルの比較

ATLAS によって生成されたスタディパッケージでは、さまざまな予測問題に対して、多くの異なる予測モデルを生成および評価することができます。そのため、特にスタディパッケージによって生成された出力用に、複数のモデルを表示するための追加の Shiny アプリが開発されました。このアプリを起動するには、`viewMultiplePlp(outputFolder)` を実行します。ここで `outputFolder` は、`execute` コマンドを実行する際に指定した分析結果を含むパスです（例えば、「Analysis_1」という名前のサブフォルダを含む必要があります）。

モデルの要約と設定の表示

インタラクティブな Shiny アプリは、図 13.21 に示す要約ページで起動します。

The screenshot shows a Shiny application interface with the following layout:

- Left sidebar (Filters):**
 - Development Database:** All
 - Validation Database:** All
 - Target Cohort:** New users of ACE inhibitors as first-line monotherapy for hypertension
 - Outcome Cohort:** All
- Top navigation:** Results, Model Settings, Population Settings, Covariate Settings, Show 10 entries, Search.
- Table:** A data grid showing model performance metrics. The columns are:
 - Analysis
 - Dev
 - Val
 - T
 - O
 - Model
 - TAR start
 - TAR end
 - AUC
 - AUPRC
 - T Size
 - O Count
 - Incidence (%)
- Bottom:** Showing 1 to 4 of 4 entries, Previous, Next.

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	0.71867	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	0.64283	0.02447	87615	148	0.16892

Figure 13.21: 各モデルの訓練に使用されたホールドアウトセットの主要な性能指標を含む Shiny の要約ページ

この要約ページの表には以下が含まれています：

- モデルに関する基本情報（例：データベース情報、分類器タイプ、リスク期間設定、対象母集団、アウトカム名）
- ホールドアウト n 対象母集団の数とアウトカム発生率
- 判別指標 : AUC, AUPRC

表の左側にはフィルターオプションがあり、開発/検証データベース、モデルの種類、リスク期間の設定および/または対象とするコホートを指定できます。例えば、対象集団「高血圧症の第一選択の単剤療法としての ACE 阻害薬の新規ユーザー」に対応するモデルを選択するには、「Target Cohort」オプションでこれを選択します。

モデルを詳細に調査するには、該当する行をクリックします。選択された行はハイライト表示されます。行が選択された状態で「Model Settings」タブをクリックしてモデルを開発する際に使用した設定を調べることができます。

同様に、他のタブでモデルを生成するために使用された母集団および共変量の設定を調べることもできます。

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: モデルを開発する際に使用した設定を表示する

モデル性能の表示

モデル行が選択されると、モデル性能も表示できます。[Performance] をクリックして閾値の性能評価の要約を表示できます（図 13.23 参照）。

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	542
Predicted Negative	42892	108

Figure 13.23: 特定の閾値における性能評価の要約

このサマリー表示は標準形式で選択された予測問題を表示し、閾値セレクタと陽性的中率 (PPV)、陰性的中率 (NPV)、感度、特異度（セクション 13.4.2 を参照）などの重要な閾値ベースのメトリクスを含むダッシュボードを表示します。図 13.23 では、感度が 83.4%（翌年に結果が判明する患者の 83.4% が 0.00482 以上のリスクを持つ）で、PPV が 1.2%（0.00482 以上のリスクを持つ患者の 1.2% が翌年に結果が判明する）であることが示されています。1 年以内のアウトカム発生率が 0.741% であるため、0.00482 以上のリスクを持つ

患者を特定すると、集団平均リスクのほぼ2倍のリスクを持つ患者サブグループが見つかります。スライダーを使用して閾値を調整し、他の値でのパフォーマンスを表示することができます。

モデル全体の識別力を確認するには、「Discrimination」タブをクリックしてROCプロット、精度-再現プロット、分布プロットを表示します。プロットの線は選択された閾値ポイントに対応しています。図13.24はROCおよび精度-再現プロットを示しています。ROCプロットは、モデルが1年内にアウトカムが発生する人と発生しない人を区別できることを示しています。しかし、精度-再現プロットを見ると性能はそれほど印象的でないよう見えます。なぜなら、結果の発生率が低いということは、偽陽性率が高いことを意味するからです。

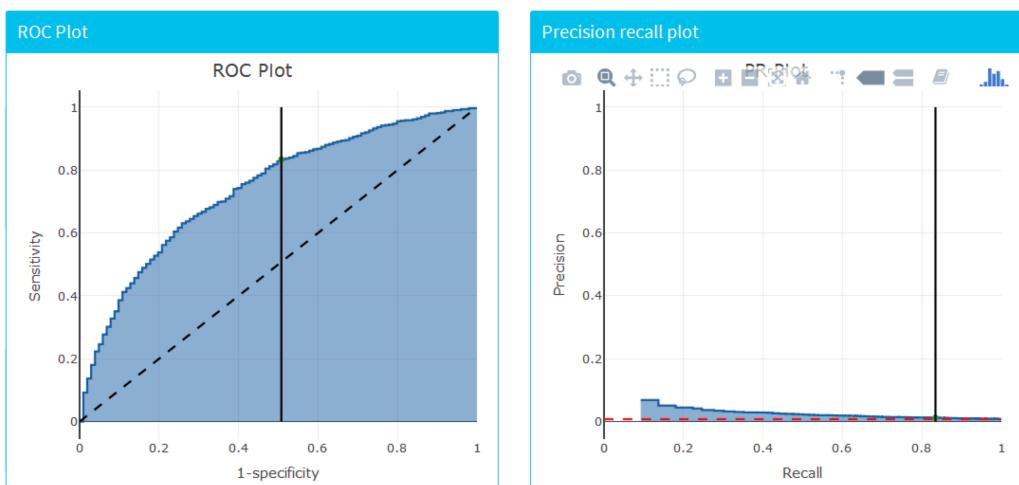


Figure 13.24: ROC および精度-再現プロットを使用してモデルの判別能力全体を評価する

図13.25は予測リスクおよび選好スコア分布を示しています。

最後に、「Calibration」タブをクリックしてモデルのキャリブレーションを確認することもできます。これにより、図13.26に示されるキャリブレーションプロットおよび人口統計学的キャリブレーションが表示されます。

1年内にアウトカムを経験したグループの予測リスクと観察されたアウトカムの割合が一致しているように見えるので、モデルはよくキャリブレーションされています。興味深いことに、人口統計学的キャリブレーションは、若年患者の場合、予測されたリスクが観察されたリスクよりも高いことを示しています。逆に80歳以上の患者の場合、モデルは観察されたリスクよりも低いリスクを予測しています。これは、若年および高齢者のために別々のモデルを開発する必要があることを示唆しているかもしれません。

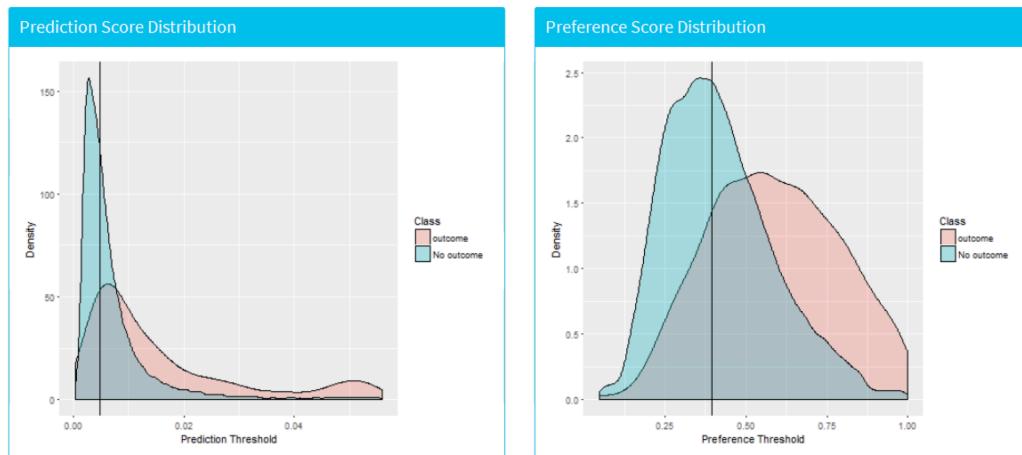


Figure 13.25: アウトカム有およびア outing の患者に対する予測リスク分布。重なりが多いほど、判別が悪化します。

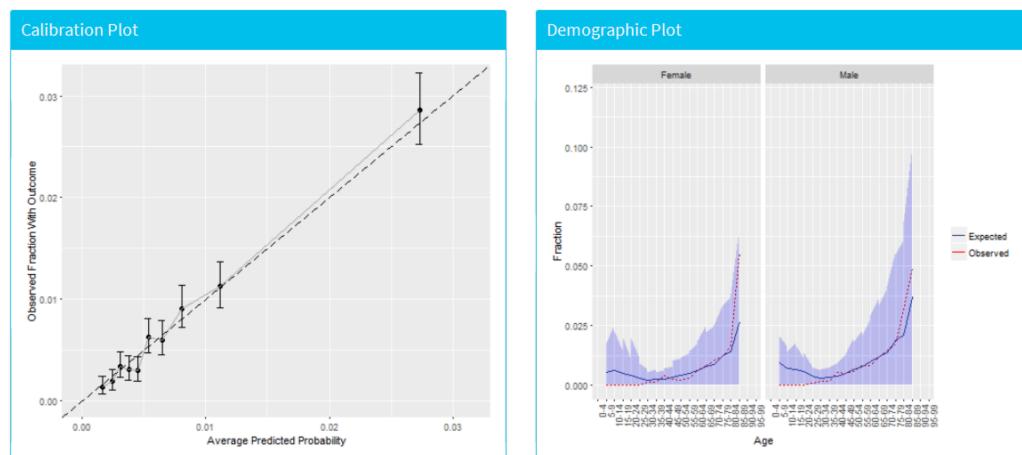


Figure 13.26: リスク層別キャリブレーションおよび人口統計学的キャリブレーション

モデルの表示

最終モデルを検査するには、左側のメニューから  Model オプションを選択します。これにより、図 13.27 に示すモデル内の各変数のプロットと図 13.28 に示すすべての候補の共変量を要約するテーブルが表示されます。変数プロットはバイナリ変数と連続変数に分かれています。X 軸はアウトカムがない患者の中での有病率/平均値、Y 軸はアウトカムがある患者の中での有病率/平均値です。従って、変数の点が対角線の上にある場合、その変数はアウトカムがある患者の方が一般的であり、点が対角線の下にある場合、その変数はアウトカムがない患者の方が一般的です。

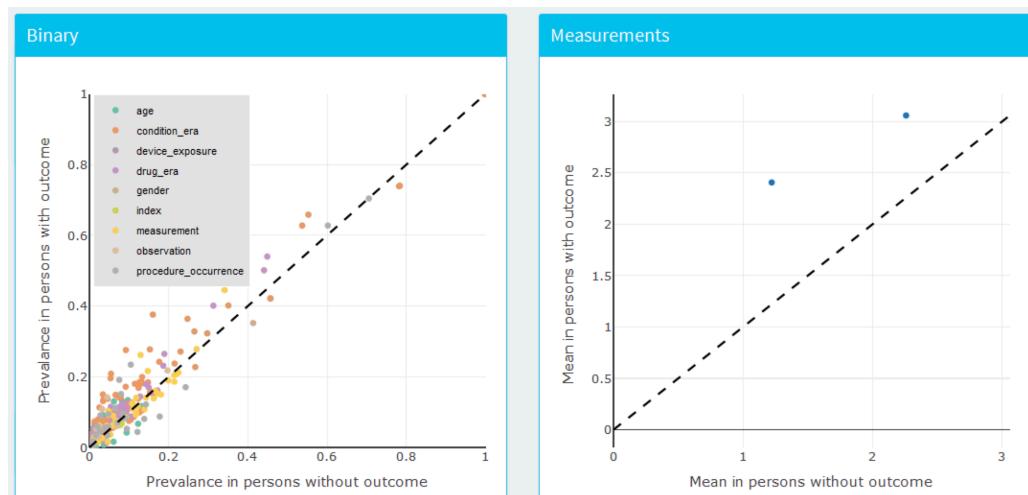


Figure 13.27: モデル要約プロット。各点はモデルに含まれる変数に対応します。

図 13.28 に示すテーブルでは、すべての候補の共変量の名前、値（一般線形モデルを使用する場合は係数、その他の場合は変数の重要性）、アウトカム平均（アウトカムがある患者の平均値）、非アウトカム平均（アウトカムがない患者の平均値）が表示されます。



予測モデルは因果モデルではなく、予測変数を原因と誤解しないようにしてください。図 13.28 のいずれかの変数を変更することでアウトカムのリスクが影響を受ける保証はありません。

13.9 患者レベルの予測に関する追加の機能

13.9.1 学術誌論文の生成

自動的にワードドキュメントを生成する機能を追加しました。このドキュメントは学術誌の草稿として利用でき、多くの生成された研究の詳細とアウトカム

Model Table				
	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: モデル詳細テーブル

を含みます。外部検証を行った場合、そのアウトカムも追加することができます。また、ターゲット集団の多くの共変量に関するデータを含む “Table 1” を任意で追加することもできます。この機能を実行することで学術誌の草稿を作成できます：

```
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>")
```

詳細は関数のヘルプページを参照してください。

13.10 まとめ



- 患者レベルの予測は、過去のデータを使用して将来の出来事を予測するモデルを開発することを目的としています。
- モデル開発に最適な機械学習アルゴリズムの選択は経験的な問題であり、手元の問題とデータによって決定されるべきです。
- PatientLevelPrediction パッケージは、OMOP CDM に格納されたデータを使用した予測モデルの開発と検証のためのベストプラクティスを実装しています。
- モデルとその性能指標の公表はインタラクティブなダッシュボードを通じて実装されています。
- OHDSI の予測フレームワークは、臨床応用の前提条件である予測モデルの大規模な外部検証を可能にします。

13.11 演習

前提条件

これらの演習では、セクション 8.4.5 で説明されているように、R、R-Studio、および Java がインストールされていることを前提としています。また、SqlRender、

DatabaseConnector、EunomiaおよびPatientLevelPredictionパッケージも必要です。これらは以下のコマンドでインストールできます：

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

Eunomia パッケージは、ローカルの R セッション内で実行される CDM 内のシミュレーションデータセットを提供します。接続の詳細は、以下のようにして取得できます。:

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

CDM データベースのスキーマは「main」です。これらの演習ではいくつかのコホートも使用します。Eunomia パッケージの `createCohorts` 関数は、これを COHORT テーブルに作成します：

```
Eunomia::createCohorts(connectionDetails)
```

問題定義

初めて NSAIDs（非ステロイド性抗炎症剤）を使用し始めた患者において、今後 1 年間に消化管（GI）出血を発症する患者を予測します。

NSAID の新規ユーザーコホートの COHORT_DEFINITION_ID は 4 です。GI 出血コホートの COHORT_DEFINITION_ID は 3 です。

演習 13.1. PatientLevelPrediction R パッケージの PatientLevelPrediction を使用して、予測に使用する共変量を定義し、CDM から PLP データを抽出します。PLP データの要約を作成します。

演習 13.2. 最終的な標本母集団を定義するために行う必要があるデザインの選択を再検討し、`createStudyPopulation` 関数を使用してこれらを指定します。選択した内容が標本母集団の最終的なサイズにどのような影響を与えるでしょうか？

演習 13.3. LASSO を使用して予測モデルを構築し、Shiny アプリケーションを使用してその性能を評価します。モデルの性能はどの程度ですか？

推奨される回答は、付録 E.9を参照ください。

第 IV 部

エビデンスの質

第 14 章

エビデンスの質

著者: Patrick Ryan & Jon Duke

14.1 信頼できるエビデンスの属性

どんな旅も出発する前に、理想の目的地がどのようなものかを思い描いておくことが役に立つでしょう。データからエビデンスへの旅を支援するために、信頼できるエビデンスの質を裏付けることができる望ましい属性を本章では強調します。

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: 信頼できる証拠の望ましい属性

信頼性の高いエビデンスは再現性があるるべきであり、特定の質問に対しても同じデータに同じ分析を適用した場合に、研究者は同一の結果が得られると期待すべきです。この最低限の要件には、エビデンスが定義されたプロセスの実行結果であり、その過程で事後的な意思決定による手動の介入があつてはならないという考え方が暗黙のうちに含まれています。さらに理想的には、信頼できるエビデンスは再現可能であるべきであり、別の研究者が特定のデータベースで同じ分析を実行した場合でも、最初の研究者と同じ結果を期待すべき

です。再現可能性とは、プロセスが完全に明記されており、人間が読める形式とコンピュータが実行可能な形式の両方であり、研究者の裁量に委ねられる余地がないことを意味します。再現性と反復性を実現する最も効率的な解決策は、入力と出力を定義した標準化された分析ルーチンを使用し、これらの手順をバージョン管理されたデータベースに対して適用することです。

私たちは、再現可能であることが示されれば、そのエビデンスが信頼できるものであると自信を持つ可能性が高くなります。例えば、ある大手民間保険会社の保険請求データベースに対する分析から生成されたエビデンスは、別の保険会社の保険請求データで再現されれば、より強固なものとなるでしょう。集団レベルの効果推定という文脈では、この特性は、一貫性に関するオースティン・ブラッドフォード・ヒル卿の因果関係に関する見解と一致します。「異なる人物、異なる場所、状況、時間において、繰り返し観察されたか？…偶然が説明なのか、真の危険性が明らかになったのかは、状況と観察を繰り返すことによってのみ答えられる場合がある」(Hill, 1965)。患者レベルの予測では、再現性は外部検証の価値と、異なるデータベースに適用した際の識別精度とキャリブレーションを観察することで、1つのデータベースで訓練されたモデルのパフォーマンスを評価する能力とみなされます。同一の分析が異なるデータベースに対して実行され、依然として一貫して類似した結果を示す状況では、私たちのエビデンスが一般化可能であるという確信がさらに深まります。OHDSI 研究ネットワークの重要な価値は、異なる集団、地理、データ収集プロセスによって表される多様性です。Madigan et al. (2013b) は、効果推定値がデータの選択に敏感であることを示しました。各データソースには固有の限界や独特の偏りがあり、単一の調査結果に対する信頼性を制限する可能性があることを認識した上で、異なるタイプのデータセットにわたって類似したパターンを観察することには大きな意味があります。なぜなら、ソース固有の偏りだけで調査結果を説明できる可能性が大幅に低くなるからです。ネットワーク研究が、米国、ヨーロッパ、アジアの複数の保険請求データベースや電子的健康記録 (EHR) データベースにわたって、一貫した集団レベルの効果推定値を示した場合、医療介入に関するより強力なエビデンスとして認められるべきであり、医療上の意思決定に影響を与える可能性はより広範囲に及びます。

信頼性の高い証拠は頑健であるべきであり、つまり、分析の中でなされる主観的な選択に過度に左右されない結果でなければなりません。特定の研究に対して、代替となる合理的な統計的手法が考えられる場合、異なる手法が類似の結果をもたらすことが確認できれば安心材料となり、逆に、食い違う結果が明らかになれば注意が必要となります (Madigan et al., 2013a)。母集団レベルの効果推定では、感度分析には、コホート比較研究や自己対照ケースシリーズ研究のデザインを適用するかどうか、といった高度な研究デザインの選択が含まれる場合もあります。また、コホート比較研究の枠組みの中で、傾向スコアマッチング、層別化、または重み付けを交絡調整の方法として実行するかどうかといった、デザインに組み込まれた分析上の考慮事項に焦点を当てる場合もあります。

最後に、しかし最も重要なこととして、エビデンスは校正されるべきである。未知の質問に対する回答を生成するエビデンス生成システムを保有しているだけ

では不十分であり、そのシステムのパフォーマンスを検証できなければ意味がありません。クローズドシステムは、既知の動作特性を持つことが期待されるべきであり、その特性はシステムが生成する結果を解釈する上でのコンテクストとして測定および伝達できるものでなければなりません。統計的アーティファクトは、95%の信頼区間が95%の包含確率を持つ、または予測確率が10%のコホート集団において10%の事象の割合が観察されるなど、明確に定義された特性を持つことを実証できるべきです。観察研究には、常に、デザイン、方法、データに関する仮説を検証する研究診断を付随させる必要があります。これらの診断は、研究の妥当性に対する主な脅威である選択バイアス、交絡、測定エラーの評価に重点を置くべきです。ネガティブコントロールは、観察研究における系統的エラーを特定し、軽減するための強力なツールであることが示されています (Schuemie et al., 2016, 2018a,b)。

14.2 エビデンスの質の理解

しかし、研究結果が十分に信頼できるものであるかどうかを、どうすれば判断できるのでしょうか？臨床現場での使用に耐えうるのでしょうか？規制当局の意思決定に利用できるのでしょうか？将来の研究の基礎として役立つのでしょうか？ランダム化比較試験、観察研究、その他の分析手法による研究であるかに関わらず、新しい研究が発表または公表されるたびに、読者はこれらの疑問を考慮しなければなりません。

観察研究や「リアルワールドデータ」の利用に関してよく挙げられる懸念事項のひとつに、データの質に関するものがあります (Botsis et al., 2010; Hersh et al., 2013; Sherman et al., 2016)。一般的に指摘されるのは、観察研究で使用されるデータはもともと研究目的で収集されたものではないため、不完全または不正確なデータ取得や、内在するバイアスに苦しむ可能性があるということです。こうした懸念から、データの品質を測定、特徴づけ、そして理想を言えば改善する方法に関する研究が増加しています (Kahn et al., 2012; Liaw et al., 2013; Weiskopf and Weng, 2013)。OHDSI コミュニティは、こうした研究の強力な推進者であり、コミュニティのメンバーは、OMOP CDM と OHDSI ネットワークにおけるデータ品質を調査する多くの研究を主導し、または参加しています (Huser et al., 2016; Kahn et al., 2015; Callahan et al., 2017; Yoon et al., 2016)。

この分野における過去10年間の調査結果を踏まえると、データ品質は完璧ではなく、今後も完璧になることはないということが明らかになっています。この考え方には、医療情報学の分野におけるパイオニアであるクレム・マクドナルド博士の次の言葉にうまく反映されています。

データの正確性が損なわれるのは、医師の頭脳からカルテにデータ
が移動する時点から始まります

したがって、コミュニティとして私たちは次のような問いかけをしなければなりません。不完全なデータが与えられた場合、信頼性の高いエビデンスをどのようにして得られるのか？

その答えは、「エビデンスの質」を全体的に見ることにあります。すなわち、データからエビデンスに至るまでの全過程を検証し、エビデンス生成プロセスを構成する各要素を特定し、各要素の質に対する信頼性をどのように構築するかを決定し、各段階で学んだことを透明性をもって伝えることです。エビデンスの質は、観察データの質だけでなく、観察分析で使用される方法、ソフトウェア、臨床定義の妥当性も考慮します。

次の章では、表 14.1 にリストされているエビデンスの質の 4 つのコンポーネントを探ります。

Table 14.1: エビデンスの質の 4 つのコンポーネント

エビデンスの質のコンポーネント	測定するもの
データの質	データが合意された構造と規約に準拠した形で、完全にキャプチャされ信憑性のある値を持つかどうか？
臨床的妥当性	実施された分析が臨床的な意図とどの程度一致しているか？
ソフトウェアの妥当性	データの変換および分析プロセスが意図した通りに機能するかどうか？
方法の妥当性	データの強みと弱点を考慮した上で、その方法論が研究の問い合わせに適しているか？

14.3 エビデンスの質の伝達

エビデンスの質に関する重要な側面は、データからエビデンスに至る過程で生じる不確実性を表現する能力です。OHDSI がエビデンスの質に関して掲げる目標は、OHDSI が生成したエビデンスには多くの点で不完全な部分があることは確かですが、その弱点と強みを常に測定し、この情報を厳格かつオープンな方法で伝達してきたという確信を医療の意思決定者に抱いてもらうことです。

14.4 まとめ



- 我々が生成するエビデンスは、再現可能、再現実験が可能、複製可能、一般化可能、頑健性、そして較正済みでなければなりません。
- エビデンスが信頼できるかどうかを判断する際には、データの質だけでなく、エビデンスの質を考慮するべきです：
 - * データの品質
 - * 臨床的妥当性

- * ソフトウェアの妥当性
 - * 方法の妥当性
- エビデンスを伝える際には、エビデンスの質に対するさまざまな課題から生じる不確実性を表現しなければなりません。

第 15 章

データ品質

著者: Martijn Schuemie, Vojtech Huser & Clair Blacketer

医療観察研究に用いられるデータのほとんどは、研究目的で収集されたものではない。例えば、電子カルテ（EHR）は患者のケアをサポートするために必要な情報を捕捉することを目的としており、保険請求データは保険者への費用配分を割り当てるために収集されています。このようなデータを臨床研究に用いることが適切であるかどうかについては、多くの疑問が呈されています。van der Lei (1991) は「データは収集された目的のみに使用されるべきである」とさえ述べています。懸念されるのは、私たちがやりたい研究のために収集されたデータではないため、そのデータの質が十分である保証がないということです。データの品質が低い（入力がゴミ）場合、そのデータを使用した研究結果の質も低い（出力もゴミ）に違いないでしょう。したがって、観察医療研究の重要な側面はデータの質を評価することであり、次の質問に答えることを目指しています。

研究目的に対して、データの品質は十分でしょうか？

データ品質（DQ）を次のように定義できます (Roebuck, 2012):

特定の使用目的に適したデータとなるような、完全性、妥当性、一貫性、適時性、正確性。

データが完璧であることはまずありませんが、目的には十分である可能性があることに留意ください。

DQ は直接観察することができませんが、それを評価する方法論が開発されています。DQ 評価には 2 つの種類があります (Weiskopf and Weng, 2013): DQ を全般的に評価する評価と、特定の研究における DQ を評価する評価です。

本章では、まず DQ の問題の原因となり得るものを検討し、その後、一般的な DQ 評価と特定の研究における DQ 評価の理論について説明し、最後に、OHDSI ツールを使用してこれらの評価を実行する方法を段階的に説明します。

15.1 データ品質問題の原因

第14章で述べたように、医師が自身の考えを記録する段階からデータの品質に対する脅威は数多く存在します。Dasu and Johnson (2003) は、データのライフサイクルにおいて次のステップを区別し、各ステップに DQ を統合することを推奨しています。これを DQ 連続体と呼んでいます。

1. データ収集と統合。考えられる問題としては、手入力の誤り、バイアス（例：保険請求におけるコーディングの誤り）、EHR でのテーブルの誤った結合、欠測値のデフォルト値への置き換えなどがあります。
2. データの保存と知識の共有。考えられる問題としては、データモデルの文書化不足やメタデータの欠如などが挙げられます。
3. データ分析。不正確なデータ変換、データの誤った解釈、不適切な方法論の使用などの問題が含まれます。
4. データの公開。下流での使用のためにデータを公開する際の問題。

私たちが使用するデータはすでに収集され統合されていることが多いため、ステップ1を改善するためにできることはほとんどありません。この章の後半で詳しく述べるように、このステップによって生成される DQ をチェックする方法があります。

同様に、私たちは特定の形式でデータを頻繁に受け取っているため、ステップ2の一部についてはほとんど影響力を持ちません。しかし、OHDSIでは、すべての観測データを共通データモデル（CDM）に変換しており、このプロセスについては私たちが管理権限を持っています。この特定のステップが DQ を低下させる可能性があるという懸念が示されたこともあります。しかし、私たちがこのプロセスを管理しているため、後述のセクション 15.2.2 で説明するように、DQ を維持するための厳格な保護策を構築することができます。いくつかの調査 (Defalco et al., 2013; Makadia and Ryan, 2014; Matcho et al., 2014; Voss et al., 2015a,b; Hripcsak et al., 2018) によれば、正しく実行されれば、CDM への変換時にほとんどエラーは発生しないことが示されています。実際、大規模なコミュニティで共有される詳細に文書化されたデータモデルがあれば、曖昧さのない明確な方法でデータの保存が容易になります。

ステップ3（データ分析）もまた、私たちの管理下にあります。OHDSIでは、このステップにおける品質問題について DQ という用語は使用せず、臨床妥当性、ソフトウェア妥当性、方法妥当性という用語を使用しています。これらの用語については、それぞれ第16、17、18章で詳しく議論しています。

15.2 一般的なデータ品質

観察研究の一般的な目的に対してデータが適しているかどうかを問うことができます。Kahn et al. (2016) は、このような一般的なデータ品質（DQ）を次の3つの要素から構成されるものと定義しています。

1. 適合性: データ値が指定された基準や形式に従っているでしょうか。3つのサブタイプに識別されます:
 - 値: 記録されたデータ要素が指定された形式に合致しているでしょうか。例えば、すべての医療専門分野は有効な専門分野でしょうか。
 - 関係: 記録されたデータが指定された関係制約に合致しているでしょうか。例えば、DRUG_EXPOSURE データの PROVIDER_ID が PROVIDER テーブルの対応するレコードと一致しているでしょうか。
 - 計算: データに対する計算結果が意図したとおりになっているでしょうか。例えば、身長と体重から計算された BMI がデータに記録されているものと等しいでしょうか。
2. 完全性: 特定の変数が存在するかどうか（例：診察室で測定された体重が記録されているか？）や、変数がすべての記録された値を含んでいるか（例：すべての人の性別が分かっているか）を参照します。
3. 妥当性: データ値は信頼できるでしょうか。3つのサブタイプが定義されています:
 - 一意性: 例えば、PERSON テーブルで各 PERSON_ID は一度しか出現しないでしょうか？
 - 非一時的: 値、分布、密度が期待される値と一致しているでしょうか？例えば、データから示唆される糖尿病の有病率は既知の有病率と一致しているでしょうか？
 - 一時的: 値の変化は期待と一致しているでしょうか？例えば、予防接種の順序は推奨事項と一致しているでしょうか。

各コンポーネントは 2 つの方法で評価できます：

- 検証では、モデルとメタデータのデータ制約、システムの前提条件、ローカルの知識に焦点を当てます。外部参照には依存しません。検証の主な特徴は、ローカル環境内のリソースを使用して、期待される値と分布を決定する能力です。
- 妥当性（バリデーション）では、関連する外部ベンチマークとのデータ値の整合性に焦点を当てます。外部ベンチマークのソースの 1 つとして、複数のデータサイトにわたる結果を組み合わせることが考えられます。

15.2.1 データ品質チェック

カーンは、データが所定の要件に適合しているかどうかをテストするデータ品質チェック（データ品質ルールと呼ばれることがある）という用語を導入しています（例えば、患者の年齢が 141 歳というありえない値になっている場合、誤った生年が入力されたか、死亡イベントが記録されていない可能性があります）。このようなチェックは、自動化された DQ ツールを作成することでソフトウェアに実装することができます。そのようなツールの 1 つに、ACHILLES (Automated Characterization of Health Information at Large-scale Longitudinal Evidence Systems) があります (Huser et al., 2018)。ACHILLES

は、CDM に準拠したデータベースの特性評価と可視化を行うソフトウェアツールです。そのため、データベースのネットワークにおける DQ の評価に使用することができます (Huser et al., 2016)。ACHILLES はスタンダードアロンツールとして利用でき、「データソース」機能としても ATLAS に統合されています。

ACHILLES は、170 以上のデータ特性評価を事前に計算します。各分析には分析 ID と分析の簡単な説明があり、その例として「715: DRUG_CONCEPT_ID による DAYS_SUPPLY の分布」や「506: 性別による死亡時の年齢の分布」などがあります。これらの分析結果はデータベースに保存され、ウェブビューアーまたは ATLAS からアクセスできます。

コミュニティが作成したもう一つのツールで、DQ を評価するものに、Data Quality Dashboard (DQD) があります。ACHILLES が特性評価を実行して CDM インスタンスの全体像を視覚的に把握できるようにするのに対し、DQD は表ごとに、またフィールドごとに、CDM 内の所定の仕様を満たさないレコード数を数値化します。合計で 1,500 を超えるチェックが実行され、それぞれが Kahn のフレームワークで整理されています。各チェックの結果は閾値と比較され、違反行の割合がその値を上回る場合は不合格とみなされます。表 15.1 は、いくつかのチェックの例を示しています。

Table 15.1: データ品質ダッシュボードのデータ品質ルールの例

違反行の割合	チェックの説明	閾値	状態
0.34	VISIT_OCCURRENCE の provider_id が 仕様に基づく期待 されるデータ型で あるかどうかを示 す yes、no の値。	0.05	FAIL
0.99	MEASUREMENT テーブルの measure- ment_source_value フィールドにある 異なるソース値の 数やパーセントが 0 にマッピングさ れている。	0.30	FAIL
0.09	DRUG_ERA テー ブルの drug_concept_id フィールドにある 値が成分クラスに 適合しない記録の 数、パーセント。	0.10	PASS

違反行の割合	チェックの説明	閾値	状態
0.02	DRUG_EXPOSURE 0.05 テーブルの DRUG_EXPOSURE_END_DATE フィールドの値が DRUG_EXPOSURE_START_DATE フィールドの日付 より前に発生する 記録の数、パーセ ント。	0.05	PASS
0.00	PROCEDURE_OCCURRENCE テーブルの procedure_occurrence_id フィールドに重複 する値がある記録 の数、パーセン ト。		PASS

このツールでは、チェックは複数の方法で整理されており、その一つはテーブル、フィールド、コンセプトレベルのチェックです。テーブルレベルのチェックは、CDM 内の高次レベルで行われるもので、例えば、必要なテーブルがすべて揃っているかどうかの判断などです。フィールドレベルのチェックは、CDM の仕様への適合性を評価するために、各テーブル内のすべてのフィールドに対して実施されます。これには、すべての主キーが実際に一意であること、すべての標準コンセプトフィールドが適切なドメインにコンセプト ID を含んでいることなど、多くの項目が含まれます。コンセプトレベルのチェックは、個々のコンセプト ID をさらに詳しく検証して実施します。これらの多くは、Kahn フレームワークの妥当性カテゴリーに分類されます。例えば、性別特有のコンセプトが誤った性別の人物に帰属されていないこと（すなわち、女性患者の前立腺癌）を確認することなどです。



ACHILLES と DQD は CDM 内のデータに対して実行されます。このようにして特定された DQ の問題は、CDM への変換に起因する可能性もありますが、ソースデータにすでに存在していた DQ の問題を反映している可能性もあります。変換に問題がある場合、通常は私たちで問題を修正することができますが、根本的なデータに問題がある場合は、問題のあるレコードを削除するしか対処方法がない場合もあります。

15.2.2 ETL（抽出-変換-読込）単体テスト

高度なデータ品質のチェックに加え、個別レベルでのデータチェックも実施すべきです。データが CDM に変換される ETL プロセスは、非常に複雑であることがよくあり、その複雑さゆえに、気づかれないままミスが発生する危険性があります。さらに、時間の経過とともにソースデータモデルが変更されたり、CDM が更新されたりすることもあり、ETL プロセスを修正する必要が生じます。ETL のように複雑なプロセスに変更を加えると予期せぬ結果を招く可能性があり、ETL のすべての側面を再考し、再評価する必要が生じます。

ETL が期待通りに動作し、その状態を維持できるようにするために、一連のユニットテストを作成することが強く推奨されます。ユニットテストとは、自動的に单一の側面をチェックする小さなコードの断片です。第 6 章で説明した Rabbit-in-a-Hat ツールを使用すると、このようなユニットテストを簡単に作成できるユニットテストフレームワークを作成することができます。このフレームワークは、ソースデータベースとターゲット CDM バージョンの ETL 用に特別に作成された R 関数の集合です。これらの関数の一部は、ソースデータスキーマに準拠した偽のデータエントリを作成するためのもので、その他の関数は CDM 形式のデータに関する期待値を指定するために使用できます。以下にユニットテストの例を示します。

```
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)
```

この例では、Rabbit-in-a-Hat によって生成されたフレームワークがソースとして読み込まれ、コードの残りの部分で使用される関数が読み込まれます。次に、人物の性別マッピングのテストを開始することを宣言します。ソース・スキーマには ENROLLMENT テーブルがあり、Rabbit-in-a-Hat によって作成された add_enrollment 関数を使用して、MEMBER_ID および GENDER_OF_MEMBER フィールドに異なる値を持つ 2 つのエントリを作成します。最後に、ETL 実行後には、さまざまな期待値を持つ 2 つのエントリが PERSON テーブルに存在しているはずであるという期待値を指定します。

ENROLLMENT テーブルには他にも多くのフィールドがありますが、このテストのコンテキストでは、それらのフィールドの値についてはあまり気にする必要はありません。ただし、それらの値（生年月日など）を空のままにしておくと、ETL がレコードを破棄したりエラーを発生させたりする可能性があります。この問題を克服し、テストコードを読みやすく保つために、add_enrollment 関数は、ユーザーによって明示的に指定されていないフィールド値にデフォルト値（White Rabbit スキャンレポートで観測された最も一般的な値）を割り当てます。

同様のユニットテストを ETL の他のすべてのロジックに対して作成することもでき、通常は数百のテストが作成されます。テストの定義が完了したら、フレームワークを使用して 2 つの SQL ステートメントセットを生成します。1 つは偽のソースデータを作成し、もう 1 つは ETL 化されたデータに対するテストを作成します。

```
insertSql <- generateInsertSql(databaseSchema = "source_schema")
testSql <- generateTestSql(databaseSchema = "cdm_test_schema")
```

全体のプロセスは図 15.1 に示されています。

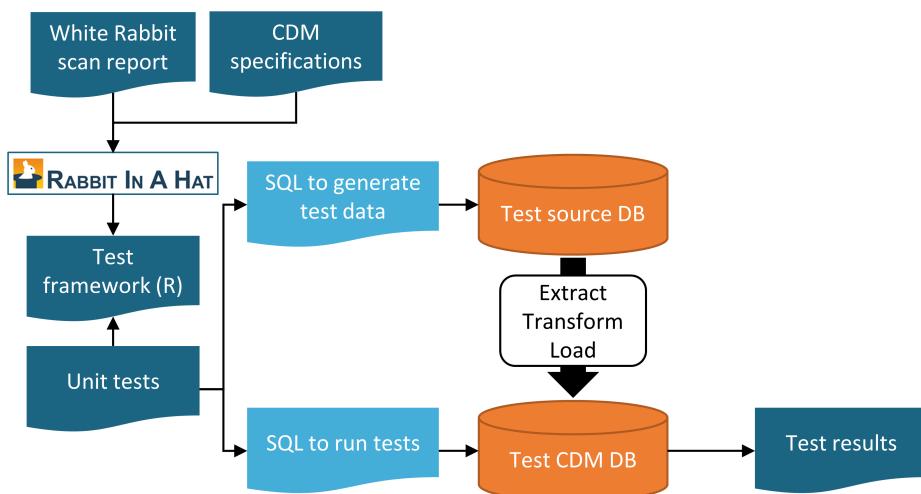


Figure 15.1: Rabbit-in-a-Hat テストフレームワークを使用した ETL (Extract-Transform-Load) プロセスの単体テスト

テスト用の SQL は、表 15.2 のようなテーブルを返します。このテーブルでは、先に定義した 2 つのテストに合格したことがわかります。

Table 15.2: ETL 単体テスト結果の例

ID	説明	状態
101	Person gender mappings	PASS
101	Person gender mappings	PASS

これらの単体テストの力は、ETL プロセスが変更されたときに簡単に再実行できることです。

15.3 研究特有のチェック

この章では、これまで一般的な DQ チェックに焦点を当ててきました。このようなチェックは、データを研究に用いる前に実行すべきです。これらのチェックは調査の質問とは関係なく実行されるため、研究固有の DQ 評価を行うことを推奨しています。

これらの評価の一部は、調査に特に関連する DQ ルールという形式を取ることができます。例えば、関心のある曝露に関するレコードの少なくとも 90% が曝露期間を指定しているというルールを課すことが考えられます。

標準的な評価は、ACHILLES で研究に最も関連するコンセプトを検討することであり、例えば、コホート研究の定義で指定されたものなどがあります。コードが観察される割合が時間とともに突然変化する場合は、DQ の問題が示唆される場合があります。いくつかの例は、この章で後ほど説明します。

別の評価方法としては、研究用に開発されたコホート定義を使用して生成されたコホートの有病率や経時的な有病率の変化を再検討し、それらが外部臨床知識に基づく予測と一致しているかどうかを確認することが挙げられます。例えば、新薬への曝露は市場に投入される前には存在しないはずであり、投入後は時間の経過とともに増加する可能性が高いです。同様に、アウトカムの有病率は、その集団における疾患の有病率として知られている内容と一致しているはずです。研究がデータベースのネットワーク全体で実施された場合、データベース間でコホートの有病率を比較することができます。あるデータベースではコホートの有病率が非常に高いが、別のデータベースでは欠落している場合、DQ の問題がある可能性があります。このような評価は、第 16 章で説明されているように、臨床的妥当性のコンセプトと重複していることに留意ください。一部のデータベースで予期せぬ有病率が見られるのは、DQ の問題ではなく、コホートの定義が、対象とする疾患状態を正確に捉えていないため、あるいは、異なる患者集団を捉えているデータベース間で、これらの疾患状態が当然異なるためである可能性があります。

15.3.1 マッピングのチェック

私たちの管理下で明確に該当するエラーの可能性として、ソースコードから標準コンセプトへのマッピングが挙げられます。ボキャブラリのマッピングは入念に作成されており、コミュニティのメンバーによって指摘されたマッピングのエラーは、ボキャブラリの課題追跡システム¹に報告され、今後のリリースで修正されます。しかし、すべてのマッピングを手作業で完全にチェックすることは不可能であり、エラーが残っている可能性は依然としてあります。そのため、調査を行う際には、その調査に最も関連性の高いコンセプトのマッピングを確認することをお勧めします。幸い、これは非常に簡単に実行できます。なぜなら、CDM では標準コンセプトだけでなくソースコードも保存されているからです。研究で使用されたコンセプトにマッピングされたソースコードだ

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

けでなく、そうでないソースコードも確認することができます。

対応づけられたソースコードをレビューする方法のひとつに、R パッケージ MethodEvaluation の `checkCohortSourceCodes` 関数を使用する方法があります。この関数は、ATLAS によって作成されたコホート定義を入力として使用し、コホート定義で使用されている各コンセプトセットについて、そのセット内のコンセプトに対応づけられたソースコードがどれであるかをチェックします。また、特定のソースコードに関連する時間的な問題を特定するのに役立つよう、これらのコードの経時的な有病率も計算します。図 15.2 の出力例は、「うつ病性障害」と呼ばれるコンセプトセットの一部内訳を示しています。対象のデータベースにおけるこのコンセプトセットで最も頻度の高いコンセプトは、コンセプト 440383（「うつ病性障害」）です。このデータベースでは、ICD-9 コード 3.11、ICD-10 コード F32.8、F32.89 の 3 つのソースコードがこのコンセプトにマッピングされていることが分かります。左側では、このコンセプト全体がまず徐々に増加し、その後急激に減少していることが分かります。個々のコードを調べると、この減少は、減少の時期に ICD-9 コードが使用されなくなったことによるものであることが分かります。これは ICD-10 コードが使用され始めた時期と一致していますが、ICD-10 コードの合計の有病率は ICD-9 コードの有病率よりもはるかに低くなっています。この特定の例は、ICD-10 コードの F32.9（「大うつ病性障害、単一エピソード、特定不能」）もまた、このコンセプトにマッピングされるべきであったという事実によるものです。この問題は、その後、ボキャブラリで解決されました。

% 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` 関数のサンプル出力

前述の例では、マッピングされていないソースコードが偶然発見されたことを示していますが、一般的に、存在するマッピングの確認よりも、欠落しているマッピングの特定の方が困難です。どのソースコードがマッピングされるべきだが、されていないのかを知る必要があります。この評価を半自動的に行う方法としては、MethodEvaluation R パッケージの `findOrphanSourceCodes` 関数

を使用する方法があります。この関数を使用すると、簡単なテキスト検索でボキャブラリの中からソースコードを検索し、それらのソースコードが特定のコンセプトまたはそのコンセプトの下位層のいずれかにマッピングされているかどうかを確認することができます。結果として得られたソースコードのセットは、次に、手元の CDM データベースに表示されているものだけに制限されます。例えば、研究では「壊疽性疾患」(439928) という概念と、その下位層すべてを使用して、壊疽のすべての出現箇所を特定しました。この検索結果が、実際に壊疽を示すすべてのソースコードを含んでいるかどうかを評価するために、いくつかの用語（例えば「壊疽」）を使用して、ソースコードを特定するためにコンセプトと SOURCE_TO_CONCEPT_MAP テーブルの説明を検索しました。次に、自動検索を使用して、データに表示される各壊疽ソースコードが、実際に直接または間接的（上位層経由）にコンセプト「壊疽性疾患」にマッピングされているかどうかを評価しました。この評価の結果は図 15.3 に示されています。ICD-10 の J85.0（「肺壊疽および壊死」）は、4324261（「肺壊死」）というコンセプトにのみマッピングされており、これは「壊疽性障害」の下位層ではありません。

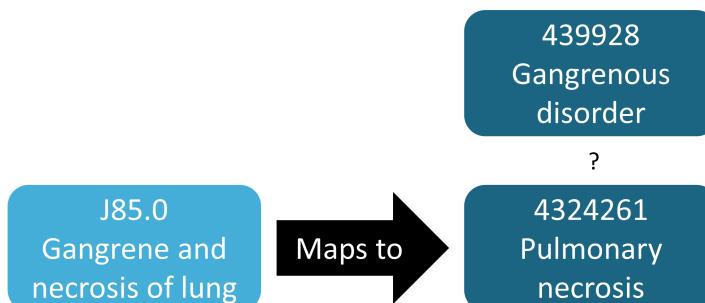


Figure 15.3: 孤立したソースコードのサンプル

15.4 実践における ACHILLES

ここでは、CDM 形式のデータベースに対して ACHILLES を実行する方法を説明します。

まず、R にサーバーへの接続方法を指示する必要があります。ACHILLES は、`DatabaseConnector` パッケージを使用しており、このパッケージは `createConnectionDetails` と呼ばれる関数を提供しています。`createConnectionDetails` と入力すると、さまざまなデータベース管理システム (DBMS) に必要な特定の設定を行うことができます。例えば、以下のコードを使用して PostgreSQL データベースに接続することができます。

```
user = "joe",
password = "supersecret")

cdmDbSchema <- "my_cdm_data"
cdmVersion <- "5.3.0"
```

最後の 2 行では、`cdmDbSchema` 変数と CDM のバージョンを定義しています。これらは、CDM 形式のデータがどこに存在し、どのバージョンの CDM が使用されているかを R に伝えるために使用します。Microsoft SQL Server では、データベーススキーマではデータベースとスキーマの両方を指定する必要があることに留意ください。例えば、`cdmDbSchema <- 「my_cdm_data.dbo」` となります。

次に、ACHILLES を実行します：

```
result <- achilles(connectionDetails,
                      cdmDatabaseSchema = cdmDbSchema,
                      resultsDatabaseSchema = cdmDbSchema,
                      sourceName = "My database",
                      cdmVersion = cdmVersion)
```

この関数は、`resultsDatabaseSchema` 内に複数のテーブルを作成します。ここでは、CDM データと同じデータベーススキーマに設定しています。ACHILLES データベース特性評価を表示することができます。これは、ATLAS を ACHILLES 結果データベースにポイントするか、ACHILLES 結果を JSON ファイルのセットにエクスポートすることで実行できます。

```
exportToJson(connectionDetails,
              cdmDatabaseSchema = cdmDatabaseSchema,
              resultsDatabaseSchema = cdmDatabaseSchema,
              outputPath = "achillesOut")
```

JSON ファイルは `achillesOut` サブフォルダに書き込まれ、AchillesWeb ウェブアプリケーションと併用して結果を調査することができます。例えば、図 15.4 は ACHILLES データ密度プロットを示しています。このプロットは、データの大部分が 2005 年から始まっていることを示しています。しかし、1961 年頃のレコードもいくつか存在しているように見えますが、これはおそらくデータのエラーです。

別の例を図 15.5 に示します。これは、糖尿病の診断コードの有病率に急激な変化が生じていることを示しています。この変化は、この特定の国における償還規則の変更と時期が一致しており、より多くの診断につながっていますが、おそらく基礎となる集団における真の有病率の増加ではないでしょう。

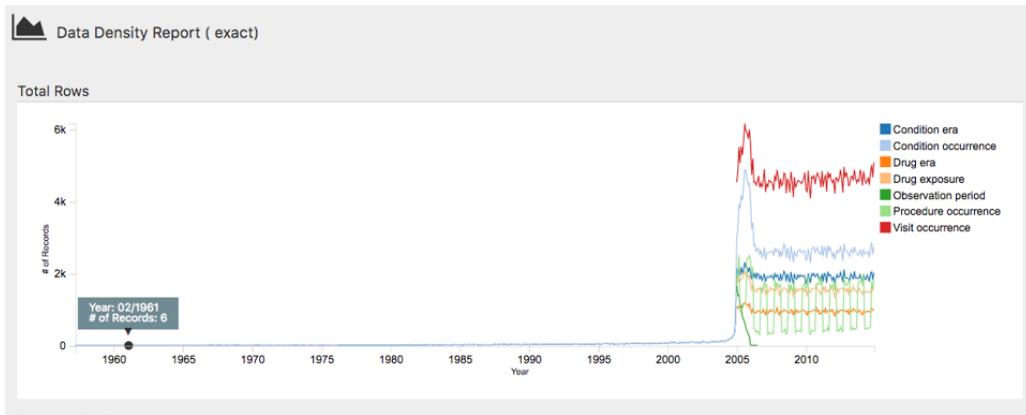


Figure 15.4: ACHILLES ウエブビューウーでのデータ密度プロット

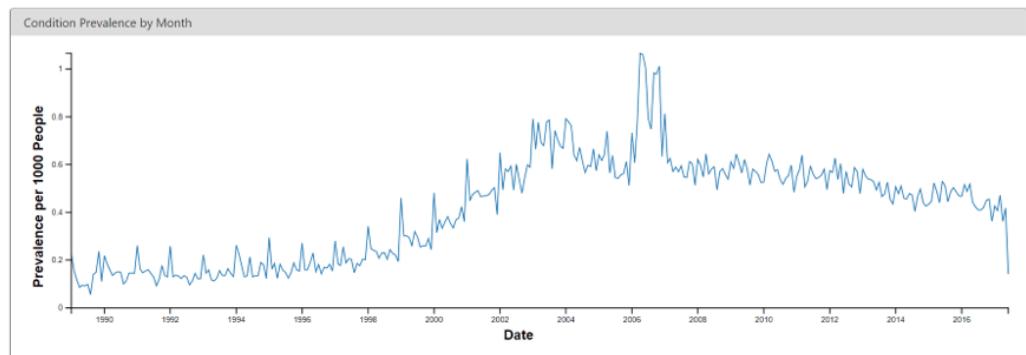


Figure 15.5: ACHILLES ウエブビューウーでの月次糖尿病の診断コードの有病率

15.5 Data Quality Dashboard の実践

ここでは、CDM 形式のデータベースに対してデータ品質ダッシュボードを実行する方法を説明します。これを行うには、セクション 15.4 で説明されている CDM 接続に対して、一連のチェックを実行します。現時点では、DQD は CDM v5.3.1 のみをサポートしているため、接続する前にデータベースが正しいバージョンであることを確認してください。ACHILLES の場合と同様に、R にデータの検索先を指示するために、変数

```
cdmDbSchema
```

を作成する必要があります。

```
cdmDbSchema <- "my_cdm_data.dbo"
```

次に、Dashboard を実行します…

```
DataQualityDashboard::executeDqChecks(connectionDetails = connectionDetails,  
                                         cdmDatabaseSchema = cdmDbSchema,  
                                         resultsDatabaseSchema = cdmDbSchema,  
                                         cdmSourceName = "My database",  
                                         outputFolder = "My output")
```

上記の関数は、指定されたスキーマ上で利用可能なすべてのデータ品質チェックを実行します。その後、resultsDatabaseSchema にテーブルを書き込みます。ここでは、CDM と同じスキーマに設定しています。このテーブルには、CDM テーブル、CDM フィールド、チェック名、チェックの説明、Kahn のカテゴリーおよびサブカテゴリー、違反行の数、閾値レベル、チェックの合否など、各チェック実行に関するすべての情報が含まれます。この関数は、テーブルに加えて、outputFolder として指定された場所に JSON ファイルも書き込みます。この JSON ファイルを使用して、結果を検査するためのウェビビューアーを起動することができます。

```
viewDqDashboard(jsonPath)
```

変数 jsonPath は、上記の executeDqChecks 関数を呼び出す際に指定した outputFolder にある、ダッシュボードの結果を含む JSON ファイルへのパスである必要があります。

最初にダッシュボードを開くと、図 15.6 に示すような概要テーブルが表示されます。このテーブルには、コンテキスト別に分類された各 Kahn カテゴリで実行されたチェックの合計数、それぞれの合格数と合格率、および全体の合格率が表示されます。

左側のメニューで Results (結果) をクリックすると、実行された各チェックのドリルダウン結果が表示されます（図 15.7）。この例では、個々の CDM テー

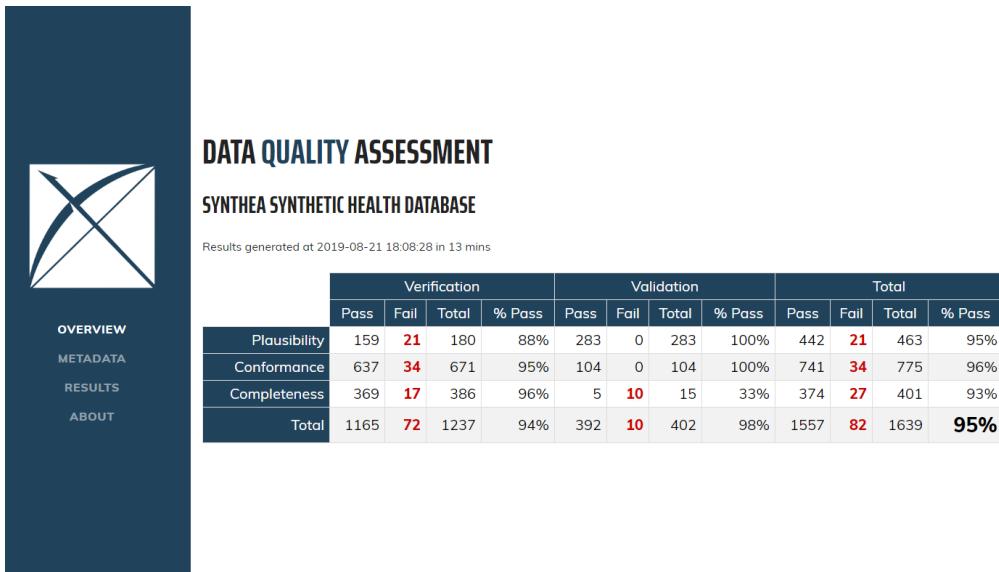


Figure 15.6: Data Quality Dashboard におけるデータ品質チェックの概要

ブルの完全性、すなわち、指定されたテーブルに少なくとも 1 つのレコードを持つ CDM 内の人数とパーセンテージを決定するために実行されたチェックを示す表が表示されます。この場合、ダッシュボードは 5 つのテーブルがすべて空であるとカウントし、失敗と見なします。アイコンをクリックすると、リストされた結果を生成するためにデータに対して実行された正確なクエリを表示するウィンドウが開きます。これにより、ダッシュボードでエラーとみなされた行を簡単に特定できます。

15.6 特定の研究チェックの実践

次に、付録 B.4 で提供されている血管性浮腫コホート定義に特化したいくつかのチェックを実行します。接続の詳細はセクション 15.4 で説明されているように設定済みであり、コホート定義の JSON と SQL はそれぞれ「cohort.json」と「cohort.sql」というファイルに保存済みであると仮定します。JSON と SQL は、ATLAS コホート定義機能のエクスポートタブから取得できます。

```
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

Audit Log							Column visibility	CSV
Show 5 entries							Search:	
	Status	Context	Category	Subcategory	Level	Description	% Records	
[+]	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: Data Quality Dashboard におけるデータ品質チェックの詳細

出力ファイルをウェブブラウザで開くことができます（図 15.8）。ここでは、血管性浮腫のコホート定義には”Inpatient or ER (入院または救急室ビジット) ”と”Angioedema (血管性浮腫) ”という二つのコンセプトセットがあります。この例のデータベースでは、ビジットは”ER” および”IP” というデータベース固有のソースコードによって同定され、ETL 中に標準コンセプトにマッピングされました。また、血管性浮腫は一つの ICD-9 コードと二つの ICD-10 コードによって同定されました。それぞれのコードのスパークライインを見ると、二つのコーディングシステムのどの時点で切り替わったかが明らかになりますが、コンセプトセット全体としてはその時点での不連続性はありません。

次に、標準コンセプトコードにマッピングされていない孤立したソースコードを検索できます。ここでは、標準コンセプト「Angioedema」を検索し、名前の一部として”Angioedema” または提供する同義語を含むコードやコンセプトを探します。

% 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: 血管性浮腫のコホート定義で使用されるソースコード

コード	説明	語彙 ID	全体のカウント
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

データで実際に使用されている潜在的なオーファンコード（上位層も下位層もない）として見つかったのは「血管神経性浮腫、続発」のみであり、これは血管性浮腫にマッピングすべきではありません。したがって、この分析では、欠落しているコードは発見されませんでした。

15.7 まとめ



- ほとんどの観察医療データは研究のために収集されたものではありません。
- データの品質チェックは研究に不可欠な要素です。データが研究目的に十分な品質かどうかを判断するには、データの品質を評価する必要があります。
- 一般に研究目的でデータの品質を評価すべきであり、特定の研究においては特に慎重に評価すべきです。
- データ品質の一部は、Data Quality Dashboard のような大規模な事前に定義されたルールに基づいて自動的に評価することができます。
- 特定の研究に関連するコードのマッピングを評価するための他のツールも存在します。

15.8 演習

前提条件

これらの演習では、セクション 8.4.5で説明されているように、R、R-Studio、および Java がインストール済みであると想定します。また、SqlRender、DatabaseConnector、ACHILLES、およびEunomiaパッケージも必要です。これらは、以下のコマンドでインストールできます：

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

Eunomia パッケージは、ローカルの R セッション内で実行される CDM のシミュレーションデータセットを提供します。接続情報は以下のコマンドで取得できます：

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

CDM データベーススキーマは「main」です。

演習 15.1. Eunomia データベースに対して ACHILLES を実行してください。

演習 15.2. Eunomia データベースに対して Data Quality Dashboard を実行してください。

演習 15.3. DQD のチェックリストを抽出してください。

回答例は付録 E.10にあります。

第 16 章

臨床的妥当性

著者: Joel Swerdel, Seng Chan You, Ray Chen & Patrick Ryan

物質をエネルギーに変える可能性は、鳥がほとんどいない国で暗闇
の中で鳥を撃つようなものだ。アインシュタイン、1935 年

OHDSI のビジョンは、「観察研究によって健康と疾病に関する包括的な理解が得られる世界」です。レトロスペクティブデザインは、既存のデータを使用する研究の手段を提供しますが、第 14 章で説明したように、妥当性のさまざまな側面に対する脅威に満ちている可能性があります。臨床的妥当性をデータ品質や統計的手法から切り離すことは容易ではありませんが、ここでは臨床的妥当性の観点から 3 つの側面に焦点を当てます。すなわち、医療データベースの特徴、コホートの検証、エビデンスの一般化可能性です。集団レベルの推定の例に戻りましょう（第 12 章）。「ACE 阻害薬は、サイアザイド系またはサイアザイド系利尿薬と比較して血管浮腫を引き起こすか？」という質問に答えようと試みました。その例では、ACE 阻害薬はサイアザイド系またはサイアザイド系利尿薬よりも血管浮腫を引き起こすことを示唆しました。本章では、「実施された分析はどの程度臨床的意図に一致しているか？」という質問に答えることに専念します。

16.1 医療データベースの特性

私たちが発見したのは、ACE 阻害薬の処方と血管浮腫の関係であり、ACE 阻害薬の使用と血管浮腫の関係ではない可能性があることです。データの質については、すでに前章（第 15 章）で議論しました。Common Data Model (CDM) に変換されたデータベースの質は、元のデータベースを超えることはできません。ここでは、ほとんどの医療用データベースの特性について取り上げます。OHDSI で使用される多くのデータベースは、保険請求または電子的健康記録 (EHR) から派生しています。保険請求と EHR ではデータ取得プロセスが異なり、いずれも研究を主な目的としていません。保険請求レコードのデータ要素

は、医療提供者から患者に提供されたサービスが、責任当事者による支払い合意を可能にするのに十分な正当性があることを示す、臨床医と保険者間の償還、財務取引を目的として取得されます。EHR のデータ要素は、臨床ケアと管理業務をサポートするために収集され、通常は、特定の医療システム内の医療提供者が、現在のサービスを文書化し、その医療システム内での今後のフォローアップケアに必要な背景情報を提供するために必要だと考える情報のみが反映されます。それらのデータは患者の完全な病歴を表しているとは限らず、また、複数の医療システムにまたがるデータを統合しているとは限りません。

観察データから信頼性の高いエビデンスを生成するには、患者が治療を求めた瞬間から、その治療を反映するデータが分析に使用される瞬間までのデータの経過を研究者が理解することが有用です。例えば、「薬物曝露」は、臨床医による処方箋、薬局の調剤記録、病院での処置管理、または患者による服薬歴の自己申告など、さまざまな観察データから推測することができます。データのソースは、どの患者が薬を使用したか、あるいは使用しなかったか、またいつ、どのくらいの期間使用したかについて、私たちが導き出す推論の信頼性に影響を与える可能性があります。データの収集プロセスでは、無料サンプルや市販薬が記録されない場合など、曝露が過小評価される可能性もあります。また、処方箋が患者によって服用されない場合や、処方された薬が患者によって忠実に消費されない場合など、曝露が過大評価される可能性もあります。曝露と結果の確認における潜在的な偏りを理解し、さらに理想的には、これらの測定エラーを定量化して調整することで、入手可能なデータから導き出されるエビデンスの妥当性に対する信頼性を向上させることができます。

16.2 コホートバリデーション

Hripcak and Albers (2017) は、「表現型とは、生物の遺伝的構成から導かれる遺伝子型とは区別される、生物の観察可能な、潜在的に変化する状態の仕様である」と説明しています。表現型という用語は、EHR データから推測される患者特性にも適用できます。研究者たちは、構造化データと非構造化データの両方から、インフォマティクスの初期から EHR のフェノタイピングを実施してきました。その目的は、EHR の生データ、保険請求データ、またはその他の臨床的に関連するデータに基づいて、対象となるコンセプトについての結論を導き出すことです。フェノタイプアルゴリズム、すなわちフェノタイプを識別または特徴づけるアルゴリズムは、知識工学の最近の研究を含め、ドメインエキスパートや知識エンジニアによって生成されることがあります。また、多様な形態の機械学習を通じて生成されることもあります。

この説明は、臨床的妥当性を検討する際に強化すべきいくつかの属性を強調しています。1) 観察可能なもの（したがって、観察データに捕捉可能なもの）について話していることが明確であること。2) 表現型仕様には時間という概念が含まれていること（人の状態は変化しうるため）。3) 表現型アルゴリズムは、意図の実現であるのに対し、表現型は意図そのものであるという区別があること。

OHDSI では、一定期間にわたって 1 つ以上の包含基準を満たす人々の集合を

定義するために、「コホート」という用語を採用しています。「コホート定義」は、観察データベースに対してコホートを具体化するために必要な論理を表します。この点において、コホート定義（または表現型アルゴリズム）は、表現型を表すことを目的としたコホートを生成するために使用されます。

臨床的特性、集団レベルの影響の推定、患者レベルの予測など、ほとんどの観察分析では、研究プロセスの一部として1つまたは複数のコホートを確立する必要があります。これらの分析によって得られたエビデンスの妥当性を評価するには、各コホートについて、次のような質問を考慮する必要があります。コホート定義と利用可能な観察データに基づいてコホート内で特定された対象者は、真に表現型に属する対象者をどの程度正確に反映しているでしょうか？

集団レベルの推定の例（第12章）「ACE阻害薬は、サイアザイド系またはサイアザイド系利尿薬と比較して血管浮腫を引き起こすでしょうか？」に戻ると、3つのコホートを定義する必要があります。ACE阻害薬の新規使用者、サイアザイド系利尿薬の新規使用者、血管浮腫を発症した人の3つです。ACE阻害薬またはサイアザイド系利尿薬の使用がすべて完全に把握されているため、過去の（観察されていない）使用を懸念することなく、最初の観察された曝露によって「新規ユーザー」を特定できると、どの程度確信できるでしょうか？ACE阻害薬の薬物曝露記録がある人は実際にその薬物に曝露されており、薬物曝露記録のない人は実際に曝露されていないと、自信を持って推論できるでしょうか？ACE阻害薬使用の状態に分類される期間を定義する際に、不確実性はないでしょうか。薬剤の開始時にコホートへの組入れを推測する場合、または薬剤の中止時のコホートからの離脱を推測する場合のいずれにおいても不確実性はないでしょうか。「血管性浮腫」の症状発現記録を持つ患者は、実際に他の皮膚アレルギー反応と区別される皮膚下の急速な腫れを経験しているでしょうか。血管性浮腫を発症した患者のうち、コホート定義に基づいてこれらの臨床例を特定するために使用された、観察データを生み出すような医療措置を受けた患者の割合はどの程度でしょうか？薬剤誘発の可能性がある血管性浮腫事象を、食物アレルギーやウイルス感染など、他の原因によって生じることが知られている事象からどの程度明確に区別できるでしょうか？曝露状況と結果の発生率との間に時間的な関連性を導くことに自信が持てるほど、疾患の発症が十分に把握されているでしょうか？こうした疑問に答えることが、臨床的妥当性の核心です。

本章では、コホート定義の妥当性を検証する方法について説明します。まず、コホート定義の妥当性を測定するために使用される評価基準について説明します。次に、これらの評価基準を推定する2つの方法について説明します。1) 原資料の検証による臨床判定、2) 診断予測モデリングを使用する半自動化された方法であるPheEvaluator。

16.2.1 コホート評価指標

対象コホートの定義が確定すると、その妥当性を評価することができます。妥当性を評価する一般的なアプローチは、定義されたコホートの一部またはすべてを基準となる「ゴールドスタンダード」と比較し、その結果を混同行列（2×2

分割表)で表現することです。混同行列は、ゴールドスタンダードの分類とコホート定義内の適格性に基づいて対象を層別化します。図16.1は、混同行列の要素を示しています。

		Gold Standard	
		True	False
Cohort Definition	True	True Positive	False Positive
	False	False Negative	True Negative

Figure 16.1: コンフュージョンマトリックス

コホート定義による真の結果と偽の結果は、その定義がある集団に適用することで決定されます。定義に含まれる人は、その健康状態を陽性と見なされ、「真」とラベル付けされます。コホート定義に含まれない人々は、健康状態が陰性と見なされ、「偽」とラベル付けされます。コホート定義で考慮される個人の健康状態の絶対的な真実は非常に判断が難しいですが、参照用ゴールドスタンダードを確立する方法は複数あり、そのうちの2つは本書の後半で説明します。使用する手法に関わらず、これらの人々に対するラベル付けは、コホート定義で説明したものと同じです。表現型指定の二値表示におけるエラーに加えて、健康状態のタイミングも不正確である可能性があります。例えば、コホート定義が表現型に属する人々を正しくラベル付けしている場合でも、その定義が、健康状態ではない人が健康状態にある人となった日時を不正確に指定している可能性があります。このエラーは、効果測定値として生存分析結果(例えば、ハザード比)を用いる研究にバイアスを加えることになります。次のステップは、ゴールドスタンダードとコホート定義の一一致度を評価することです。ゴールドスタンダード法とコホート定義の両方で「真」と判定された人を「真陽性」と呼びます。ゴールドスタンダード法で「偽」と判定され、コホート定義で「真」と判定された人を「偽陽性」と呼びます。つまり、コホート定義では、これらの人は疾患有していないにもかかわらず、疾患有していると誤分類されています。ゴールドスタンダード法とコホート定義の両方で「偽」と判定された人は「真の陰性」と呼ばれます。ゴールドスタンダード法で「真」と判定され、コホート定義で「偽」と判定された人は「偽の陰性」と呼ばれ、すなわち、コホート定義がこれらの人を、実際にはその人が表現型に属しているにもかかわらず、その条件に該当しないと誤って分類したことになります。混同行列の4つのセルのカウント値を用いれば、ある集団における表現型の状態を分類する際のコホート定義の精度を定量化することができます。コホート定義の性能を測定するための標準的なパフォーマンス指標があります。

1. コホート定義の感度 - 集団内の表現型に真に属する人のうち、コホート定義に基づいて健康アウトカムを持つと正しく特定された人の割合はどの程度か?これは以下の式で求められます。

$$\text{感度} = \text{真陽性} / (\text{真陽性} + \text{偽陰性})$$

2. コホート定義の特異度 - 集団内の表現型に属さない人のうち、コホート定義に基づいて健康アウトカムを持たないと正しく特定された人の割合は

どの程度か？これは以下の式で求められます。

$$\text{特異度} = \text{真の陰性数} / (\text{真の陰性数} + \text{偽陽性数})$$

3. コホート定義の陽性的中率 (PPV) - コホート定義によって健康状態にあると特定された人のうち、実際にその健康状態にある人の割合はどの程度か。これは以下の式で求められます。

$$\text{PPV} = \text{真の陽性数} / (\text{真の陽性数} + \text{偽陽性数})$$

4. コホート定義の陰性的中率 (NPV) - コホート定義によって特定された健康状態ではないとされた人のうち、実際にその表現型に属さない人の割合はどの程度か？これは以下の式で求められます。

$$\text{NPV} = \text{真陰性} / (\text{真陰性} + \text{偽陰性})$$

これらの指標の満点は 100% です。観察データの性質上、満点は通常、標準からかけ離れた値となります。Rubbo et al. (2015) は、心筋梗塞のコホート定義を検証した研究をレビューしました。彼らが調査した 33 件の研究のうち、PPV の満点を得たコホート定義は 1 つのデータセットにおける 1 つのコホート定義のみでした。全体として、33 件の研究のうち 31 件が $\text{PPV} \geq 70\%$ と報告しています。しかし、33 件の研究のうち感度を報告しているのは 11 件のみ、特異度を報告しているのは 5 件のみでした。PPV は感度、特異度、有病率の関数です。有病率の値が異なるデータセットでは、感度と特異度を一定に保ったまま PPV の値が異なるものとなります。感度と特異度がなければ、不完全なコホート定義によるバイアスを補正することはできません。さらに、健康状態の誤分類は差異があるかもしれません。つまり、比較群と比較してある集団においてコホート定義が異なる結果となる場合、または両方の比較群においてコホート定義が同様の結果となる場合、差異がない場合です。以前のコホート定義の検証研究では、潜在的な差異のある誤分類をテストしていませんが、これは効果推定値に強い偏りをもたらす可能性があります。

コホート定義のパフォーマンス指標が確立された後は、これらの定義を使用する研究の結果を調整するためにそれらを活用することができます。理論的には、これらの推定値の測定誤差を用いて研究結果を調整することは十分に確立されています。しかし実際には、パフォーマンス特性入手することが困難であるために、このような調整はほとんど考慮されません。ゴールドスタンダードを決定するために使用される方法は、このセクションの後半の部分で説明されています。

16.3 ソースレコード検証

コホートの定義を検証するために一般に用いられる方法は、ソースレコードの検証による臨床判定です。これは、対象とする臨床状態または特徴を適切に分類するのに十分な知識を有する 1 人または複数の専門家の下で、個人の記録を徹底的に調査するものです。一般にカルテのレビューは以下の手順に従って行われます。

1. カルテレビューを含む調査を実施するため、必要に応じて現地の IRB (Institutional Review Board) および／または関係者から許諾を得る。
2. 評価対象のコホートの定義を用いてコホートを生成する。コホート全体を審査するのに十分なリソースがない場合は、対象者のサブセットをサンプリングし、手作業で審査する。
3. 対象者の記録を審査するのに十分な臨床的専門知識を有する 1 人または複数の人物を特定する。
4. 対象者が対象とする臨床状態または特性について陽性または陰性であるかを判定するためのガイドラインを決定する。
5. 臨床専門家がサンプル内の人々について、利用可能なすべてのデータを検証および判定し、各対象者が表現型に属するかどうかを分類する。
6. コホートの定義分類や臨床判定分類に従って対象者を混同行列に分類し、収集したデータから可能な性能特性を算出する。

チャートレビューの結果は、通常、1 つの性能特性である陽性的中率 (PPV) の評価に限定されます。これは、評価対象のコホート定義によって、望ましい状態または特性を持つと見なされる集団のみが生成されるためです。したがって、コホートのサンプル内の各人は、臨床判定に基づいて真陽性または偽陽性のいずれかに分類されます。集団全体の表現型内のすべての人（コホート定義で特定されない人を含む）に関する知識がなければ、偽陰性を特定することはできず、それにより残りの性能特性を生成するための混乱行列の残りの部分を埋めることはできません。集団全体の表現型のすべての人を特定する方法のある方法としては、データベース全体のチャートレビューが考えられますが、これは一般に、対象となる集団が小規模でない限りは実行不可能です。あるいは、腫瘍レジストリ（下記参照）など、真の症例がすべてフラグ付けされ、判定済みの包括的な臨床レジストリを利用する方法もあります。あるいは、コホート定義に該当しない人々をサンプリングし、予測される陰性のサブセットを作成し、その後、上記のチャートレビューのステップ 3 から 6 を繰り返して、これらの患者が本当に臨床的に関心のある状態や特徴を欠いているかどうかを確認することで、真の陰性または偽陰性を特定することができます。これにより、陰性的中率 (NPV) を推定することができ、表現型有病率の適切な推定値が利用可能であれば、感度と特異度を推定することができます。

ソース記録の検証による臨床判定には、多くの限界があります。前述の通り、PPV のような単一の指標の評価だけでも、カルテのレビューは非常に時間とリソースを要するプロセスになります。この限界は、完全な混乱マトリクスを記入するために集団全体を評価する実用性を著しく妨げます。さらに、上記のプロセスには複数のステップがあり、研究結果にバイアスが生じる可能性があります。例えば、EHR で記録が均等にアクセスできない場合、EHR が存在しない場合、または個々の患者の同意が必要な場合、評価対象のサブセットは真にランダムではなく、サンプリングや選択バイアスが入り込む可能性があります。さらに、手動による判定はヒューマンエラーや誤分類の影響を受けやすく、完璧に正確な評価基準とは言えない可能性があります。個人の記録のデータが曖

昧であったり、主観的であったり、質が低かったりするために、臨床審査員の間で意見が分かれることもよくあります。多くの研究では、このプロセスでは多数決による合意決定が採用されており、評価者間の意見の相違を反映しない二値分類が個人に対して行われています。

16.3.1 ソースレコード検証の例

コロンビア大学アーヴィング医療センター (CUIMC) による研究では、米国国立がん研究所 (NCI) の実現可能性調査の一環として、複数の癌に関するコホート定義の検証が行われました。この研究から、カルテレビューによるコホート定義の検証プロセス例が提供されています。この例では、前立腺癌の検証プロセスは以下の通りです。

1. OHDSI がんフェノタイピング研究のための提案を提出し、IRB の承認を取得しました。
2. 前立腺がんの集団定義を開発：ボキャブラリを調査するために ATHENA と ATLAS を使用し、前立腺悪性腫瘍（概念 ID 4163261）の発生状態の患者をすべて含み、前立腺二次新生物（概念 ID 4314337）または前立腺非ホジキンリンパ腫（概念 ID 4048666）を除く集団定義を作成しました。
3. ATLAS を使用して生成されたコホートから、手動レビュー用に 100 人の患者を無作為に抽出し、マッピングテーブルを使用して各 PERSON_ID を患者 MRN にマッピングしました。100 人の患者は、PPV のパフォーマンス指標について、望ましいレベルの統計的精度を達成するよう抽出されました。
4. 無作為に抽出されたサブセット内の人人が真陽性か偽陽性かを判断するために、入院患者と外来患者の両方のさまざまな EHR の記録を手動で確認しました。
5. 手動レビューと臨床判定は 1 人の医師によって実施されました（ただし、将来、理想的には合意と評価者間の信頼性を評価するために、より多くのレビュー担当者によってより厳密な検証研究が行われることになります）。
6. 参照基準の決定は、入手可能な電子的な患者記録のすべてに記録されている臨床記録、病理報告書、検査、投薬、処置に基づいて行われました。
7. 患者は、1) 前立腺がん、2) 前立腺がんではない、3) 判断不能、のいずれかに分類されました。
8. 前立腺がん / (前立腺がんではない + 判断不能) という計算式で、PPV の控えめな推定値が算出されました。
9. 次に、腫瘍登録を追加のゴールドスタンダードとして使用し、CUIMC 全体の集団における基準標準を特定しました。腫瘍レジストリにおいて、コホート定義により正確に特定された人数と特定されなかった人数を数え、これらの値を真陽性および偽陰性として感度を推定しました。

10. 推定された感度、陽性適中率、および有病率を用いて、このコホート定義の特異度を推定することができました。前述の通り、このプロセスは時間を要し、労力を要するものでした。各コホート定義を個別に評価し、また、すべての性能指標を特定するために、手作業によるチャートレビューと CUIMC 腫瘍レジストリとの照合を行う必要があったためです。IRB の承認プロセス自体は、腫瘍レジストリへのアクセスを得るための迅速審査にもかかわらず数週間を要し、手作業によるカルテレビューのプロセス自体にもさらに数週間を要しました。

Rubbo et al. (2015) らによる心筋梗塞 (MI) コホート定義の妥当性評価のレビューでは、研究で使用されたコホート定義、および妥当性評価の方法と報告された結果に著しい異質性があることが判明しました。著者らは、急性心筋梗塞については、利用可能なゴールドスタンダードのコホート定義はないと結論づけました。また、そのプロセスは費用と時間がかかることも指摘しています。この限界により、ほとんどの研究では検証のサンプルサイズが小さくなり、性能特性の推定値に大きなばらつきが生じることとなりました。また、33 件の研究のうち、すべての研究で陽性適中率が報告されていた一方で、感度が報告されていたのは 11 件の研究のみ、特異度が報告されていたのは 5 件の研究のみでした。前述の通り、感度と特異度の推定値がなければ、誤分類バイアスに対する統計的補正を行うことはできないのです。

16.4 PheEvaluator

OHDSI コミュニティは、診断予測モデルを用いてゴールドスタンダードを構築する別のアプローチを開発しました (Swerdel et al., 2019)。一般的な考え方とは、臨床医がソースレコードの検証で実施するのと同様の方法で健康アウトカムの確認をエミュレートすることですが、規模を拡大して適用できる自動化された方法です。このツールは、オープンソースの R パッケージである PheEvaluator として開発されています¹。PheEvaluator は、Patient Level Prediction パッケージの機能を使用しています。

プロセスは以下の通り：

1. 極めて特異的な（「xSpec」）コホートを作成する：診断予測モデルのトレーニング時にノイズの多い陽性ラベルとして使用される、対象となる結果を持つ可能性が極めて高い人物のセットを決定する。
2. 極めて感度の高い（「xSens」）コホートを作成する：結果が得られる可能性のある人をすべて含むべき集団を決定します。このコホートは、その逆数（結果が得られないと確信できる人の集合）を特定するために使用され、診断予測モデルのトレーニング時にノイズを含むネガティブラベルとして使用されます。
3. xSpec と xSens コホートを使用して予測モデルを適合：第 13 で説明したように、幅広い患者の特徴量を予測因子として使用してモデルを適合

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

し、その人物が xSpec コホート（結果が出ると考えられる人々）に属するのか、あるいは xSens コホート（結果が出ないと考える人々）の逆数に属するのかを予測することを目指します。

4. コホート定義の性能を評価するために使用される、除外された人々のセットに対して、結果の確率を推定するために適合されたモデルを適用します：モデルからの予測因子セットを個人のデータに適用して、その個人が表現型に属する確率を予測します。これらの予測を確率的なゴールドスタンダードとして使用します。
5. コホート定義の性能特性を評価します：予測確率をコホート定義の二値分類と比較します（混同行列のテスト条件）。テスト条件と真の条件の推定値を使用して、混同行列を完全に作成し、感度、特異度、予測値など、性能特性の全体的なセットを推定する。

このアプローチを使用する際の主な限界は、健康アウトカムのある人の確率の推定がデータベース内のデータに制限されることです。データベースによっては、臨床医のメモなどの重要な情報が利用できない場合があります。

診断予測モデリングでは、疾患を持つ人と持たない人を識別するモデルを作成します。患者レベルの予測（第 13 章）で説明されているように、予測モデルは対象コホートと結果コホートを使用して開発されます。対象コホートには、健康アウトカムを持つ人と持たない人が含まれます。結果コホートは、対象コホートの中で健康アウトカムを持つ人を特定します。PheEvaluator プロセスでは、予測モデルのアウトカムコホートを決定するために、非常に特異的なコホート定義である「xSpec」コホートを使用します。xSpec コホートは、定義を使用して、対象疾患の罹患確率が極めて高い人を見つけ出します。xSpec コホートは、対象の健康アウトカムについて複数の条件発生記録を持つ人々として定義されることがあります。例えば、心房細動の場合、心房細動の診断コードが 10 件以上ある人を対象とします。急性心筋梗塞のような急性疾患の場合は、心筋梗塞の発生を 5 件とし、入院による発生が少なくとも 2 件あることを条件に含めることができます。予測モデルの対象コホートは、対象とする健康アウトカムの発生可能性が低い人々と xSpec コホートの人々を合わせたものから構築されます。対象となる健康アウトカムの可能性が低い人を決定するために、データベース全体からサンプルを抽出し、通常は xSpec コホートを定義する際に使用されるコンセプトを含むレコードを持つ人物を除外することで、表現型に属する可能性を示唆する何らかの証拠を持つ人を除外します。この方法には限界があり、xSpec コホートの人物は、その疾患を持つ他の人々とは異なる特性を持っている可能性があります。また、これらの人物は、初期診断後の観察期間が平均的な患者よりも長かった可能性もあります。LASSO ロジスティック回帰を使用して、確率的なゴールドスタンダードを生成するための予測モデルを作成します (Suchard et al., 2013)。このアルゴリズムは簡潔なモデルを生成し、通常、データセット全体に存在する可能性がある共線性の共変量の多くを削除します。PheEvaluator ソフトウェアの現行バージョンでは、アウトカムの状態（はい/いいえ）は、その人に関するすべてのデータ（すべての観察期間）に基づいて評価され、コホート開始日の正確性は評価されません。

16.4.1 PheEvaluator による検証例

PheEvaluator を使用して、急性心筋梗塞を患ったことがある人を特定する必要がある研究で使用されるコホート定義の完全なパフォーマンス特性を評価することができます。

PheEvaluator を使用して心筋梗塞のコホート定義をテストする手順は以下の通りです。

ステップ 1: xSpec コホートの定義

MI の可能性が高いものを特定します。心筋梗塞またはその下位層のコンセプトを持つコンディション発生レコードで、5 日以内の入院ビジットから 1 回以上の MI 発生が記録され、365 日以内の患者レコードで 4 回以上の MI 発生が記録されているものが必要とされました。図 16.2 は、ATLAS における MI のこのコホート定義を示しています。

ステップ 2: xSens コホートの定義

次に、極めて感度の高いコホート (xSens) を開発します。このコホートは、MI については、病歴の任意の時点での心筋梗塞のコンセプトを含む少なくとも 1 つの疾患発生記録を持つ人々として定義することができます。図 16.3 は、ATLAS における MI の xSens コホート定義を示しています。

ステップ 3: 予測モデルの適合

関数 `createPhenoModel` は、評価コホートにおいて対象の健康アウトカムとなる確率を評価するための診断予測モデルを開発します。この関数を使用するには、ステップ 1 と 2 で開発した xSpec コホートと xSens コホートを利用します。xSpec コホートは、関数の `xSpecCohort` パラメータとして入力します。xSens コホートは、モデリングプロセスで使用されるターゲットコホートから除外すべきであることを示すために、`exclCohort` パラメータとして関数に入力します。この除外方法を使用すると、健康アウトカムの可能性が低い人物を特定することができます。このグループを「ノイズネガティブ」な人々、すなわち、健康アウトカムがネガティブである可能性が高いが、健康結果がポジティブである人も若干含まれる可能性があるグループと考えることができます。また、xSens コホートを関数の `prevCohort` パラメータとして使用することもできます。このパラメータは、母集団における健康結果のおおよその有病率を決定するプロセスで使用されます。通常、データベースから抽出した多数のランダムサンプルから、データベースにおける結果の有病率とほぼ同等の割合で、対象とするアウトカムを持つ人を含む人々の集団が生成されるはずです。ここで説明した方法を用いると、人々のランダムサンプルはもはや存在せず、結果を持つ人々と結果を持たない人々の割合をリセットして予測モデルを再キャリブレーションする必要があります。

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...

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...

with a Visit occurrence of: Inpatient Visit

where event starts between 0 days Before and 5 days After

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...

where event starts between 1 days After and 365 days After

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: 心筋梗塞の極めて特異的なコホート定義 (xSpec)

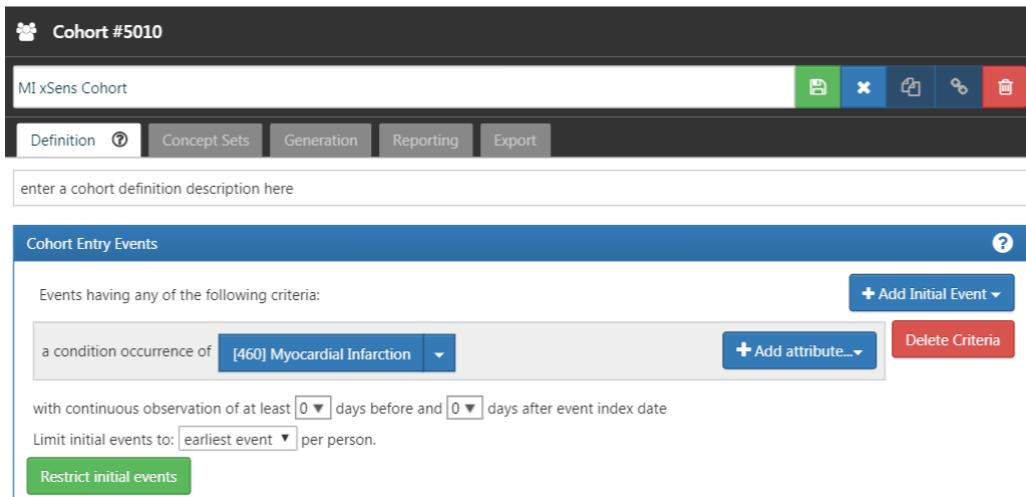


Figure 16.3: 心筋梗塞の極度に感度の高いコホート定義 (xSens)

xSpec コホートを定義するために使用されたすべてのコンセプトは、モデリングプロセスから除外する必要があります。これを実行するには、`excludedConcepts` パラメータを xSpec 定義で使用されたコンセプトのリストに設定します。例えば、MI の場合、心筋梗塞のコンセプトと、そのすべての下位層のコンセプトを使用して、ATLAS でコンセプトセットを作成します。この例では、`excludedConcepts` パラメータを 4329847 (心筋梗塞の概念 ID) に設定し、さらに `addDescendantsToExclude` パラメータも TRUE に設定して、除外されたコンセプトの下位層も除外されるようにします。

モデリングプロセスに含まれる人物の特徴を指定するために使用できるパラメータがいくつかあります。モデリングプロセスに含まれる人物の年齢を指定するには、`lowerAgeLimit` をモデルで希望する年齢の下限に、`upperAgeLimit` を上限に設定します。計画中の研究のコホート定義を特定の年齢グループに対して作成する場合は、この設定を行うとよいでしょう。例えば、研究で使用するコホート定義が小児の 1 型糖尿病である場合、診断予測モデルの開発に使用する年齢を、例えば 5 歳から 17 歳に限定したいと考えるかもしれません。そうすることで、テスト対象のコホート定義によって選択された人々により密接に関連する可能性が高い特徴量を持つモデルが作成されます。また、性別パラメータを男性または女性のコンセプト ID に設定することで、モデルに含める性別を指定することもできます。デフォルトでは、パラメータは男性と女性の両方を含めるように設定されています。この機能は、前立腺がんなどの性別特有の健康アウトカムに役立つ場合があります。`startDate` および `endDate` パラメータをそれぞれ日付範囲の下限および上限に設定することで、その人物のレコードにおける最初のビジットに基づいて、人物の包含期間を設定することができます。最後に、`mainPopnCohort` パラメータを使用して、対象および結果コホート内のすべての人物が選択される大規模な母集団コホートを指定することができます。ほとんどの場合、このパラメータは 0 に設定され、対象およびアウトカムコホート内の人物の選択に制限がないことを示します。ただし、この

パラメータがより優れたモデルの構築に役立つ場合もあります。例えば、健康アウトカムの発生率が極めて低い場合、おそらく 0.01% 以下の場合は。次の例を参照してください:

```
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", # 書き込み権限が必要
  trainOutFile = "5XMI_train",
  exclCohort = 1770120, #xSens コホート
  prevCohort = 1770119, # 有病率決定のコホート
  modelAnalysisId = "20181206V1",
  excludedConcepts = c(312327, 314666),
  addDescendantsToExclude = TRUE,
  cdmShortName = "myCDM",
  mainPopnCohort = 0, # 全人口を使用
  lowerAgeLimit = 18,
  upperAgeLimit = 90,
  gender = c(8507, 8532),
  startDate = "20100101",
  endDate = "20171231")
```

この例では、「my_results」データベースで定義されたコホートを使用し、コホートテーブルの場所 (cohortDatabaseSchema、cohortDatabaseTable - 「my_results.cohort」) と、モデルに条件、薬物曝露などを知らせる場所 (cdmDatabaseSchema - 「my_cdm_data」) を指定しました。モデルに含まれる対象者は、CDM における初回ビギット日が 2010 年 1 月 1 日から 2017 年 12 月 31 日の間の人です。また、xSpec コホートを作成するために使用されたコンセプト ID 312327、314666、およびそれらの下位層は、除外しています。これらの初回ビギット時の年齢は 18 歳から 90 歳の間です。上記のパラメータを使用した場合、このステップで出力される予測モデルの名前は次のようにになります。「c:/temp/lr_results_5XMI_train_myCDM_ePPV0.75_20181206V1.rds」

ステップ 4: 評価コホートの作成

関数 `createEvalCohort` は、パッケージ関数 `applyModel` を使用して、対象とする健康アウトカムの予測確率をそれぞれ持つ多数の人々からなるコホートを作成します。この関数では、`xSpec` コホートを指定する必要があります (`xSpecCohort` パラメータを `xSpec` コホート ID に設定します)。また、前のステップで行ったように、評価コホートに含まれる人々の特性を指定することもできます。これには、下限および上限の年齢（それぞれ、`lowerAgeLimit` および `upperAgeLimit` 引数として年齢を設定）、性別（`gender` パラメータを男性および/または女性のコンセプト ID に設定）、開始日および終了日（それぞれ `startDate` および `endDate` 引数として日付を設定）、および対象とする母集団から対象者を選択する際に使用する母集団の ID として `mainPopnCohort` を設定することによって指定できます。

例えば：

```
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")
```

この例では、パラメータにより、関数がモデルファイル「c:/temp/lr_results_5XMI_train_myCDM」を使用して評価コホートファイル「c:/temp/lr_results_5XMI_eval_myCDM_ePPV0.75_2018」を生成することが指定されています。このステップで作成されたモデルファイルと評価用コホートファイルは、次のステップで提供されるコホート定義の評価に使用されます。

ステップ 5: コホート定義の作成とテスト

次のステップは、評価対象のコホート定義を作成し、テストすることです。望ましい性能特性は、対象とする研究課題に対処するためのコホートの使用目的によって異なる場合があります。特定の研究課題には非常に感度の高いアルゴリズムが必要となる場合もありますが、より特異的なアルゴリズムが必要となる場合もあります。PheEvaluator を使用してコホート定義の性能特性を決定するプロセスを図 16.4 に示します。

図 16.4 のパート A では、私たちは、テスト対象となるコホート定義の人物を調査し、コホート定義に含まれる評価コホート（前のステップで作成）の人物（人物 ID 016、019、022、023、025）と、含まれない評価コホートの人物（人物 ID 017、018、020、021、024）を見つけました。これらの対象者/非対象者それぞれについて、予測モデルを使用して健康アウトカムの確率を事前に決定していました ($p(O)$)。

真陽性、真陰性、偽陽性、偽陰性の値は、以下のように推定しました（図 16.4 のパート B）：

1. コホート定義に評価コホートに属する人物が含まれていた場合、すなわち、コホート定義がその人物を「陽性」とみなした場合。健康アウトカムの予測確率は、その人物が真陽性に寄与するカウント数の期待値を示し、1 から確率を引いた値は、その人物が偽陽性に寄与するカウント数の期待値を示します。人物ごとのすべてのカウント数の期待値を合計します。例えば、PersonId 016 の健康結果の存在の予測確率は 99% であり、 $0.99 - 0.01 = 0.98$ が真陽性（カウントの期待値に 0.99 を追加）に追加され、 $1.00 - 0.99 = 0.01$ が偽陽性（0.01 の期待値）に追加されました。この処理は、コホート定義に含まれる評価コホートの全人物（すなわち、PersonIds 019、022、023、および 025）に対して繰り返されました。
2. 同様に、コホート定義が評価コホートに属する人物を含んでいなかった場合、すなわちコホート定義がその人物を「陰性」とみなした場合、その人物の表現型に対する予測確率を 1 から引いた値が「真陰性」に寄与するカウントの期待値となり、それに加えられます。また、並行して、表現型に対する予測確率は「偽陰性」に寄与するカウントの期待値となり、それに加えられます。例えば、PersonId 017 の健康アウトカムの存在に対する予測確率は 1%（および、対応する健康アウトカムの不在は 99%）であり、 $1.00 - 0.01 = 0.99$ が真陰性に、 0.01 が偽陰性に追加されました。この手順を、コホート定義に含まれない評価コホートの全対象者（すなわち、PersonIds 018、020、021、024）に対して繰り返しました。

これらの値を評価コホート内の全対象者について加算した後、4 つのセルに各セルの期待値を記入し、感度、特異度、陽性適中率などの PA の性能特性の点推定値を作成することができました（図 16.4 のパート C）。これらの期待セルカウントは、推定値の分散を評価するために使用することはできず、点推定値のみに使用できることを強調しておきます。この例では、感度、特異度、陽性適中率、陰性適中率はそれぞれ 0.99、0.63、0.42、0.99 でした。

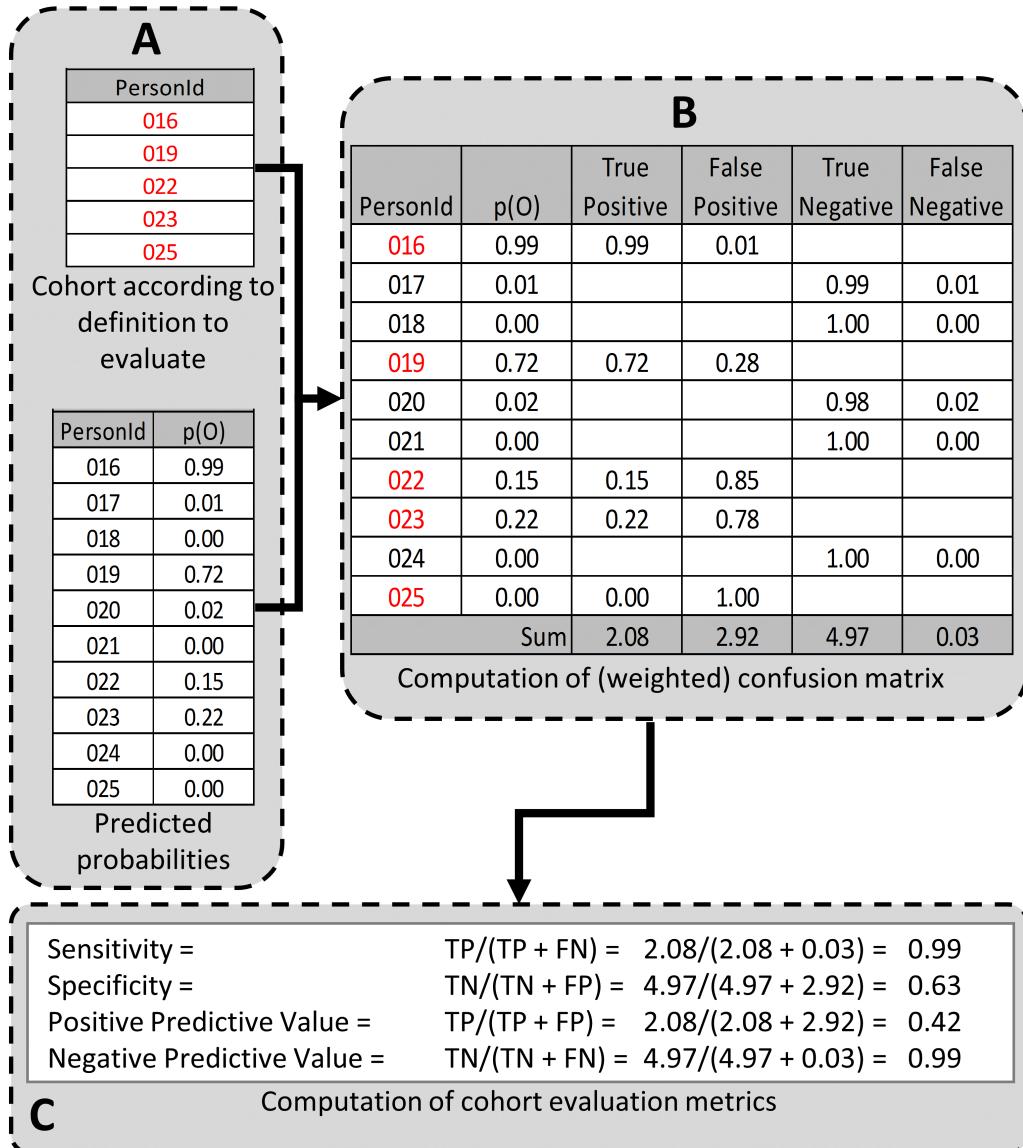


Figure 16.4: PheEvaluator を使用したコホート定義の性能特性の決定 $p(O) =$ 結果の確率; TP = 真陽性; FN = 偽陰性; TN = 真陰性; FP = 偽陽性

コホート定義の性能特性を決定するには、関数 `testPhenotype` を使用します。この関数は、モデルと評価コホートを作成した前の 2 つのステップからの出力を使用します。この例では、`createPhenoModel` 関数からの RDS ファイル出力である「c:/temp/lr_results_5XMI_train_myCDM_ePPV0.75_20181206V1.rds」に `modelFileName` パラメータを設定します。この例では、`createEvalCohort` 関数から出力された RDS ファイル「c:/temp/lr_results_5XMI_eval_myCDM_ePPV0.75_20181206V1.rds」に結果ファイル名パラメータを設定します。研究で使用するコホート定義をテストするために、`cohortPheno` をそのコホート定義のコホート ID に設定します。`phenText` パラメータを、コホート定義として人が読める説明文で設定します。例えば、「MI 発症、入院患者」などです。`testText` パラメータを、`xSpec` 定義の人が読める説明文で設定します。例えば、「5 x MI」などです。このステップの出力は、テストされたコホート定義の性能特性を含むデータフレームです。

cutPoints

パラメータの設定は、性能特性の結果を導き出すために使用される値のリストです。性能特性は通常、図 1 で説明されているように「期待値」を使用して計算されます。期待値に基づく性能特性を取得するには、

cutPoints

パラメータのリストに「EV」を含めます。また、特定の予測確率、すなわちカットポイントに基づく性能特性を確認したい場合もあります。例えば、予測確率が 0.5 以上の場合は健康状態が良好と見なされ、予測確率が 0.5 未満の場合は健康状態が不良と見なされる場合の性能特性を確認したい場合、`cutPoints` パラメータのリストに「0.5」を追加します。例えば：

```
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")
```

この例では、予測閾値の幅広い範囲 (cutPoints) が提供されており、期待値 (「EV」) も含まれています。パラメータ設定を前提として、このステップからの出力は、期待値計算を用いた場合と同様に、各予測閾値におけるパフォーマンス特性 (感度、特異度など) を提供します。評価には、前のステップで作成された評価コホートの予測情報が使用されます。このステップで作成されたデータフレームは、検証用に CSV ファイルに保存することができます。このプロセスを使用すると、表 16.1 は、5 つのデータセットにおける MI の 4 つのコホート定義の性能特性を示します。Cutrona 氏らによって評価されたものと同様のコホート定義「 $\geq 1 \times \text{HOI}$, In-Patient」では、平均 PPV は 67% (範囲: 59%~74%) であることが分かりました。

Table 16.1: pheEvaluator を使用して複数のデータセット上で心筋梗塞を診断するための診断条件コードを用いた 4 つのコホート定義の性能特性 Sens – 感度; PPV – 陽性適中率; Spec – 特異度; NPV – 陰性適中率; Dx Code – コホートの診断コード。

Phenotype Algorithm	Database	Sens	PPV	Spec	NPV
$\geq 1 \times \text{HOI}$	CCAE	0.761	0.598	0.997	0.999
	Optum1862	0.723	0.530	0.995	0.998
	OptumGE60	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
$\geq 2 \times \text{HOI}$	CCAE	0.585	0.769	0.999	0.998
	Optum1862	0.495	0.693	0.998	0.996
	OptumGE60	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
$\geq 1 \times \text{HOI}$, In-Patient	CCAE	0.674	0.737	0.999	0.998
	Optum1862	0.623	0.693	0.998	0.997
	OptumGE60	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
	OptumGE60	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 エビデンスの一般化可能性

コホートは、特定の観察データベースの文脈内で明確に定義され、十分に評価される可能性がありますが、臨床的有効性は、結果が対象とする母集団に一般化可能とみなされる程度によって制限されます。同じテーマに関する複数の観察研究は、異なる結果をもたらす可能性があり、その原因是、研究デザインや分析方法だけでなく、データソースの選択にもある Madigan et al. (2013b) は、データベースの選択が観察研究の結果に影響を与えることを実証しました。彼らは、10 の観察研究データベースにわたる 53 の薬物とアウトカムの組み合わせ、および 2 つの研究デザイン（コホート研究と自己対照ケースシリーズ）について、結果の異質性を系統的に調査しました。研究デザインを一定に保ったにもかかわらず、効果推定値にかなりの異質性が観察されました。OHDSI ネットワーク全体を見ると、観察データベースは、対象とする集団（例えば、小児対高齢者、民間保険加入者対公的保険失業者）、データ収集のケア環境（例えば、入院患者対外来患者、プライマリケア対二次医療/専門医療）、データ収集プロセス（例えば、保険請求、EHR、臨床レジストリ）、ケアの基盤となる全国と地域の医療システムにおいて、かなり異なっています。これらの相違は、疾患や医療介入の効果を研究する際に観察される異質性として確認される場合があり、また、ネットワーク研究におけるエビデンスとなる各データソースの品質に対する信頼性に影響を与える可能性もあります。OHDSI ネットワーク内のすべてのデータベースは CDM に標準化されていますが、標準化によって集団全体に存在する真の固有の異質性が減少するわけではなく、単にネットワーク全体にわたる異質性を調査し、より深く理解するための一貫した枠組みが提供されるだけであることを理解しておくことが重要です。OHDSI 研究ネットワークは、世界中のさまざまなデータベースに同じ解析プロセスを適用できる環境を提供しており、研究者は他の方法論的側面を一定に保ちながら、複数のデータソースにわたる結果を解釈することができます。OHDSI ネットワーク研究におけるオープンサイエンスへの協調的アプローチでは、参加するデータパートナーの研究者が臨床分野の知識を持つ研究者や分析の専門知識を持つ方法論者と協力して作業を行います。これは、ネットワーク全体のデータの臨床的有効性に対する理解を集合的に高めるための方法のひとつであり、このデータを使用して生成されたエビデンスに対する信頼性を構築するための基盤となるべきものです。

16.6 まとめ



- 臨床的妥当性は、基礎となるデータソースの特性を理解し、分析内のコホートの性能特性を評価し、研究の対象集団への一般化可能性を評価することによって確立できます。
- コホートの定義は、コホートの定義と利用可能な観察データに基づいてコホート内で特定された人が、真に表現型に属する人をどの程度正確に反映しているかという観点で評価できます。

- コホート定義の検証には、感度、特異度、陽性適中率など、複数の性能特性を推定して、測定誤差を完全に要約し、調整できるようになります。
- ソースレコードの検証と PheEvaluator による臨床判定は、コホート定義の検証を推定するための 2 つの代替アプローチです。
- OHDSI ネットワーク研究は、データソースの異質性を調査し、実証結果の一般化可能性を拡大して、リアルワールドのエビデンスの臨床的有効性を改善するメカニズムを提供します。

第 17 章

ソフトウェアの妥当性

著者: Martijn Schuemie

ソフトウェアの妥当性に関する中心的な問題は

ソフトウェアが期待通りに動作しているか？

ソフトウェアの妥当性は、エビデンスの質にとって不可欠な要素です。つまり、私たちの分析ソフトウェアが期待通りに機能してこそ、信頼性の高いエビデンスを生成できるのです。セクション 17.1.1 で説明されているように、すべての研究をソフトウェア開発の演習と見なすことが不可欠であり、共通データモデル (CDM) のデータから推定値や表形式の数値などの結果に至るまでの分析全体を実行する自動スクリプトを作成します。このスクリプト、およびこのスクリプトで使用されるソフトウェアはすべて検証されなければなりません。セクション 8.1 で説明されているように、分析全体をカスタムコードとして記述することも、OHDSI Methods Library で利用可能な機能を用いることもできます。Methods Library を使用する利点は、その妥当性を確保するためにすでに多大な注意が払われているため、分析全体の妥当性を確立する負担が軽減されることです。

この章では、まず有効な分析コードの記述に関するベストプラクティスについて説明します。次に、ソフトウェア開発プロセスとテストを通じて、Methods Library がどのように検証されているかについて説明していきます。

17.1 研究コードの妥当性

17.1.1 再現性のための自動化の必要性

従来、観察研究はプロセスというよりも旅路として捉えられることがよくありました。データベースの専門家がデータベースからデータセットを抽出し、それをデータ分析者に渡します。データ分析者はスプレッドシートエディタや他のインタラクティブなツールで開き、分析作業を開始します。最終的に結

果が生成されますが、その生成過程はほとんど保存されません。旅路の目的地に到達しても、そこへ至るまでの正確なステップを辿ることはできません。このようなやり方は、再現できないというだけでなく、透明性にも欠けるため、まったく受け入れられないのです。結果を導くために何が行われたのかが正確にわからないため、ミスがなかったことを検証することもできません。したがって、エビデンスを生み出す分析はすべて完全に自動化されなければなりません。自動化とは、分析が単一のスクリプトとして実装されることを意味し、テーブルや図表を含め、CDM 形式のデータベースから結果まで、単一のコマンドで分析全体をやり直せるようにしなければなりません。分析は任意の複雑さで実施でき、おそらくは単一のカウントのみを生成するか、あるいは何百万もの研究課題に対する経験的に校正された推定値を生成することになるでしょう。しかし、同じ原則が適用されます。スクリプトは他のスクリプトを呼び出すことができ、さらに下位レベルの分析プロセスを呼び出すこともできます。分析スクリプトは、どのようなコンピュータ言語でも実装できますが、OHDSI では R 言語が推奨されています。R の DatabaseConnector パッケージのおかげで、CDM 形式のデータに直接接続でき、また、OHDSI Methods Library の他の R パッケージを通じて、多くの高度な分析機能を利用できます。

17.1.2 プログラミングのベストプラクティス

観察分析は、最終結果を得るまでに多くのステップを必要とするため、非常に複雑になる可能性があります。この複雑性により、分析コードの維持が難しくなり、エラーが発生する可能性が高まるだけでなく、エラーに気づきにくくなる可能性もあります。幸いにも、コンピュータプログラマーは長年にわたり、複雑な問題に対処できるコードを書くためのベストプラクティスを開発してきました。そのコードは、読みやすく、再利用、適応、検証が容易になっています (Martin, 2008)。これらのベストプラクティスについて詳しく説明すると、多くの書籍が書けるほどですので、ここでは以下の 4 つの重要な原則に焦点を当てます。

- 抽象化：すべてを実行する巨大なスクリプトを 1 つ書くのではなく、コードの行と行の間の依存関係がどこからどこへでも及ぶ（例えば、10 行目に設定された値が 1,000 行目で使用される）いわゆる「スパゲッティ・コード」を避けるために、コードを「関数」と呼ばれる単位で整理することができます。関数は明確な目的を持つべきであり、例えば「ランダムなサンプルを抽出する」といったものです。関数を作成すれば、その関数が何を行うのかの詳細を考えることなく、より大きなスクリプトでその関数を使用することができます。
- カプセル化：抽象化を機能させるには、関数の依存関係を最小限に抑え、明確に定義する必要があります。例のサンプリング関数は、いくつかの引数（例えばデータセットとサンプルサイズ）と 1 つの出力（例えばサンプル）を持つべきです。関数の動作に影響を与えるものは他に何もあってはなりません。いわゆる「グローバル変数」、つまり関数外で設定される変数は、関数の引数ではありませんが、関数内で使用されるため、避けるべ

きです。

- わかりやすい命名：変数や関数はわかりやすい名前を付けるべきであり、コードはほとんど自然言語のように読みやすくすべきです。例えば、`x <- sample(y, 100)` の代わりに、`sampledPatients <- takeSample(patients, sampleSize = 100)` と記述できます。省略したくなる衝動を抑えるようにしてください。最新の言語では、変数名や関数名の長さに制限はありません。
- 再利用：明確で、うまくカプセル化された関数を書くことの利点のひとつは、それらを再利用できることが多いということです。これは時間を節約するだけでなく、コードの量が減ることによって複雑さが減り、エラーの可能性も少なくなることを意味します。

17.1.3 コードの検証

ソフトウェアコードの妥当性を検証する方法はいくつかありますが、観察研究を実施するコードには特に次の 2 つの方法が有効です。

- コードレビュー：1 人がコードを書き、別の 1 人がそのコードをレビューする。
- ダブルコーディング：2 人がそれぞれ独立して分析コードを書き、その後、2 つのスクリプトの結果を比較する。

コードレビューには通常、作業量が少ないという利点がありますが、欠点としては、レビュー者がエラーを見逃す可能性があることです。一方、ダブルコーディングは通常、非常に手間がかかりますが、エラーを見逃す可能性は低く、不可能ではないことです。ダブルコーディングのもう一つの欠点は、多くの些細な恣意的な選択（例えば、「exposure end」を `exposure end date` を含むと解釈すべきか、それともそうでないか）が必要なため、2 つの別々の実装ではほとんどの場合に異なる結果が得られることです。その結果、本来独立しているはずの 2 人のプログラマーが、分析結果を一致させるために協力する必要が生じ、独立性が損なわれることになります。

ユニットテストなどの他のソフトウェア検証手法は、研究が通常、入力（CDM のデータ）と出力（研究結果）の間に高度な複雑な関係があるため、1 回限りの活動であり、これらの手法はあまり適切ではありません。これらの手法は、Methods Library で適用されていることに注意してください。

17.1.4 Methods Library の使用

OHDSI Methods Library は、多数の関数を提供しており、ほとんどの観察研究は数行のコードを記述するだけで実施することができます。Methods Library を使用することで、研究コードの妥当性を立証する負担のほとんどが Library に移行されます。Methods Library の妥当性は、そのソフトウェア開発プロセスと広範なテストによって保証されています。

17.2 Methods Library のソフトウェア開発プロセス

OHDSI Methods Library は OHDSI コミュニティによって開発されています。Libraryへの変更提案は、GitHub の issue tracker (例えば、CohortMethod issue tracker¹) と OHDSI フォーラムの 2つの場所で議論されます²。いずれも一般公開されています。コミュニティのメンバーは誰でもライブラリにソフトウェアコードを寄与することができますが、リリースされたソフトウェアのバージョンに組み込まれる変更の最終承認は、OHDSI 集団レベルの推定ワークグループのリーダー (Marc Suchard 博士と Martijn Schuemie 博士) および OHDSI 患者レベルの予測ワークグループのリーダー (Peter Rijnbeek 博士と Jenna Reps 博士) のみが行います。

ユーザーは GitHub のリポジトリにあるマスターブランチから直接、または「drat」と呼ばれるシステムを通じて、R に Methods Library をインストールすることができます。Methods Library のパッケージの多くは R の Comprehensive R Archive Network (CRAN) を通じて入手でき、この数は今後さらに増える見込みです。

OHDSI では、Methods Library のパフォーマンスの正確性、信頼性、一貫性を最大限に高めるため、適切なソフトウェア開発やテスト方法を採用しています。重要なのは、Methods Library が Apache License V2 の条件に基づいてリリースされているため、Methods Library の基盤となるすべてのソースコード (R、C++、SQL、Java のいずれであっても) は、OHDSI コミュニティのすべてのメンバーに、また一般公開されています。したがって、Methods Library に具現化されたすべての機能は、その正確性、信頼性、一貫性に関して、継続的な評価と改善の対象となります。

17.2.1 ソースコード管理

Methods Library のソースコードはすべて、GitHub を通じて一般公開されているソースコードバージョン管理システム「git」で管理されています。OHDSIMethods Library のリポジトリはアクセス制御されています。世界中の誰もがソースコードを閲覧でき、OHDSI コミュニティのメンバーであれば誰でも、いわゆるプルリクエストを通じて変更を提出することができます。OHDSI 人口レベル推定作業部会および患者レベル予測作業部会のリーダーシップのみが、こうしたリクエストを承認し、マスターブランチに変更を加え、新しいバージョンをリリースすることができます。GitHub リポジトリ内では、コード変更の継続的なログが管理されており、コードとドキュメントの変更のあらゆる側面が反映されています。これらのコミットログは、一般公開されています。

新しいバージョンは、OHDSI 集団レベルの推定ワークグループや患者レベルの予測ワークグループのリーダーシップにより、必要に応じてリリースされます。新しいリリースは、前のリリースのバージョン番号よりも大きなパッケージの

¹<https://github.com/OHDSI/CohortMethod/issues>

²<http://forums.ohdsi.org/>

バージョン番号（パッケージ内の DESCRIPTION ファイルで定義されている）をマスター・ブランチにプッシュすることから始まります。これにより、パッケージの確認とテストが自動的に開始されます。すべてのテストに合格すると、新しいバージョンがバージョン管理システムに自動的にタグ付けされ、パッケージが OHDSI ドラフトリポジトリに自動的にアップロードされます。新しいバージョンは、3 つのコンポーネントからなるバージョン番号で表されます。

- 新しいマイクロバージョン（例：4.3.2 から 4.3.3）は、バグ修正のみを示します。新しい機能はなく、前方および後方互換性が保証されます
- 新しいマイナーバージョン（例：4.3.3 から 4.4.0）は、機能追加を示します。後方互換性のみが保証されます
- 新しいメジャーバージョン（例：4.4.0 から 5.0.0）は、大幅な改訂を示します。互換性については保証されません。

17.2.2 ドキュメンテーション

Methods Library 内のすべてのパッケージは、R の内部ドキュメントフレームワークを通じて文書化されています。各パッケージには、そのパッケージで利用可能なすべての関数が記載されたパッケージマニュアルがあります。関数のドキュメントと関数の実装を一致させるため、Roxygen2 ソフトウェアを使用して、関数のドキュメントとソースコードを 1 つのファイルにまとめています。パッケージマニュアルは、R のコマンドラインインターフェースからオンデマンドで利用でき、パッケージリポジトリでは PDF として、またウェブページとしても利用できます。さらに、多くのパッケージには、そのパッケージの特定の使用事例を強調するビネットも用意されています。すべてのドキュメントは、Methods Library ウェブサイトで閲覧できます³。

Methods Library のソースコードはすべてエンドユーザーが利用できます。コミュニティからのフィードバックは、GitHub の課題追跡システムと OHDSI フォーラムを使用して促進されます。

17.2.3 現行および過去のアーカイブバージョンの利用可能性

Methods Library パッケージの現在および過去のバージョンは、2 つの場所で入手できます。まず、GitHub バージョン管理システムには各パッケージの開発の全履歴が含まれており、各時点でのパッケージの状態を再現して取得することができます。最も重要なのは、GitHub で各リリースバージョンにタグ付けがされていることです。次に、リリースされた R ソースパッケージは、OHDSI の GitHub drat リポジトリに保存されています。

³<https://ohdsi.github.io/MethodsLibrary/>

17.2.4 メンテナンス、サポート、およびリタイアメント

Methods Library の各最新バージョンは、OHDSI によりバグレポート、修正、パッチに関して積極的にサポートされています。GitHub の課題追跡システムや OHDSI フォーラムを通じて問題を報告することができます。各パッケージにはパッケージマニュアルがあり、0 個、1 個、または複数のビネットがあります。オンラインビデオチュートリアルが利用可能であり、また、対面式のチュートリアルも隨時提供されています。

17.2.5 有資格者

OHDSI コミュニティのメンバーは、複数の統計分野を代表しており、複数の地域にまたがる学術機関、非営利団体、業界関連機関に所属しています。OHDSI 集団レベルの推定ワークグループや OHDSI 患者レベルの予測ワークグループのすべてのリーダーは、認定された学術機関で博士号を取得しており、査読付き学術誌に多数の論文を発表しています。

17.2.6 物理的および論理的セキュリティ

OHDSI Methods Library は GitHub⁴システムでホストされています。GitHub のセキュリティ対策については、<https://github.com/security>を参照してください。OHDSI コミュニティのすべてのメンバーが Methods Library に変更を加えるには、ユーザー名とパスワードが必要です。また、マスタープランチに変更を加えられるのは、集団レベルの推定ワークグループと患者レベルの予測ワークグループのリーダーのみです。ユーザーアカウントは、標準的なセキュリティポリシーや機能要件に基づいてアクセスが制限されています。

17.2.7 災害復旧

OHDSI Methods Library は GitHub システム上でホストされています。GitHub の災害復旧施設については、<https://github.com/security> に説明があります。

17.3 Methods Library のテスト

Methods Library で実行されるテストには、パッケージ内の個々の関数に対するテスト（いわゆる「ユニットテスト」）と、シミュレーションを使用したより複雑な機能に対するテストの 2 種類があります。

17.3.1 ユニットテスト

OHDSI により、ソースコードを既知のデータおよび既知の結果に対してテストできるように、多数の自動検証テストが維持、更新されています。各テストは、

⁴<https://github.com/>

いくつかの単純な入力データの指定から始まり、この入力に対してパッケージのいずれかの関数を実行し、出力が期待通りのものであるかどうかを評価します。単純な関数では、期待される結果は明白であることがほとんどです（例えば、少数の対象者のみを対象としたサンプルデータで傾向スコアのマッチングを行う場合など）。より複雑な関数では、Rで利用可能な他の関数との組み合わせて、期待される結果が生成されることがあります（例えば、当社の大規模回帰エンジンである Cyclops は、R の他の回帰ルーチンを使用して単純な問題の結果を比較することでテストします）。私たちは、これらのテストを実行可能なソースコードの行の合計 100% をカバーすることを目指しています。

これらのテストは、パッケージに変更が加えられた場合（具体的には、変更がパッケージリポジトリにプッシュされた場合）に自動的に実行されます。テスト中にエラーが検出された場合は、ワークグループのリーダーに自動的に電子メールが送信され、パッケージの新しいバージョンのリリース前に解決されなければなりません。これらのテストのソースコードおよび期待される結果は、必要に応じて確認および他のアプリケーションで使用することができます。これらのテストは、エンドユーザーおよび/またはシステム管理者にも利用可能であり、インストールプロセスの一部として実行することで、Methods Library のインストールに関する正確性、信頼性、一貫性に関する追加の文書および客観的な証拠を提供することができます。

17.3.2 シミュレーション

より複雑な機能については、入力に対して期待される出力がどのようなものであるべきかが常に明らかであるとは限りません。このような場合、特定の統計モデルを基に入力を生成し、機能がこの既知のモデルに沿った結果を生成するかどうかを検証するために、シミュレーションが使用されることがあります。例えば、SelfControlledCaseSeriesパッケージでは、シミュレーションデータにおける一時的な傾向を検出し、適切にモデル化できることを検証するために、シミュレーションが用いられます。

17.4 まとめ



- 再現性と透明性を確保するため、観察研究は CDM のデータから結果まで、分析全体を実行する自動スクリプトとして実施すべきです。
- カスタムスタディコードは、抽象化、カプセル化、明確な命名、コードの再利用など、最良のプログラミング慣行に従うべきです。
- カスタムスタディコードは、コードレビューまたはダブルコーディングにより検証することができます。
- Methods Library は、観察研究で使用できる検証済みの機能を提供しています。
- Methods Library は、有効なソフトウェアを作成すること目的としたソフトウェア開発プロセスとテストにより検証されています。

第 18 章

方法の妥当性

著者: Martijn Schuemie

方法の妥当性を検討する際、次の質問に答えようとなります。

この方法は、この質問に答えるために妥当ですか？

「方法」には研究デザインだけでなく、データやデザインの実施も含まれます。したがって、方法の妥当性は、ある意味で包括的なものです。多くの場合、データ品質、臨床的妥当性、ソフトウェアの妥当性が良くなければ、方法の妥当性を良好に保つことはできません。方法の妥当性を検討する前に、エビデンスの質に関して、これらの側面はすでに個別に対処しておく必要があります。

方法の妥当性を確立する上での中心的な活動は、分析における重要な仮定が満たされているかどうかを評価することです。例えば、傾向スコアマッチングによって 2 つの母集団を比較可能になるという前提を立てますが、それが事実であるかどうかを評価する必要があります。可能な場合には、これらの前提を検証するための実証的テストを実施すべきです。例えば、マッチング後の 2 つの母集団が、幅広い特性において実際に比較可能であることを示す診断を生成することができます。OHDSI では、分析が実施されるたびに生成および評価されるべき、多くの標準診断を開発してきました。

本章では、集団レベルの推定で使用される手法の妥当性に焦点を当てます。まず、研究デザインに特化した診断法を簡単に説明し、次に、集団レベルの推定を行う研究のすべてではないにしても、ほとんどに適用できる診断法について説明します。最後に、OHDSI ツールを使用してこれらの診断法を実行する方法を段階的に説明します。本章の締めくくりとして、OHDSI Methods Benchmark と OHDSI Methods Library への応用について、高度なトピックも紹介します。

18.1 デザイン特有の診断

各研究デザインには、そのデザインに特有の診断法があります。これらの診断法の多くは、OHDSI Methods Libraryの R パッケージで実装されており、すぐに利用できます。例えば、セクション 12.9 では、CohortMethod パッケージで生成される幅広い診断法がリストアップされており、以下が含まれます。

- コホートの初期の比較可能性を評価するための傾向スコア分布。
- モデルから除外すべき潜在変数を特定するための傾向モデル。
- 傾向スコア調整によりコホートが比較可能になったかどうかを評価するための共変量バランス（ベースライン共変量で測定）。
- さまざまな分析ステップで除外された対象者の数を観察するための脱落。これは、対象とする初期コホートから得られる結果の一般化可能性について情報を提供する可能性があります。
- 質問に答えるのに十分なデータが利用可能かどうかを評価するための検出力。
- 典型的な発症までの時間を評価し、Cox モデルの基礎となる比例性の仮定が満たされているかどうかを評価するための Kaplan Meier 曲線。

他の研究デザインでは、それらのデザインの異なる仮説を検証するために、異なる診断が必要となります。例えば、自己対照ケースシリーズ（SCCS）デザインでは、観察の終了が結果に依存しないという必要な仮定を確認することができます。この仮定は、心筋梗塞などの深刻で致死の可能性もある事象の場合には、成立していないことがよくあります。この仮定が成り立つかどうかは、図 18.1 に示すプロットを生成することで評価できます。このプロットは、打ち切りとなったもの、打ち切りとならなかったものの観察期間終了までの時間をヒストグラムで示しています。私たちのデータでは、データ収集の最終日（データベース全体で観察が終了した日、例えば抽出日や研究終了日）に観察期間が終了したものを持ち切らなければならず、それ以外を打ち切りとみなします。図 18.1 では、2 つの分布の間にわずかな違いしか見られず、私たちの仮説が正しいことが示唆されています。

18.2 推定のための診断

デザイン固有の診断に加え、因果効果の推定法全般に適用できる診断もいくつあります。これらの多くは、すでに答えがわかっている研究上の仮説であるコントロール仮説の使用に依存しています。コントロール仮説を使用することで、デザインが真実と一致した結果を生み出しているかどうかを評価することができます。コントロールは、ネガティブコントロールとポジティブコントロールに分けることができます。

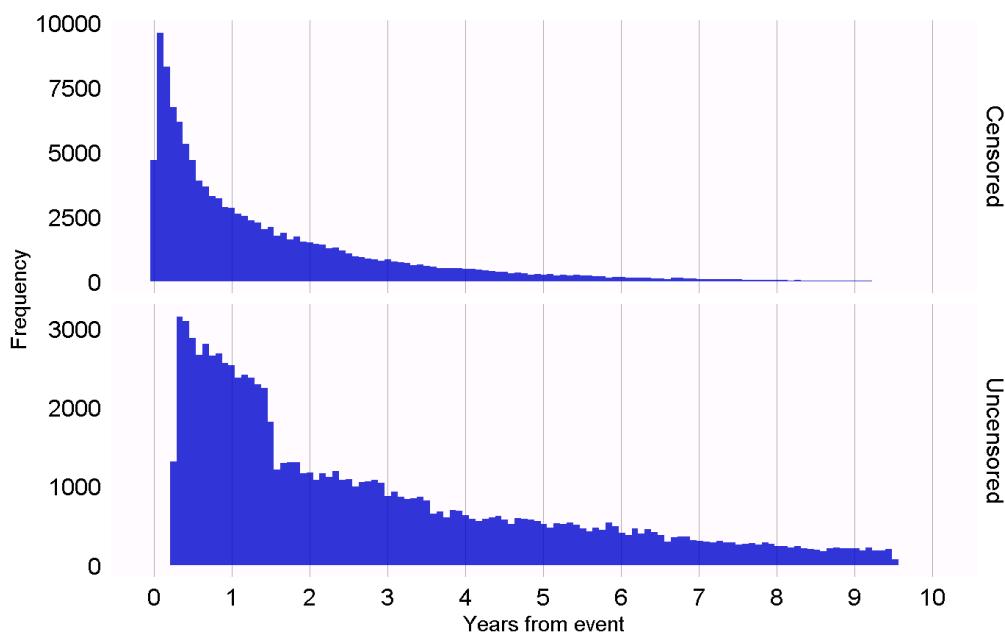


Figure 18.1: 打ち切りありと打ち切りなしとされた対象者の観察終了までの時間。

18.2.1 ネガティブコントロール

ネガティブコントロールとは、因果関係が存在しないと考えられる曝露と結果の組み合わせであり、交絡 (Lipsitch et al., 2010)、選択バイアス、測定誤差 (Arnold et al., 2016) を検出する方法として推奨されているネガティブコントロールまたは「偽陰性エンドポイント」(Prasad and Jena, 2013) が含まれます。たとえば、小児期の疾患と後の多発性硬化症 (MS) との関係を調査したある研究 (Zaadstra et al., 2008) では、著者は MS の原因とは考えられていない 3 つのネガティブコントロール（腕の骨折、脳震盪、扁桃摘出）を含めています。この 3 つのコントロールのうち 2 つは MS と統計的に有意な関連性を示しており、この研究にはバイアスが生じた可能性を示唆していました。

私たちは、関心のある仮説と比較可能なネガティブコントロールを選択すべきであり、通常は、関心のある仮説と同じ曝露を持つ曝露-アウトカムの組み合わせ（いわゆる「アウトカムコントロール」）または同じ結果を持つ曝露-アウトカムの組み合わせ（「曝露コントロール」）を選択します。ネガティブコントロールは、さらに以下の基準を満たすべきです。

- 曝露がアウトカムを引き起こすべきではない。因果関係を考える一つの方法は、仮説を否定するものを考えることです。患者が曝露されなかった場合と曝露された場合とを比較して、結果が引き起こされる（または防止される）可能性があるだろうか？時には、これは明らかです。例えば、ACE 阻害薬は血管性浮腫を引き起こすことが知られています。しかし、時にはそれほど明白ではないこともあります。例えば、高血圧を引き起こす可能

性のある薬物は、間接的に高血圧の結果である心血管疾患を引き起こす可能性があります。

- 曝露はアウトカムを予防または治療すべきではありません。これは、真の効果量（例えばハザード比）が1であると考えるのであれば、存在しないはずの因果関係の一つにすぎません。
- ネガティブコントロールはデータ内に存在すべきであり、理想的には十分な数であるべきです。この目的を達成するために、有病率に基づいてネガティブコントロールの候補を優先付けします。
- ネガティブコントロールは理想的には独立しているべきです。例えば、ネガティブコントロールが互いに祖先（例えば、「巻き爪」と「足の巻き爪」）であったり、兄弟（例えば、「左大腿骨の骨折」と「右大腿骨の骨折」）であったりすることは避けるべきです。
- ネガティブコントロールは、ある程度の偏りの可能性があることが理想的です。例えば、社会保障番号の最後の数字は基本的にランダムな数字であり、交絡を示す可能性は低いでしょう。したがって、ネガティブコントロールとして使用すべきではありません。

また、ネガティブコントロールは、注目する曝露とアウトカムのペアと同じ交絡構造を持つべきであるという意見もあります (Lipsitch et al., 2010)。しかし、この交絡の構造は不明であると私たちは考えています。現実に見られる変数間の関係は、人々が想像するよりもはるかに複雑であることがよくあります。また、交絡因子の構造が分かっていたとしても、その交絡因子とまったく同じ構造を持ち、かつ直接的な因果効果を持たないネガティブコントロールが存在する可能性は低いでしょう。このため、OHDSIではこのようなセットは、関心のある仮説に存在するものも含め、多くの異なるタイプのバイアスを表すと想定して、多数のネガティブコントロールに依存しています。

曝露とアウトカムの間に因果関係がないことは、ほとんど文書化されていません。その代わり、関係性のエビデンスがないことは関係性がないことを意味するという仮定がしばしばなされます。曝露とアウトカムの両方が広範に研究されている場合、この仮定はより妥当性が高くなります。例えば、全く新しい薬物に対するエビデンスがないことは、関係性がないことを意味するのではなく、知識がないことを意味する可能性が高いです。この原則を念頭に、私たちはネガティブコントロールを選択するための半自動化された手順を開発しました (Voss et al., 2016)。簡単に説明すると、文献、製品ラベル、および自発報告から得られた情報は自動的に抽出され、統合されてネガティブコントロールの候補リストが作成されます。このリストはその後、自動抽出が正確であったことを確認するためだけでなく、生物学的妥当性などの追加の基準を課すためにも、手動でレビューする必要があります。

18.2.2 ポジティブコントロール

真の相対リスクが 1 より小さい場合、または 1 より大きい場合の方法の動作を理解するには、帰無仮説が真実ではないと考えられるポジティブコントロールを使用する必要があります。残念ながら、観察研究における実際のポジティブコントロールには、3 つの理由から問題が生じがちです。第一に、ほとんどの研究状況では、例えば 2 つの治療効果を比較する場合など、その特定の状況に該当するポジティブコントロールが不足しています。第二に、ポジティブコントロールが利用可能であったとしても、効果の大きさが正確にわからないことがあります。また、多くの場合、測定対象の母集団に依存します。第三に、治療が特定の結果を引き起こすことが広く知られている場合、例えば望ましくない結果のリスクを軽減するための措置が取られるなど、その治療を処方する医師の行動に影響を与え、その結果、評価手段としてのポジティブコントロールが役に立たなくなることがあります (Noren et al., 2014)。

そのため、OHDSI では合成したポジティブコントロール (Schuemie et al., 2018a) を使用しています。これは、曝露のリスクにさらされる期間中に、アウトカムを追加でシミュレーション発生をさせることで、ネガティブコントロールを修正して作成します。例えば、ACE 阻害薬への曝露中に、ネガティブコントロールのアウトカム「陷入爪」が n 回発生したと仮定します。曝露中にさらに n 回のシミュレーション発生を追加すると、リスクは 2 倍になります。これはネガティブコントロールであるため、対照群と比較した相対リスクは 1 ですが、追加の発生後は 2 になります。

重要な問題として、交絡因子の保存が挙げられます。ネガティブコントロールでは強い交絡が示されるかもしれません、ランダムに追加のアウトカムを発生させれば、これらから得られる結果は交絡されません。したがって、ポジティブコントロールの交絡に対処する能力の評価については楽観的になることができます。交絡を維持するには、新しい結果が元の結果と同様に、ベースラインの被験者固有の共変量と類似した関連性を示すようにする必要があります。これを達成するために、各アウトカムについて、曝露前に得られた共変量を用いて曝露中のアウトカムに関する生存率を予測するモデルをトレーニングします。これらの共変量には、人口統計学的データ、すべての記録された診断、薬物曝露、測定値、医療プロセッサー（処置）が含まれます。正則化ハイパープラメータを選択するために 10 分割の交差検証を用いる L1 正則化ポアソン回帰 (Suchard et al., 2013) により、予測モデルを適合させます。次に、曝露中のシミュレーション結果を予測率を用いてサンプリングし、真の効果の大きさを望ましい大きさにまで増大させます。その結果、ポジティブコントロールには、実際の結果とシミュレーション結果の両方が含まれます。

図 18.2 は、このプロセスを示しています。この手順では、いくつかの重要なバイアスの原因をシミュレーションしますが、すべてを捕捉するわけではありません。例えば、測定誤差の影響の一部は存在しません。合成されたポジティブコントロールは、一定の陽性適中率と感度を意味しますが、現実には必ずしもそうではない場合があります。

各コントロールについて、単一の真の「効果の大きさ」を参照していますが、異

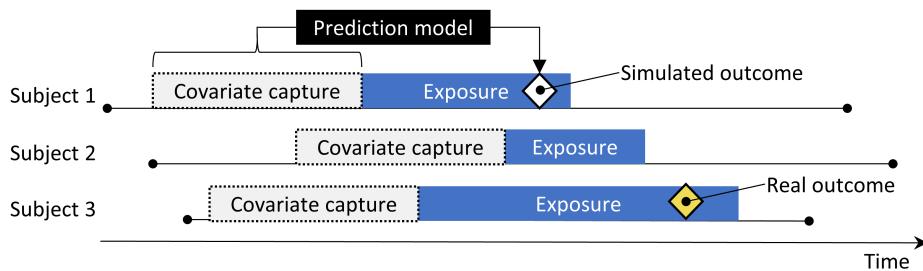


Figure 18.2: ネガティブコントロールからのポジティブコントロールの合成

なる手法では治療効果の異なる統計値を推定します。因果効果がないと考えるネガティブコントロールでは、相対リスク、ハザード比、オッズ比、罹患率比(条件付き、制限付き)、平均治療効果(ATT)や全体平均治療効果(ATE)など、すべての統計値は1となります。ポジティブコントロールを作成するプロセスでは、限界効果が達成される時点まで、この比率が一定に保たれる患者を条件としたモデルを使用し、時間経過や患者間の一定の発生率比で結果を統合します。したがって、真の効果量は、治療群における限界発生率比として保持されることが保証されます。合成に使用されるモデルが正しいという前提の下では、これは条件付き効果量およびATEにも当てはまります。結果はすべてまれであるため、オッズ比は相対リスクとほぼ同じになります。

18.2.3 実証的評価

ネガティブコントロールとポジティブコントロールに対する特定の手法の推定値に基づき、さまざまな指標を計算することで、その運用特性を理解することができます。例えば、

- ROC曲線下面積(AUC)：ポジティブコントロールとネガティブコントロールを識別する能力。
- カバー率：真の効果量が95%信頼区間に収まる頻度。
- 平均精度：精度は $1/()^2$ として計算され、精度が高いほど信頼区間が狭くなる。精度の歪んだ分布を考慮するために幾何平均を使用します。
- 平均二乗誤差(MSE)：効果量の推定値の対数と真の効果量の対数との間の平均二乗誤差。
- 第1種の過誤：ネガティブコントロールの場合、帰無仮説が棄却された頻度($\alpha = 0.05$)。これは偽陽性率、もしくは「1 - 特異度」と同等です。
- 第2種の過誤：ポジティブコントロールの場合、どのくらいの頻度で帰無仮説が棄却されなかったか($\alpha = 0.05$)。これは偽陰性率、もしくは「1 - 感度」と同等である。
- 推定なし：推定値を算出できなかったコントロールはいくつあったか？推定値が算出できない理由は様々であり、例えば傾向スコアマッチング後

に被験者が残らなかった場合や、結果を持つ被験者が残らなかった場合などである。

ユースケースに応じて、これらの操作特性が目的に適しているかどうかを評価することができます。例えば、シグナル検出を行いたい場合は、第1種の過誤と第2種の過誤を考慮する必要があります。また、 α 閾値を修正する場合は、代わりにAUCを検討することができます。

18.2.4 P値のキャリブレーション

しばしば、第1種の過誤 ($\alpha = 0.05$) は 5% よりも大きくなります。言い換えれば、実際には帰無仮説が真であるにもかかわらず、5% よりも高い確率で帰無仮説を棄却してしまう可能性が高いということです。その理由は、p 値はランダム誤差、すなわちサンプルサイズが限られていることによる誤差のみを反映しているためです。系統的誤差、例えば交絡による誤差は反映されません。OHDSI は、p 値を補正して第1種の過誤を名目上、回復するためのプロセスを開発しました (Schuemie et al., 2014)。ネガティブコントロールの実際の効果推定値から経験的帰無分布を導出します。これらのネガティブコントロールの推定値は、帰無仮説が真である場合に期待される値を示唆するものであり、経験的帰無分布を推定するためにそれらを使用します。

具体的には、各推定値のサンプリングエラーを考慮して、推定値にガウス確率分布を当てはめます。第 i 番目のネガティブコントロールとアウトカムの組から推定された対数効果推定値（相対リスク、オッズ、または発生率比）を θ_i で表し、対応する推定標準誤差を $\hat{\tau}_i$ 、 $i = 1, \dots, n$ で表します。 θ_i を真の対数効果量とし（ネガティブコントロールでは 0 と仮定）、 β_i を対 i に関する真の（ただし未知の）バイアス、すなわち、コントロールが非常に大きかった場合に研究がコントロール i に対して返すであろう推定値の対数と真の効果量の対数の差とします。標準的な p 値の計算と同様に、 $\hat{\theta}_i$ は、 $\theta_i + \beta_i$ を平均とし、 $\hat{\tau}_i^2$ を標準偏差とする正規分布に従うと仮定します。従来の p 値の計算では、 β_i は常にゼロと仮定されていましたが、我々は、 μ を平均とし σ^2 を分散とする正規分布から生じる β_i を仮定します。これは、帰無（バイアス）分布を表します。最尤法により μ と σ^2 を推定します。つまり、以下の仮定を置きます。

$$\beta_i \sim N(\mu, \sigma^2) \text{かつ} \hat{\theta}_i \sim N(\theta_i + \beta_i, \tau_i^2)$$

ここで $N(a, b)$ は平均値 a 、分散 b のガウス分布を示し、 μ と σ^2 を次の尤度を最大にすることにより求めます：

$$L(\mu, \sigma | \theta, \tau) \propto \prod_{i=1}^n \int p(\hat{\theta}_i | \beta_i, \theta_i, \hat{\tau}_i) p(\beta_i | \mu, \sigma) d\beta_i$$

ここから最大尤度推定 $\hat{\mu}$ と $\hat{\sigma}$ を得ます。キャリブレートされた p 値を実証的な帰無分布を用いて計算します。新薬-アウトカムペアの効果推定 $\hat{\theta}_{n+1}$ を取り、

対応する推定標準誤差 $\hat{\tau}_{n+1}$ を用います。前述の仮定の下で β_{n+1} が同じ帰無分布から発生したとして、次が得られます。

$$\hat{\theta}_{n+1} \sim N(\hat{\mu}, \hat{\sigma} + \hat{\tau}_{n+1})$$

ここでは、 $\hat{\theta}_{n+1}$ は $\hat{\mu}$ より小さく、新しいペアのキャリブレーションされた片側 P 値は、

$$\phi \left(\frac{\theta_{n+1} - \hat{\mu}}{\sqrt{\hat{\sigma}^2 + \hat{\tau}_{n+1}^2}} \right)$$

ここでは $\phi(\cdot)$ は、標準正規分布の累積分布関数を表します。また、 $\hat{\theta}_{n+1}$ が $\hat{\mu}$ より大きいとき、キャリブレーションされた片側 P 値は、

$$1 - \phi \left(\frac{\theta_{n+1} - \hat{\mu}}{\sqrt{\hat{\sigma}^2 + \hat{\tau}_{n+1}^2}} \right)$$

18.2.5 信頼区間のキャリブレーション

同様に、通常、95% 信頼区間のカバー率は 95% 未満であることが観察されます。真の効果量は、95% 信頼区間に収まるのは 95% 未満です。信頼区間キャリブレーション (Schuemie et al., 2018a) では、ポジティブコントロールも活用することで、p 値キャリブレーションの枠組みを拡張します。通常、必ずしもそうとは限りませんが、キャリブレーションされた信頼区間は名義尺度の信頼区間よりも広くなり、標準的な手順では考慮されないがキャリブレーションでは考慮される問題（未測定の交絡、選択バイアス、測定誤差など）を反映しています。

厳密には、 β_i (i に関するバイアス) は再びガウス分布から得られると仮定しますが、今回は平均と標準偏差が真のエフェクトサイズである θ_i と線形関係にあるものを使用します：

$$\beta_i \sim N(\mu(\theta_i), \sigma^2(\theta_i))$$

ここでは、

$$\begin{aligned}\mu(\theta_i) &= a + b \times \theta_i \text{ かつ} \\ \sigma(\theta_i)^2 &= c + d \times |\theta_i|\end{aligned}$$

a, b, c, d は、未観測の β_i を積分した以下の周辺尤度を最大化することで推定します：

$$l(a, b, c, d | \theta, \hat{\theta}, \hat{\tau}) \propto \prod_{i=1}^n \int p(\hat{\theta}_i | \beta_i, \theta_i, \hat{\tau}_i) p(\beta_i | a, b, c, d, \theta_i) d\beta_i,$$

そしてこれにより最尤推定値 $(\hat{a}, \hat{b}, \hat{c}, \hat{d})$ を求めます。

系統誤差モデルを用いてキャリブレーションされた信頼区間を計算します。再び $\hat{\theta}_{n+1}$ を新しい対象結果に対する効果推定値の対数とし、 $\hat{\tau}_{n+1}$ を対応する推定標準誤差とします。上記の仮定から、 β_{n+1} が同じ系統誤差モデルから生じると仮定すると、次のようにになります：

$$\hat{\theta}_{n+1} \sim N(\theta_{n+1} + \hat{a} + \hat{b} \times \theta_{n+1}, \hat{c} + \hat{d} \times |\theta_{n+1}| + \hat{\tau}_{n+1}^2).$$

この式を θ_{n+1} について解くことで、キャリブレーションされた 95% 信頼区間の下限を求めます：

$$\Phi \left(\frac{\theta_{n+1} + \hat{a} + \hat{b} \times \theta_{n+1} - \hat{\theta}_{n+1}}{\sqrt{(\hat{c} + \hat{d} \times |\theta_{n+1}|) + \hat{\tau}_{n+1}^2}} \right) = 0.025,$$

ここで、 $\Phi(\cdot)$ は標準正規分布の累積分布関数を表します。確率 0.975 についても同様に上限を求めます。確率 0.5 を用いてキャリブレーションされた点推定値を定義します。

p 値 キャリブレーションと信頼区間のキャリブレーションの両方が EmpiricalCalibration パッケージで実装されています。

18.2.6 医療機関をまたいだ複製

別のある方法検証の形として、異なる母集団、異なる医療システム、および/または異なるデータ収集プロセスを表す複数の異なるデータベースで研究を実施することが挙げられます。これまでの研究では、異なるデータベースで同じ研究デザインを実施すると、効果量の推定値が大幅に異なることが示されています (Madigan et al., 2013b)。これは、異なる母集団では効果が大幅に異なるか、または異なるデータベースで見られる異なるバイアスを研究デザインが適切に考慮していないことを示唆しています。実際、信頼区間の経験的キャリブレーションによりデータベース内の残差交絡を考慮することで、研究間の異質性を大幅に低減できることがわかっています (Schuemie et al., 2018a)。

データベース間の異質性を表現する一つの方法として、 I^2 スコアがあります。これは、偶然ではなく異質性に起因する研究間の総変動の割合を表します

(Higgins et al., 2003)。 I^2 の値を単純に分類することは、すべての状況に適切であるとはいえないが、25%、50%、75% の I^2 値にそれぞれ「低」、「中程度」、「高」という形容詞を仮に割り当てることはできます。大規模傾向スコア調整を用いた新規ユーザーコホートデザインを使用して、多くのうつ病治療の効果を推定した研究 (Schuemie et al., 2018b) では、推定値の 58%のみが I^2 が 25%未満であることが観察され、実証的なキャリブレーション後、この値は 83%に増加しました。



データベース間の異質性を観察すると、推定値の妥当性に疑問が生じます。残念ながら、その逆は当てはまりません。異質性が観察されないからといって、偏りのない推定値が保証されるわけではありません。すべてのデータベースが同様の偏りを共有し、したがってすべての推定値が一貫して誤っている可能性もあります。

18.2.7 感度分析

研究を計画する際には、不確実なデザイン上の選択肢がしばしば存在します。例えば、層化傾向スコアマッチングを用いるべきでしょうか？層化を使用する場合、層をいくつに分けるべきでしょうか？適切なリスクにさらされた期間はどのくらいでしょうか？このような不確実性に直面した場合、解決策の一つは、さまざまな選択肢を評価し、デザイン上の選択肢に対する結果の感度を観察することです。さまざまな選択肢の下で推定値が同じままであれば、その研究は不確実性に強いと言えるでしょう。

この感度分析の定義は、例えば Rosenbaum (2005) が「さまざまな規模の隠れた偏りによって研究の結論がどのように変化するかを評価する」と定義している感度分析の定義と混同すべきではありません。

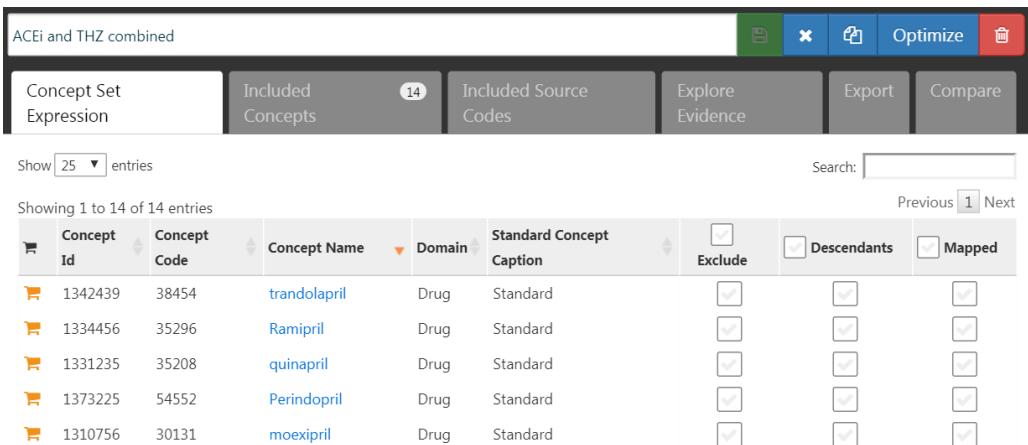
18.3 実践におけるメソッド検証

ここでは、第 12 章の例を基に、ACE 阻害薬 (ACEi) が血管性浮腫および急性心筋梗塞 (AMI) のリスクに及ぼす影響について、サイアザイド系およびサイアザイド系利尿薬 (THZ) と比較して調査します。その章では、すでに、用いたデザインであるコホート法に特有の診断の多くを調査しています。ここでは、他のデザインが使用されていた場合にも適用されていた可能性のある追加の診断を適用します。セクション 12.7 で説明されているように ATLAS を使用して研究が実施された場合、これらの診断には ATLAS によって生成された研究 R パッケージに含まれる Shiny アプリが利用できます。セクション 12.8 で説明されているように、代わりに R を使用して研究が実施された場合、次の部で説明されているように、さまざまなパッケージで利用可能な R 関数を使用する必要があります。

18.3.1 ネガティブコントロールの選択

私たちは、因果効果は存在しないと考えられるネガティブコントロール、すなわち曝露とアウトカムの組み合わせを選択しなければなりません。私たちの例の研究のような効果の比較の推定では、対象とする曝露も比較対照の曝露も原因ではないと考えられるネガティブコントロールの結果を選択します。私たちは、コントロールに表れるさまざまなバイアスを確実に反映させ、また実証的なキャリブレーションを可能にするために、十分な数のネガティブコントロールを必要とします。経験則として、通常は 50~100 件のネガティブコントロールを目標とします。これらのコントロールは完全に手作業で作成することも可能ですが、幸いにも ATLAS には文献、製品ラベル、自発報告からのデータを使用してネガティブコントロールを選択する機能が備わっています。

ネガティブコントロールの候補リストを作成するには、まず、関心のある曝露をすべて含むコンセプトセットを作成する必要があります。このケースでは、図 18.3 に示すように、ACEi および THZ クラスのすべての成分を選択します。



The screenshot shows the ATLAS interface with the title "ACEi and THZ combined". Below the title are several tabs: "Concept Set Expression" (selected), "Included Concepts" (14), "Included Source Codes", "Explore Evidence", "Export", and "Compare". There are also buttons for "Optimize" and a trash can icon.

Below the tabs, there is a search bar with "Show 25 entries" and a "Search:" field. A message indicates "Showing 1 to 14 of 14 entries". At the bottom right, there are "Previous" and "Next" buttons.

The main area displays a table with the following columns: Concept Id, Concept Code, Concept Name, Domain, Standard Concept Caption, Exclude, Descendants, and Mapped. The table contains the following data:

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: 対象とする曝露および比較対照の曝露を定義するコンセプトを含むコンセプトセット

次に、「Explore Evidence」タブに移動し、**▶ Generate** ボタンをクリックします。エビデンス概要の生成には数分かかりますが、その後、**View Evidence** ボタンをクリックできます。これにより、図 18.4 に示されるように、結果のリストが表示されます。

このリストには、条件のコンセプトと、その条件を私たちが定義した曝露のいずれかと関連付けるエビデンスの概要が示されます。例えば、さまざまな戦略を用いて PubMed で曝露とアウトカムを関連付ける文献の数、対象とする曝露の製品ラベルで、条件を可能性のある有害事象として記載しているものの数、自発報告の数を確認できます。デフォルトでは、候補となるネガティブコントロールが最初に表示されるようにリストがソートされます。次に、「ソート順」によってソートされ、これは観察データベースの集合におけるその状態の有病

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: SYNPNUF 5% ▾

Column visibility Copy CSV Show 15 ▾ entries Filter:

Showing 1 to 15 of 13,787 entries Previous 1 2 3 4 5 ... 920 Next

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 Count (Exact Concept Match)	Product Label (Parent Concept Match)	Product Label (Exact Concept Match)
Rift valley fever	Y	13,781	0	0	0	0	0	0	0
Obstruction due to foreign body accidentally left in operative wound AND/OR body cavity during a procedure	Y	13,780	0	0	0	0	0	0	0
Infection by Shigella	Y	13,766	0	0	0	0	0	0	0

▼ Suggested Negative Control
No (12777)
Yes (1010)
▼ Found in Publications
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)
Vec (Parent) (1949)

Figure 18.4: 文献、製品ラベル、および自発的な報告から見つかったエビデンスの概要を示すコントロール候補の結果

率を表します。ソート順が高いほど、有病率も高くなります。これらのデータベースにおける有病率は、調査を実施したいデータベースにおける有病率と一致しない可能性もありますが、近似値としては妥当であると考えられます。

次のステップは、候補リストを手動で確認することです。通常はリストの上から始め、最も頻度の高い条件から順に確認し、十分であると納得できるまで作業を続けます。この作業を行う一般的な方法としては、リストを CSV（カンマ区切り）ファイルにエクスポートし、セクション 18.2.1 で述べた基準を考慮しながら臨床医に確認してもらいます。

今回の研究例では、付録 C.1 にリストされた 76 のネガティブコントロールを選択します。

18.3.2 コントロールを含めること

ネガティブコントロールのセットを定義したら、それらを調査に含める必要があります。まず、ネガティブコントロール条件のコンセプトをアウトカムコホートに変換するためのロジックを定義する必要があります。セクション 12.7.3 では、ATLAS がユーザーが選択するいくつかのオプションに基づいて、そのようなコホートを作成する方法について説明しています。多くの場合、単にネガティブコントロールのコンセプトまたはその下位層のいずれかの出現に基づいてコホートを作成することを選択します。研究が R で実施される場合、SQL (構造化問い合わせ言語) を使用してネガティブコントロールコホートを構築することができます。第 9 章では、SQL および R を使用してコホートを作成する方

法について説明しています。適切な SQL および R を記述する方法については、読者の練習問題とします。

OHDSI ツールは、ネガティブコントロールから派生したポジティブコントロールを自動的に生成して含める機能も提供しています。この機能は、セクション 12.7.3 で説明されている ATLAS の「評価設定」セクションにあり、MethodEvaluation パッケージの `synthesizePositiveControls` 関数に実装されています。ここでは、生存モデルを使用して、真の効果サイズが 1.5、2、4 の 3 つのポジティブコントロールを各ネガティブコントロールに対して生成します。

```
library(MethodEvaluation)
# ターゲットの曝露 (ACEi = 1) のみを使用して、
# すべてのネガティブコントロールの曝露-アウトカムのペアを含むデータフレームを作成
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"))
```

注意すべきは、推定研究のデザインで使用されたリスク時間設定を模倣しなければならないということです。`synthesizePositiveControls` 関数は、曝露とネガティブコントロールのアウトカムに関する情報を抽出し、曝露とアウトカムの組み合わせごとにアウトカムモデルを適合させ、結果を統合します。ポジティブコントロールのアウトカムコホートは、`cohortDbSchema` および `cohortTable` で指定されたコホートテーブルに追加されます。結果の `pcs` データフレームには、統合されたポジティブコントロールの情報が含まれます。

次に、効果を推定するために使用したのと同じ研究を実行して、ネガティブコントロールとポジティブコントロールの効果も推定する必要があります。ATLAS の比較ダイアログでネガティブコントロールのセットを設定すると、ATLAS はこれらのコントロールの推定値を計算するように指示されます。同様に、評価設定でポジティブコントロールを生成するように指定すると、それらも分析に含まれます。R では、ネガティブコントロールとポジティブコントロールは、他の結果と同様に処理されます。OHDSI Methods Library のすべての推定パッケージは、多くの効果を効率的に推定することを容易に可能にします。

18.3.3 実証されたパフォーマンス

図 18.5 は、私たちの研究例に含まれているネガティブコントロールとポジティブコントロールについて、真の効果サイズ別に層別した推定効果サイズを示しています。このプロットは、ATLAS によって生成された研究 R パッケージに付属する Shiny アプリに含まれており、MethodEvaluation パッケージの plotControls 関数を使用して生成することができます。コントロールの数は、推定値を生成したりポジティブコントロールを統合したりするのに十分なデータがなかったため、定義された数よりも少ない場合が多いことに注意してください。

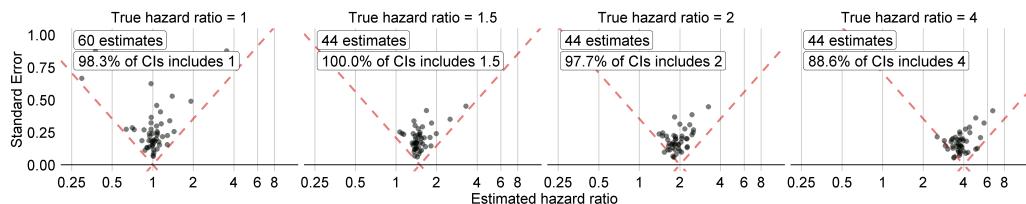


Figure 18.5: ネガティブコントロール（真のハザード比 = 1）およびポジティブコントロール（真のハザード比 > 1）に関する推定値。各点はコントロールを表します。点線の下にある推定値は、真の効果サイズを含まない信頼区間を持ちます。

これらの推定値をもとに、MethodEvaluation パッケージの computeMetrics 関数を使用して、表 18.1 に示すメトリクスを計算することができます。

Table 18.1: ネガティブコントロールとポジティブコントロールの推定値から得られたメソッドのパフォーマンスマトリクス

メトリクス	値
ROC 曲線下面積 (AUC)	0.96
カバー率	0.97
平均精度	19.33
平均二乗誤差 (MSE)	2.08
第 1 種の過誤	0.00

メトリクス	値
第 2 種の過誤	0.18
推定なし	0.08

カバー率と第 1 種の過誤は、それぞれ 95% と 5% という公称値に非常に近く、AUC も非常に高いことがわかります。これは必ずしもそうなるものではありません。

図 18.5 では、真のハザード比が 1 である場合の信頼区間すべてに 1 が含まれていませんが、表 18.1 の第 1 種の過誤は 0% です。これは例外的な状況であり、Cyclops パッケージの信頼区間が尤度プロファイリングを用いて推定されることによるものです。この方法は従来の方法よりも正確ですが、非対称な信頼区間になる可能性があります。一方、p 値は対称な信頼区間を仮定して計算され、これは第 1 種の過誤を計算する際に使用されたものです。

18.3.4 P 値のキャリブレーション

ネガティブコントロールの推定値を使用して、p 値を調整することができます。これは Shiny アプリでは自動的に行われ、R では手動で行うことができます。セクション 12.8.6 で説明したように、要約オブジェクト `summ` を作成したと仮定すると、経験的なキャリブレーション効果のプロットを作成することができます。

```
# 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)
```

図 18.6 では、陰影部分が破線で示された領域とほぼ完全に重なっていることがわかります。これは、ネガティブコントロールではほとんどバイアスが観察されなかったことを示しています。対象とするアウトカム (AMI) の 1 つは、破線と陰影部分の上にあり、未補正および補正後の p 値の両方において帰無仮説を棄却できないことを示しています。もう一方のアウトカム (血管性浮腫) は、明らかにネガティブコントロールから際立っており、未補正および補正済みの p 値がいずれも 0.05 未満である領域内に収まっています。

補正済みの p 値を計算することができます。

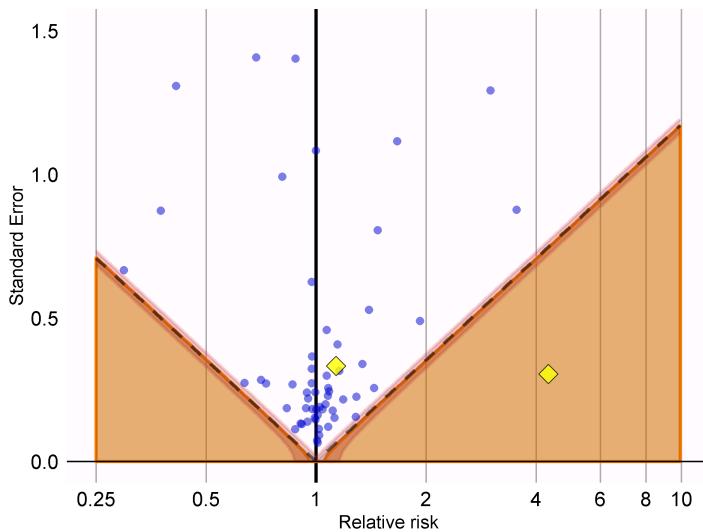


Figure 18.6: P 値のキャリブレーション：破線以下の推定値は従来の $p < 0.05$ 。網掛け部分の推定値はキャリブレーションされた $p < 0.05$ 。網掛け部分の端の狭い帯は 95% 信用区間。点はネガティブコントロール。ダイアモンド型は関心のあるアウトカムを示します。

```
null <- fitNull(logRr = ncEstimates$logRr,
                 seLogRr = ncEstimates$seLogRr)
calibrateP(null,
            logRr= oiEstimates$logRr,
            seLogRr = oiEstimates$seLogRr)
```

```
## [1] 1.604351e-06 7.159506e-01
```

そして、キャリブレーションされていない p 値と比較してみましょう。

```
oiEstimates$p
```

```
## [1] [1] 1.483652e-06 7.052822e-01
```

予想通り、バイアスはほとんど観察されなかったため、未補正および補正後の p 値は非常に類似しています。

18.3.5 信頼区間のキャリブレーション

同様に、ネガティブコントロールとポジティブコントロールの推定値を用いて、信頼区間をキャリブレーションすることができます。Shiny アプリは、キャリブレーションされた信頼区間を自動的に報告します。R では、EmpiricalCalibration パッケージの `fitSystematicModelError` および

`calibrateConfidenceInterval` 関数を使用して、信頼区間をキャリブレーションすることができます。詳細は、「信頼区間の経験的キャリブレーション」のヴィネットを参照してください。

キャリブレーション前の推定ハザード比（95% 信頼区間）は、血管性浮腫とAMIでそれぞれ 4.32 (2.45 - 8.08) と 1.13 (0.59 - 2.18) です。補正されたハザード比はそれぞれ 4.75 (2.52 - 9.04) および 1.15 (0.58 - 2.30) です。

18.3.6 データベース間の異質性

1つのデータベース（この場合は IBM MarketScan Medicaid (MDCD) データベース）で分析を実施したのと同様に、共通データモデル (CDM) に準拠する他のデータベースでも同じ分析コードを実行することができます。図 18.7 は、血管性浮腫の結果について、合計 5 つのデータベースにわたるフォレストプロットとメタ分析推定値（ランダム効果を仮定）(DerSimonian and Laird, 1986) を示しています。この図は、EvidenceSynthesisパッケージの `plotMetaAnalysisForest` 関数を使用して生成されました。

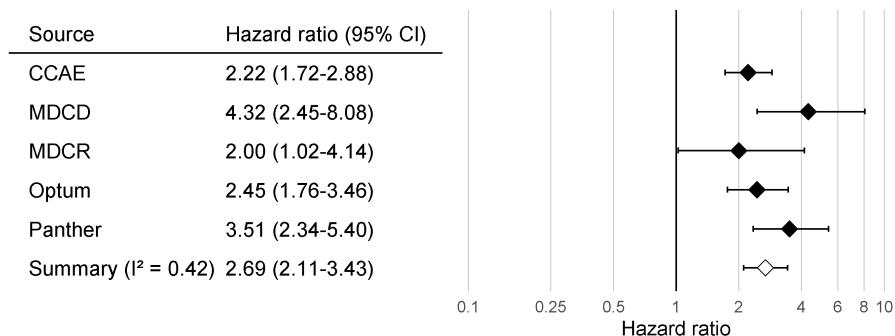


Figure 18.7: 血管浮腫のリスクについて ACE 阻害薬とサイアザイドおよびサイアザイド様利尿薬を比較した場合の、5 つの異なるデータベースからの効果推定値と 95% 信頼区間 (CI)、およびメタ解析による推定値

すべての信頼区間が 1 より大きいため、何らかの影響があるという点では一致していることが示唆されますが、 I^2 はデータベース間の異質性を示唆しています。しかし、図 18.8 で示したようにキャリブレーション済みの信頼区間を使用して I^2 を計算すると、この異質性は、ネガティブコントロールとポジティブコントロールを通じて各データベースで測定されたバイアスによって説明できることが分かります。経験的なキャリブレーションは、このバイアスを適切に考慮しているようです。

18.3.7 感度分析

分析におけるデザインの選択肢のひとつは、傾向スコアで変数比マッチングを使用することでした。しかし、傾向スコアで層別化を使用することも可能でした。この選択肢については不明な点があるため、両方を使用することにしまし

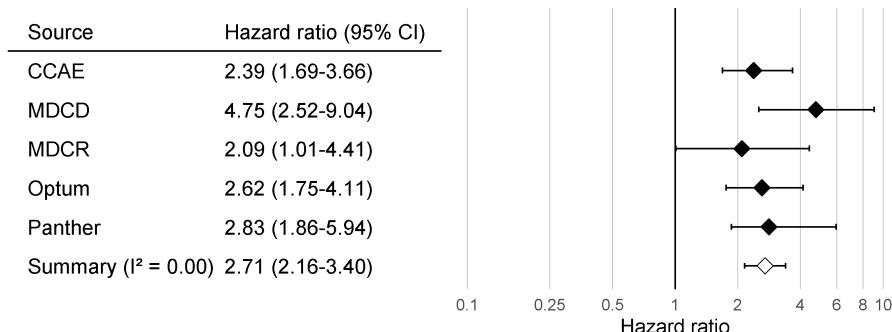


Figure 18.8: キャリブレーションされた血管浮腫のリスクについて ACE 阻害薬とサイアザイドおよびサイアザイド様利尿薬を比較した場合の、5つの異なるデータベースからの効果推定値と 95%信頼区間 (CI)、およびメタ解析による推定値

た。表 18.2 は、変数比マッチングと層別化（10 の均等サイズの層）を使用した場合の AMI と血管性浮腫に対する効果サイズ推定値を示している（キャリブレーションありとキャリブレーションなしの両方）。

Table 18.2: 2 つの分析におけるキャリブレーション前と
キャリブレーション後のハザード比 (95% 信頼区間)

アウトカム	調整方法	キャリブレーションなし	キャリブレーションあり
血管性浮腫	マッチング	4.32 (2.45 - 8.08)	4.75 (2.52 - 9.04)
血管性浮腫	層別化	4.57 (3.00 - 7.19)	4.52 (2.85 - 7.19)
急性心筋梗塞	マッチング	1.13 (0.59 - 2.18)	1.15 (0.58 - 2.30)
急性心筋梗塞	層別化	1.43 (1.02 - 2.06)	1.45 (1.03 - 2.06)

マッチングと層化分析による推定値は、強い一致を示しており、層化分析の信頼区間はマッチングの信頼区間に完全に収まっています。このことは、このデザイン選択に関する不確実性が推定値の妥当性に影響を与えないことを示唆しています。層化はより高い効果量（より狭い信頼区間）をもたらすように見えますが、これは驚くことではありません。マッチングではデータの損失が生じますが、層化では生じないためです。この代償として、層内の残差交絡によりバイアスが増加する可能性がありますが、キャリブレーションされた信頼区間にバイアスの増加が反映されているというエビデンスは見当たりません。



研究診断により、研究を完全に実施する前でもデザインの選択肢を評価することができます。すべての研究診断を生成し、確認してからプロトコルを確定しすることをお勧めします。P-ハッキング（望ましい結果を得るためにデザインを調整すること）を避けるため、対象となる効果量の推定値

をブラインドした状態で実施する必要があります。

18.4 OHDSI メソッド評価ベンチマーク

推薦される方法は、適用される文脈の中で、その手法のパフォーマンスを経験的に評価することですが、関心のある曝露とアウトカムのペア（例えば、同じ曝露または同じアウトカムを用いる）と研究で使用されるデータベースに類似したネガティブコントロールとポジティブコントロールを使用して、一般的な手法のパフォーマンスを評価することも価値があります。これが、OHDSI メソッド評価ベンチマークが開発された理由です。このベンチマークは、慢性または急性のアウトカム、長期または短期の曝露など、幅広い範囲のコントロール質問でパフォーマンスを評価します。このベンチマークの結果は、メソッドの全体的な有用性を実証するのに役立ち、コンテクスト固有に実証評価が利用できない（または利用できない）場合、メソッドのパフォーマンスに関する事前評価を形成するために用いることができます。このベンチマークは、慎重に選択された 200 件のネガティブコントロールから構成されており、これらは 8 つのカテゴリーに層別することができます、各カテゴリーのコントロールは同じ曝露または同じアウトカムを共有しています。これらの 200 件のネガティブコントロールから、セクション 18.2.2 で説明されているように、600 件の合成されたポジティブコントロールが導かれます。手法を評価するには、すべてのコントロールについて効果量の推定値を生成するためにその手法を用いる必要があり、その後に、セクション 18.2.3 で説明されている評価基準を計算することができます。ベンチマークは公開されており、MethodEvaluation パッケージの「OHDSI Methods Benchmark vignette」で説明されているように展開することができます。

私たちは、OHDSI Methods Library に収められているすべての手法をこのベンチマークで実行し、手法ごとにさまざまな分析を選択しました。例えば、コホート法は傾向スコアマッチング、層化、重み付けを用いて評価されました。この実験は、4 つの大規模な医療観察データベース上で実行されました。オンラインの Shiny アプリで閲覧できる結果¹によると、いくつかの手法では AUC（ポジティブコントロールとネガティブコントロールを区別する能力）が高い値を示しているものの、ほとんどの手法では図 18.9 に示されているように、第 1 種の過誤が高く、95% 信頼区間のカバー率が低いことが示されています。

このことは、経験的評価とキャリブレーションの必要性を強調しています。経験的評価が実施されていない場合、これはほぼすべての公表された観察研究に当てはまりますが、私たちは図 18.9 の結果から事前に情報を得て、真の効果の大きさが 95% 信頼区間には含まれていない可能性が高いと結論づけなければなりません。

また、Methods Library におけるデザインの評価では、経験則に基づくキャリ

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

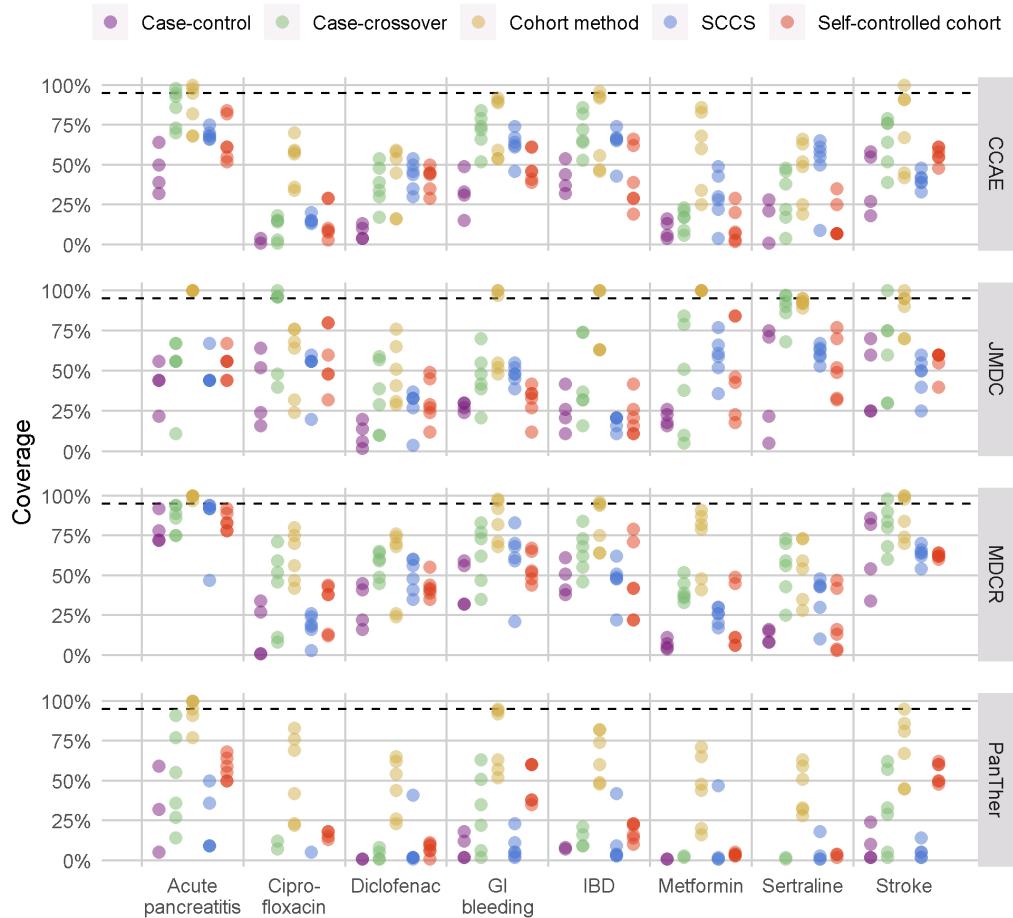


Figure 18.9: Methods Library の Methods に対する 95% 信頼区間のカバー率。各ドットは分析選択の特定セットの性能を表します。点線は名目上の性能(95% カバー率)を示します。SCCS = 自己対照症ケースシリーズ、GI = 消化管、IBD = 炎症性腸疾患。

ブレーションによって第 1 種の過誤と信頼区間が名目上の値に回復することが示されていますが、多くの場合、第 2 種の過誤が増し、精度が低下するという代償を伴います。

18.5 まとめ



- 手法の妥当性は、その手法の前提条件が満たされているかどうかによって決まります。
- 可能な場合、これらの前提条件は研究診断を用いて経験的に検証されるべきです。

- コントロール仮説、すなわち答えが既知の質問は、特定の研究デザインが真と一致する回答を生み出すかどうかを評価するために使用されるべきです。
- 多くの場合、p 値や信頼区間は、コントロール仮説を用いて測定された名目上の特性を示しません。
- これらの特性は、経験的なキャリブレーションによって多くの場合、名目上の特性に復元することができます。
- 研究診断は、研究者が p-hacking を回避するために関心のある効果に対して盲検性を維持する限り、分析デザインの選択を導き、プロトコルを適応させるために使用することができます。

第 V 部

OHDSI 研究

第 19 章

研究の段階

著者: Sara Dempster & Martijn Schuemie

ここでは、OHDSI ツールを用いた観察研究のデザインと実施に関する一般的な段階的ガイドを提供することを目的としています。研究プロセスの各段階を個別に説明し、その後、一般的な手順を説明し、場合によっては、OHDSI の書の前の章で説明した主な研究タイプ (1) 特性評価、(2) 集団レベルの推定 (PLE)、(3) 患者レベルの予測 (PLP) の特定の側面について議論します。そのため、前章で取り上げた多くの要素を、初心者にも理解できる形で統合します。同時に、この章は、必要に応じて他の章でより詳細な資料をさらに深く追求するオプションとともに、実践的な高度な説明を求める読者にとって、独立した章としても利用できます。最後に、いくつかの重要な例を随所で示します。

さらに、OHDSI コミュニティが推奨する観察研究のためのガイドラインとベストプラクティスを要約します。ここで取り上げる原則のいくつかは、観察研究のための他の多くのガイドラインにも見られるベストプラクティス推奨事項と共通する一般的なものです。一方、他の推奨プロセスは、OHDSI フレームワークにより特化したものです。したがって、OHDSI ツールスタックによって可能となる OHDSI 固有のアプローチを強調します。

本章では、OHDSI ツール、R、SQL のインフラが読者にとって利用可能であることを前提としているため、本章ではこのインフラのセットアップに関する側面については一切説明しません（第 8 章）および第 9 章を参照）。また、読者は OMOP CDM のデータベースを使用して、主に自身の施設でデータを対象とした研究を実施することに関心を持っているものと想定しています（OMOP ETL については第 6 章を参照）。ただし、以下で説明するように研究パッケージが準備されれば、原則として他の施設に配布して実施することも可能であることを強調しておきます。OHDSI ネットワーク研究の実施に関する追加的な考慮事項（組織および技術的な詳細を含む）については、第 20 章で詳しく説明します。

19.1 一般的なベストプラクティスガイドライン

19.1.1 観察研究の定義

観察研究とは、定義上、患者は単に観察されるだけで、特定の患者の治療に介入する試みは行われない研究である。時には、レジストリ研究のように特定の目的のために観察データが収集されることもあるが、多くの場合、これらのデータは、特定の研究課題以外の何らかの目的のために収集される。後者のタイプのデータとしてよく見られる例としては、電子的健康記録（EHR）や保険請求データなどがある。観察研究は、データの二次利用とよく呼ばれます。観察研究を実施する際の基本的な指針は、研究の疑問を明確に説明し、研究実施前にアプローチを完全に特定することです。この点において、観察研究は臨床試験と変わりません。ただし、臨床試験では、特定の疑問に対する答えを主目的として、通常は治療介入の有効性および/または安全性に関する疑問について、患者を募集し、追跡調査します。観察研究で採用される分析方法が臨床試験で使用されるものとは異なる点は数多くあります。とりわけ、PLE の観察研究では無作為化が行われないため、因果関係の推論を行うことを目的とする場合は交絡因子を制御するアプローチが必要となります（OHDSI が支援する PLE の研究デザインや多くの特性について集団をバランスさせることにより観察された交絡因子を排除する方法など、詳細については、第 12 章および第 18 章を参照ください。

19.1.2 事前に規定した研究デザイン

観察研究のデザインとパラメータの事前規定は、望ましい結果を得るために無意識または意識的にアプローチを進化させて、さらなるバイアスをもたらすこと为了避免るために極めて重要です。この傾向は、p-ハッキングと呼ばれることもあります。EHR や保険請求データなどのデータは、時に研究者に無限の可能性を感じさせ、調査の方向性を迷走させることができます。そのため、データの二次利用では、一次利用よりも研究の詳細を事前に完全に規定しない誘惑が強くなります。したがって、既存のデータが容易に入手できるように見える場合でも、科学的な調査の厳格な構造を維持することが重要となります。事前規定の原則は、最終的に臨床実践や規制上の決定に影響を与える可能性があるため、厳格な結果や再現可能な結果を確保する上で、PLE や PLP において特に重要です。探索的な理由のみで実施される特性評価研究の場合でも、詳細に規定された計画を策定することが望ましいです。そうでなければ、進化する研究デザインや分析プロセスを文書化、説明、再現することが困難になります。

19.1.3 プロトコル

観察研究計画は、研究実施前に作成されるプロトコルという形式で文書化されるべきです。少なくとも、プロトコルには、主要な研究課題、アプローチ、およびその課題に対する回答に用いられる評価基準が記載されます。研究対象集団は、他の研究者が完全に再現できる程度に詳細に記述されるべきです。さら

に、すべての方法または統計手順、評価基準、表、グラフなどの予想される研究結果の形式が記述されるべきです。プロトコルには、研究の実行可能性や統計的パワーを評価するための事前分析のセットが記載されることがよくあります。さらに、プロトコルには、感度分析と呼ばれる主要な研究課題のバリエーションに関する記述が含まれる場合もあります。感度分析は、研究デザインの選択が研究結果全体に及ぼす潜在的な影響を評価するために設計されており、可能な限り事前に記述されるべきです。時には、プロトコルが完了した後で、予期せぬ問題が発生し、プロトコルの修正が必要になる場合があります。このような事態が生じた場合、プロトコル自体に変更内容と変更理由を記録することが極めて重要です。特に、PLE または PLP の場合、完成した研究プロトコルは、独立したプラットフォーム (clinicaltrials.gov や OHDSI の studyProtocols sandbox など) に記録することが理想的です。そうすれば、そのバージョンや修正をタイムスタンプ付きで個別に追跡することができます。また、研究実施前に、機関またはデータソースの所有者がプロトコルの確認と承認を行う機会を必要とする場合もよくあります。

19.1.4 標準化された分析

OHDSI のユニークな利点は、観察研究で繰り返し尋ねられる質問（第 2、7、11、12、13 章）は、実際にはいくつかの主要なカテゴリーに分類できることを認識し、繰り返される側面を自動化することでプロトコル開発と研究実施プロセスを合理化するツールのサポート方法にあります。多くのツールは、遭遇するであろう大半の使用事例に対応する少数の研究デザインまたは評価基準をパラメータ化するように設計されています。例えば、研究者は研究対象集団と少数の追加パラメータを指定し、異なる薬剤および/またはアウトカムについて反復的に多数の比較研究を実施します。研究者の質問が一般的なテンプレートに当てはまる場合、研究対象集団やプロトコルに必要なその他のパラメータの基本的な記述の多くを自動生成する方法があります。歴史的に、これらのアプローチは OMOP 実験から着想を得たもので、多くの異なる研究デザインやパラメータを反復することで、観察研究デザインが薬剤と有害事象の既知の因果関係をどの程度再現できるかを評価しようとするものです。

OHDSI のアプローチは、これらのステップを共通の枠組みとツール内で比較的簡単に実行できるようにすることで、プロトコル内に実行可能性と研究診断を含めることをサポートします（下記セクション 19.2.4 を参照）。

19.1.5 研究パッケージ

標準化されたテンプレートやデザインのもう一つの利点は、研究者がプロトコルの形で研究が完全に詳細に記述されていると考えていても、研究を実行するための完全なコンピュータコードを生成するには、実際には十分に指定されていない要素があるかもしれないということです。OHDSI フレームワークによって可能になる関連の基本原則は、コンピュータコードの形で文書化された、完全に追跡可能で再現可能なプロセスを生成することであり、これはしばしば「研究パッケージ」と呼ばれます。OHDSI のベストプラクティスは、このような

スタディパッケージを git 環境で記録することです。このスタディパッケージには、コードベースのすべてのパラメータとバージョンスタンプが含まれています。前述の通り、観察研究は公衆衛生の決定や政策に影響を与える可能性のある質問を投げかけることが多いものです。したがって、調査結果にもとづいて行動を起こす前に、異なる研究者によって複数の環境で調査を再現することが理想的です。このような目標を達成する唯一の方法は、調査を完全に再現するために必要なすべての詳細を明確にマッピングし、推測や誤解釈に委ねないことです。このベストプラクティスをサポートするために、OHDSI ツールは、書面による文書形式のプロトコルをコンピュータまたは機械で読み取り可能な調査パッケージに変換するのを支援するように設計されています。このフレームワークのトレードオフとして、すべてのユースケースやカスタマイズされた分析が既存の OHDSI ツールで容易に処理できるわけではないという点が挙げられます。しかし、コミュニティが成長し進化するにつれ、より幅広いユースケースに対応する機能が追加されています。コミュニティに関わる人であれば誰でも、新しいユースケースに基づく新しい機能についての提案を行うことができます。

19.1.6 CDM に基づくデータ

OHDSI の研究は、観察データベースが OMOP 共通データモデル（CDM）に変換されることを前提としています。OHDSI のすべてのツールや下流の解析ステップでは、データ表現が CDM の仕様を満たしているという前提を置いています（第 4 章を参照）。したがって、この前提を満たすための ETL（抽出-変換-読込）プロセス（第 6 章を参照）が、特定のデータソースについて十分に文書化されていることも重要です。このプロセスは、アーティファクトや異なるサイト間のデータベース間の差異をもたらす可能性があるためです。OMOP CDM の目的は、サイト固有のデータ表現を減らす方向に向かうことですが、これは完璧なプロセスからはほど遠く、コミュニティが改善を求めている困難な領域です。したがって、研究を実施する際には、自施設またはネットワーク研究を実施する際には外部施設において、OMOP CDM に変換されたソースデータに精通している人々と協力することが重要です。

CDM に加えて、OMOP 標準化ボキャブラリシステム（第 5 章）も、OHDSI フレームワークを使用してさまざまなデータソース間の相互運用性を実現する上で重要な要素です。標準化ボキャブラリは、他のすべてのソースボキャブラリシステムがマッピングされる各ボキャブラリドメイン内の標準コンセプトセットを定義しようとするものです。これにより、薬剤、コンディション、プロシージャ（処置）について異なるソースボキャブラリシステムを持つ 2 つの異なるデータベースでも、CDM に変換されると比較可能になります。OMOP のボキャブラリには、特定のコホート定義に適切なコードを特定する際に役立つ階層構造も含まれています。繰り返しになりますが、OMOP CDM へのデータベースの電子タグ付け（ETLing）と OMOP ボキャブラリの使用によるメリットを最大限に享受するためには、ボキャブラリのマッピングを実施し、下流のクエリで OMOP 標準化ボキャブラリのコードを使用することが推奨されます。

19.2 詳細な研究手順

19.2.1 問いを定義する

最初のステップは、研究の関心を、観察研究で対処できる正確な質問に変換することです。たとえば、あなたが臨床の糖尿病研究者で、2型糖尿病(T2DM)の患者に提供されるケアの質を調査したいとします。この大きな目的を、第7章で最初に説明した3つのタイプの質問のいずれかに該当する、はるかに具体的な質問に分解できます。

特性評価研究では、「特定の医療環境において、軽度のT2DM患者と重度のT2DM患者に対する処方方法は、現在推奨されている内容に適合しているか?」という問い合わせることができます。この問いは、特定の治療法の有効性について、他の治療法との相対的な因果関係を問うものではありません。これは、既存の一連の臨床ガイドラインと比較して、データベース内の処方慣行を単に特性評価するものです。

また、T2DM治療の処方ガイドラインが、T2DMと心臓病の両方を患っている患者のような特定の患者群にとって最適であるかどうかについても疑問に思うかもしれません。このような疑問は、PLE研究に置き換えることができます。具体的には、心不全などの心血管系イベントの予防における2種類の異なるT2DM治療薬の比較効果の問い合わせることができます。異なる薬剤を投与されている2つのコホートで心不全による入院の相対リスクを調べる研究を計画することもできますが、両方のコホートはT2DMと心臓病の診断を受けているものとします。

あるいは、軽度のT2DMから重度のT2DMへと進行する患者を予測するモデルを開発することもできます。これはPLPの問い合わせとして設定でき、重度のT2DMへの移行リスクが高い患者を特定して、より注意深いケアを行うのに役立ちます。

純粹に実用的な観点から、研究の問い合わせを定義するには、問い合わせに答えるために必要なアプローチがOHDSIツールセット内の利用可能な機能に適合するかどうかを評価する必要があります(現在のツールで対応可能な質問タイプの詳しい説明については、第7章を参照ください)。もちろん、独自の分析ツールを設計したり、現在利用可能なツールを修正して、他の質問に答えることも常に可能です。

19.2.2 データの可用性と質を確認する

特定の研究課題に着手する前に、データの質を確認し(第15章を参照)、どのフィールドにデータが入力され、そのデータがどのケア設定をカバーしているかという観点から、特定の医療観察データベースの性質を十分に理解することが推奨されます。これにより、特定のデータベースでは研究課題が実行不可能となる可能性がある問題を迅速に特定することができます。以下では、発生する可能性のある一般的な問題をいくつか指摘します。

軽度の T2DM から重度の T2DM への進行を予測するモデルを開発するという、上記の例に戻りましょう。理想的には、T2DM の重症度は、患者の血糖値を過去 3 ヶ月間の平均値で反映する検査室での測定値であるグリコヘモグロビン (HbA1c) 値を検査することで評価できるでしょう。これらの値は、すべての患者で利用できるとは限らないでしょう。患者の一部でも利用できない場合は、T2DM の重症度に関する他の臨床的基準を特定し、代わりに使用できるかどうかを検討する必要があります。あるいは、HbA1c 値が患者の一部のみで利用できる場合、この一部のみに焦点を当てることで、研究に望ましくないバイアスが生じないかどうかを評価する必要もあります。欠損データの追加的な議論については、第 7 章を参照ください。

もう一つの一般的な問題は、特定のケア環境に関する情報の不足です。上述の PLE の例では、推奨された結果は心不全による入院でした。特定のデータベースに入院患者の情報が全く含まれていない場合、異なる 2 型糖尿病治療アプローチの比較効果をみるには、別の結果を考慮する必要があるかもしれません。他のデータベースでは外来患者の診断データが利用できない場合もあり、その場合はコホート設計を考慮する必要があります。

19.2.3 研究対象集団

研究対象集団または対象集団を定義することは、あらゆる研究における基本的なステップです。観察研究では、関心のある研究対象集団を代表するグループをコホートと呼んでいます。コホートに選択されるために必要な患者特性は、臨床的疑問に関連する研究対象集団によって決定されます。単純なコホートの例としては、18 歳以上の患者で、かつ医療記録に T2DM の診断コードがある患者が挙げられます。このコホート定義には、AND 論理で結合された 2 つの基準があります。多くの場合、コホート定義には、より複雑な入れ子状の布尔論理や、特定の研究期間や患者のベースライン期間に必要な期間などの追加の時間的基準で結合された、さらに多くの基準が含まれています。

洗練されたコホート定義を作成するには、適切な科学文献のレビューと、特定のデータベースの解釈における課題を理解している臨床および技術の専門家からの助言が必要となります。観察データを使用する際には、これらのデータは患者の病歴の完全な全体像を提供するものではなく、むしろ、情報の記録時に生じるヒューマンエラーやバイアスによって正確性が損なわれる可能性のある、ある時点におけるスナップショットであることを念頭に置くことが重要です。特定の患者は、観察期間と呼ばれる限られた期間のみ追跡されることがあります。特定のデータベースやケア環境、研究対象の疾患や治療について、臨床研究者は最も一般的なエラーの原因を回避するための提案を行うことができます。わかりやすい例を挙げると、T2DM 患者の特定における一般的な問題として、T1DM 患者が T2DM の診断コードで誤って分類されることがあります。T1DM 患者は基本的に異なるグループであるため、T2DM 患者を調査対象とする研究に T1DM 患者グループが意図せず含まれてしまうと、結果が歪められてしまう可能性があります。T2DM コホートの定義を厳密に行うためには、T1DM 患者が誤って表現されることを避けるために、糖尿病治療としてインスリンの処方

のみを受けた患者を除外することが望ましいでしょう。しかし同時に、単に医療記録に T2DM の診断コードを持つすべての患者の特徴に興味がある場合もあるでしょう。この場合、誤って T1DM とコード化された患者を除外しようとして、さらに条件を追加することは適切ではないかもしれません。

調査対象集団の定義が説明されたら、OHDSI ツールの ATLAS は、関連するコホートを作成するのに適した出発点となります。ATLAS とコホート生成プロセスについては、第 8 および第 10 章で詳しく説明されています。簡単に説明すると、ATLAS は、詳細な包含基準を定義してコホートを生成するためのユーザーインターフェース (UI) を提供します。ATLAS でコホートが定義されると、ユーザーはプロトコルに組み込むために、その詳細な定義を人間が読める形式で直接エクスポートすることができます。何らかの理由で ATLAS インスタンスが観察医療データベースに接続されていない場合でも、ATLAS を使用してコホート定義を作成し、その基礎となる SQL コードを直接エクスポートして、SQL データベースサーバー上で個別に実行される研究パッケージに組み込むことができます。ATLAS を直接使用することが可能な場合は、ATLAS の使用をお勧めします。なぜなら、ATLAS は、コホート定義の SQL コード作成以上の利点を提供しているからです（下記参照）。最後に、まれに、ATLAS UI でコホート定義を実装できず、手動でカスタム SQL コードを必要とする場合もあります。

ATLAS UI では、多数の選択基準に基づくコホートの定義が可能です。コホートへの組入れとコホートからの離脱の基準、およびベースラインの基準は、OMOP CDM のあらゆるドメイン（コンディション、薬剤、プロシージャなど）に基づいて定義することができます。各ドメインには標準コードを指定する必要があります。さらに、これらのドメインに基づく論理フィルタ、および研究期間を定義する時間ベースのフィルタ、ベースラインの時間枠を ATLAS 内で定義することができます。ATLAS は、各基準のコードを選択する際に特に役立ちます。ATLAS には、コホート定義に必要なコードセットを構築するために使用できるボキャブラリブラウジング機能が組み込まれています。この機能は OMOP 標準ボキャブラリのみに依存しており、ボキャブラリ階層内のすべての下位層を含めるオプションがあります（第 5 章を参照）。この機能を使用するには、ETL プロセス（第 6 章参照）中にすべてのコードが標準コードに適切にマッピングされている必要があります。包含基準で使用する最適なコードセットが明確でない場合は、コホート定義で探索的分析を行うことが妥当である場合があります。あるいは、異なるコードセットを使用してコホートを定義するさまざまな可能性を考慮するために、より正式な感度分析を検討することもできます。

ATLAS がデータベースに接続するように適切に設定されている場合、定義されたコホートを生成するための SQL クエリは ATLAS 内で直接実行することができます。ATLAS は各コホートに一意の ID を自動的に割り当て、この ID は将来の使用のためにバックエンドデータベースでコホートを直接参照するのにも使用できます。コホートは ATLAS 内で直接使用して発生率調査を実行することもできますし、PLE または PLP 調査パッケージ内のコードを使用してバックエンドデータベースで直接指定することもできます。特定のコホートについて、ATLAS は、そのコホートに属する個人の患者 ID、インデックス日付、コホート離脱日のみを保存します。この情報があれば、特性評価、PLE または PLP 研究

に必要な患者の基本属性や共変量など、その他のすべての属性や共変量を導き出すのに十分です。

コホートが作成されると、患者の人口統計学的特性の要約と、最も頻繁に観察された薬剤や状態の頻度を、デフォルトで ATLAS 内で直接作成し、表示することができます。

実際には、ほとんどの研究では、複数のコホートまたは複数のコホートセットを指定し、それらをさまざまな方法で比較して新たな臨床的洞察を得る必要があります。PLE および PLP では、OHDSI ツールがこれらの複数のコホートを定義するための構造化されたフレームワークを提供します。例えば、PLE の比較効果研究では、通常、少なくとも 3 つのコホート、すなわち、対象コホート、比較対象、アウトカムコホートを定義します（第 12 章を参照）。さらに、完全な PLE の比較効果研究を実施するには、ネガティブコントロールアウトカムとポジティブコントロールアウトカムを持つコホートも必要となります。OHDSI ツールセットは、これらのネガティブコントロールおよびポジティブコントロールコホートの生成を迅速化し、場合によっては自動化する方法を提供します。この点については、第 18 章で詳しく説明しています。

最後に、研究のためのコホートを定義する際には、OHDSI コミュニティで進行中の、頑健で検証済みの表現型のライブラリを定義する作業が役立つ可能性があります。表現型とは、本質的にはエクスポート可能なコホート定義のことです。これらの既存のコホート定義のいずれかが研究に適している場合、ATLAS インスタンスに json ファイルをインポートすることで、正確な定義を取得することができます。

19.2.4 実施可能性と研究診断

コホートが定義され生成されたら、利用可能なデータソースで研究の実行可能性を検討するためのより正式なプロセスを実施し、その調査結果を最終的なプロトコルに要約することができます。研究の実行可能性の評価には、多くの探索的かつ時には反復的な活動が含まれる場合があります。ここでは、いくつかの一般的な側面について説明します。

この段階における主な活動は、生成したコホートが希望する臨床的特性と一致していることを確認し、予期せぬ特性があればそれを指摘するために、コホート内の特性の分布を徹底的に調査することです。上記の 2 型糖尿病の例に戻ると、この単純な 2 型糖尿病コホートを、受けた他のすべての診断の頻度を再検討して特徴づけることで、1 型糖尿病患者やその他の予期せぬ問題を抱える患者も捕捉しているという問題を指摘できる可能性があります。このようなステップを、コホート定義の臨床的妥当性の品質チェックとして、研究プロトコルに新たにコホートを特徴づける段階として組み込むことは、良い方法です。実施の観点では、この最初のステップを実行する最も簡単な方法は、ATLAS でコホートが作成された際にデフォルトで生成される可能性のあるコホートの人口統計および主要な薬剤や疾患を調査することでしょう。ATLAS 内で直接コホートを作成するオプションが利用できない場合は、手動で SQL を実行するか、R

の特徴量抽出パッケージを使用してコホートを特徴付けることができます。実際には、より大規模な PLE 研究や PLP 研究では、これらのステップを特徴量抽出ステップとともに研究パッケージに組み込むことができます。

PLE や PLP の実行可能性を評価するもう一つの一般的な重要なステップは、対象コホートと比較対照コホートにおけるコホート規模とアウトカムのカウントの評価です。ATLAS の発症率機能を使用してこれらのカウントを特定し、他の箇所で説明されているように、検出力計算を行うことができます。

PLE 研究に強く推奨されるもう一つのオプションは、傾向スコア (PS) のマッチング手順と関連する研究診断を完了し、対象グループと比較対照グループの集団間に十分な重複があることを確認することです。これらの手順については、第 12 章で詳しく説明されています。さらに、これらの最終的にマッチングされたコホートを使用して、統計的有効性を計算することができます。

OHDSI コミュニティでは、利用可能なサンプルサイズを考慮した最小検出相対リスク (MDRR) を報告した上で研究を実施し、その後に統計的有効性を検証するケースもあります。このアプローチは、多数のデータベースやサイトを対象とした自動化された研究をハイスクールペットで実施する場合に、より有用である可能性があります。このシナリオでは、任意のデータベースにおける研究の有効性は、事前フィルタリングよりもすべての分析が完了した後に検証する方が適切であると考えられます。

19.2.5 プロトコルと研究パッケージの最終化

これまでのすべてのステップの準備が完了したら、最終的なプロトコルをまとめます。このプロトコルには、詳細なコホートの定義と研究デザイン情報を含めるべきであり、ATLAS からエクスポートすることが理想的です。付録 D では、PLE 研究のフルプロトコルのサンプル目次を提供しています。この目次は、OHDSI github でも入手できます。このサンプルは、包括的なガイドおよびチェックリストとして提供していますが、一部のセクションは、お客様の研究には関連しない場合もあります。

図 19.1 に示されているように、人間が読める形式での最終的な研究プロトコルの作成は、最終的な研究パッケージに組み込まれるすべての機械可読の研究コードの準備と並行して行う必要があります。これらの後者のステップは、以下の図では研究実施と呼ばれています。これには、ATLAS からの最終的な研究パッケージのエクスポートや、必要に応じてカスタムコードを開発することが含まれます。

その後、完成したスタディパッケージを使用して、プロトコルに記載されている予備的な研究診断ステップのみを実行することができます。例えば、2つの治療法の比較効果をする新規ユーザーコホートを用いた PLE 研究の場合、予備的な研究診断ステップの実行には、コホートの作成、傾向スコアの作成、照合を行い、対象集団と比較対照集団が研究実施に十分な重複性を有していることを確認する必要があります。これが決定されると、照合された対象コホートと比較対照コホートをアウトカムコホートと交差させて検出力計算を行い、アウ

トカムのカウントを取得し、この計算結果をプロトコルに記載することができます。これらの研究診断の結果に基づいて、最終的な結果モデルの実行に進むかどうかを決定することができます。特性評価や PLP 研究では、この段階で同様のステップを完了する必要がある場合がありますが、ここではすべてのシナリオを概説するわけではありません。

重要なのは、この段階で最終的なプロトコルを臨床協力者や利害関係者に確認してもらうことを推奨します。

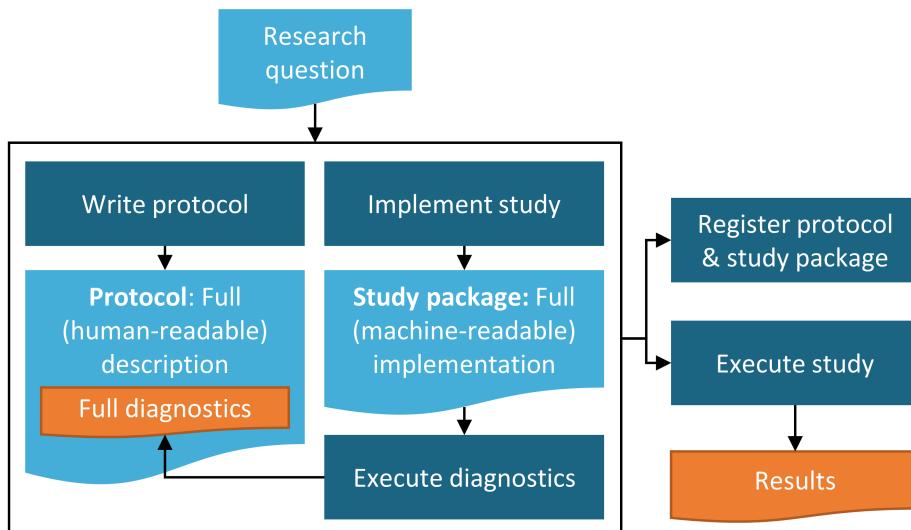


Figure 19.1: 研究プロセスのダイアグラム

19.2.6 研究の実施

これまでのすべてのステップが完了すれば、試験の実施は理想的には単純明快なはずです。もちろん、プロトコルに記載された方法やパラメータに忠実であるかを確認するために、コードやプロセスをレビューする必要があります。また、研究パッケージをテストおよびデバッグし、自身の環境で適切に実行されることを確認する必要があるかもしれません。

19.2.7 解釈と報告

サンプルサイズが十分で、データの質も妥当な、よく定義された研究では、結果の解釈は多くの場合、単純明快です。同様に、最終結果の記述以外の最終報告書の作成作業のほとんどはプロトコールの計画と作成時に完了しているため、報告書や出版用原稿の最終的な記述多くの場合、単純明快です。

しかし、解釈がより困難になり、慎重なアプローチが必要となる一般的な状況もいくつかあります。

1. サンプルサイズが有意性の境界線上にある場合、信頼区間が大きくなる

2. PLE に特異的な場合：ネガティブコントロールを用いた p 値のキャリブレーションにより、大幅なバイアスが明らかになる可能性がある
3. 研究実施中に予期せぬデータ品質の問題が明らかになる

どのような研究においても、上記の懸念事項について報告し、それに応じて研究結果の解釈を修正するかどうかは、研究者の裁量に委ねられます。プロトコル開発プロセスと同様に、最終報告書の発表や論文投稿の前に、臨床専門家や利害関係者による研究結果と解釈のレビューを行うことを推奨します。

19.3 まとめ



- 研究には明確に定義された問い合わせを検討すべきです。
- データの品質、完全性、関連性の事前チェックを適切に行いましょう。
- 可能であれば、プロトコルの開発プロセスにソースデータベースの専門家を含めることを推奨します。
- 事前にプロトコルに研究計画を記載します。
- 書面によるプロトコルと並行して研究パッケージコードを生成し、最終的な研究を実行する前に、実行可能性や研究診断を行い、記述しましょう。
- 研究は実施に先立ち登録し、必要に応じて承認を得るべきです。
- 最終報告書または論文の原稿は、臨床専門家やその他の関係者による査読を受けるべきです。

第 20 章

OHDSI ネットワーク研究

著者: Kristin Kostka, Greg Klebanov & Sara Dempster

OHDSI のミッションは、観察研究を通じて質の高いエビデンスを生み出すことです。このミッションを達成する主な方法は、共同研究です。これまでの章では、OHDSI コミュニティが質の高い再現可能な研究を促進するための標準やツールを作成していることを説明しました。これには、OMOP 標準化ボキャブラリ、共通データモデル (CDM)、分析方法パッケージ、ATLAS、レトロスペクティブなデータベース研究を実施するための研究ステップ (第 19 章) などが含まれます。OHDSI ネットワーク研究は、地理的に分散した多数のデータにわたって透明性、一貫性、再現性を備えた研究を実施する方法の集大成です。本章では、OHDSI ネットワーク研究の構成要素、ネットワーク研究の実施方法、ARACHNE 研究ネットワークなどの実現技術について説明します。

20.1 OHDSI 研究ネットワークとして

OHDSI 研究ネットワークは、医療における観測データ研究の進展を目指す研究者たちの国際的な協力体制です。現在、このネットワークは OMOP 共通データモデルに標準化された 100 以上のデータベースで構成されており、総計 10 億件以上の患者記録を網羅しています。OHDSI はオープンなネットワークであり、患者レベルのデータを有する世界中の医療機関が、データを OMOP CDM に変換し、ネットワークの研究に参加することで、このネットワークに参加することができます。データ変換が完了すると、協力者は、OHDSI プログラムマネージャーが管理するデータネットワークの一斉調査でサイト情報を報告するよう求められます。OHDSI ネットワークの各サイトは自動的に参加しています。強制的な義務はありません。各機関は、それぞれのネットワーク研究に参加するかどうかを選択できます。各研究では、データはファイアウォールの内側にあるサイト内に保管されます。ネットワークのサイト間で患者レベルのデータをプールすることはありません。共有されるのは集計結果のみです。



データ保持者が OHDSI ネットワークに参加するメリット

- 無料ツールへのアクセス: OHDSI は、データの特性評価や標準化された分析（臨床コンセプトのブラウジング、コホートの定義および特性評価、集団レベルの推定や患者レベルの予測研究の実行など）のための無料のオープンソースツールを公開しています。
- 一流の研究コミュニティへの参加: ネットワーク研究の作成と公開、さまざまな分野のリーダーや利害関係者グループとの共同作業。
- 医療のベンチマークの機会: ネットワーク研究により、データパートナー全体で臨床特性評価や品質改善のベンチマークが可能になります。

20.2 OHDSI ネットワーク研究

前章（第19章）では、CDM を使用した研究実施に関する一般的な設計上の考慮事項について説明しました。一般に、研究は単一の CDM または複数の CDM で実施されます。単一機関の CDM データ内、または多数の機関にわたって実施することができます。このセクションでは、複数の機関にわたって分析を拡大し、ネットワーク研究を実施する理由について説明します。

20.2.1 OHDSI ネットワーク研究を実施する動機

観察研究の典型的な使用例としては、「リアルワールド」の環境における治療の有効性の比較や安全性を調査することが挙げられます。より具体的には、臨床試験の結果の一般化可能性に関する懸念に対処するため、市販後の環境で臨床試験を再現することが目的となる場合があります。別のシナリオでは、ある治療法が適応外使用されているため、臨床試験では比較されたことのない 2 つの治療法を比較する研究を実施したい場合もあるでしょう。あるいは、臨床試験では観察するだけの力がなかった、市販後の稀な安全性アウトカムを調査する必要があるかもしれません。これらの研究課題に対処するには、施設で 1 つ、あるいは 2 つのデータベースで単一の観察研究を実施するだけでは不十分かもしれません。なぜなら、特定の患者グループのみに意味のある回答が得られるからです。観察研究の結果は、服薬アドヒアランス、遺伝的多様性、環境要因、全体的な健康状態など、データソースの場所によって異なる多くの要因に影響を受ける可能性があります。したがって、ネットワークで観察研究を実施する典型的な動機は、データソースと潜在的な研究対象者の多様性を高め、その結果がどの程度一般化できるかを理解することです。言い換えれば、研究結果は複数の施設で再現できるか、それとも異なるか、異なる場合、その理由について何らかの洞察が得られるか、ということです。したがって、ネットワーク研究は、幅広い設定とデータソースを調べることで、観察研究の結果に対する「リアルワールド」の要因の影響を調べる機会を提供します。

20.2.2 OHDSI ネットワーク研究の定義



どのような研究がネットワーク研究と見なされるのでしょうか？ OHDSI 研究は、異なる機関の複数の CDM で実施された場合に、OHDSI ネットワーク研究となります。

OHDSI のアプローチによるネットワーク研究では、OMOP CDM と標準化されたツールや研究パッケージを使用します。OHDSI の標準化された分析は、アティファクトを削減し、ネットワーク研究の効率性と拡張性を向上させることを目的として設計されています。

ネットワーク研究は、OHDSI 研究コミュニティの重要な一部です。しかし、OHDSI 研究をパッケージ化して OHDSI ネットワーク全体で共有することは義務付けられていません。単一の機関内で OMOP CDM および OHDSI Methods Library を使用して研究を実施したり、研究対象を一部の機関に限定したりすることも可能です。このような研究貢献もコミュニティにとって同様に重要です。研究を単一のデータベースで実行するように設計するか、限定されたパートナー間で研究を実施するか、OHDSI ネットワーク全体に研究への参加を呼びかけるかは、各研究者の裁量に委ねられます。本章では、OHDSI コミュニティが実施するオープンなネットワーク研究について説明します。

オープンな OHDSI ネットワーク研究の要素：OHDSI ネットワーク研究をオープンに実施する場合、完全に透明性の高い研究を行うことを約束することになります。OHDSI 研究を特徴づける要素はいくつかあります。これには以下が含まれます。

- すべての文書、研究コード、その後の結果は、OHDSI GitHub で一般公開されます。
- 研究者は、実施する分析の範囲と意図を詳細に記した公開研究プロトコルを作成し、公開しなければなりません。
- 研究者は、CDM に準拠したコードを含む研究パッケージ（通常は R または SQL）を作成しなければなりません。
- 研究者は、OHDSI ネットワーク研究の共同研究者を募るために、OHDSI コミュニティコールに参加することが推奨されます。
- 分析終了後、集計された研究結果は OHDSI GitHub で公開されます。
- 可能な場合は、研究者は研究 R Shiny アプリケーションを data.ohdsi.org に公開することが推奨されます。

次のセクションでは、独自のネットワーク研究を作成する方法と、ネットワーク研究を実施するための独自の設計や実務的な考慮事項について説明します。

20.2.3 OHDSI ネットワーク研究の設計上の考慮事項

OHDSI ネットワーク全体で実行する研究を設計するには、研究コードの設計や作成方法のパラダイムシフトが必要です。通常、研究では対象となるデータセットを念頭に置いて設計します。その際、分析に使用するデータについて真実であるとわかっている内容に関するコードを記述することができます。例えば、血管性浮腫コホートを構築する場合、CDM に表示されている血管性浮腫のコンセプトコードのみを選択することができます。ただし、データが特定のケア環境（例：プライマリケア、外来診療）や特定の地域（例：米国中心）のものである場合、これは問題となる可能性があります。コードの選択によって、コホートの定義が偏ってしまう可能性があるからです。

OHDSI ネットワーク研究では、もはや自分のデータのみを対象とした研究パッケージを設計・構築するわけではありません。世界中の複数の施設で実施する研究パッケージを構築することになります。自分の施設以外の参加施設の基礎データを見ることは決してありません。OHDSI ネットワーク研究では結果ファイルのみを共有します。研究パッケージで収集できるデータは、CDM のドメインで利用可能なもののみです。観察医療データが収集される医療環境の多様性を反映させるためには、コンセプトセットの作成に包括的なアプローチが必要となります。OHDSI 研究パッケージでは、全施設で同じコホート定義が使用されることがよくあります。つまり、ネットワーク内の適格なデータ（例えば、保険請求データ中心のデータや電子的健康記録（EHR）固有のデータ）のサブセットのみを表すようなバイアスのかかったコホート定義にならないよう、全体的な視点で考える必要があります。複数の CDM に移植可能な包括的なコホート定義を作成することが推奨されます。OHDSI 研究パッケージでは、すべての機関で同じパラメータ化されたコードセットを使用しています。データベース層への接続とローカルでの結果の保存のためのわずかなカスタマイズのみです。後ほど、多様なデータセットから臨床所見を解釈する際の影響について説明します。

臨床コーディングのばらつきに加えて、各地域の技術インフラのばらつきも想定して設計する必要があります。研究コードはもはや単一の技術環境で実行されるものではありません。各 OHDSI ネットワークサイトは、独自のデータベース層を選択します。つまり、特定のデータベース方言に研究パッケージをハードコードすることはできないということです。研究コードは、その方言の演算子に簡単に修正できる SQL の種類にパラメータ化する必要があります。幸い、OHDSI コミュニティには、ATLAS、DatabaseConnector、SqlRenderなどのソリューションがあり、異なるデータベース方言にわたって CDM 準拠のスタディパッケージを一般化するのに役立ちます。OHDSI の研究者は、他のネットワーク研究機関に協力を求め、異なる環境でスタディパッケージを実行できるかどうかをテストし、検証することが推奨されています。コーディングエラーが発生した場合は、OHDSI の研究者は OHDSI フォーラムを利用してパッケージの議論やデバッグを行うことができます。

20.2.4 OHDSI ネットワーク研究のためのロジスティクスに関する考察

OHDSI はオープンサイエンスのコミュニティであり、OHDSI 中央調整センターは、共同研究者がコミュニティ研究を主導し、参加することを可能にするコミュニティインフラストラクチャを提供しています。OHDSI ネットワーク研究には、必ず主任研究者を必要とし、OHDSI コミュニティ内の共同研究者であれば誰でもその任に就くことができます。OHDSI ネットワーク研究では、主任研究員、共同研究者、参加ネットワークデータパートナー間の調整が必要となります。各機関は、必要に応じて、研究プロトコルが承認され、ローカル CDM で実行することが許可されていることを確認するためのデューデリジェンスを独自に実施しなければなりません。データ分析者は、研究を実行するための適切な権限を付与するために、ローカル IT チームの支援を要請する必要があるかもしれません。各機関における研究チームの規模と範囲は、提案されたネットワーク研究の規模と複雑さ、OMOP CDM と OHDSI ツールスタックの当該サイトでの導入の成熟度によって決まります。OHDSI ネットワーク研究の実施経験の度合いも、必要な人員に影響します。

各研究において、機関での初期活動には以下が含まれます。

- 必要に応じて、研究を機関審査委員会（または同等の委員会）に登録する
- 必要に応じて、研究実施の承認を機関審査委員会から受ける
- 承認済みの CDM にスキーマを読み書きするためのデータベースレベルの権限を取得する
- 研究パッケージを実行するための機能的な RStudio 環境の構成を確認する
- 研究コードに技術的な異常がないか確認する
- 技術的な制約内でパッケージを実行するために必要な依存関係のある R パッケージを許可し、インストールするために、現地の IT チームと協力する



** データ品質とネットワーク調査：第 6 章で説明したように、品質管理は ETL（抽出-変換-読込）プロセスの基本かつ反復的な要素です。これはネットワーク調査プロセスとは別に定期的に行う必要があります。ネットワーク調査の場合、調査責任者は参加サイトのデータ品質レポートの確認を依頼したり、カスタム SQL クエリを設計して、データソースに潜在する変動要因を把握することができます。OHDSI で実施されているデータ品質の取り組みの詳細については、第 15 章を参照ください。

各機関には、研究パッケージを実行するローカルのデータアノリストが配置されます。この担当者は、研究パッケージのアウトプットをレビューし、機密情報が送信されていないことを確認する必要があります。CDM 内のデータはすべて匿名化されていますが、念のため確認します。集団レベルの効果推定 (PLE)

や患者レベルの予測 (PLP) などの構築済みの OHDSI メソッドを使用する場合、特定の分析に必要な最小セル数の設定が可能です。データアナリストは、これらの閾値を確認し、それが現地のガバナンス方針に従っていることを確認する必要があります。

データアナリストは、研究結果を共有する際には、結果の送信方法や結果の外部公開に関する承認プロセスの順守など、現地のガバナンス方針のすべてに従う必要があります。OHDSI ネットワーク研究では、患者レベルのデータは共有されません。言い換えれば、異なる施設から得られた患者レベルのデータが中央環境に集約されることはありません。研究パッケージは、集約結果（例：要約統計、点推定値、診断プロットなど）として設計された結果ファイルを作成しますが、患者レベルの情報を共有することはできません。多くの組織では、参加する研究チームのメンバー間でデータ共有契約を締結する必要はありません。しかし、関係する機関やデータソースによっては、特定の研究チームメンバー間でより正式なデータ共有契約を締結する必要がある場合もあります。ネットワーク研究への参加に関心のあるデータ所有者の方は、OHDSI コミュニティ研究に参加するために満たすべき要件について、現地の管理チームに相談することをお勧めします。

20.3 OHDSI ネットワーク研究の実行

OHDSI ネットワーク研究を実行するには、以下の三つの一般的な段階があります：

- 研究デザインと実行可能性
- 研究実行
- 結果の公表と発表

20.3.1 研究デザインと実現可能性

研究の実行可能性の段階（または事前研究段階）では、研究の問い合わせを定義し、研究プロトコルを通じてこの問い合わせに答えるためのプロセスを記述します。この段階では、参加施設全体で研究プロトコルを実行する実行可能性の評価に重点が置かれます。

実行可能性の評価段階の結果として、ネットワークでの実施に備えて公表された最終的なプロトコルと研究パッケージが作成されます。正式なプロトコルには、指定された研究責任者（多くの場合、論文発表の目的で連絡先著者となる）を含む研究チームの詳細と、研究のタイムラインに関する情報が記載されます。プロトコルは、追加のネットワーク機関が CDM データに関する研究パッケージ全体を検討、承認、実施するための重要な要素です。プロトコルには、研究対象集団、使用される方法、結果の保存および分析方法、さらに研究完了後の結果の公表方法（論文発表、学術会議での発表など）に関する情報を記載する必要があります。

実行可能性の段階は、明確に定義されたプロセスではありません。これは、提案された研究の種類に大きく依存する一連の活動です。少なくとも、研究責任者は、必要な薬剤曝露、プロシージャ（処置）情報、疾患状態または人口統計学的情報のある対象患者集団を含む関連ネットワーク機関を特定するために時間を費やします。可能であれば、研究責任者は、対象コホートを設計するため、暫定的に自身の CDM を使用すべきです。ただし、ネットワーク研究を実施するために、研究責任者が実際の患者データを含む OMOP CDM にアクセスできる必要はありません。研究責任者は、合成データ（例えば、CMS Synthetic Public Use Files、Mitre 社の SyntheticMass、Synthea）を使用して対象コホート定義を設計し、OHDSI ネットワークサイトの共同研究者に対して、このコホートの実行可能性の検証を依頼することができます。実行可能性の活動には、ATLAS からのコホート定義の JSON ファイルを使用してコホートを作成し特性を明らかにするよう共同研究者に依頼することや、研究用 R パッケージを検証すること、第 19 章で説明されている初期の研究診断を実行することが含まれます。同時に、必要に応じて、研究代表者は組織機関で OHDSI 研究を承認するための組織固有のプロセスを開始することもあります。すなわち、内部の機関審査委員会の承認などです。実行可能性の段階で、これらの組織固有の活動を完了させることは、研究代表者の責任です。

20.3.2 研究の実行

実行可能性の検討を完了した後、研究は実行段階に進みます。この期間は、OHDSI ネットワークの機関が分析への参加を選択できる期間です。この段階では、これまで検討してきた設計やロジスティクスの考慮事項が最も重要になります。

研究責任者が OHDSI コミュニティに働きかけて、OHDSI ネットワーク研究の新規の実施を正式に発表し、参加施設の募集を正式に開始した時点で、研究は実行段階に移行します。研究責任者は、OHDSI GitHub に研究プロトコルを公開します。研究責任者は、OHDSI コミュニティの週例電話会議や OHDSI フォーラムで研究を発表し、参加施設と共同研究者を募ります。参加を希望する施設が現れると、研究責任者は各施設と直接連絡を取り、研究プロトコルとコードが公開されている GitHub リポジトリの情報を提供するとともに、研究パッケージの実行方法に関する指示を行います。理想的には、ネットワーク研究はすべての施設で並行して実施されるため、最終結果は同時に共有され、どの施設のチームメンバーも他のチームの知見によってバイアスを受けることがないようにします。

各施設では、研究チームが研究参加の承認を得るための施設内手続きに従い、研究パッケージを実施し、外部に結果を共有するよう確認します。これには、特定のプロトコルに対する機関審査委員会による免除または承認、または同等の承認を得ることが含まれる可能性が高いです。研究実施が承認されると、施設のデータサイエンティストや統計学者は研究リーダーの指示に従って OHDSI 研究パッケージにアクセスし、OHDSI ガイドラインに従って標準化された形式で結果を生成します。各参加施設は、データ共有に関する規則について、内部

の機関プロセスに従うことになります。機関審査委員会またはその他の機関承認プロセスから承認または免除が得られない限り、施設は結果を共有してはなりません。

研究責任者は、結果の受け取り方法（SFTP または安全な Amazon S3 バケット経由など）と結果の提出期限を伝える責任を負います。施設は、送信方法が内部プロトコルに準拠していない場合、その旨を指定することができ、それに応じて回避策が開発される場合があります。

実行段階において、妥当な調整が必要な場合は、統合研究チーム（研究責任者および参加施設チームを含む）が結果を繰り返し検討することがあります。プロトコルの範囲や程度が承認された内容を超える形で発展した場合は、参加施設が自施設にその旨を伝え、研究責任者と協力してプロトコルを更新し、地域の機関審査委員会による審査と再承認のためにプロトコルを再提出することが求められます。

最終的には、研究責任者およびサポートするデータサイエンティスト/統計学者が、必要に応じて、施設間で結果を集約し、メタ分析を実施する責任を負うことになります。OHDSI コミュニティは、複数のネットワーク施設から共有された結果ファイルを単一の回答に集約するための検証済みの方法論を有しています。EvidenceSynthesisパッケージは、分散研究における複数のデータ施設など、複数のソースにわたるエビデンスと研究診断を結合するためのルーチンを含む、無料で利用可能な R パッケージです。これには、メタ分析とフォレストプロットを実行する機能も含まれています。

研究責任医師は、参加施設の状況を監視し、参加施設と定期的に連絡を取り合うことで、パッケージの実行の際の障害を排除する必要があります。研究の実施方法は、各施設で一律ではありません。データベース層（アクセス権／スキーマ権限など）や、その環境における分析ツール（必要なパッケージのインストール不可、R によるデータベースへのアクセス不可など）に関連する課題が生じる可能性があります。参加施設が主導権を握り、研究実施の障害となる問題を伝達します。現地の CDM で発生した問題の解決に役立つ適切なリソースを確保するかどうかは、最終的には参加施設の判断に委ねられます。

OHDSI 研究は迅速に実施できますが、すべての参加施設が研究を実施し、結果を公表するための適切な承認を得るために、妥当な期間を確保することが推奨されます。OHDSI ネットワークの新しい施設では、データベースの権限や分析ライブラリの更新などの環境設定の問題に対処する必要があるため、参加する最初のネットワーク研究が通常よりも長くなる可能性があります。OHDSI コミュニティからサポートを受けることができます。問題が発生した場合は、OHDSI フォーラムに投稿できます。

研究責任者はプロトコルに研究マイルストーンを設定し、事前に終了予定日を伝えて、研究全体のスケジュール管理を支援すべきです。スケジュールが遵守されない場合、研究責任者は参加施設に研究スケジュールの最新情報を伝え、研究実施の全体的な進捗を管理する責任があります。

20.3.3 結果の普及と公開

結果の公表と公開段階では、研究責任者は他の参加者と協力して、原稿の作成やデータ可視化の最適化など、さまざまな管理業務を行います。研究が実施され、結果が研究責任者によりさらに分析できるよう一元的に保存されたら、研究責任者は参加センターによるレビュー用に研究結果全体（例えば、Shiny Application）の作成と公表を行います。研究責任者が OHDSI 研究骨格（Atlas によって生成されたもの、または GitHub のコードを手動で修正したもの）を使用している場合、Shiny アプリケーションは自動的に作成されます。研究責任者がカスタムコードを作成している場合、研究責任者は OHDSI フォーラムを利用して、研究パッケージ用の Shiny アプリケーション作成の支援を求めるすることができます。



OHDSI ネットワーク研究をどこで発表するかお悩みですか？JANE（ジャーナル/著者名推定ツール）をご利用ください。このツールは、抄録を基に関連性と適合性のある出版物を検索します [^janeUrl]。

原稿が作成されると、参加している各共同研究者は、その成果物が外部での出版プロセスに従っていることを確認するため、その内容を審査することが推奨されます。少なくとも、参加している施設は出版を誰がリードするか指名すべきです。指名された人は、原稿の準備と提出の間に内部プロセスが遵守されていることを確認します。研究をどのジャーナルに投稿するかは研究リーダーの裁量に委ねられるますが、研究開始時の共同協議の結果であるべきです。OHDSI 研究の共著者は、ICMJE の著者の資格に関するガイドライン¹を満たすことが期待されています。結果の発表は、研究者が選択する任意のフォーラム（OHDSI シンポジウム、他の学術会議、学術誌など）で行うことができます。また、研究者は、毎週開催される OHDSI コミュニティの電話会議や世界各地で開催される OHDSI シンポジウムで OHDSI ネットワーク研究を発表することも推奨されています。

20.4 展望: ネットワーク研究の自動化を利用する

現在のネットワーク研究プロセスはマニュアルで行われており、研究チームのメンバーは、研究設計、コードや結果の共有において、さまざまなメカニズム（Wiki、GitHub、電子メールなど）を使用して共同作業を行っています。このプロセスは一貫性や拡張性に欠けるため、OHDSI コミュニティは研究プロセスのシステム化に積極的に取り組んでいます。

ARACHNE は、ネットワーク研究の実施プロセスを合理化し自動化するプラットフォームです。ARACHNE は OHDSI 標準を使用し、複数の組織にわたって一貫性があり、透明性が高く、安全で、コンプライアンスに準拠した観察研究プロセスを確立します。ARACHNE は、データのアクセスや解析結果の交換の

¹<http://www.icmje.org/recommendations/browse/roles-and-responsibilities/defining-the-role-of-authors-and-contributors.html>

Network Study Workflow

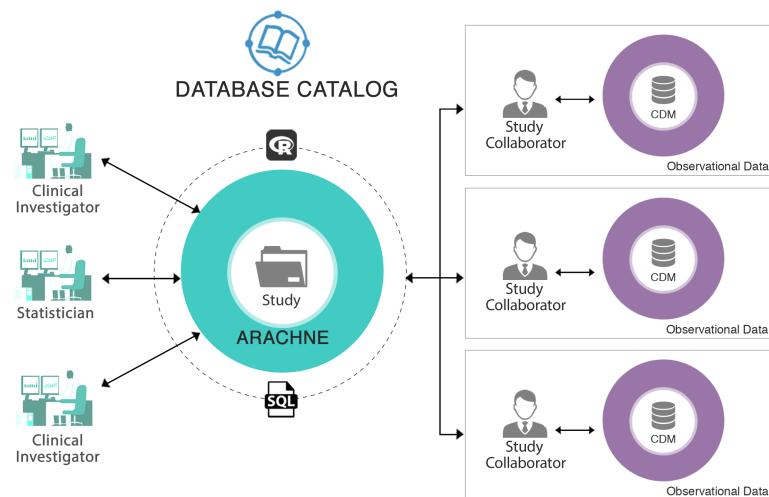


Figure 20.1: ARACHNE ネットワーク研究プロセス

ための通信プロトコルを標準化すると同時に、制限付きコンテンツの認証と承認を可能にします。データ提供者、研究員、スポンサー、データサイエンティストといった参加組織を単一の共同研究チームにまとめ、エンドツーエンドの観察研究の調整を促進します。このツールは、データ管理者によって管理される承認ワークフローを含む、完全な標準ベースの R、Python、SQL 実行環境の構築を可能にします。ARACHNE は、ACHILLES レポートや ATLAS 設計アーティファクトのインポート機能、自己完結型パッケージの作成、それらの複数サイトにわたる自動実行機能など、他の OHDSI ツールとのシームレスな統合を提供するように構築されています。将来的なビジョンは、最終的には複数のネットワークを相互にリンクし、単一ネットワーク内の組織間だけでなく、複数のネットワークにわたる組織間でも研究を実施できるようにすることです。

20.5 OHDSI ネットワーク研究のベストプラクティス

ネットワーク研究を実施する際には、OHDSI コミュニティが OHDSI ネットワーク研究のベストプラクティスを遵守できるよう支援いたします。

研究デザインと実現可能性：ネットワーク研究を実施する際には、研究デザインが単一のデータタイプに偏っていないことを確認ください。全施設で一貫した母集団を代表するコホート定義を調和させる作業は、データのタイプがどの程度異質であるか、また、各研究施設が OMOP CDM へのデータ変換に関する標準化された規定をどの程度厳密に遵守しているかによって、その難易度が異なります。この点が非常に重要な理由は、ネットワークの各施設間におけるデータ収集、表現、変換の相違を、臨床的に意味のある相違と区別する必要があるためです。特に、比較効果研究では、施設間で曝露コホートとアウトカム

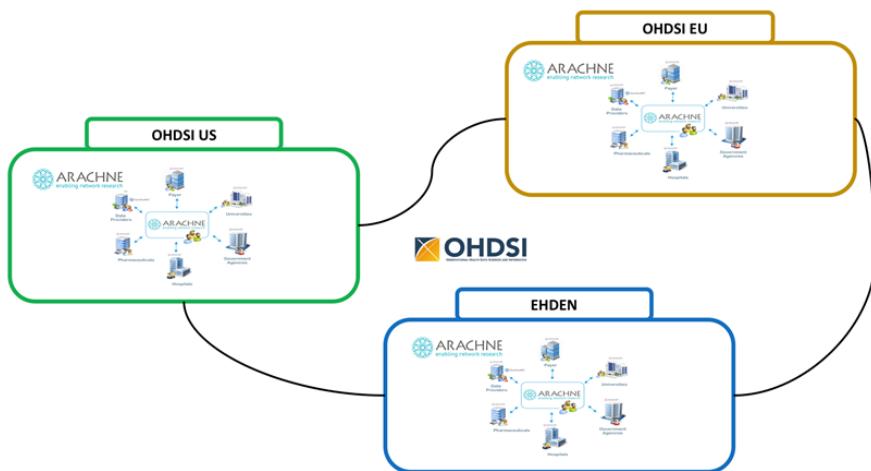


Figure 20.2: ARACHNE のネットワークのネットワークス

コホートの定義を一致させることが課題となります。例えば、薬物曝露情報はさまざまなデータソースから得られますが、誤分類の可能性はそれぞれ異なります。医療保険プランの薬局調剤請求は裁定される可能性があり、つまり、薬の請求がある場合、その人が処方箋を調剤された可能性が非常に高いことを意味します。しかし、EHR に入力された処方箋は、その処方箋が調剤されたか使用されたかを判断する他のデータとのリンクがない場合、その情報しか利用できない可能性があります。医師が処方箋を書いた記録と、薬剤師が処方箋を調剤した時間、患者が薬局で薬を受け取った時間、患者が実際に最初の薬を飲んだ時間との間に時間差がある可能性があります。この測定誤差は、あらゆる分析ユースケースの結果に潜在的にバイアスを生じさせる可能性があります。したがって、研究プロトコルの策定時には、データベース参加の妥当性を評価するための実行可能性を調査することが重要です。

研究の実行：可能な場合、研究責任者は、ATLAS、OHDSI Methods Library、OHDSI Study Skeletons を活用し、標準化された分析パッケージができるだけ多く使用した研究コードを作成することが推奨されます。研究コードは、常に OHDSI パッケージを使用した CDM 準拠のデータベース層に依存しない方法で作成すべきです。すべての関数と変数をパラメータ化すること（例：データベース接続、ローカルハードドライブパスを固定しない、特定のオペレーティングシステムを想定する）。参加施設を募集する際には、研究責任者は各ネットワーク施設が CDM に準拠しており、OMOP 標準ボキャブラリを定期的に更新していることを確認すべきです。研究責任者は、各ネットワーク施設が CDM のデータ品質チェックを実施し、文書化していることを確認するためのデューデリジェンスを行うべきです（例えば、ETL が THEMIS のビジネスルールと規約に従っていること、正しいデータが正しい CDM テーブルとフィールドに配置されていることなどを確認する）。各データアーリストは、研究パッケージを実行する前に、ローカルの R パッケージを OHDSI の最新パッケージバージョンに更新することが推奨されます。

結果と普及：研究責任者は、結果を共有する前に、各サイトがローカルのガバナンスルールに従っていることを確認すべきです。オープンで再現可能な科学とは、設計や実行されたすべてが利用可能になることを意味します。OHDSI ネットワーク研究は、すべての文書とその後の結果を OHDSI GitHub リポジトリまたは data.ohdsi.org R Shiny サーバーに公開し、完全に透明性のあるものとなっています。論文を作成する際には、研究責任者は OMOP CDM の原則と標準化されたボキャブラリを再確認し、OHDSI ネットワークの各サイト間でデータがどのように異なりうるかをジャーナルが理解できるようにする必要があります。例えば、保険請求データベースと EHR を使用するネットワーク研究を実施している場合、ジャーナルの査読者から、複数のデータタイプにわたってコホート定義の完全性をどのように維持したかを説明するように求められることがあります。査読者は、第 4 章で説明されている OMOP の観察期間が、適格性ファイル（保険請求データベースに存在し、ある人が保険事業者によって保険の対象となっている期間と対象外となっている期間を特定するファイル）と比較してどのようにになっているかを理解したいと思うかもしれません。これは、データベース自体のアーティファクト要素に焦点を当てるなどを本質的に求めているものであり、CDM がレコードをどのように観察に変換するかの ETL に焦点を当てています。この場合、ネットワーク研究のリーダーは、OMOP CDM の観察期間がどのように作成されるかを参照し、ソースシステム内のエンカウンターを使用して観察がどのように作成されるかを説明することが役立つかもしれません。原稿の考察では、EHR データには、対象期間中のすべての支払い済み受診を反映する保険請求データとは異なり、別の EHR を使用する医療従事者に受診した場合は記録されないという限界があることを認める必要があるかもしれません。これは、データがシステムに存在する方法に起因するものです。臨床的に意味のある差異ではありませんが、OMOP が観察期間表を導出する方法に不慣れな人にとっては混乱を招く可能性があります。この慣例を明確にするために、考察のセクションで説明することが望ましいでしょう。同様に、研究責任者は、OMOP 標準のボキャブラリが提供するボキャブラリサービスについて説明することが有用であると考えるかもしれません。このサービスにより、臨床コンセプトがどこで捕捉されたとしても同一に保たれるようになります。ソースコードを標準コンセプトにマッピングする際には常に何らかの決定が下されますが、THEMIS の規約や CDM の品質チェックは、情報がどこに送られるべきか、またデータベースがどの程度その原則に従っているかに関する情報を提供するのに役立ちます。

20.6 まとめ



- OHDSI 研究は、異なる機関の複数の CDM で実施されると、OHDSI ネットワーク研究となります。
- OHDSI ネットワーク研究は、すべての人に公開されています。ネットワーク研究のリーダーは誰でも構いません。OMOP 準拠のデータベースを保有している人は、参加して結果を寄与することができます

す。

- ネットワーク研究の実施でお困りですか？OHDSI 研究育成委員会に相談し、研究の設計と実施にお役立てください。
- 共有は思いやりです。すべての研究文書、コード、結果は、OHDSI GitHub または R Shiny アプリケーションで公開されています。研究責任者は、OHDSI イベントで研究発表を行うよう推奨されています。

第 A 章

用語集

ACHILLES データベースレベルの特性評価レポート。

ARACHNE 連携ネットワーク研究のオーケストレーションおよび実行を可能にするために開発されている OHDSI プラットフォーム。

ATLAS 患者レベルの臨床データからリアルワールドエビデンスを創生するための観察データの分析のデザインおよび実行を支援するために、参加施設にインストールされるウェブベースのアプリケーション。

バイアス (Bias) 誤差（真の値と推定値の差）の期待値。

ブーリアン変数 (Boolean) 2つの値（真または偽）のみを持つ変数。

医療施設 (Care site) 医療提供が実施される一意に識別された制度上（物理的または組織的）の単位（外来、病棟、病院、クリニックなど）。

症例対照（研究）(Case control) 集団レベルの効果推定のためのレトロスペクティブ研究デザインの一種。症例対照研究には、ターゲットとなるアウトカムを有する「症例」と有しない「対照」をマッチングさせる。過去に遡って、症例と対照における曝露のオッズを比較する。

因果効果 (Causal effect) 集団レベルの推定が関心を寄せるもの。「因果効果」を、ターゲットとする集団における「個人レベル因果効果」の平均とする定義もある。個人レベル因果効果は、個人が曝露された場合のアウトカムと、されなかった場合（または A に曝露された場合と B に曝露された場合）のアウトカムとの対比である。

特性評価 (Characterization) コホートまたは全データベースの記述的研究。詳細は第 11 章参照。

保険請求データ (Claims data) 医療保険会社への請求目的で作成されたデータ。

臨床試験 (Clinical trial) 介入臨床研究。

コホート (Cohort) ある期間内に、1つ以上の選択基準を満たす個人の集団。詳細は第10章参照。

コンセプト (Concept) 医学用語で定義された表現（コードが付随する）（例：SNOMED CT）。詳細は第5章参照。

コンセプトセット (Concept set) 様々な分析で再利用可能な構成要素として使用できるコンセプトのリストによる表現。詳細は第10章参照。

共通データモデル (Common Data Model, CDM) 分析のポータビリティ（同じ分析を変更なしで複数のデータセットで実行できる）を可能にするための医療データの表現規約。詳細は第7章参照。

比較効果 (Comparative Effectiveness) 関心のあるアウトカムに対する2つの異なる曝露の効果の比較。詳細は第12章参照。

コンディション (Condition) 医療従事者が観察したまたは患者が報告した診断、徵候、または症状。

交絡 (Confounding) 主たる関心の曝露が、アウトカムと関連する他の要因と混同されるときに発生する推定された関連性尺度の歪み（不正確さ）。

共変量 (Covariate) 独立変数として統計モデルで使用されるデータ要素（例：体重）。

データ品質 (Data quality) そのデータが特定の用途に適していると判断するためのデータの完全性、妥当性、一貫性、適時性、正確性の状態。

デバイス (Device) 化学作用を超えた機序によって診断または治療の目的で使用される異物または器具。デバイスには、植込み物（例：ペースメーカー、ステント、人工関節）、医療機器および補助材料（例：包帯、松葉杖、注射器）、医療処置で使用される他の器具（例：縫合糸、除細動器）および臨床ケアで使用される材料（例：接着剤、生体用材料、歯科材料、外科材料）が含まれる。

薬剤 (Drug) 人に投与されたときに特定の生理学的効果を発揮するように調製された生化学物質。薬剤には処方薬と市販薬、ワクチン、バイオ製剤が含まれる。局所的に摂取または適用する放射性デバイスは薬剤に含まない。

ドメイン (Domain) 共通データモデルのテーブルにおける標準化フィールドに対して使用が許容される一連のコンセプト一式の定義。例えば、「コンディション (Condition)」ドメインは患者の状態を説明するコンセプトを含み、これらのコンセプトは CONDITION_OCCURRENCE および CONDITION_ERA テーブルの condition_concept_id フィールドにのみ格納できる。

電子的健康記録 (Electronic Health Record, EHR) 医療の過程で生成され、電子システムに記録されるデータ。

疫学（Epidemiology） 一定の集団における健康および疾患の分布、パターン、および決定要因の研究。

エビデンスに基づく医療（Evidence-based medicine） 個々の患者の医療に関する意思決定において実証的および科学的エビデンスを使用すること。

ETL（抽出-変換-読込）（Extract-Transform-Load） データをある形式から別の形式に変換するプロセス（例：ソース形式から共通データモデルへの変換）。詳細は第6章参照。

マッチング（Matching） 多くの集団レベルの効果推定のためのアプローチは、曝露された患者のアウトカムを曝露されていない患者（または曝露Bに対して曝露A）と比較することによって、曝露の因果効果を特定しようとする。これらの2つの患者群が曝露以外の点で異なる可能性があるため、「マッチング」は曝露および非曝露群を設定する際に、少なくとも測定された患者特性に関してできる限り同じにしようとする。

メジャーメント（測定）（Measurement） 患者または患者の検体の体系的かつ標準化された診察または検査を通じて得られる構造化された値（数値またはカテゴリカル）。

測定誤差（Measurement error） 記録された測定値（例：血圧、患者の年齢、治療期間）が対応する真の測定値と異なる場合に発生する。

メタデータ（Metadata） 他のデータについて説明し、情報を提供するデータの一式。メタデータには、記述メタデータ、構造メタデータ、管理メタデータ、参照メタデータ、統計メタデータがある。

Methods Library 観察研究を実行するために OHDSI コミュニティによって開発された一連の R パッケージ。

モデルの誤特定（Model misspecification） 多くの OHDSI で用いる方法は、比列ハザード回帰やランダムフォレストなどの統計モデルを採用している。データ生成メカニズムが想定モデルから逸脱する限り、モデルは「誤特定」である。

ネガティブコントロール（Negative control） 曝露がアウトカムを引き起こさないまたは予防しないと信じられる曝露-アウトカムの組み合わせ。効果推定が真実に沿った結果を生成するかどうかを評価するために使用される。詳細は第18章参照。

オブザベーション（観察）（Observation） 診察、問診または処置のコンテキストで得られた患者に関する臨床上の事実。

観察期間（Observation period） 患者がソースシステム内で臨床イベントの有無にか関わらず（状態の良い患者が医療に関与しない場合を含めて）臨床イベントを記録する可能性がある期間。

観察研究（Observational study） 研究者が介入を制御しない研究。

OHDSI SQL R パッケージ SqlRender を使用して様々な他の SQL ダイアレクト（方言）に自動変換できる SQL ダイアレクト。OHDSI SQL は主に SQL Server SQL のサブセットであるが、追加のパラメータ化が可能である。詳細は第 9 章参照。

オープンサイエンス（Open science） 科学研究（パブリケーション、データ、物理的サンプル、ソフトウェアを含む）およびその普及を、要求のあった社会、アマチュア、専門家のすべてのレベルでアクセス可能にする運動。詳細は第 3 章参照。

アウトカム（Outcome） 分析の焦点となる観察。例えば、患者レベルの予測モデルは、アウトカム「脳卒中」を予測するのかもしれない。または集団レベルの推定は、薬剤がアウトカム「頭痛」に及ぼす因果効果を推定するかもしれない。

患者レベルの予測（Patient-level prediction） ベースライン特性に基づいて将来のアウトカムを経験する患者固有の確率を生成する予測モデルの開発と適用。

フェノタイプ（Phenotype） 身体的特性の説明。これには、体重や髪の色のような可視的な特徴だけでなく、全体的な健康状態、病歴、行動も含まれる。

集団レベル推定（Population-level estimation） 因果効果の研究。平均（集団レベル）の効果の大きさを推定する。

ポジティブコントロール（Positive control） 曝露がアウトカムを引き起こすまたは予防すると信じられる曝露-アウトカムの組み合わせ。効果推定方法が真実に沿った結果を生成するかどうかを評価するために使用される。詳細は第 18 章参照。

処置（Procedure） 患者に対して診断または治療目的で医療従事者によって命じられまたは実行される活動またはプロセス。

傾向スコア（Propensity score, PS） 観察研究において、2 つの治療群間の均衡をとり無作為化を模倣するために集団レベル推定で用いられる単一の指標。傾向スコアは、観察された一連のベースライン共変量の関数として患者が関心のある治療を受ける確率を示す。最もよく使われるのは、二値アウトカムはターゲット治療を受ける群を 1、比較対照治療を受ける群を 0 に設定したロジスティック回帰モデルを使用した計算である。詳細は第 12 章参照。

プロトコル（Protocol） 研究のデザインを完全に指定するドキュメントで、人が読んで理解できるもの。

Rabbit-in-a-Hat ソース形式から共通データモデルへの ETL を定義を支援するインタラクティブなソフトウェアツール。White Rabbit によって生成されたデータベースプロファイルを入力として使用する。詳細は第 7 章参照。

選択バイアス（Selection bias） データ内の患者集団が統計分析を歪める形で母集団の患者から逸脱した場合に発生するバイアス。

自己対照デザイン（Self-controlled designs） 同一患者内で異なる曝露期間中のアウトカムを比較する研究デザイン。

感度分析（Sensitivity analysis） 不確実性が存在する分析に関する選択の影響を評価するために研究の主たる分析のバリエーション。

SNOMED 臨床ドキュメントおよび報告書で使用するため、コード、用語、同義語および定義を提供する体系化されたコンピュータ処理可能な医療用語集。

研究診断（Study diagnostics） 特定の分析アプローチが特定の研究質問に対する回答に使用できるかどうか（あるいは、妥当かどうか）を判断することを目的とする分析手順のセット。詳細は第 18 章参照。

研究パッケージ（Study package） 研究を完全に実行するコンピュータ実行プログラム。詳細は第 17 章参照。

ソースコード（Source code） ソースデータベースで使用されるコード。例えば、ICD-10 コード。

標準コンセプト（Standard Concept） 妥当であると指定され、共通データモデルに含めることができるコンセプト。

THEMIS 共通データモデル仕様に関して高い粒度と詳細を持つデータ形式に取り組む OHDSI ワークグループ。

ビジット（Visit） 医療システム内で特定の設定の医療施設において、1 人以上の医療従事者から継続して医療サービスを受ける期間。

ボキャブラリ（Vocabulary） 通常アルファベット順に並べられ、定義または翻訳された単語や語句のリスト。詳細は第 5 章参照。

White Rabbit 共通データモデルへの ETL を定義する前にデータベースをプロファイルリングするソフトウェアツール。詳細は第 6 章参照。

第 B 章

コホート定義

この付録には、本書全体で使用されるコホート定義が含まれています。

B.1 ACE 阻害薬

初回イベントコホート

以下のいずれかを持つ人：

- ・その人の履歴において初回の ACE 阻害薬（表 B.1）への曝露

かつ、インデックス日から遡って少なくとも 365 日前からインデックス日 0 日後の間の連続した観察があり、初期イベントを以下に限定します：その個人における全てのイベント。

適格コホートとしてイベントを次のように限定します：その個人における全てのイベント。

終了日の考え方

カスタム薬剤曝露期間の終了基準：ここでは、指定されたコンセプトセットで見つかったコードから薬剤曝露期間を作成します。インデックスイベントが曝露期間内で見つかった場合、コホート終了日はその曝露期間の終了日を使用します。そうでない場合、そのインデックスイベントを含む観察期間の終了日を使用します。

ACE 阻害薬（表 B.1）の曝露期間終了日を使用

- ・曝露期間の間隔は 30 日を許容
- ・曝露期間終了後に 0 日を追加

コホート圧縮の考え方

30日間のギャップによりコホートを圧縮します。

コンセプトセット定義

Table B.1: ACE 阻害薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1308216	リシノプリル	いいえ	はい	いいえ
1310756	モエキシプリル	いいえ	はい	いいえ
1331235	キナプリル	いいえ	はい	いいえ
1334456	ラミプリル	いいえ	はい	いいえ
1335471	ベナゼプリル	いいえ	はい	いいえ
1340128	カプトプリル	いいえ	はい	いいえ
1341927	エナラプリル	いいえ	はい	いいえ
1342439	トランドラプリル	いいえ	はい	いいえ
1363749	フォシノプリル	いいえ	はい	いいえ
1373225	ペリンドプリル	いいえ	はい	いいえ

B.2 ACE 阻害薬単剤療法新規ユーザー

初回イベントコホート

以下のいずれかを持つ人：

- ・その人の履歴において初回の ACE 阻害薬（表 B.2）への曝露

かつ、インデックス日から遡って少なくとも 365 日前からインデックス日 0 日後の間に連続した観察があり、初期イベントを以下に限定します：その個人における最も早いイベント。

選択ルール

選択基準 #1：治療開始前 1 年間に高血圧の診断を受けています。

以下の全ての基準を満たします：

- ・インデックス開始日から遡って 365 日前からインデックス開始日 0 日後の間に少なくとも 1 回の高血圧性障害（表 B.3）のコンディションが出現します。

選択基準 #2：病歴に高血圧治療薬の使用がありません。

以下の全ての基準を満たします：

- インデックス開始日 1 日前までのすべての日に始まる高血圧薬（表 B.4）の薬物使用が完全に 0 回です。

選択基準 #3 : ACE 単剤療法のみを受けており、併用治療を行っていません。

以下のすべての基準を満たします :

- インデックス開始日 0 日前から 7 日後の間に始まる高血圧薬（表 B.4）の明確な薬物曝露期間の出現がちょうど 1 回です。

適格コホートとしてイベントを次のように限定します：その個人における最も早いイベント。

終了日の考え方

カスタム薬剤曝露期間の終了の基準：この考え方は、指定されたコンセプトセットで見つかったコードから薬剤曝露期間を作成します。インデックスイベントが曝露期間内で見つかった場合、コホートの終了日はその曝露期間の終了日を使用します。そうでない場合、インデックスイベントを含む観察期間の終了日を使用します。

ACE 阻害薬の曝露期間の終了日 (表 B.2)

- 薬剤曝露間隔が 30 日を許容します。
- 薬剤曝露終了後 0 日追加します。

コホート圧縮の考え方

0 日間のギャップサイズで期間によりコホート圧縮します。

コンセプトセット定義

Table B.2: ACE 阻害薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1308216	リシノプリル	いいえ	はい	いいえ
1310756	モエキシプリル	いいえ	はい	いいえ
1331235	キナプリル	いいえ	はい	いいえ
1334456	ラミプリル	いいえ	はい	いいえ
1335471	ベナゼプリル	いいえ	はい	いいえ
1340128	カプトプリル	いいえ	はい	いいえ
1341927	エナラプリル	いいえ	はい	いいえ
1342439	トランドラプリル	いいえ	はい	いいえ
1363749	フォシノプリル	いいえ	はい	いいえ
1373225	ペリンドプリル	いいえ	はい	いいえ

Table B.3: 高血圧性障害

コンセプト ID	コンセプト名	除外	下位層	マッピング元
316866	高血圧性障害	いいえ	はい	いいえ

Table B.4: 高血圧治療薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
904542	トリアムテレン	いいえ	はい	いいえ
907013	メトラゾン	いいえ	はい	いいえ
932745	ブメタニド	いいえ	はい	いいえ
942350	トルセミド	いいえ	はい	いいえ
956874	フロセミド	いいえ	はい	いいえ
970250	スピロノラクトン	いいえ	はい	いいえ
974166	ヒドロクロロチアジド	いいえ	はい	いいえ
978555	インダパミド	いいえ	はい	いいえ
991382	アミロリド	いいえ	はい	いいえ
1305447	メチルドパ	いいえ	はい	いいえ
1307046	メトプロロール	いいえ	はい	いいえ
1307863	ベラパミル	いいえ	はい	いいえ
1308216	リシノプリル	いいえ	はい	いいえ
1308842	バルサルタン	いいえ	はい	いいえ
1309068	ミノキシジル	いいえ	はい	いいえ
1309799	エプレレノン	いいえ	はい	いいえ
1310756	モエキシプリル	いいえ	はい	いいえ
1313200	ナドロール	いいえ	はい	いいえ
1314002	アテノロール	いいえ	はい	いいえ
1314577	ネビボロール	いいえ	はい	いいえ
1317640	テルミサルタン	いいえ	はい	いいえ
1317967	アリスキレン	いいえ	はい	いいえ
1318137	ニカルジピン	いいえ	はい	いいえ
1318853	ニフェジピン	いいえ	はい	いいえ
1319880	ニソルジピン	いいえ	はい	いいえ
1319998	アセブトロール	いいえ	はい	いいえ
1322081	ベタキソロール	いいえ	はい	いいえ
1326012	イスラジピン	いいえ	はい	いいえ
1327978	ペンブトロール	いいえ	はい	いいえ
1328165	ジルチアゼム	いいえ	はい	いいえ
1331235	キナプリル	いいえ	はい	いいえ
1332418	アムロジピン	いいえ	はい	いいえ
1334456	ラミブリル	いいえ	はい	いいえ
1335471	ベナゼブリル	いいえ	はい	いいえ
1338005	ビソプロロール	いいえ	はい	いいえ

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1340128	カプトプリル	いいえ	はい	いいえ
1341238	テラゾシン	いいえ	はい	いいえ
1341927	エナラプリル	いいえ	はい	いいえ
1342439	トランドラプリル	いいえ	はい	いいえ
1344965	グアンファシン	いいえ	はい	いいえ
1345858	ピンドロール	いいえ	はい	いいえ
1346686	エプロサルタン	いいえ	はい	いいえ
1346823	カルベジロール	いいえ	はい	いいえ
1347384	イルベサルタン	いいえ	はい	いいえ
1350489	プラゾシン	いいえ	はい	いいえ
1351557	カンデサルタン	いいえ	はい	いいえ
1353766	プロプラノロール	いいえ	はい	いいえ
1353776	フェロジピン	いいえ	はい	いいえ
1363053	ドキサゾシン	いいえ	はい	いいえ
1363749	フォシノプリル	いいえ	はい	いいえ
1367500	ロサルタン	いいえ	はい	いいえ
1373225	ペリンドプリル	いいえ	はい	いいえ
1373928	ヒドララジン	いいえ	はい	いいえ
1386957	ラベタロール	いいえ	はい	いいえ
1395058	クロルタリドン	いいえ	はい	いいえ
1398937	クロニジン	いいえ	はい	いいえ
40226742	オルメサルタン	いいえ	はい	いいえ
40235485	アジルサルタン	いいえ	はい	いいえ

B.3 急性心筋梗塞（AMI）

初回イベントコホート

以下のいずれかを持つ人：

- 急性心筋梗塞（表 B.5）のコンディション出現

かつ、インデックス日から遡って少なくとも 0 日前からインデックス日 0 日後の間の連続した観察があり、インデックスイベントを以下に限定します：その個人における全てのイベント。

主要イベントがありとなるのは、以下のいずれかの基準を満たす人：

- インデックス日から遡るすべての日から 0 日後の間に始まり、ビジット終了日がインデックス開始日 0 日後からインデックス開始日後の全ての日の間に終了すると一致する入院または救急室ビジット（表 B.6）のビジット出現が少なくとも 1 件。

インデックスイベントのコホート次のように限定します：その個人における全てのイベント。

終了日の考え方

日付オフセット終了基準：

このコホート定義の終了日は、インデックスイベントの開始日から 7 日後とします。

コホート圧縮の考え方

180 日間のギャップサイズで期間によりコホートを圧縮します。

コンセプトセット定義

Table B.5: 急性心筋梗塞

コンセプト ID	コンセプト名	除外	下位層	マッピング元
314666	陳旧性心筋梗塞	はい	はい	いいえ
4329847	心筋梗塞	いいえ	はい	いいえ

Table B.6: 入院または救急室ビジット

コンセプト ID	コンセプト名	除外	下位層	マッピング元
262	救急室および入院ビジット	いいえ	はい	いいえ
9201	入院ビジット	いいえ	はい	いいえ
9203	救急室ビジット	いいえ	はい	いいえ

B.4 血管性浮腫

初回イベントコホート

以下のいずれかを持つ人：

- 血管性浮腫のコンディション出現（表 B.7）

イベント発生日の前と後少なくとも 0 日間の連続した観察期間を持ち、初回イベントを次のように限定します：その個人における全てのイベント。

主要イベントがありとなるのは、以下のいずれかの基準を満たす人：

- 入院または救急室ビジットインデックス日前全ての日と後 0 日の間に開始し、インデックス日前 0 日と後全ての日の間に終了する（表 B.8）で特定されるビジットが少なくとも 1 回発生します。

初回イベントのコホートを次のように限定します：各個人の全てのイベント。

適格なコホートを次のように限定します：各個人の全てのイベント。

終了日の考え方

このコホート定義の終了日はインデックスイベントの開始日から 7 日後とします。

コホート圧縮の考え方

30 日間のギャップサイズで期間によりコホートを圧縮します。

コンセプトセット定義

Table B.7: 血管性浮腫

コンセプト ID	コンセプト名	除外	下位層	マッピング元
432791	血管性浮腫	いいえ	はい	いいえ

Table B.8: 入院または救急室ビジット

コンセプト ID	コンセプト名	除外	下位層	マッピング元
262	救急室および入院ビジット	いいえ	はい	いいえ
9201	入院ビジット	いいえ	はい	いいえ
9203	救急室ビジット	いいえ	はい	いいえ

B.5 サイアザイド様利尿薬単剤療法の新規ユーザー使用者

初回イベントコホート

以下のいずれかを持つ人：

- ・その人の履歴において初回のサイアザイドまたはサイアザイド様利尿薬（表 B.9）への曝露

かつ、インデックス日からの遡って少なくとも 365 日間からインデックス日 0 日後の間の連続した観察があり、初回イベントを次のように限定します：その個人における最も早いイベント。

選択ルール

選択基準 1：治療前の 1 年間に高血圧の診断をうけています。

以下の全ての基準を満たします：

- インデックス開始日から遡って 365 日前からインデックス開始日 0 日後の間に少なくとも 1 回の 高血圧性障害 (表 B.10) のコンディションが出現します。

選択基準 #2：病歴に高血圧治療薬の使用がありません。

以下の全ての基準を満たします：

- インデックス開始日 1 日前までのすべての日に始まる高血圧薬 (表 B.11) の薬物使用が完全に 0 回です。

選択基準 #3：ACE 単剤療法のみを受けており、併用治療を行っていません。

以下のすべての基準を満たします：

- インデックス開始日 0 日前から 7 日後の間に始まる高血圧薬 (表 B.11) の明確な薬物曝露期間の出現がちょうど 1 回です。

適格コホートとしてイベントを次のように限定します：その個人における最も早いイベント。

終了日の考え方

カスタム薬剤曝露期間の終了の基準：この考え方は、指定されたコンセプトセットで見つかったコードから薬剤曝露期間を作成します。インデックスイベントが曝露期間内で見つかった場合、コホートの終了日はその曝露期間の終了日を使用します。そうでない場合、インデックスイベントを含む観察期間の終了日を使用します。

サイアザイドまたはサイアザイド様利尿薬（表 B.9）の曝露期間の終了日

- 薬剤曝露間隔が 30 日を許容します。
- 薬剤曝露終了後 0 日追加します。

コホート圧縮の考え方

0 日間のギャップによりコホートを圧縮します。

コンセプトセット定義

Table B.9: サイアザイドまたはサイアザイド様利尿薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
907013	メトラゾン	いいえ	はい	いいえ

コンセプト ID	コンセプト名	除外	下位層	マッピング元
974166	ヒドロクロロチアジド	いいえ	はい	いいえ
978555	インダパミド	いいえ	はい	いいえ
1395058	クロルタリドン	いいえ	はい	いいえ

Table B.10: 高血圧性障害

コンセプト ID	コンセプト名	除外	下位層	マッピング元
316866	高血圧性障害	いいえ	はい	いいえ

Table B.11: 高血圧治療薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
904542	トリアムテレン	いいえ	はい	いいえ
907013	メトラゾン	いいえ	はい	いいえ
932745	ブメタニド	いいえ	はい	いいえ
942350	トルセミド	いいえ	はい	いいえ
956874	フロセミド	いいえ	はい	いいえ
970250	スピロノラクトン	いいえ	はい	いいえ
974166	ヒドロクロロチアジド	いいえ	はい	いいえ
978555	インダパミド	いいえ	はい	いいえ
991382	アミロライド	いいえ	はい	いいえ
1305447	メチルドパ	いいえ	はい	いいえ
1307046	メトプロロール	いいえ	はい	いいえ
1307863	ベラパミル	いいえ	はい	いいえ
1308216	リシノプリル	いいえ	はい	いいえ
1308842	バルサルタン	いいえ	はい	いいえ
1309068	ミノキシジル	いいえ	はい	いいえ
1309799	エプレレノン	いいえ	はい	いいえ
1310756	モエキシプリル	いいえ	はい	いいえ
1313200	ナドロール	いいえ	はい	いいえ
1314002	アテノロール	いいえ	はい	いいえ
1314577	ネビボロール	いいえ	はい	いいえ
1317640	テルミサルタン	いいえ	はい	いいえ
1317967	アリストレン	いいえ	はい	いいえ
1318137	ニカルディピン	いいえ	はい	いいえ
1318853	ニフェジピン	いいえ	はい	いいえ
1319880	ニソルジピン	いいえ	はい	いいえ
1319998	アセブトロール	いいえ	はい	いいえ
1322081	ベタキソール	いいえ	はい	いいえ
1326012	イスラジピン	いいえ	はい	いいえ

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1327978	ペンブトロール	いいえ	はい	いいえ
1328165	ジルチアゼム	いいえ	はい	いいえ
1331235	キナブリル	いいえ	はい	いいえ
1332418	アムロジピン	いいえ	はい	いいえ
1334456	ラミブリル	いいえ	はい	いいえ
1335471	ベナゼブリル	いいえ	はい	いいえ
1338005	ビソプロロール	いいえ	はい	いいえ
1340128	カプトブリル	いいえ	はい	いいえ
1341238	テラゾシン	いいえ	はい	いいえ
1341927	エナラブリル	いいえ	はい	いいえ
1342439	トランドラブリル	いいえ	はい	いいえ
1344965	グアンファシン	いいえ	はい	いいえ
1345858	ピンドロール	いいえ	はい	いいえ
1346686	エプロサルタン	いいえ	はい	いいえ
1346823	カルベジロール	いいえ	はい	いいえ
1347384	イルベサルタン	いいえ	はい	いいえ
1350489	プラゾシン	いいえ	はい	いいえ
1351557	カンデサルタン	いいえ	はい	いいえ
1353766	プロプラノロール	いいえ	はい	いいえ
1353776	フェロジピン	いいえ	はい	いいえ
1363053	ドキサゾシン	いいえ	はい	いいえ
1363749	フォシノブリル	いいえ	はい	いいえ
1367500	ロサルタン	いいえ	はい	いいえ
1373225	ペリンドブリル	いいえ	はい	いいえ
1373928	ヒドララジン	いいえ	はい	いいえ
1386957	ラベタロール	いいえ	はい	いいえ
1395058	クロルタリドン	いいえ	はい	いいえ
1398937	クロニジン	いいえ	はい	いいえ
40226742	オルメサルタン	いいえ	はい	いいえ
40235485	アジルサルタン	いいえ	はい	いいえ

B.6 高血圧のための第一選択治療を開始する患者

初回イベントコホート

以下のいずれかを持つ人：

- ・その人の履歴において初回の第一選択高血圧治療薬（表 B.12）への曝露

かつ、インデックス日から遡って少なくとも 365 日前からインデックス日 365 日後の間の連続した観察があり、初期イベントを次のように限定します：その個人における最も早いイベント。

選択ルール

以下の全ての基準を満たすこと：

- インデックス開始日 1 日前までのすべての日に少なくとも 1 回の高血圧治療薬（表 B.13）の薬剤への曝露がちょうど 0 回出現します。
- かつ、インデックス開始日からさかのぼって 365 日と後 0 日までの間に高血圧性障害（表 B.14）のコンディションが少なくとも 1 回出現します。

初回イベントのコホートを次のように限定します：その個人における最も早いイベント。適格コホートとしてイベントを次のように限定します：その個人における最も早いイベント。

終了日の考え方

終了日の考え方を選択されません。デフォルトでは、コホート終了日はインデックスイベントを含む観察期間の終了日になります。

コホート圧縮の考え方

0 日間のギャップによりコホートを圧縮します。

コンセプトセット定義

Table B.12: 第一選択高血圧治療薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
907013	メトラゾン	NO	YES	NO
974166	ヒドロクロロチアジド	NO	YES	NO
978555	インダパミド	NO	YES	NO
1307863	ベラパミル	NO	YES	NO
1308216	リシノプリル	NO	YES	NO
1308842	バルサルタン	NO	YES	NO
1310756	モエキシプリル	NO	YES	NO
1317640	テルミサルタン	NO	YES	NO
1318137	ニカルジピン	NO	YES	NO
1318853	ニフェジピン	NO	YES	NO
1319880	ニソルジピン	NO	YES	NO
1326012	イスラジピン	NO	YES	NO
1328165	ジルチアゼム	NO	YES	NO
1331235	キナプリル	NO	YES	NO
1332418	アムロジピン	NO	YES	NO
1334456	ラミプリル	NO	YES	NO
1335471	ベナゼプリル	NO	YES	NO

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1340128	カプトプリル	NO	YES	NO
1341927	エナラプリル	NO	YES	NO
1342439	トランドラプリル	NO	YES	NO
1346686	エプロサルタン	NO	YES	NO
1347384	イルベサルタン	NO	YES	NO
1351557	カンデサルタン	NO	YES	NO
1353776	フェロジピン	NO	YES	NO
1363749	ホシノプリル	NO	YES	NO
1367500	ロサルタン	NO	YES	NO
1373225	ペリンドプリル	NO	YES	NO
1395058	クロルタリドン	NO	YES	NO
40226742	オルメサルタン	NO	YES	NO
40235485	アジルサルタン	NO	YES	NO

Table B.13: 高血圧治療薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
904542	トリアムテレン	NO	YES	NO
907013	メトラゾン	NO	YES	NO
932745	ブメタニド	NO	YES	NO
942350	トルセミド	NO	YES	NO
956874	フロセミド	NO	YES	NO
970250	スピロノラクトン	NO	YES	NO
974166	ヒドロクロロチアジド	NO	YES	NO
978555	インダパミド	NO	YES	NO
991382	アミロイド	NO	YES	NO
1305447	メチルドパ	NO	YES	NO
1307046	メトプロロール	NO	YES	NO
1307863	ベラパミル	NO	YES	NO
1308216	リシノプリル	NO	YES	NO
1308842	バルサルタン	NO	YES	NO
1309068	ミノキシジル	NO	YES	NO
1309799	エプレレノン	NO	YES	NO
1310756	モエキシプリル	NO	YES	NO
1313200	ナドロール	NO	YES	NO
1314002	アテノロール	NO	YES	NO
1314577	ネビボロール	NO	YES	NO
1317640	テルミサルタン	NO	YES	NO
1317967	アリスキレン	NO	YES	NO
1318137	ニカルジピン	NO	YES	NO
1318853	ニフェジピン	NO	YES	NO
1319880	ニソルジピン	NO	YES	NO

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1319998	アセブトロール	NO	YES	NO
1322081	ベタキソロール	NO	YES	NO
1326012	イスラジピン	NO	YES	NO
1327978	ベンブトロール	NO	YES	NO
1328165	ジルチアゼム	NO	YES	NO
1331235	キナプリル	NO	YES	NO
1332418	アムロジピン	NO	YES	NO
1334456	ラミプリル	NO	YES	NO
1335471	ベナゼプリル	NO	YES	NO
1338005	ビソプロロール	NO	YES	NO
1340128	カプトプリル	NO	YES	NO
1341238	テラゾシン	NO	YES	NO
1341927	エナラプリル	NO	YES	NO
1342439	トランドラプリル	NO	YES	NO
1344965	グアンファシン	NO	YES	NO
1345858	ピンドロール	NO	YES	NO
1346686	エプロサルタン	NO	YES	NO
1346823	カルベジロール	NO	YES	NO
1347384	イルベサルタン	NO	YES	NO
1350489	プラゾシン	NO	YES	NO
1351557	カンデサルタン	NO	YES	NO
1353766	プロプラノロール	NO	YES	NO
1353776	フェロジピン	NO	YES	NO
1363053	ドキサゾシン	NO	YES	NO
1363749	ホシノプリル	NO	YES	NO
1367500	ロサルタン	NO	YES	NO
1373225	ペリンドプリル	NO	YES	NO
1373928	ヒドララジン	NO	YES	NO
1386957	ラベタロール	NO	YES	NO
1395058	クロルタリドン	NO	YES	NO
1398937	クロニジン	NO	YES	NO
40226742	オルメサルタン	NO	YES	NO
40235485	アジルサルタン	NO	YES	NO

Table B.14: 高血圧性障害

コンセプト ID	コンセプト名	除外	下位層	マッピング元
316866	高血圧性障害	NO	YES	NO

B.7 追跡期間が 3 年以上ある高血圧のための第一選択治療を開始する患者

コホート定義 B.6 と同じだが、インデックス日から遡って少なくとも 365 日前から 1095 日後の連続した観察がある。

B.8 ACE 阻害薬の使用

初回イベントコホート

以下のいずれかを持つ人：

- ACE 阻害薬（表 B.15）の薬物への曝露

かつ、インデックス日から遡って少なくとも 0 日前から 0 日後の間の連続した観察があり、初回イベントを次のように限定します：その個人のすべてのイベント。

適格なコホートを個人ごとのすべてのイベントに限定します。

終了日の考え方

この考え方は、指定されたコンセプトセットで見つかったコードから薬剤曝露期間を作成します。インデックスイベントが曝露期間内で見つかった場合、コホートの終了日はその曝露期間の終了日を使用します。そうでない場合、インデックスイベントを含む観察期間の終了日を使用します。

ACE 阻害薬の曝露期間の終了日（表 B.15）

- 薬剤曝露間隔が 30 日を許容します。
- 薬剤曝露終了後 0 日追加します。

コホート圧縮の考え方

30 日間のギャップサイズで期間によりコホート圧縮します。

コンセプトセット定義

Table B.15: ACE 阻害薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1308216	リシノプリル	NO	YES	NO
1310756	モエキシプリル	NO	YES	NO
1331235	キナプリル	NO	YES	NO
1334456	ラミプリル	NO	YES	NO

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1335471	ベナゼプリル	NO	YES	NO
1340128	カプトプリル	NO	YES	NO
1341927	エナラプリル	NO	YES	NO
1342439	トランドラプリル	NO	YES	NO
1363749	ホシノプリル	NO	YES	NO
1373225	ペリンドプリル	NO	YES	NO

B.9 アンジオテンシン受容体拮抗薬（ARB）の使用

コホート定義 B.8 と同じですが、アンジオテンシン受容体拮抗薬（ARB）（表 B.16）が ACE 阻害薬（表 B.15）の代わりに使用されます。

コンセプトセット定義

Table B.16: アンジオテンシン受容体拮抗薬（ARB）

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1308842	バルサルタン	NO	YES	NO
1317640	テルミサルタン	NO	YES	NO
1346686	エプロサルタン	NO	YES	NO
1347384	イルベサルタン	NO	YES	NO
1351557	カンデサルタン	NO	YES	NO
1367500	ロサルタン	NO	YES	NO
40226742	オルメサルタン	NO	YES	NO
40235485	アジルサルタン	NO	YES	NO

B.10 サイアザイドおよびサイアザイド様利尿薬の使用

コホート定義 B.8 と同じですが、サイアザイドおよびサイアザイド様利尿薬（表 B.17）が ACE 阻害薬（表 B.15）の代わりに使用されます。

コンセプトセット定義

Table B.17: サイアザイドおよびサイアザイド様利尿薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
907013	メトラゾン	NO	YES	NO
974166	ヒドロクロロチアジド	NO	YES	NO
978555	インダパミド	NO	YES	NO
1395058	クロルタリドン	NO	YES	NO

コンセプト ID	コンセプト名	除外	下位層	マッピング元
----------	--------	----	-----	--------

B.11 ジヒドロピリジン系カルシウムチャネル遮断薬 (DCCB) の使用

コホート定義 B.8 と同じですが、ジヒドロピリジン系カルシウムチャネル遮断薬 (DCCB) (表 B.18) が ACE 阻害薬 (表 B.15) の代わりに使用されます。

コンセプトセット定義

Table B.18: ジヒドロピリジン系カルシウムチャネル遮断薬 (DCCB)

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1318137	ニカルジピン	NO	YES	NO
1318853	ニフェジピン	NO	YES	NO
1319880	ニソルジピン	NO	YES	NO
1326012	イスラジピン	NO	YES	NO
1332418	アムロジピン	NO	YES	NO
1353776	フェロジピン	NO	YES	NO

B.12 非ジヒドロピリジン系カルシウムチャネル遮断薬 (NDCCB) の使用

コホート定義 B.8 と同じですが、非ジヒドロピリジン系カルシウムチャネル遮断薬 (NDCCB) (表 B.19) が ACE 阻害薬 (表 B.15) の代わりに使用されます。

コンセプトセット定義

Table B.19: 非ジヒドロピリジン系カルシウムチャネル遮断薬 (NDCCB)

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1307863	ベラパミル	NO	YES	NO
1328165	ジルチアゼム	NO	YES	NO

B.13 ベータ遮断薬使用

コホート定義 B.8 と同じですが、ベータ遮断薬（表 B.20）が ACE 阻害剤（表 B.15）の代わりに使用されます。

コンセプトセット定義

Table B.20: ベータ遮断薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1307046	メトプロロール	NO	YES	NO
1313200	ナドロール	NO	YES	NO
1314002	アテノロール	NO	YES	NO
1314577	ネビボロール	NO	YES	NO
1319998	アセブトロール	NO	YES	NO
1322081	ベタキソロール	NO	YES	NO
1327978	ベンブトロール	NO	YES	NO
1338005	ビソプロロール	NO	YES	NO
1345858	ピンドロール	NO	YES	NO
1346823	カルベジロール	NO	YES	NO
1353766	プロプラノロール	NO	YES	NO
1386957	ラベタロール	NO	YES	NO

B.14 ループ利尿薬使用

ACE 阻害剤使用 B.8 と同じですが、ループ利尿薬（表 B.21）が ACE 阻害剤（表 B.15）の代わりに使用されます。

コンセプトセット定義

Table B.21: ループ利尿薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
932745	ブメタニド	NO	YES	NO
942350	トルセミド	NO	YES	NO
956874	フロセミド	NO	YES	NO

B.15 カリウム保持性利尿薬使用

ACE 阻害剤使用 B.8 と同じですが、カリウム保持性利尿薬（表 B.22）が ACE 阻害剤（表 B.15）の代わりに使用されます。

コンセプトセットの定義

Table B.22: カリウム保持性利尿薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
904542	トリアムテレン	NO	YES	NO
991382	アミロライド	NO	YES	NO

B.16 アルファ 1 遮断薬使用

ACE 阻害剤使用 B.8 と同じですが、アルファ 1 遮断薬 (表 B.23) が ACE 阻害剤 (表 B.15) の代わりに使用されます。

コンセプトセットの定義

Table B.23: アルファ 1 遮断薬

コンセプト ID	コンセプト名	除外	下位層	マッピング元
1341238	テラゾシン	NO	YES	NO
1350489	プラゾシン	NO	YES	NO
1363053	ドキサゾシン	NO	YES	NO

第 C 章

ネガティブコントロール

この付録には、本書のさまざまな章で使用されるネガティブコントロールが含まれています。

C.1 ACE 阻害薬とサイアザイド・サイアザイド様利尿薬

Table C.1: ACE 阻害剤 (ACEi) とサイアザイドおよびサイアザイド様利尿薬 (THZ) を比較する場合のネガティブコントロールアウトカム

コンセプト ID	コンセプトの名前
434165	子宮頸部スメア異常
436409	瞳孔異常
199192	感染を伴わない体幹の擦過傷および/または摩擦熱傷
4088290	乳房欠損
4092879	腎臓欠損
44783954	胃酸逆流
75911	後天性外反母趾
137951	後天性角質変性症
77965	後天性ばね指
376707	急性結膜炎
4103640	切断足
73241	肛門および直腸ポリープ
133655	前腕の熱傷
73560	踵骨の骨棘
434327	大麻乱用
4213540	頸部の体性機能障害
140842	皮膚の質感の変化
81378	膝蓋骨軟骨軟化症

コンセプト ID	コンセプトの名前
432303	コカイン乱用
4201390	人工肛門あり
46269889	クローン病による合併症
134438	接触性皮膚炎
78619	膝の打撲傷
201606	クローン病
76786	膝関節運動障害
4115402	睡眠困難
45757370	再建乳房の不均衡
433111	空腹の影響
433527	子宮内膜症
4170770	類表皮囊胞
4092896	糞便内容物異常
259995	開口部内の異物
40481632	ガングリオン囊胞
4166231	遺伝的素因
433577	槌状趾
4231770	遺伝性血栓症
440329	合併症を伴わない帯状疱疹
4012570	ハイリスクの性行動
4012934	ホモシチン尿症
441788	ヒトパピローマウイルス感染
4201717	人工回腸あり
374375	耳垢塞栓
4344500	肩関節インピンジメント症候群
139099	嵌入爪
444132	膝損傷
196168	月経不順
432593	クワシオルコル
434203	打撲の後遺症
438329	自動車事故の後遺症
195873	白色帯下
4083487	網膜ドルーゼン
4103703	メレナ（黒色便）
4209423	ニコチニン依存症
377572	内耳に対する騒音の影響
40480893	非特異的ツベルクリンテスト反応
136368	非毒性多結節性甲状腺腫
140648	皮膚糸状菌による爪白癬
438130	オピオイド乱用
4091513	放屁
4202045	ウイルス感染後疲労症候群
373478	老視

コンセプト ID	コンセプトの名前
46286594	ライフスタイルに関連する問題
439790	精神疼痛
81634	下垂乳房
380706	正乱視
141932	老人性角化症
36713918	腰部の体性機能障害
443172	大きな開放創のない顔の刺
81151	足首の捻挫
72748	肩腱板のストレイン損傷
378427	涙液不足
437264	タバコ依存症候群
194083	膿炎および外陰膿炎
140641	尋常性疣贅
440193	手首下垂
4115367	手関節痛

第 D 章

プロトコルテンプレート

1. 目次
2. 略語一覧
3. 要約
4. 変更と更新
5. マイルストーン
6. 研究の根拠と背景
7. 研究目的
 - 主要仮説
 - 二次仮説
 - 主要目的
 - 二次目的
8. 研究方法
 - 研究デザイン
 - データソース
 - 研究対象集団
 - 曝露
 - 結果（アウトカム）
 - 共変量
9. データ解析計画
 - リスク期間の計算
 - モデル仕様
 - データベース間の効果推定の統合
 - 実施する解析
 - 出力
 - エビデンス評価
10. 研究診断
 - サンプルサイズと検出力
 - コホートの比較可能性
 - 系統的誤差の評価

11. 研究方法の強みと限界
12. 研究対象者の保護
13. 有害事象と有害反応の管理および報告
14. 研究結果の普及およびコミュニケーション計画
15. 付録：ネガティブコントロール
16. 参考文献

第 E 章

回答例

この付録には、本書の演習に対する回答例が含まれています。

E.1 共通データモデル

演習 4.1

演習で説明されている内容に基づくと、ジョンのレコードは表 E.1 のようになります。

Table E.1: PERSON テーブル

カラム名	値	説明
PERSON_ID	2	一意の整数。
GENDER_CONCEPT_ID	8507	男性のコンセプト ID は 8507。
YEAR_OF_BIRTH	1974	
MONTH_OF_BIRTH	8	
DAY_OF_BIRTH	4	
BIRTH_DATETIME	1974-08-04 00:00:00	時間が不明な場合は 0 時 (00:00:00) を使用。
DEATH_DATETIME	NULL	
RACE_CONCEPT_ID	8516	アフリカ系アメリカ人のコンセプト ID は 8516。38003564 は「非ヒスパニック」を示す。
ETHNICITY_CONCEPT_ID	38003564	
LOCATION_ID		住所は不明。
PROVIDER_ID		主治医が不明。
CARE_SITE		主たる医療施設が不明。

カラム名	値	説明
PERSON_SOURCE_VALUE	NULL	提供されていない。
GENDER_SOURCE_VALUE	Man	説明で使用されたテキスト。
GENDER_SOURCE_CONCEPT_ID	0	
RACE_SOURCE_VALUE	African American	説明で使用されたテキスト。
RACE_SOURCE_CONCEPT_ID	0	
ETHNICITY_SOURCE_VALUE	NULL	
ETHNICITY_SOURCE_CONCEPT_ID	0	

演習 4.2

演習で説明されている内容に基づくと、ジョンのレコードは表 E.2 のようになります。

Table E.2: OBSERVATION_PERIOD テーブル

カラム名	値	説明
OBSERVATION_PERIOD_ID	2	一意の整数。
PERSON_ID	2	これは PERSON テーブルのジョンのレコードへの外部キー。
OBSERVATION_PERIOD_START_DATE	2015-01-01	加入日の日付。
OBSERVATION_PERIOD_END_DATE	2019-07-01	データ抽出日以降のデータが存在することは期待されない。
PERIOD_TYPE_CONCEPT_ID	44814722	44814724 は「保険加入期間」を示す。

演習 4.3

演習で説明されている内容に基づくと、ジョンのレコードは表 E.3 のようになります。

Table E.3: DRUG_EXPOSURE テーブル

カラム名	値	説明
DRUG_EXPOSURE_ID	1001	一意の整数。
PERSON_ID	2	これは PERSON テーブルのジョンのレコードへの外部キー。
DRUG_CONCEPT_ID	19078461	提供された NDC コードは標準コンセプト 19078461 にマッピングされる。
DRUG_EXPOSURE_START_DATE	2019-05-01	薬剤への曝露開始日。
DRUG_EXPOSURE_START_DATETIME	2019-05-01 00:00:00	時間が不明なため 0 時を使用。
DRUG_EXPOSURE_END_DATE	2019-05-31	開始日 + 処方日数に基づく。
DRUG_EXPOSURE_END_DATETIME	2019-05-31 00:00:00	時間が不明なため 0 時を使用。
VERBATIM_END_DATE	NULL	提供されていない。
DRUG_TYPE_CONCEPT_ID	38000177	38000177 は「書かれた処方箋」を示す。
STOP_REASON	NULL	
REFILLS	NULL	
QUANTITY	NULL	提供されていない。
DAYS_SUPPLY	30	演習に記述されている通り。
SIG	NULL	提供されていない。
ROUTE_CONCEPT_ID	4132161	4132161 は「経口」を示す。
LOT_NUMBER	NULL	提供されていない。
PROVIDER_ID	NULL	提供されていない。
VISIT_OCCURRENCE_ID	NULL	ビジットに関する情報は提供されなかった。
VISIT_DETAIL_ID	NULL	
DRUG_SOURCE_VALUE	76168009520	提供された NDC コード。
DRUG_SOURCE_CONCEPT_ID	583945	583945 は薬剤のソースコードの値を表す (NDC コード 「76168009520」)。
ROUTE_SOURCE_VALUE	NULL	

演習 4.4

一連のレコードを見つけるには、CONDITION_OCCURRENCE テーブルをクエリする必要があります：

```
library(DatabaseConnector)
connection <- connect(connectionDetails)
sql <- "SELECT *
FROM @cdm.condition_occurrence
WHERE condition_concept_id = 192671;"

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 ...
```

演習 4.5

一連のレコードを見つけるには、CONDITION_OCCURRENCE テーブルの CONDITION_SOURCE_VALUE フィールドを使用してクエリを実行する必要があります：

```
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 ...
```

演習 4.6

この情報は OBSERVATION_PERIOD テーブルに保存されています：

```
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 標準化ボキャブラリ

演習 5.1

コンセプト ID 192671 (“消化管出血”)

演習 5.2

ICD-10CM コード :

- K29.91 “出血を伴う胃十二指腸炎、詳細不明”
- K92.2 “消化管出血、詳細不明”

ICD-9CM コード :

- 578 “消化管出血”
- 578.9 “消化管出血、詳細不明”

演習 5.3

MedDRA 基本語 (Preferred Terms, PT) :

- “消化管出血” (コンセプト ID 35707864)
- “腸出血” (コンセプト ID 35707858)

E.3 ETL (Extract-Transform-Load)

演習 6.1

- A) データの専門家と CDM の専門家が協力して ETL の設計を行います。
- B) 医学知識を持つ人がコードマッピングを作成します。
- C) エンジニアが ETL を実装します。
- D) 全員が品質管理に関与します。

演習 6.2

カラム	値	回答
PERSON_ID	A123B456	このカラムは整数型のデータタイプを持っているため、ソースのレコード値を数値に変換する必要があります。
GENDER_CONCEPT_ID	8532	
YEAR_OF_BIRTH	NULL	生年月日の月や日が不明な場合は推測しません。PERSON は、生年月日の月や日がなくても存在可能です。生まれた年の情報がない場合には、その PERSON は除外する必要があります。この人は生年月日の年がないため除外する必要がありました。
MONTH_OF_BIRTH	NULL	
DAY_OF_BIRTH	NULL	
RACE_CONCEPT_ID	0	人種は WHITE であり、これは 8527 にマッピングされるべきです。
ETHNICITY_CONCEPT_ID	8527	民族の情報が提供されていないため、これは 0 にマッピングされるべきです。
PERSON_SOURCE_VALUE	A123B456	
GENDER_SOURCE_VALUE	F	
RACE_SOURCE_VALUE	WHITE	
ETHNICITY_SOURCE_VALUE	NONE PROVIDED	

演習 6.3

カラム	値
VISIT_OCCURRENCE_ID	1
PERSON_ID	11
VISIT_START_DATE	2004-09-26

カラム	値
VISIT_END_DATE	2004-09-30
VISIT_CONCEPT_ID	9201
VISIT_SOURCE_VALUE	inpatient

E.4 データ分析のユースケース

演習 7.1

1. 特性評価
2. 患者レベルの予測
3. 集団レベルの推定

演習 7.2

そうではないかもしれません。ジクロフェナク曝露コホートと比較可能な非曝露コホートを定義することは、多くの場合不可能です。なぜなら、人々はジクロフェナクを理由があって服用するからです。このことは、異なる対象の間の比較を妨げます。ジクロフェナクコホートのそれぞれの患者の中で曝露されていない時間を特定することで、個人内での比較がある程度可能かもしれません。ここでも同様の問題が発生します。こうした時間は比較できないことが多く、ある時間に曝露されている理由と別の時間に曝露されていない理由が異なるからです。

E.5 SQL と R

演習 9.1

人数を計算するには、単純に PERSON テーブルをクエリすればよいです:

```
library(DatabaseConnector)
connection <- connect(connectionDetails)
sql <- "SELECT COUNT(*) AS person_count
FROM @cdm.person;"

renderTranslateQuerySql(connection, sql, cdm = "main")
```

```
##    PERSON_COUNT
## 1          2694
```

演習 9.2

セレコキシブの処方を少なくとも 1 回受けた人の人数を計算するには、DRUG_EXPOSURE テーブルをクエリします。セレコキシブの成分を含むすべての薬剤を見つけるために、CONCEPT_ANCESTOR および CONCEPT テーブルを結合します：

```
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
```

COUNT(DISTINCT(person_id)) を使用して、重複することない人数を求めるごとに注意してください。これは、各患者が複数の処方を受けたかもしれないからです。また、「celecoxib」の検索で、大文字小文字を区別しないように LOWER 関数を使用していることにも注意してください。

代わりに、すでに成分レベルにまとめられている DRUG_ERA テーブルを使用することもできます：

```
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
```

演習 9.3

曝露期間中の診断の数を計算するには、以前のクエリを拡張して CONDITION_OCCURRENCE テーブルに結合します。消化管出血を意味するすべてのコンディションのコンセプトを見つけるために、CONCEPT_ANCESTOR テーブルに結合します：

```
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")
```

```
##   DIAGNOSE_COUNT
## 1      41
```

この場合、DRUG_EXPOSURE テーブルではなく DRUG_ERA テーブルを使用することが重要です。なぜなら、同じ成分を含む薬剤の曝露が重なる可能性がありますが、薬剤の曝露期間は重なることがないからです。これにより重複して数えることを避けることができます。例えば、ある人が同時に 2 種類のセレコキシブを含む薬剤を受け取ったとしましょう。これは 2 つの薬剤曝露として記録され、その期間中に発生する診断は 2 回カウントされてしまうでしょう。2 つの曝露は 1 つの重なりのない薬剤曝露期間に統合されます。

E.6 コホートの定義

演習 10.1

以下の要件をコード化する初期イベント基準を作成します：

- ・ジクロフェナクの新規ユーザー
- ・年齢は 16 歳以上
- ・曝露前に少なくとも 365 日間の連続した観察期間があること

完了したときには、コホート組入れイベントのセクションは図 E.1 のようにな

ります。

Cohort Entry Events

Events having any of the following criteria:

- + Add Initial Event
- + Add attribute...
- Delete Criteria

a drug era of **diclofenac**

X for the first time in the person's history

X with age in years at era start **Greater or Equal To** **16**

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

Figure E.1: ジクロフェナクの新規ユーザーのコホート組入れ・エントリー・イベント設定

ジクロフェナクのコンセプトセットは図 E.2のように、成分「Diclofenac」とそのすべての下位層を含むので、ジクロフェナクを含むすべての薬剤を含むことになります。

Concept Set Expression Included Concepts 11473 Included Source Codes Export Import

Name: diclofenac

Show 25 entries Search:

Showing 1 to 1 of 1 entries Previous 1 Next

	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"/>	

Classification Non-Standard Standard

Figure E.2: ジクロフェナクのコンセプトセット

次に、図 E.3に示されるように、NSAID の曝露の履歴がないことを求めます。

NSAIDs のコンセプトセットは、NSAIDs クラスとそのすべての下位層を含めるように、図 E.4のようになるはずで、よって、NSAID を含むすべての薬剤を含むことになります。

さらに、図E.5のよう、以前にがんの診断がないことも要求します。

「広範な悪性腫瘍」のコンセプトセットは、上位コンセプト「悪性腫瘍」とそのすべての下位層を含み、図 E.6のようになるはずです。

最後に、図 E.7のように。曝露中断をコホート離脱基準として定義します（30日間のギャップを許容します）。

Inclusion Criteria

New inclusion criteria

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 + Add attribute... 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

Limit qualifying events to: earliest event per person.

Copy Delete Delete Criteria

Figure E.3: NSAID 曝露の履歴がないことの要求

Concept Set Expression Included Concepts 23112 Included Source Codes Export Import

Name: NSAIDs

Show 25 entries Search: Previous 1 Next

Showing 1 to 1 of 1 entries

	Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	Exclude	Descendants	Mapped
21603933	M01A	ANTIINFLAMMATORY AND ANTIRHEUMATIC PRODUCTS, NON-STEROIDS	Drug	Classification	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Classification Non-Standard Standard

Figure E.4: NSAIDs のコンセプトセット

Inclusion Criteria

New inclusion criteria

Without prior diagnosis of cancer

Excluding subjects with prior cancer diagnosis

having [all] of the following criteria:

with exactly 0 using all occurrences of: a condition occurrence of Broad malignancies + Add attribute... where event starts between All days Before and 0 days Before
index start date add additional constraint
 restrict to the same visit occurrence
 allow events from outside observation period

Limit qualifying events to: earliest event per person.

Copy Delete Delete Criteria

Figure E.5: 以前にがんの診断がないことの要求

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

	Concept Id	Concept Code	Concept Name	Domain	Standard Concept Caption	Exclude	Descendants	Mapped
	443392	363346000	Malignant neoplastic disease	Condition	Standard	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Classification Non-Standard Standard

Figure E.6: 広範な悪性腫瘍のコンセプトセット

Cohort Exit

Event Persistence:
Event will persist until: end of a continuous drug exposure ▼

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: diclofenac ▼

- Persistence window: allow for a maximum of 30 days between exposure records when inferring the era of persistence exposure
- Surveillance window: add 0 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: コホート離脱日の設定

演習 10.2

読みやすくするため、ここでは SQL を 2 つのステップに分けます。まず、すべての心筋梗塞コンディションの出現を同定し、それらを一時テーブル「#diagnoses」に格納します：

```
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 -- 心筋梗塞
)
AND condition_concept_id NOT IN (
    SELECT descendant_concept_id
    FROM @cdm.concept_ancestor
    WHERE ancestor_concept_id = 314666 -- 陳旧性心筋梗塞
);"

renderTranslateExecuteSql(connection, sql, cdm = "main")
```

次に、入院または救急室ビジット時に出現したもののみを選択し、特定の COHORT_DEFINITION_ID（ここでは「1」を選択しました）を使用します：

```
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); -- 入院または救急室ビジット;"

renderTranslateExecuteSql(connection, sql, cdm = "main")
```

コンディションの日がビジット開始日と終了日の間になることを要求する代わりに、コンディションとビジットを VISIT_OCCURRENCE_ID に基づいて結合するアプローチも考えられることに注意してください。この方法は、コンディ

ションが入院または救急室ビジットに関連して記録されたことを保証するので、より正確かもしれません。しかし、多くの観察データベースは、ビジットと診断を関連させて記録しないため、ここでは代わりに日付を使用することを選びました。これにより、感度が高くなりますが、特異度は低くなる可能性があります。

また、ここではコホート終了日は考慮していないことに注意してください。通常、コホートがアウトカムを定義するために使用される場合、関心を持つのはコホートの開始日だけであり、(不明確な) コホート終了日を作成する必要はありません。

一時テーブルが不要になったらクリーンアップすることをお勧めします:

```
sql <- "TRUNCATE TABLE #diagnoses;
DROP TABLE #diagnoses;

renderTranslateExecuteSql(connection, sql)
```

E.7 特性評価

演習 11.1

ATLAS で  Data Sources をクリックし、興味のあるデータソースを選択します。図 E.8 のように、薬剤曝露レポートを選択し、「Table (表)」タブを選択して「celecoxib (セレコキシブ)」を検索することができます。ここでは、この特定のデータベースがセレコキシブのさまざまな製剤の曝露を含むことがわかります。これらの薬剤をクリックすると、例えばその薬剤の年齢や性別の分布など、より詳細なビューを得ることができます。

演習 11.2

 Cohort Definitions をクリックして「New cohort (新規コホート)」を作成します。コホートに意味のある名前（例：「Celecoxib new users (セレコキシブ新規ユーザー)」）を付け、「Concept Sets (コンセプトセット)」タブに移動します。「New Concept Set (新規コンセプトセット)」をクリックし、コンセプトセットに意味のある名前（例：「Celecoxib (セレコキシブ)」）を付けます。 Search モジュールを開き、「Celecoxib (セレコキシブ)」を検索し、クラスを「Ingredient (成分)」、標準コンセプトを「Standard (標準)」とするように限定し、 をクリックして、図 E.9 に示されるように、コンセプトセットにコンセプトを追加します。

図 E.9 の左最上部に表示されている左矢印をクリックしてコホート定義に戻ります。「+Add Initial Event (初回イベントを追加)」をクリックしてから「Add Drug Era (薬剤曝露期間を追加)」をクリックします。薬剤曝露期間基準のため

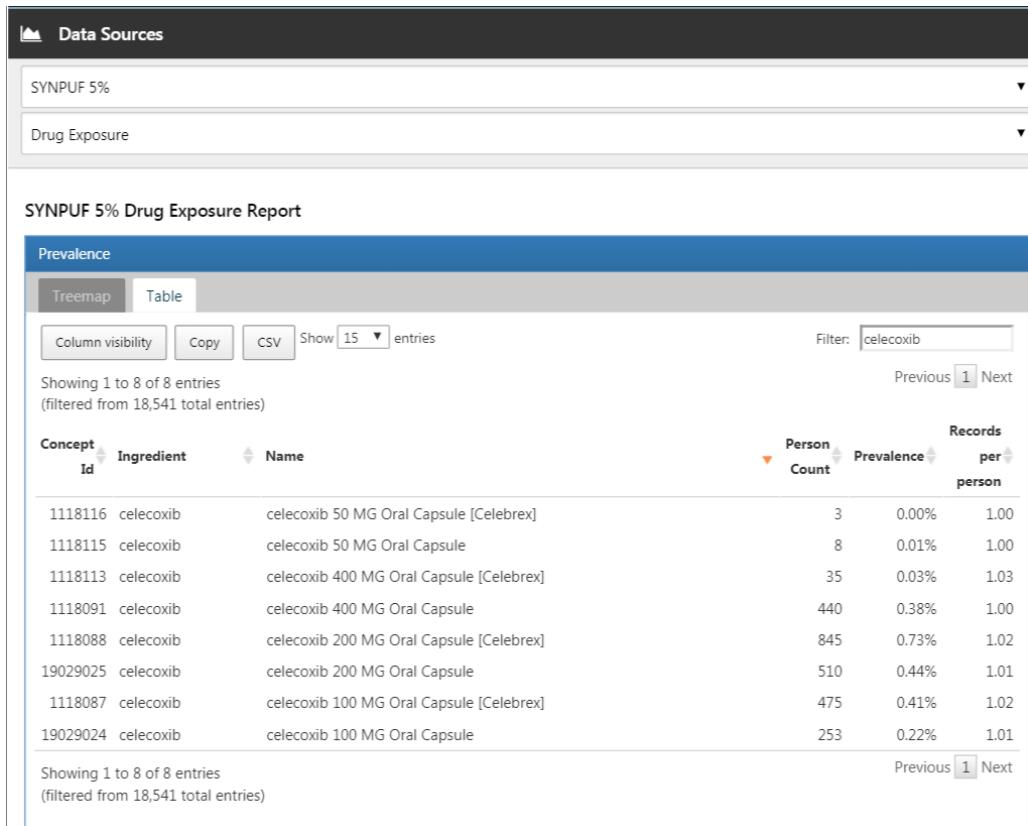


Figure E.8: データソースの特性評価

◀ Celecoxib new users ▶ Celecoxib

Search

Search Import

celecoxib 

Advanced Options

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

	Id	Code	Name	Class	RC	DRC	Domain	Vocabulary
	1118084	140587	celecoxib	Ingredient	2,587	5,184	Drug	RxNorm

Showing 1 to 1 of 1 entries Previous 1 Next

Vocabulary

- RxNorm Extension (1376)
- NDC (1337)
- SPL (449)
- DPD (167)
- SNOMED (75)

Class

- Ingredient (7)**
 - Clinical Drug Form (5)
 - Clinical Drug Comp (5)
 - Lab Test (5)
- Medicine (1)

Domain

- Drug (3570)
- Measurement (18)
- Observation (1)
- Meas Value (1)

Standard Concept

- Non-Standard (1831)
- Standard (1292)**
- Classification (467)

Figure E.9: 成分「Celecoxib (セレコキシブ)」の標準コンセプトの選択

に以前に作成したコンセプトセットを選択します。「Add attribute… (属性を追加…)」をクリックして「Add First Exposure Criteria (最初の曝露基準を追加)」を選択します。インデックス日の前に少なくとも 365 日の連続する観察期間が必要と設定します。結果は図 E.10 のようになります。選択基準、コホート離脱、コホート期間の選択はそのままにします。 をクリックしてコホート定義を保存することを忘れないでください。 をクリックして終了します。

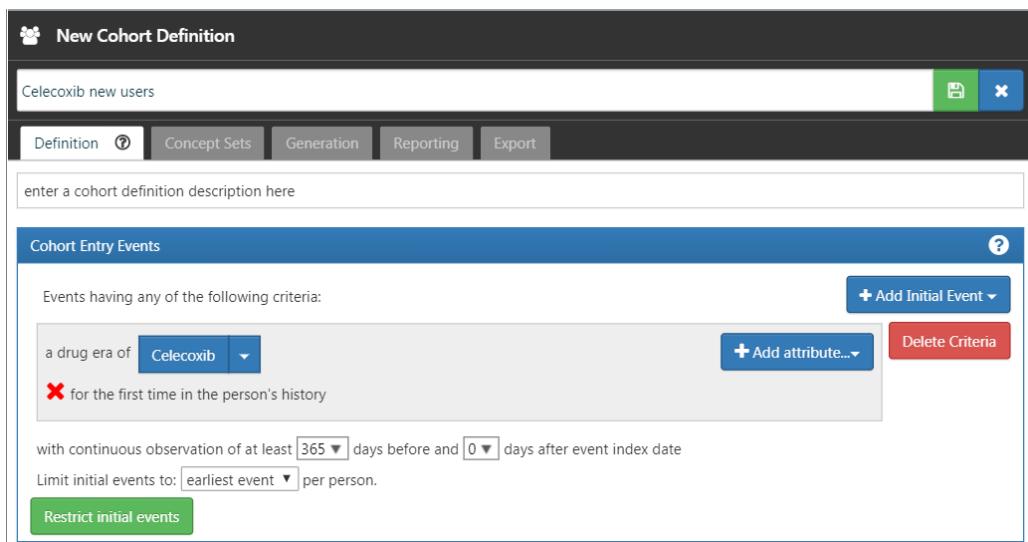


Figure E.10: セレコキシブ新規ユーザーの単純なコホート定義

コホートを定義したので、その特性評価ができます。 をクリックして「New Characterization (新規特性評価)」を選択します。特性評価に意味のある名前（例：「Celecoxib new users characterization (セレコキシブ新規ユーザーの特性評価)」）を付けます。コホート定義の下で、「Import」をクリックして最近作成したコホート定義を選択します。「Feature Analyses (特徴量分析)」の下で、「Import (インポート)」をクリックし、少なくとも 1 つのコンディション分析と 1 つの薬剤分析を選択します。たとえば、「Drug Group Era Any Time Prior (任意の期間前の薬剤グループ曝露期間)」と「Condition Group Era Any Time Prior (任意の期間前のコンディショングループ期間)」を選択します。特性評価の定義は図 E.11 のようになっているはずです。 をクリックして特性評価の設定を保存してください。

「Executions (実行)」タブをクリックし、1 つのデータソースについて「Generate (作成)」をクリックします。作成が完了するまで時間を要する場合があります。完了すると、「View latest results (最新の結果を表示)」をクリックできます。結果画面は、図 E.12 のように見えるはずで、図には例えば痛みや関節症が一般的に観察されることを示しており、これらはセレコキシブの適応症として意外ではないでしょう。リストの下の方には、予期しないコンディションが表示されることがあります。

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	Actions
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: 特性評価設定

Characterization #69

Celecoxib new users characterization

Design Executions Utilities

Executions > Reports for SYNPUF 5%

Date: 08/23/2019 12:53 PM Design: -1840810470 Results: 2 reports

Filter panel

Cohorts Analyses Domains

Celecoxib new users Condition Group Era Any Time P Condition, Drug

CONDITION / Condition Group Era Any Time Prior

Export Show 10 entries Search:

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: 特性評価の結果

演習 11.3

Cohort Definitions をクリックして「New cohort (新規コホート)」を作成します。コホートに意味が分かりやすい名前（例：「GI bleed (消化管出血)」）を付け、「Concept Sets (コンセプトセット)」タブに移動します。「New Concept Set (新しいコンセプトセット)」をクリックし、コンセプトセットに意味のある名前（例：「GI bleed (消化管出血)」）を付けます。 Search モジュールを開き、「Gastrointestinal hemorrhage (消化管出血)」を検索し、一番上のコンセプトの横にある をクリックしてコンセプトセットにコンセプトを追加します（図 E.13 参照）。

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)								
DRG (2)								
NIDERT (1)								
Class	4100660	27719009	Acute gastrointestinal hemorrhage	Clinical Finding	0	9,852	Condition	SNOMED
Clinical Finding (17)								

Figure E.13: ”Gastrointestinal hemorrhage (消化管出血)” の標準コンセプトの選択

図 E.13 の左最上部に表示されている左矢印をクリックしてコホート定義に戻ります。「Concept Sets (コンセプトセット)」タブを再度開き、消化管出血のコンセプトの横にある “Descemdamts (下位層)” をチェックします（図 E.14 参照）。

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"/>

Classification Non-Standard Standard

Figure E.14: ”Gastrointestinal hemorrhage (消化管出血)” のすべての下位層を追加

“Definition (定義)” タブに戻り、“+Add Initial Event (初回イベントを追加)” をクリックしてから “Add Condition Occurrence (コンディション出現を追加)” をクリックします。先に作成したコンディション出現基準に関するコンセプトセットを選択します。結果は図 E.15 のようになっているはずです。選択基準、コホート離脱、コホート期間のセクションはそのままにします。 をクリックしてコホート定義を保存し、 をクリックして終了します。

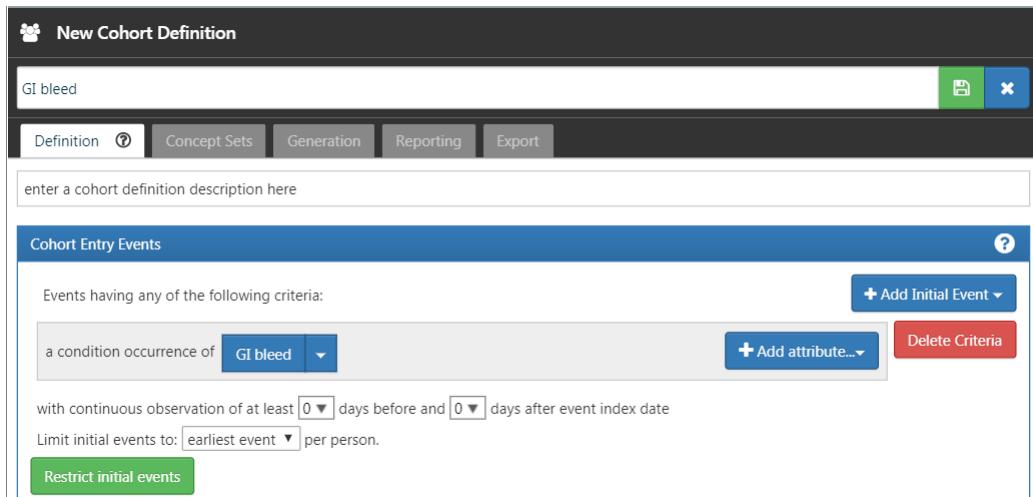


Figure E.15: 単純な消化管出血コホート定義

これでコホートが定義されたので、発生率が計算できます。 **Incidence Rates** をクリックして「New Analysis (新規分析)」を選択します。分析に意味の通じる名前（例：「Incidence of GI bleed after celecoxib initiation (セレコキシブ開始後の消化管出血発生率)」）を付けます。「Add Target Cohort (ターゲットコホートを追加)」をクリックし、セレコキシブ新規ユーザーコホートを選択します。「Add Outcome Cohort (アウトカムコホートを追加)」をクリックし、作成した消化管出血コホートを追加します。リスク期間の終了を開始日から 1095 日後に設定します。分析は図 E.16 のようになっているはずです。 をクリックして分析設定を保存してください。

「Generation (作成)」タブをクリックし、「Generation (作成)」をクリックします。データソースを 1 つ選択し、「Generation (作成)」をクリックします。完了すると、計算された発生率と発生割合が表示されます（図 E.17 参照）。

E.8 集団レベルの推定

演習 12.1

デフォルトの共変量セットを指定しますが、比較している 2 つの薬剤を含むすべての下位層を除外しなければなりません。そうしないと、傾向スコアモデルが完全に予測可能になってしまいます：

New Incidence Rate Analysis

Incidence of GI bleed after celecoxib initiation

Definition Concept Sets Generation Utilities

Study Cohorts

Target Cohorts	Outcome Cohorts
#1771701:Celecoxib new users	#1771702:GI bleed

Add Target Cohort Add Outcome Cohort

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

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 Please select a qualifying inclusion criteria to edit.

Figure E.16: 発生率の分析

Showing target cohort: Celecoxib new users and outcome cohort: GI bleed

Generate Export Analysis to CSV

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

Figure E.17: 発生率の結果

```
library(CohortMethod)
nsaids <- c(1118084, 1124300) # セレコキシブ、ジクロフェナク
covSettings <- createDefaultCovariateSettings(
  excludedCovariateConceptIds = nsaids,
  addDescendantsToExclude = TRUE)

# データの読み込み：
cmData <- getCohortMethodData(
  connectionDetails = connectionDetails,
  cdmDatabaseSchema = "main",
  targetId = 1,
  comparatorId = 2,
  outcomeIds = 3,
  exposureDatabaseSchema = "main",
  exposureTable = "cohort",
  outcomeDatabaseSchema = "main",
  outcomeTable = "cohort",
  covariateSettings = covSettings)
summary(cmData)
```

```
## CohortMethodData オブジェクトのまとめ
##
## 治療群のコンセプトID：1
## 比較群のコンセプトID：2
## アウトカムコンセプトID：3
##
## 治療群の人数：1800
## 比較群の人数：830
##
## アウトカムのカウント：
##   イベント 数 人数
## 3       479 479
##
## 共変量：
## 共変量の数：389
## 非ゼロ共変量値の数：26923
```

演習 12.2

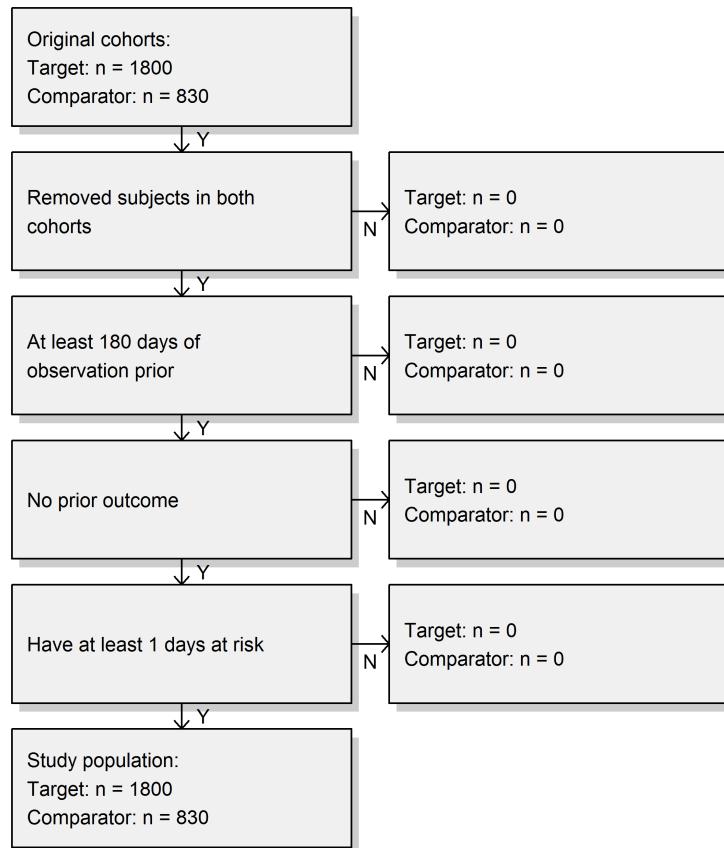
仕様に従って研究対象集団を作成し、脱落を示す図を出力します：

```
studyPop <- createStudyPopulation(
  cohortMethodData = cmData,
  outcomeId = 3,
  washoutPeriod = 180,
```

```

removeDuplicateSubjects = "remove all",
removeSubjectsWithPriorOutcome = TRUE,
riskWindowStart = 0,
startAnchor = "cohort start",
riskWindowEnd = 99999)
drawAttritionDiagram(studyPop)

```



元のコホートと比較して研究対象が失われなかったことが見てとれます。なぜなら、ここで使用した限定要件がすでにコホートの定義に適用されているためです。

演習 12.3

Cox 回帰モデルを使用して単純なアウトカムをフィットさせます：

```
model <- fitOutcomeModel(population = studyPop,
                           modelType = "cox")
model

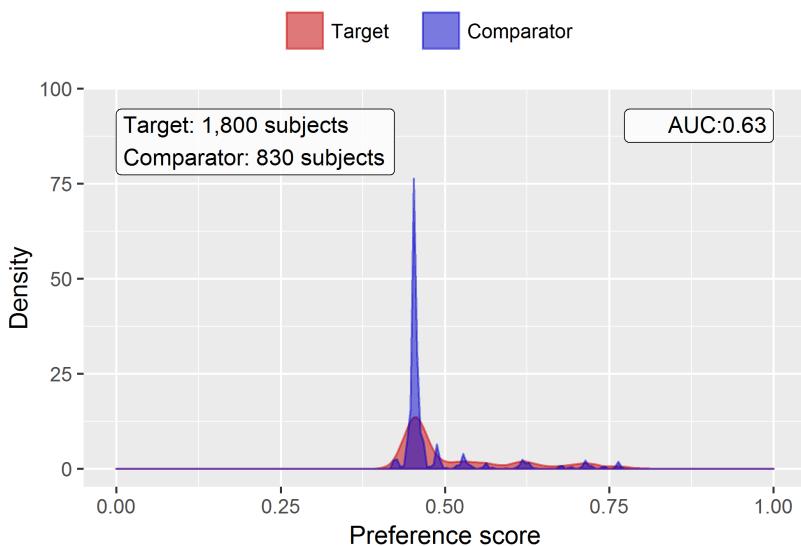
## モデルタイプ : cox
## 階層化 : FALSE
## 共変量の使用 : FALSE
## 反トリートメント重み付けの使用 : FALSE
## ステータス : OK
##
##      推定値 下限 .95 上限 .95 ログ相対リスク ログ相対リスク標準誤差
## 治療    1.34612   1.10065   1.65741          0.29723          0.1044
```

セレコキシブのユーザーとジクロフェナクのユーザーが交換可能でない可能性が高く、ベースラインの違いにより、すでにアウトカムのリスクが異なる可能性があります。この分析のようにこれらの違いを調整しない場合、バイアスのある推定値が計算される可能性があります。

演習 12.4

抽出したすべての共変量を使用し、研究集団に対して傾向スコアモデルをフィットさせます。その後、傾向スコアの分布を示します：

```
ps <- createPs(cohortMethodData = cmData,
                 population = studyPop)
plotPs(ps, showCountsLabel = TRUE, showAucLabel = TRUE)
```



この分布には、いくつかのスパイクがあり少し奇妙に見えることに注意してく

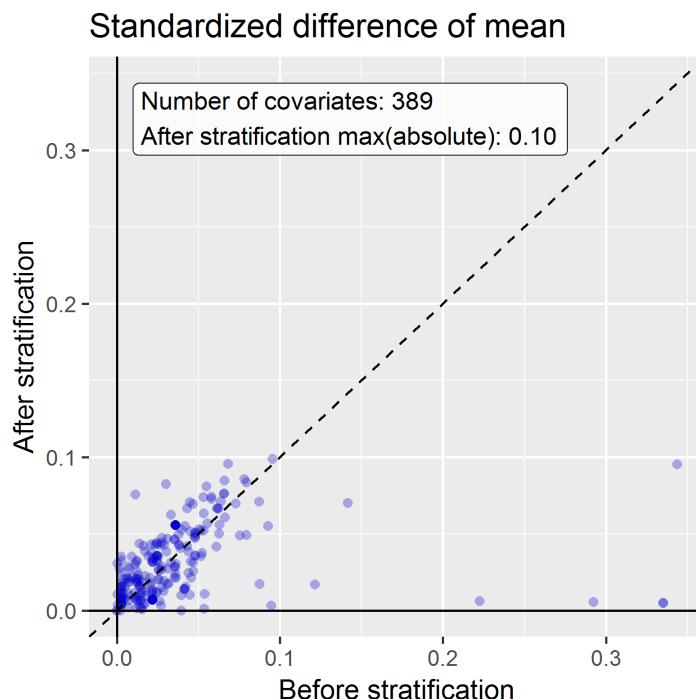
ださい。これは、非常に小さなシミュレーションデータセットを使用しているためです。実際の傾向スコアの分布はより滑らかであることが通常です。

傾向スコアモデルは 0.63 の AUC を達成し、ターゲットのコホートと比較群コホートの間に違いがあることを示唆しています。両グループの間にかなりの重複があることがわかり、傾向スコア調整によって両方をより比較可能にすることができます。

演習 12.5

傾向スコアに基づいて集団を層別化し、層別化前後の共変量バランスを計算します：

```
strataPop <- stratifyByPs(ps, numberOfStrata = 5)
bal <- computeCovariateBalance(strataPop, cmData)
plotCovariateBalanceScatterPlot(bal,
                                showCovariateCountLabel = TRUE,
                                showMaxLabel = TRUE,
                                beforeLabel = " 層別化前",
                                afterLabel = " 層別化後")
```



さまざまなベースライン共変量が、層別化前（x 軸）には大きな (>0.3) 標準化平均差 (standardized mean difference, SMD) を示していることがわかります。層別化後のバランスが向上し、標準化平均差は最大でも ≤ 0.1 です。

演習 12.6

傾向スコアによる層別化し、Cox 回帰モデルをフィットさせます：

```
adjModel <- fitOutcomeModel(population = strataPop,
                             modelType = "cox",
                             stratified = TRUE)
adjModel

## モデルタイプ : cox
## 階層化 : TRUE
## 共変量の使用 : FALSE
## 逆確率重みづけ(IPTW)の使用 : FALSE
## ステータス : OK
##
##          推定値 下限 .95 上限 .95 ログ相対リスク ログ相対リスク標準誤差
## 治療    1.13211   0.92132   1.40008           0.12409      0.1068
```

調整後の推定値が未調整の推定値より低くなり、95%信頼区間が 1 を含むようになりましたことがわかります。これは、2 つの曝露群間のベースラインの違いを調整することによって、バイアスを減ったためです。

E.9 患者レベルの予測

演習 13.1

共変量設定のセットを指定し、データベースからデータを抽出するために getPlpData 関数を使用します：

```
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 オブジェクトのまとめ
##
## At riskであるコホートコンセプト ID : -1
## アウトカムコンセプトID : 3
##
## 人数 : 2630
##
## 結果のカウント :
##   イベント 数 人数
## 3      479    479
##
## 共変量 :
## 共変量の数 : 245
## 非ゼロ共変量値の数 : 54079

```

演習 13.2

関心の対象であるアウトカムの研究対象集団を（この場合は抽出したデータに対して唯一のアウトカム）を作成します。ここでは、NSAID を使用し始める前にそのアウトカムが出現した対象を除外し、364 日のリスク期間があることを必要条件とします。

```

population <- createStudyPopulation(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

```

この場合、アウトカムがすでに出現した対象を除外することと 364 日以上のリスク期間を要求することにより、数名が失われました。

演習 13.3

LASSO モデルを実行するために、まずモデル設定オブジェクトを作成し、その後 runPlp 関数を呼び出します。この場合、人単位で分割し、データの 75% をトレーニングデータとして使用し、25% をデータで評価します。：

```
lassoModel <- setLassoLogisticRegression(seed = 0)

lassoResults <- runPlp(population = population,
                        plpData = plpData,
                        modelSettings = lassoModel,
                        testSplit = 'person',
                        testFraction = 0.25,
                        nfold = 2,
                        splitSeed = 0)
```

この例では、LASSO のクロスバリデーションと訓練用・テスト用データ分割の両方に対して乱数によるシード値を設定し、複数回の実行で結果が同じになるようにしていることに注意してください。

Shiny アプリを使用して結果を表示することができます：

```
viewPlp(lassoResults)
```

これにより、図 E.18に示されるようにアプリが起動されます。ここで、テスト用データセットの AUC が 0.645 であり、ランダムな推測よりも優れているものの、臨床における実装には十分ではないかもしれないことがわかります。

E.10 データ品質

演習 15.1

ACHILLES を実行するには：

```
library(ACHILLES)
result <- achilles(connectionDetails,
                      cdmDatabaseSchema = "main",
                      resultsDatabaseSchema = "main",
                      sourceName = "Eunomia",
                      cdmVersion = "5.3.0")
```

The screenshot shows a Shiny application interface titled "PatientLevelPrediction Explorer". The top navigation bar includes tabs for "Internal Validation" and "External Validation". Below the navigation is a sub-navigation bar with tabs: "Evaluation Summary" (which is selected and highlighted in blue), "Characterization", "ROC", "Calibration", "Demographics", "Preference", "Box Plot", and "Settings". The main content area is titled "Evaluation Summary" and contains a table of metrics. The table has columns for "Metric", "test", and "train". The rows show the following data:

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

There are also buttons to "Show 25 entries" and a search bar.

Figure E.18: Shiny アプリによる患者レベルの予測の表示

演習 15.2

データ品質ダッシュボード (Data Quality Dashboard, DQD) を実行するには:

```
DataQualityDashboard::executeDqChecks(
  connectionDetails,
  cdmDatabaseSchema = "main",
  resultsDatabaseSchema = "main",
  cdmSourceName = "Eunomia",
  outputFolder = "C:/dataQualityExample")
```

演習 15.3

データ品質チェックのリストを見るには:

```
DataQualityDashboard::viewDqDashboard(
  "C:/dataQualityExample/Eunomia/results_Eunomia.json")
```

E.11

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.
- 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., Bertoch, 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., Hripcak, 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.
- Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1 edition.
- 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 Academ 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.

- Ramcharran, 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., Branen, 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 Flraig, 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., Ramcharran, 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., Hripcak, 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., Hripcak, 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., Hripcak, 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., 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., Hripcsak, 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., Hripcsak, 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 Vandebroucke, 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. (2018). 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., Person, 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.
- 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

- ACE 阻害薬, 270
ACHILLES, 315
adaboost, 264
agnostic SQL, see SqlRender
analysis implementation, 115
APHRODITE, 164
arachne, 403
ATHENA, 64
ATLAS, 118, 273
 characterization features, 199
 インストール, 120
 コホートパスウェイ, 119
 コホート定義, 119
 コンセプトセット, 118
 ジョブ, 119
 セキュリティ, 120
 データソース, 118
 ドキュメント, 120
 フィードバック, 120
 プロファイル, 119
 ボキャブラリ検索, 118
 患者レベル予測, 119
 特性の評価, 119
 発生率, 119
 設定, 119
 集団レベル推定, 119
attrition diagram, 251
AUC, 269
- back-propagation, 266
balance, see covariate balance
baseline time, 187
best practice for network research,
 404
- between-database heterogeneity, 367
bigknn, 264
Bill of Mortality, 61
- caliper, 219
 scale, 234
- case-control design, 220
- case-crossover design, 221
- case-time-control design, 221
- CDM , see Common Data Model
- characterization, 109, 187
 cohort, 188
 database level, 188
 treatment pathways, 188
- classification concept, 68
- Clem McDonald, 309
- clinical equipoise, 219
- cohort
 entry event, 162
 exit criteria, 162
 inclusion criteria, 162
 probabilistic design, 164
 rule-based design, 161
- cohort method, 216
- colliders, 219
- Common Data Model, 33
 スケーラビリティ, 35
 ソースコード, 35
 デザインの原則, 34
 データモデル図, 34
 データ保護, 35
 データ損失防止, 35
 ドメイン, 35

- 後方互換性, 35
- 技術の中立性, 35
- 目的適合性, 35
- community, 7, 310
- comparative effect estimation, 215
- comparative effectiveness, see comparative effect estimation
- comparator cohort, 216
- concept, 64
 - class, 67
 - code, 70
 - hierarchy, 74
 - identifier, 65
 - mapping, 72
 - relationship, 72
- concept set, 163
- conditioned model, 236
- confidence interval calibration, 366
- confounder, 217
- control hypotheses, 122
- convolutional neural network, 266
- counterfactual, 215
- covariate balance, 219
 - example, 249
- Cox proportional hazards model, see Cox regression
- Cox regression, 217
- cross-validation, 263
- Cyclops, 263
- data profiling, see White Rabbit
- data quality, 313
- DatabaseConnector, 132
 - creating a connection, 141
 - querying, 142
- decision boundary, 261
- decision tree, 265
- deep learning, 266
- descriptive statistics, see characterization
- design considerations for network research, 398
- direct effect estimation, 215
- disease natural history, see characterization
- domain
 - concept, 66
- drug utilization, 188
- ETL, see 抽出、変換、ロード (ETL)
- implementations, 100
- quality control, 101
- 単体テスト, 318
- ETL design, see Rabbit-In-A-Hat
- evidence quality, 307
- FAIR, 27
- feature analyses, 195
- FeatureExtraction, 201
- forum, 13
- gradient boosting, 263
- high correlation, 239
- hyper-parameter, 263
- incidence, 189
 - proportion, 190
 - rate, 190
- index date, 187
- instrumental variables, 219
- k-nearest neighbors, 264
- Kaplan-Meier plot, 252
- LASSO, 263
- logistic regression, 217, 263
- logistical considerations for an OHDSI network study, 399
- methods library, 120
- minimum detectable relative risk (MDRR), 251
- mission, 6
- model viewer app, 293
- naive bayes, 264
- nesting cohort
 - case-control design, 220
- network study, 396
- neural network, 266
- no free lunch, 262

- objectives, 7
- OHDSI Methods Benchmark, 377
- OHDSI SQL, see SqlRender
- outcome cohort
 - case-control design, 220
 - case-crossover design, 221
 - cohort method, 216
 - SCCS design, 222
 - self-controlled cohort design, 220
- p-value calibration, 365
- Pallas system, 63
- patient-level prediction, 111
- perceptron, 266
- person-time, 190
- phenotype library, 165
- PheEvaluator, 338
- Poisson regression, 217
- population-level estimation, 110, 215
- post-index time, 187
- power, 251
- preference score, 219
 - example, 248
- propensity model, 218
 - example, 249
- propensity score, 217
 - matching, 218
 - stratification, 218
 - trimming, 233
 - weighting, 218
- python, 264, 265
- quality improvement, see characterization
- Query Library, 132
- QueryLibrary, 148
- R, 132
 - installation, 124
- Rabbit-In-A-Hat, 86
- random forest, 264
- randomized trial, 217
- recurrent neural networks, 266
- regularization, 263
- reliable evidence, 310
- ROC, 269
- running network research, 400
- safety surveillance, 215
- self-controlled case series (SCCS) design, 222
- self-controlled cohort design, 220
- sensitivity analysis, 368
- sklearn, 264
- source code mapping, see Usagi
- SQL, 131
- SQL Query Library, see Query Library
- SqlRender, 132
 - debugging, 140
 - parameterization, 133
 - supported functions, 135
 - translation, 134
- standard concept, 68
- Standard SQL Dialect, see SqlRender
- standardized vocabularies, 61
 - download, 64
 - search, 64
- stratified model, conditioned model を参照236
- strongly ignorable, 218
- structured query language, see SQL
- study diagnostics, 390
- study feasibility
 - single study, 390
- study-a-thon, 25
- supervised learning(教師あり学習), 260
- survival plot, see Kaplan-Meier plot
- system requirements, 123
- target cohort
 - case-control design, 220
 - case-crossover design, 221
 - cohort method, 216
 - SCCS design, 222
 - self-controlled cohort design, 220
- tools deployment, 127

- Amazon AWS, 128
 Broadsea, 128
 treatment utilization, see characterization
 TRIPOD, 258
 Usagi, 93
 variable ratio matching, 218
 variance, 263
 vignette, 123
 vision, 6
 vocabulary, 67
 White Rabbit, 82
 workgroups, 15
 xgboost, 263
 アウトカムコホート, 258
 アウトカムステータス, 260
 インデックス日, 270
 オープンサイエンス, 23
 open standards, 25
 オープンソース, 26
 オープンディスクース, 27
 オープンデータ, 26
 キャリブレーション, 270
 クラス, 260
 コホート, 160
 コホート定義, 160
 コミュニティ
 コミュニティコール, 15
 コンセプト
 祖先, 76
 コードセット, 160
 ソースデータ, see 生データ
 ソースレコード検証, 335
 データ品質
 研究特有のチェック, 320
 チェック, 315
 バリデーション, 315
 妥当性, 315
 完全性, 315
 検証, 315
 準拠, 315
 ネイティブデータ, *see* 生データ
 ネガティブコントロール, 361
 プロトコル, 384
 ポジティブコントロール, 362
 ラベル, 260
 リスク期間, 258
 リレーションナルデータモデル, *see* Common Data Model
 予後アウトカム, 257
 予測モデル, 257
 予測モデルの評価, 266
 交差検証, 267
 偽陰性, 268
 偽陽性, 268
 共変量, 260
 共通データモデル
 標準化テーブル, 42
 規約, 35
 基準日, 260
 実証的キャリブレーション, 365
 実証的評価, 364
 対象コホート, 258
 性能指標, 267
 患者レベルの予測, 257
 感度, 267
 方法の妥当性, 359
 検証
 内部検証, 266
 外部検証, 266
 時間的検証, 267
 空間的検証, 267
 機械学習, 258
 欠損データ, 114, 261
 特異度, 267
 生データ, 81
 真陰性, 268
 真陽性, 268
 研究ネットワーク, 395
 研究パッケージ, 385
 研究診断, 360
 精度, 267

- 臨床意思決定, 257
 - 血管性浮腫, 270
 - 表現型, 160
 - 観察研究の限界, 113
 - 診断アウトカム, 257
 - 識別力, 269
- 適合率-再現率曲線下の面積, 269
- 陽性予測値, 267