



Reproducing Graham et al. 2014 using the CohortMethod package

Martijn Schuemie



CohortMethod package

ATLAS x Achilles: TRUVEN_CCAE x ATLAS x OHDSI Forums x OHDSI/CohortMethod: A x

GitHub, Inc. [US] | <https://github.com/OHDSI/CohortMethod>

Apps Reference JnJ Epi Analytics OHDSI EMIF Distractions Music Cloud dplyr M5x100 / per stuk - F Formatting Guide to Bayesian Methods: A Validation Auto release

This repository Search Pull requests Issues Marketplace Explore

OHDSI / CohortMethod Unwatch 54 Star 19 Fork 15

Code Issues 7 Pull requests 0 Insights Settings

An R package for performing new-user cohort studies in an observational database in the OMOP Common Data Model. Edit

Add topics

445 commits 9 branches 10 releases 7 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

msuchard trying repo slug

- .settings Mysterious new EOL prob
- R Fixed typo causing bound
- data The treatment variable
- extras Fixed typo causing bound
- inst Reran multi-analysis vign
- man Power calculations now a
- src Now using default select
- standalone/src Fixed end of line charact
- tests Added option to choose
- vignettes Minor edits to vignettes
- .Rbuildignore enable drat
- .cproject Mysterious new EOL problems

GitHub repository:

- Installation instructions
- Package manual
- Vignettes

<https://github.com/OHDSI/CohortMethod>

R script implementing the study

```
#####
# Study: OHDSI estimation tutorial: Graham replication: dabigatran vs warfarin for risk of ischemic stroke
#####
```

```
#####
# Cohort Method Installation & Load
#####
```

```
# Uncomment to install Cohort Method
# install.packages("drt")
# devtools::install("https://github.com/ohdsi/cohortMethod")
# install.packages("CohortMethod")
```

```
# Load the Cohort Method library
library(CohortMethod)
library(SqlRender)
```

```
#####
# Data extraction
#####
```

```
# Manually modified this
connectionDetails <- createConnectionDetails(dbs = "pdw",
                                             server = "JRDUASPSCT01",
                                             port = 17001)
```

```
cdmDatabaseSchema <- "CDM_Truen_MOCR_V415_dbo"
resultDatabaseSchema <- cdmDatabaseSchema
exposureTable <- "cohort"
outcomeTable <- "cohort"
cdmVersion <- "5"
```

```
options("Rtmpdir") = "%/Rtmp/"
workFolder <- "%/Rtmp/GrahamStudy"
```

```
#####
# TODO: NOTE: If you want to use your code in a distributed network study
# you will need to create a temporary cohort table with common cohort IDs.
# The code below ASSUMES you are only running in your local network
# where common cohort IDs have already been assigned in the cohort table.
#####
```

```
# Get all OHDSI estimation tutorial - Graham replication: covariates to exclude in PS model Concept IDs for exclusion
excludedConcepts <- c()
```

```
# Get all OHDSI estimation tutorial - Graham replication: covariates to include in PS model Concept IDs for inclusion
includedConcepts <- read.csv("inst/csv/GrahamIncludedConcepts.csv")$concept_id
```

```
# Define which types of covariates must be constructed
covariateSettings <- createCovariateSettings(useCovariateDemographics = TRUE,
                                           useCovariateDemographicGender = TRUE,
                                           useCovariateDemographicRace = TRUE,
                                           useCovariateDemographicActivity = TRUE,
                                           useCovariateDemographicAge = TRUE,
                                           useCovariateDemographicYear = FALSE,
                                           useCovariateDemographicMonth = FALSE,
                                           useCovariateConditionOccurrence = TRUE,
                                           useCovariateConditionOccurrence365d = TRUE,
                                           useCovariateConditionOccurrence30d = TRUE,
                                           useCovariateConditionOccurrence1800d = TRUE,
                                           useCovariateConditionEra = FALSE,
                                           useCovariateConditionEraEver = FALSE,
                                           useCovariateConditionEraOverlap = FALSE,
                                           useCovariateConditionGroup = TRUE,
                                           useCovariateConditionGroupMeddra = TRUE,
                                           useCovariateConditionGroupSynonym = FALSE,
                                           useCovariateDrugExposure = FALSE,
                                           useCovariateDrugExposure365d = FALSE,
                                           useCovariateDrugExposure30d = FALSE,
                                           useCovariateDrugEra = TRUE,
                                           useCovariateDrugEra365d = TRUE,
                                           useCovariateDrugEra30d = FALSE,
                                           useCovariateDrugEraOverlap = FALSE,
                                           useCovariateDrugEraEver = FALSE,
                                           useCovariateDrugEraOverlap = FALSE,
                                           useCovariateDrugGroup = TRUE,
                                           useCovariateProcedureOccurrence = FALSE,
                                           useCovariateProcedureOccurrence365d = FALSE,
                                           useCovariateProcedureOccurrence30d = FALSE,
                                           useCovariateProcedureGroup = FALSE,
                                           useCovariateObservation = FALSE,
                                           useCovariateObservation365d = FALSE,
                                           useCovariateObservation30d = FALSE,
                                           useCovariateObservationCount365d = FALSE,
                                           useCovariateMeasurement = FALSE,
                                           useCovariateMeasurement365d = FALSE,
                                           useCovariateMeasurement30d = FALSE,
                                           useCovariateMeasurementCount365d = FALSE,
```

```
useCovariateMeasurementBelow = FALSE,
useCovariateMeasurementAbove = FALSE,
useCovariateConceptCounts = FALSE,
useCovariateRiskScores = TRUE,
useCovariateRiskScoresCharlson = FALSE,
useCovariateRiskScoresDCHI = FALSE,
useCovariateRiskScoresCHAQD2 = TRUE,
useCovariateRiskScoresCHAQD2AS = FALSE,
useCovariateInteractionYear = FALSE,
useCovariateInteractionMonth = FALSE,
excludeCovariateConcepts = excludedConcepts,
includeCovariateConcepts = includedConcepts,
deleteCovariatesSmallCount = 100)
```

```
cohortMethodData <- getRxCohortMethodData(connectionDetails = connectionDetails,
                                           cdmDatabaseSchema = cdmDatabaseSchema,
                                           oracleTempSchema = resultDatabaseSchema,
                                           targetId = 2040,
                                           comparatorId = 2650,
                                           outcomeIds = 2651,
                                           studyStartDate = "",
                                           studyEndDate = "",
                                           exposureDatabaseSchema = resultDatabaseSchema,
                                           exposureTable = exposureTable,
                                           outcomeDatabaseSchema = resultDatabaseSchema,
                                           outcomeTable = outcomeTable,
                                           cdmVersion = cdmVersion,
                                           excludeDrugsFromCovariates = FALSE,
                                           covariateSettings = covariateSettings)
```

```
saveCohortMethodData(cohortMethodData, file.path(workFolder, "cohortMethodData"))
```

```
# cohortMethodData <- loadCohortMethodData(file.path(workFolder, "cohortMethodData"))
```

```
summary(cohortMethodData)
```

```
#####
# Defining the study population
#####
```

```
studyPop <- createStudyPopulation(cohortMethodData = cohortMethodData,
                                outcomeId = 2651,
                                firstExposureOnly = FALSE,
                                washoutPeriod = 0,
                                removeSubjectsWithPriorOutcome = TRUE,
                                minDaysAtRisk = 1,
                                riskWindowStart = 1,
                                addExposureDaysToStart = FALSE,
                                riskWindowEnd = 0,
                                addExposureDaysToEnd = TRUE)
```

```
saveRDS(studyPop, file.path(workFolder, "studyPop.rds"))
```

```
# Return the attrition table for the study population
getAttritionTable(studyPop)
```

```
# Return the attrition diagram
drawAttritionDiagram(studyPop, treatmentLabel = "Target", comparatorLabel = "Comparator")
```

```
#####
# Propensity scores
#####
```

```
# Fitting a propensity model
ps <- createP(cohortMethodData = cohortMethodData,
             population = studyPop,
             control = createControl(type = "auto",
                                    startingVariance = 0.01,
                                    noiseLevel = "quiet",
                                    tolerance = 2e-07,
                                    cOfRegression = 10,
                                    threads = 20))
```

```
saveRDS(ps, file.path(workFolder, "ps.rds"))
```

```
# ps <- readRDS(file.path(workFolder, "ps.rds"))
# Compute the area under the receiver operator curve (AUC) for the propensity score model
computePauc(ps)
```

```
# Plot the propensity score distribution
plotP(ps, treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin", fileName = file.path(workFolder, "ps.png"))
```

```
# Inspect the propensity model
propensityModel <- getPModel(ps, cohortMethodData)
head(propensityModel)
```

```
matchedPop <- matchOnP(ps, caliper = 0.25, caliperScale = "standardized", maxRatio = 1)
```

```
plotP(matchedPop, ps)
```

```
# Get the attrition table and diagram
getAttritionTable(matchedPop)
drawAttritionDiagram(matchedPop)
```

```
# Compute and plot the covariate balance
balance <- computeCovariateBalance(matchedPop, cohortMethodData)
plotCovariateBalanceCovariatePlot(balance)
plotCovariateBalanceCovariatePlot(balance)
```

```
#####
# Outcome Model
#####
```

```
# Get all Concept IDs for exclusion:
omExcludedConcepts <- c()
```

```
# Get all Concept IDs for inclusion:
omIncludedConcepts <- c()
```

```
outcomeModel <- fitOutcomeModel(cohortMethodData = cohortMethodData,
                                population = matchedPop,
                                stratified = FALSE,
                                modelType = "cox",
                                useCovariates = FALSE,
                                includeCovariates = omIncludedConcepts,
                                excludeCovariates = omExcludedConcepts)
```

```
# Display the outcome model
outcomeModel
```

```
summary(outcomeModel)
```

```
# Generate the Kaplan Meier plot
plotKaplanMeier(matchedPop, treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin", includeZero = FALSE, fileName = file.path(workFolder, "kaplanMeier.png"))
```



Step 1: Getting the necessary data from the database

- Target and comparator cohorts of interest
 - Index date
 - End of cohort eligibility (e.g. end of exposure)
 - End of observation
- Outcome(s) of interest
 - Dates of occurrence
- Covariates

Take as-is from ATLAS



Covariates (Graham et al.)

Graham et al. used several covariates, including

- Demographics
- Medical history
 - Various conditions
 - CHADS
 - HAS-BLED
- Medication use

These were hand-picked for this study



Connecting to the database

```
connectionDetails <- createConnectionDetails(dbms = "pdw",  
                                              server = "JRDUSAPCTL01",  
                                              port = 17001)  
  
cdmDatabaseSchema <- "CDM_Truven_MDCR_V415.dbo"  
cdmVersion <- "5"
```



Loading the list of selected covariate concept IDs

```
includedConcepts <- read.csv("inst/csv/GrahamIncludedConcepts.csv")$Concept.ID
```



Covariate settings

```
covariateSettings <- createCovariateSettings(useCovariateDemographics = TRUE,  
                                             useCovariateDemographicsGender = TRUE,  
                                             useCovariateDemographicsRace = TRUE,  
                                             useCovariateDemographicsEthnicity = TRUE,  
                                             useCovariateDemographicsAge = TRUE,  
                                             useCovariateDemographicsYear = FALSE,  
                                             useCovariateDemographicsMonth = FALSE,  
                                             ...  
                                             useCovariateRiskScores = TRUE,  
                                             useCovariateRiskScoresCharlson = FALSE,  
                                             useCovariateRiskScoresDCSI = FALSE,  
                                             useCovariateRiskScoresCHADS2 = TRUE,  
                                             useCovariateRiskScoresCHADS2VAsC = FALSE,  
                                             excludedCovariateConceptIds = c(),  
                                             includedCovariateConceptIds = includedConcepts,  
                                             deleteCovariatesSmallCount = 100)
```

Including only the concept IDs
we just loaded



Loading all data

```
cohortMethodData <- getDbCohortMethodData(connectionDetails = connectionDetails,  
                                           cdmDatabaseSchema = cdmDatabaseSchema,  
                                           targetId = 2649,  
                                           comparatorId = 2650,  
                                           outcomeIds = 2651,  
                                           exposureTable = "cohort",  
                                           outcomeTable = "cohort",  
                                           cdmVersion = cdmVersion,  
                                           excludeDrugsFromCovariates = FALSE,  
                                           covariateSettings = covariateSettings)  
  
saveCohortMethodData(cohortMethodData, "s:/GrahamStudy/cohortMethodData")
```



Loading all data

```
cohortMethodData <- getDbCohortMethodData(connectionDetails = connectionDetails,  
                                           cdmDatabaseSchema = cdmDatabaseSchema,  
                                           targetId = 2649,  
                                           comparatorId = 2650,  
                                           outcomeIds = 2651,  
                                           exposureTable = "cohort",  
                                           outcomeTable = "cohort",  
                                           cdmVersion = cdmVersion,  
                                           excludeDrugsFromCovariates = FALSE,  
                                           covariateSettings = covariateSettings)  
  
saveCohortMethodData(cohortMethodData, "cohortMethodData")
```

Using the cohorts generated by ATLAS



Loading all data

```
cohortMethodData <- getDbCohortMethodData(connectionDetails = connectionDetails,  
                                           cdmDatabaseSchema = cdmDatabaseSchema,  
                                           targetId = 2649,  
                                           comparatorId = 2650,  
                                           outcomeIds = 2651,  
                                           exposureTable = "cohort",  
                                           outcomeTable = "cohort",  
                                           cdmVersion = cdmVersion,  
                                           excludeDrugsFromCovariates = FALSE,  
                                           covariateSettings = covariateSettings)
```

```
saveCohortMethodData(cohortMethodData, connectionDetails = connectionDetails,
```

The covariate settings we specified earlier



Inspecting the data

```
summary(cohortMethodData)
```

CohortMethodData object summary

Treatment concept ID: 2649

Comparator concept ID: 2650

Outcome concept ID(s): 2651

Treated persons: 19046

Comparator persons: 50918

Outcome counts:

	Event count	Person count
2651	8063	8063

Covariates:

Number of covariates: 827

Number of non-zero covariate values: 2428813



Step 2: Defining the study population

```
studyPop <- createStudyPopulation(cohortMethodData = cohortMethodData,  
                                outcomeld = 2651,  
                                removeSubjectsWithPriorOutcome = TRUE,  
                                minDaysAtRisk = 1,  
                                riskWindowStart = 1,
```

Remove subjects with outcomes prior to
index date



Step 2: Defining the study population

```
studyPop <- createStudyPopulation(cohortMethodData = cohortMethodData,  
                                  outcomeld = 2651,  
                                  removeSubjectsWithPriorOutcome = TRUE,  
                                  minDaysAtRisk = 1,  
                                  riskWindowStart = 1,  
                                  addExposureDaysToStart = FALSE,  
                                  riskWindowEnd = 0,  
                                  addExposureDaysToEnd = TRUE)
```

Define risk window: start on day after index date, end at end of cohort eligibility (end of exposure)

Graham: "Follow-up began on the day after the first qualifying anticoagulant prescription fill"



Step 3: Creating a propensity model

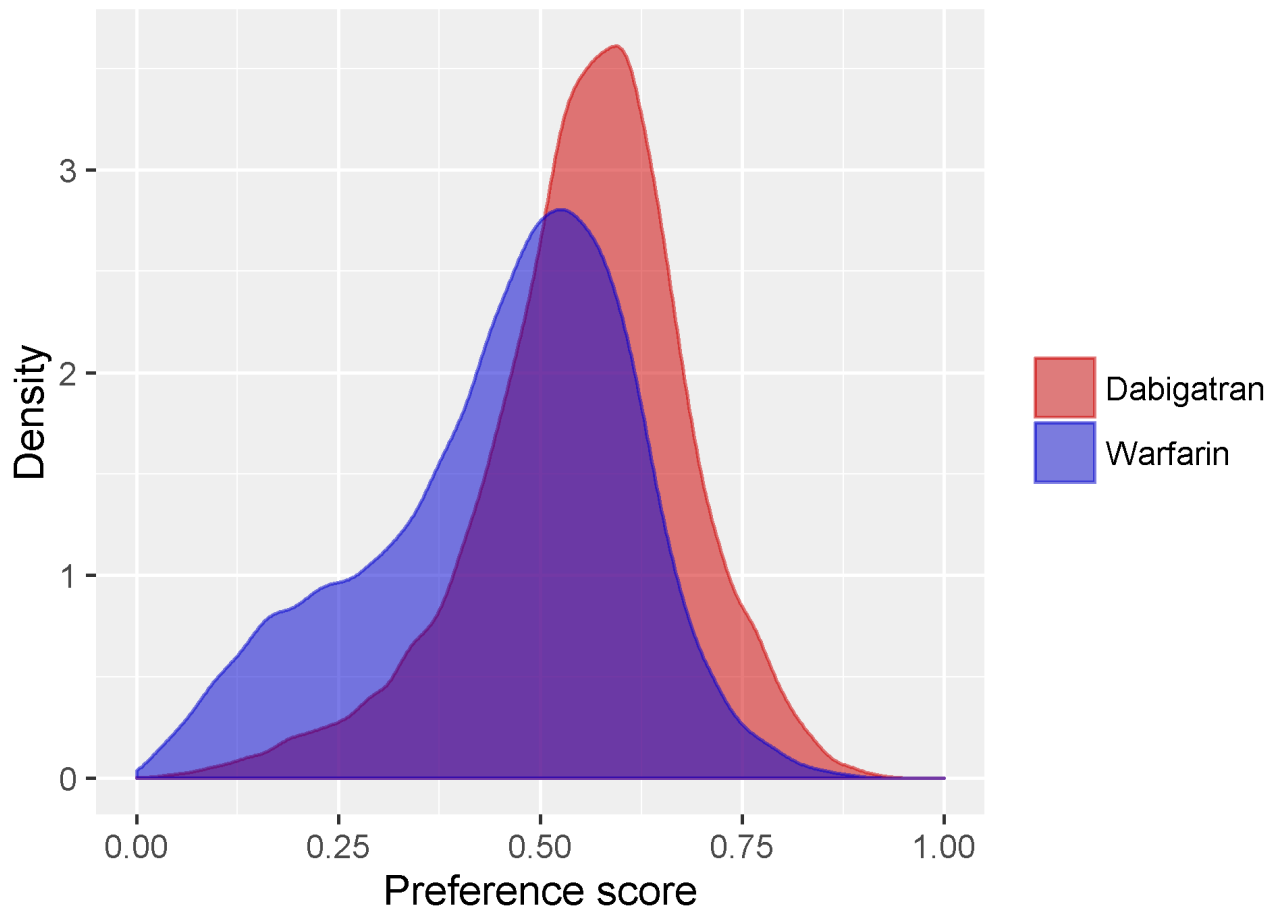
```
ps <- createPs(cohortMethodData = cohortMethodData,  
               population = studyPop)
```

```
saveRDS(ps, "s:/GrahamStudy/ps.rds")
```



Plot propensity score distribution

```
plotPs(ps , treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin")
```





Step 4: Matching

```
matchedPop <- matchOnPs(ps,  
  caliper = 0.25,  
  caliperScale = "standardized",  
  maxRatio = 1)
```

Select up to 1 comparator per target subject
(1-on-1 matching)

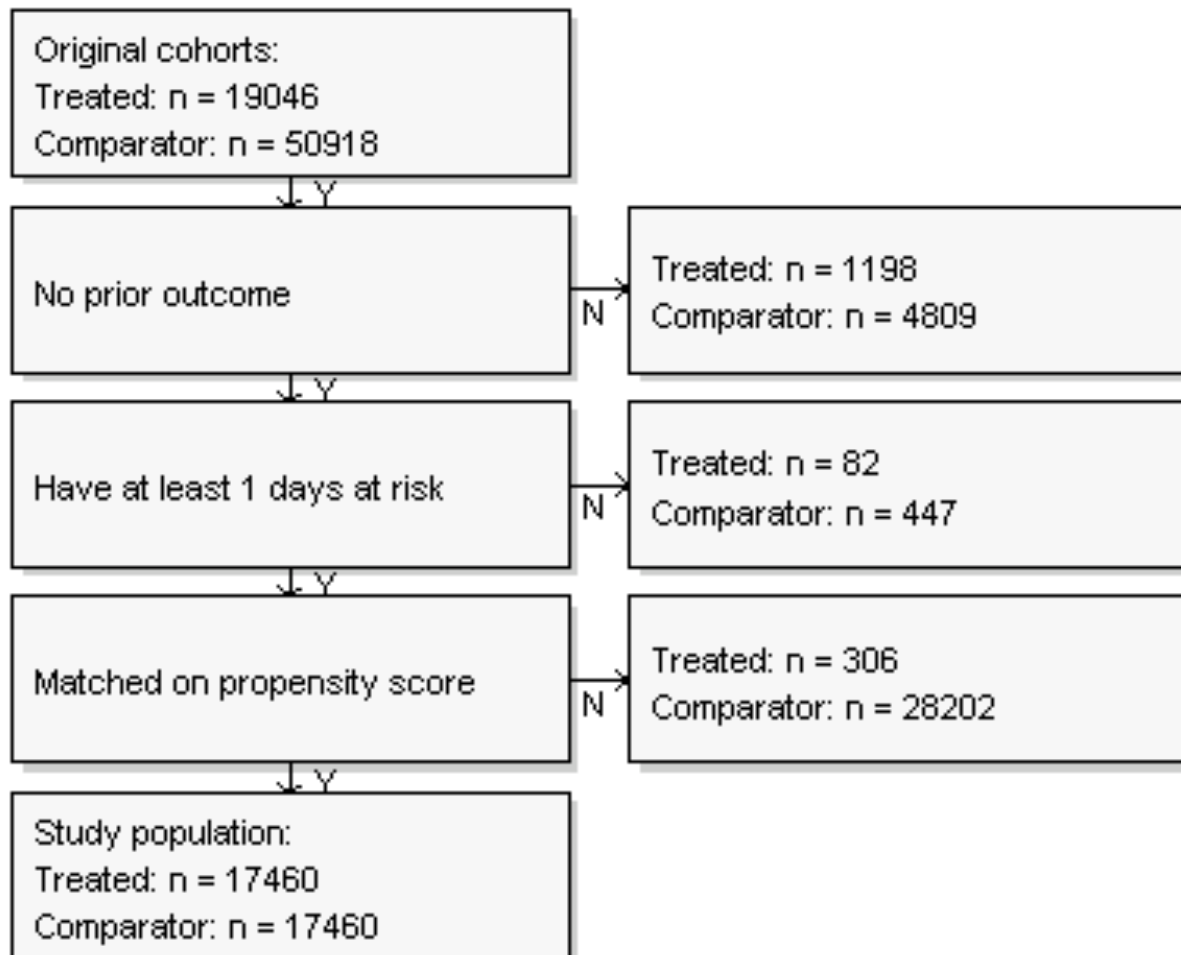
Graham: “Dabigatran users were propensity score matched to warfarin users in a 1:1 ratio with the use of a greedy matching algorithm.”

No caliper was mentioned, but it is probably a good idea to use one



Attrition

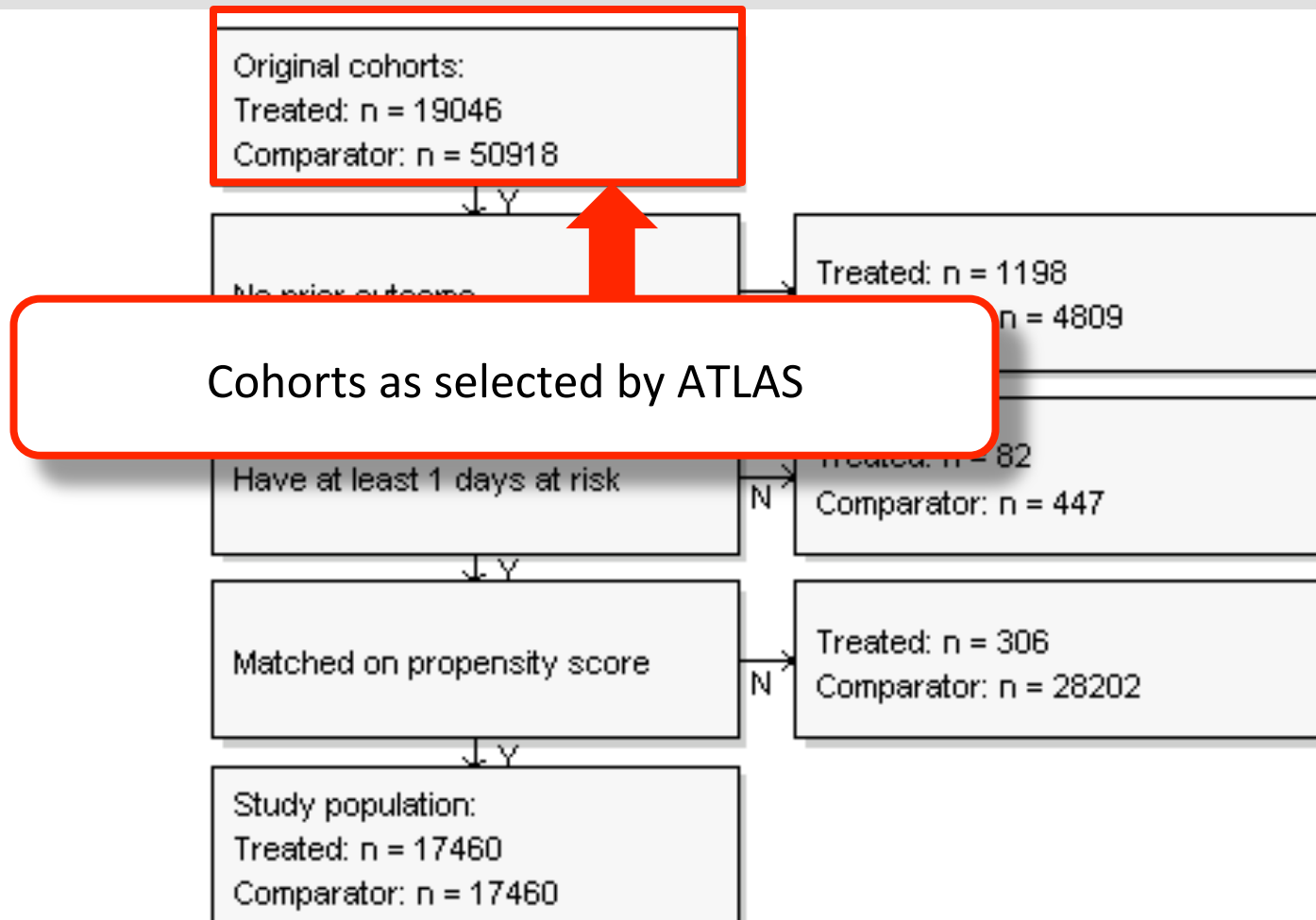
```
drawAttritionDiagram(matchedPop , treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin")
```





Attrition

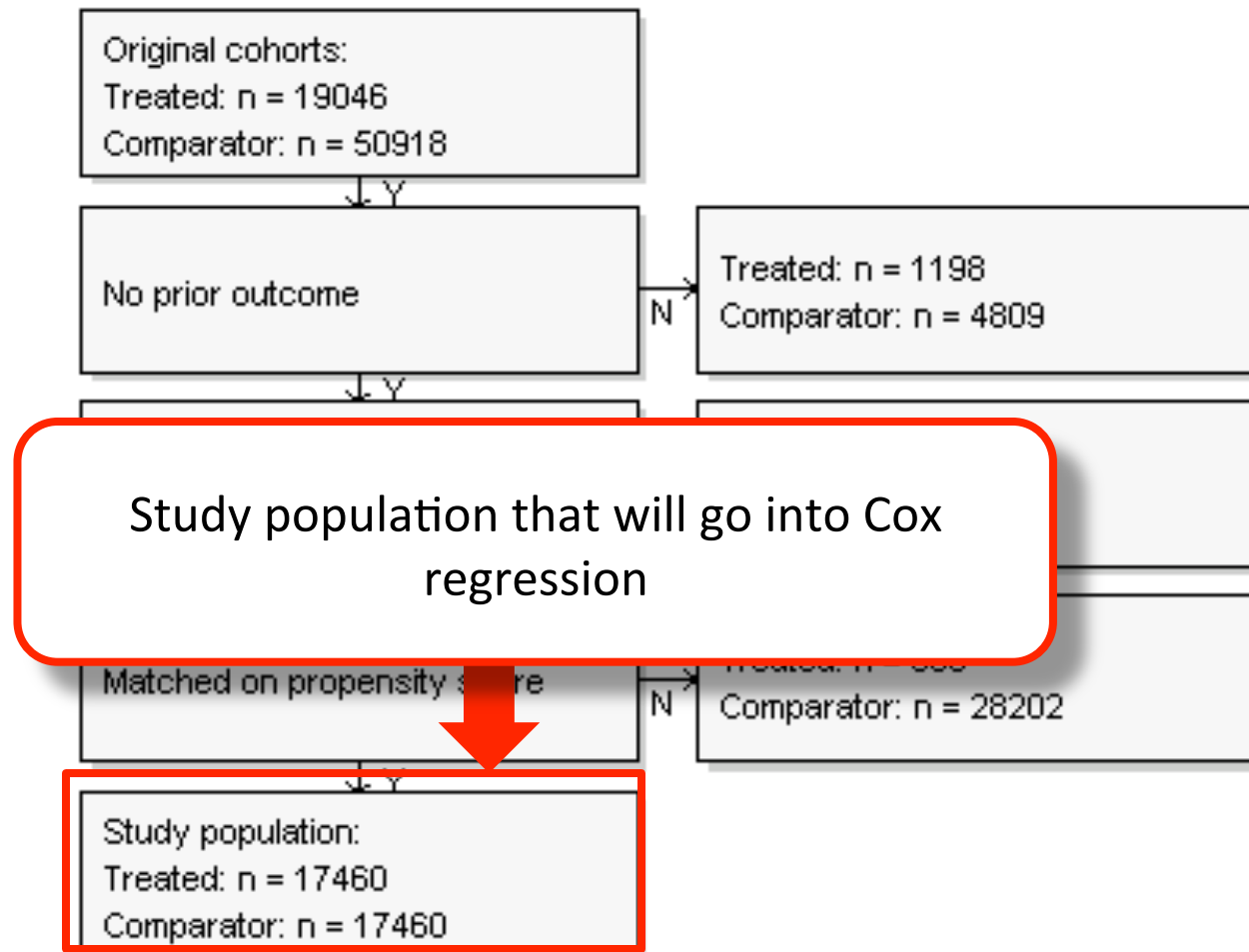
```
drawAttritionDiagram(matchedPop , treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin")
```





Attrition

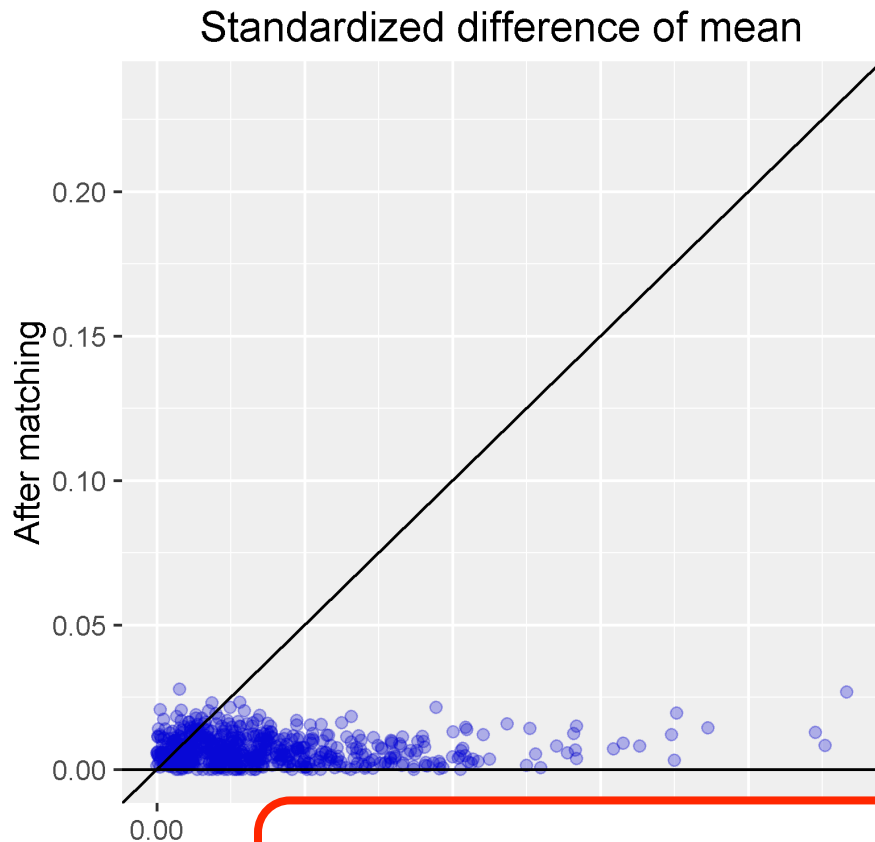
```
drawAttritionDiagram(matchedPop , treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin")
```





Covariate balance

```
balance <- computeCovariateBalance(matchedPop, cohortMethodData)  
plotCovariateBalanceScatterPlot(balance)
```



Most covariates are binary:

$$\frac{\text{abs}(P_{\text{target group}} - P_{\text{comparator group}})}{\text{standard deviation}}$$

Graham: “A standardized mean difference of ≤ 0.1 indicates a negligible difference.”



Step 5: Fitting the outcome model

```
outcomeModel <- fitOutcomeModel(population = matchedPop,  
                                modelType = "cox",  
                                stratified = FALSE,  
                                useCovariates = FALSE)
```

Graham: “Cox proportional hazards regression was used to compare time to event in dabigatran compared with warfarin (reference) cohorts.

Ambiguous: Did they condition the model on the matched set (recommended)?



Inspect the outcome model

```
summary(outcomeModel)
```

Model type: cox
Stratified: FALSE
Use covariates: FALSE
Status: OK

	Estimate	lower .95	upper .95	logRr	seLogRr
treatment	0.89626	0.71863	1.11829	-0.10952	0.1128

Population counts

	treatedPersons	comparatorPersons	treatedExposures	comparatorExposures
Count	17460	17460	17460	17460

Outcome counts

	treatedPersons	comparatorPersons	treatedExposures	comparatorExposures
Count	164	155	164	155

Time at risk

	treatedDays	comparatorDays
Days	4912947	3954046



Inspect the outcome model

```
summary(outcomeModel)
```

Model type: cox

Stratified: FALSE

Use covariates: FALSE

Status: OK

	Estimate	lower .95	upper .95	logRr	seLogRr
treatment	0.89626	0.71863	1.11829	-0.10952	0.1128

Population counts

	treatedPersons	comparatorPersons	treatedExposures	comparatorExposures
Count	164	155	164	155

Point estimate and 95% confidence interval

	treatedPersons	comparatorPersons	treatedExposures	comparatorExposures
Count	164	155	164	155

Time at risk

	treatedDays	comparatorDays
Days	4912947	3954046



Inspect the outcome model

```
summary(outcomeModel)
```

Model type: cox
Stratified: FALSE
Use covariates: FALSE
Status: OK

	Estimate	lower .95	upper .95	logRr	seLogRr
treatment	0.89626	0.71863	1.11		
Population counts					
	treatedPersons		com		es
Count	17460		1746		
Outcome counts					
	treatedPersons		comparatorPersons	treat	Exposures
Count	164		155	164	comparatorExposures
Time at risk					
	treatedDays		comparatorDays		
Days	4912947		3954046		

Target group (dabigatran) has more outcomes, but also more time at risk



Inspect the outcome model

```
summary(outcomeModel)
```

Model type: cox
Stratified: FALSE
Use covariates: FALSE
Status: OK

	Estimate	lower .95	upper .95
treatment	0.89626	0.71863	1.11829

Population counts

	treatedPersons	comparatorPersons
Count	17460	17460

Outcome counts

	treatedPersons	comparatorPersons	treatedExposures	comparatorExposures
Count	164	155	164	155

Time at risk

	treatedDays	comparatorDays
Days	4912947	3954046

Graham:

IR_{dabigatran} = 11.3

IR_{warfarin} = 13.9

HR_{adjusted} = 0.80 (0.67–0.96)

IR_{dabigatran} = 12.2

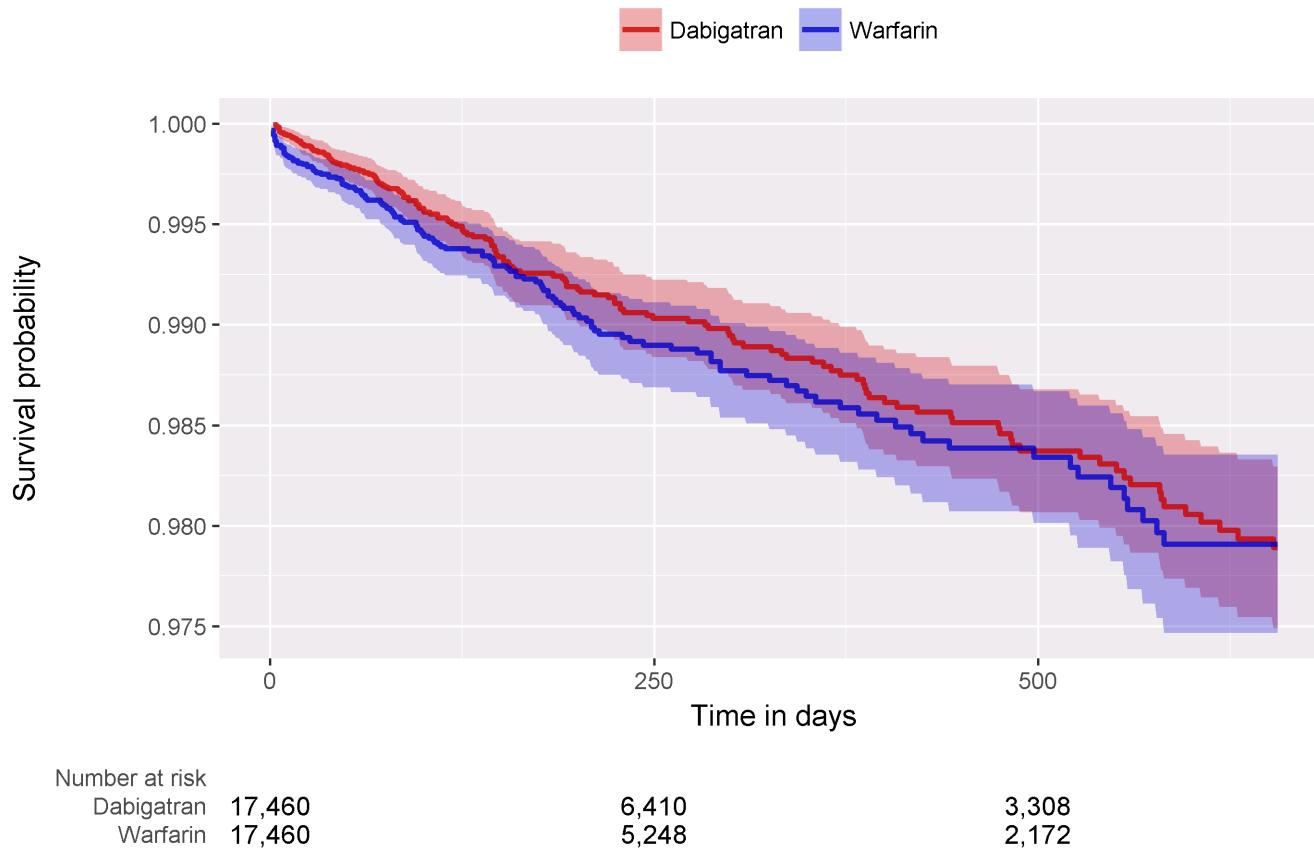
IR_{warfarin} = 14.3

HR_{adjusted} = 0.90 (0.72 – 1.12)



Kaplan Meier plot

```
plotKaplanMeier(matchedPop,  
  treatmentLabel = "Dabigatran",  
  comparatorLabel = "Warfarin")
```

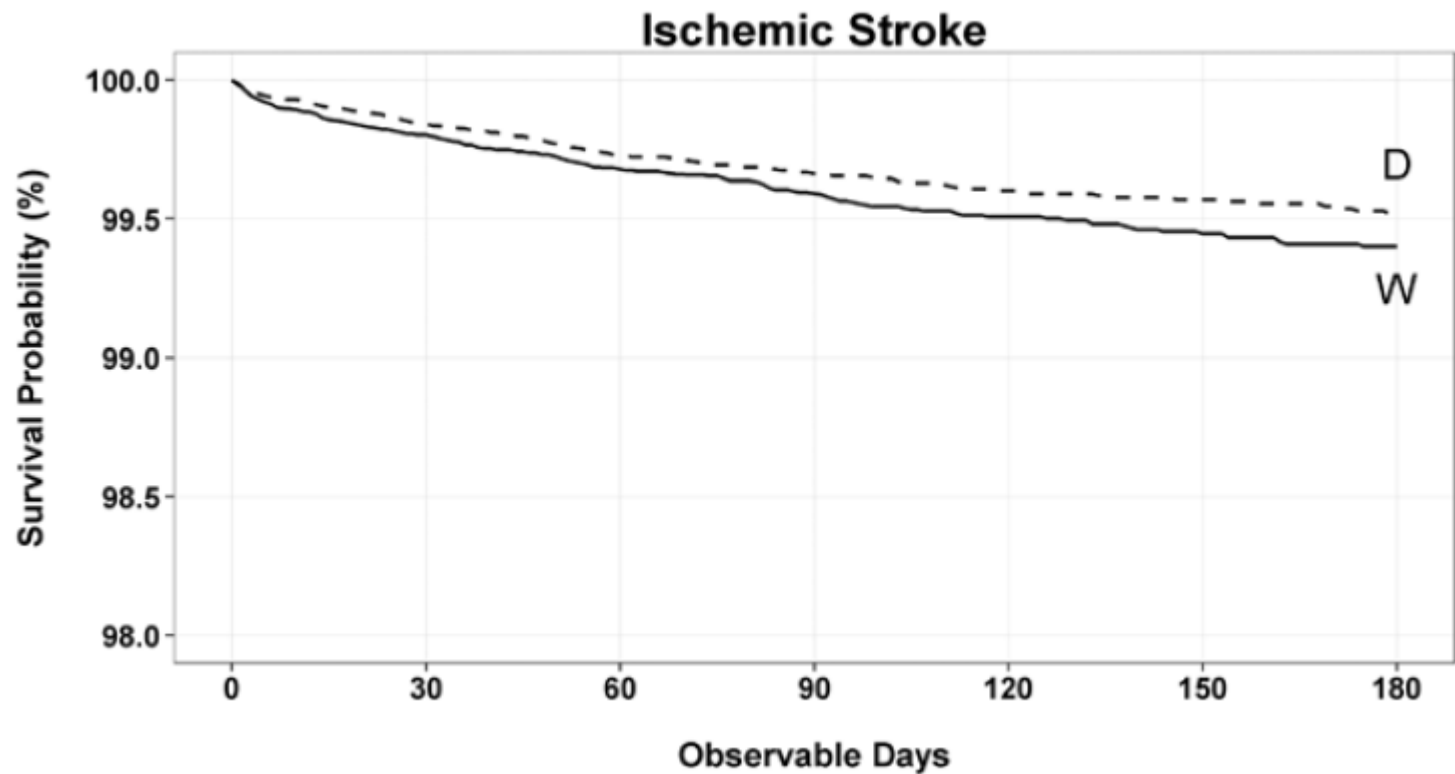




Kaplan Meier plot

```
plotKaplanMeier(matchedPop,
```

Graham:



Time in days



Execution steps

1. Getting the necessary data from the database
 2. Defining the study population
 3. Creating a propensity model
 4. Matching
 5. Fitting the outcome model
- + generating various diagnostics



Handpicking covariates is not recommended!

- Might miss an important confounder (proxy)
- Subjective (non-reproducible)

Better:

- Include all pre-defined covariates
- Let data (and regularization) decide which ones are important
- Bonus: more work for the computer, but less work for you!

Covariate settings

[illegible]



Covariate settings

Constructing covariates for

- Demographics
- All drugs & classes
- All conditions & groups
- All procedures
- All measurements
- All observations
- Risk scores



```
covariateSettings <- createCovariateSettings(
  useCovariateDemographics = TRUE,
  useCovariateConditionOccurrence = TRUE,
  useCovariateConditionOccurrence365d = TRUE,
  useCovariateConditionOccurrence30d = TRUE,
  useCovariateConditionOccurrenceInpt180d = TRUE,
  ...
  useCovariateRiskScores = TRUE,
  useCovariateRiskScoresCharlson = TRUE,
  useCovariateRiskScoresDCSI = TRUE,
  useCovariateRiskScoresCHADS2 = TRUE,
  useCovariateInteractionYear = FALSE,
  useCovariateInteractionMonth = FALSE,
  excludedCovariateConceptIds = excludeConceptIds,
  deleteCovariatesSmallCount = 100)
```



Covariate settings

```
covariateSettings <- createCovariateSettings(useCovariateDemographics = TRUE,  
                                             useCovariateConditionOccurrence = TRUE,  
                                             useCovariateConditionOccurrence365d = TRUE,  
                                             useCovariateConditionOccurrence30d = TRUE,  
                                             useCovariateConditionOccurrenceInpt180d = TRUE,  
                                             ...  
                                             useCovariateRiskScores = TRUE,  
                                             useCovariateRiskScoresCharlson = TRUE,  
                                             useCovariateRiskScoresDCSI = TRUE,  
                                             useCovariateRiskScoresCHADS2 = TRUE,  
                                             useCovariateInteractionYear = FALSE,  
                                             useCovariateInteractionMonth = FALSE,  
                                             excludedCovariateConceptIds = excludeConceptIds,  
                                             deleteCovariatesSmallCount = 100)
```

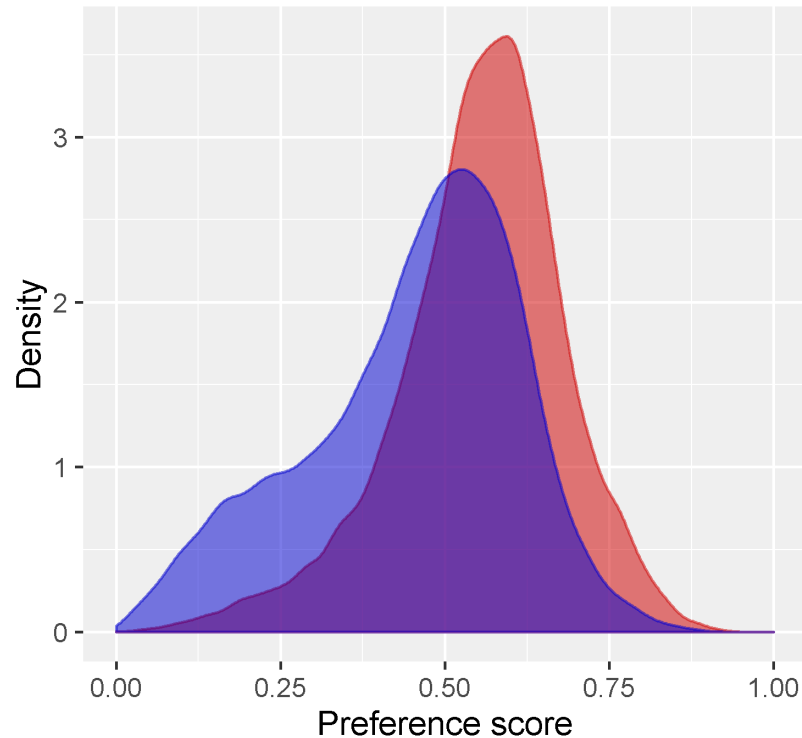
Have to explicitly exclude
dabigatran and warfarin



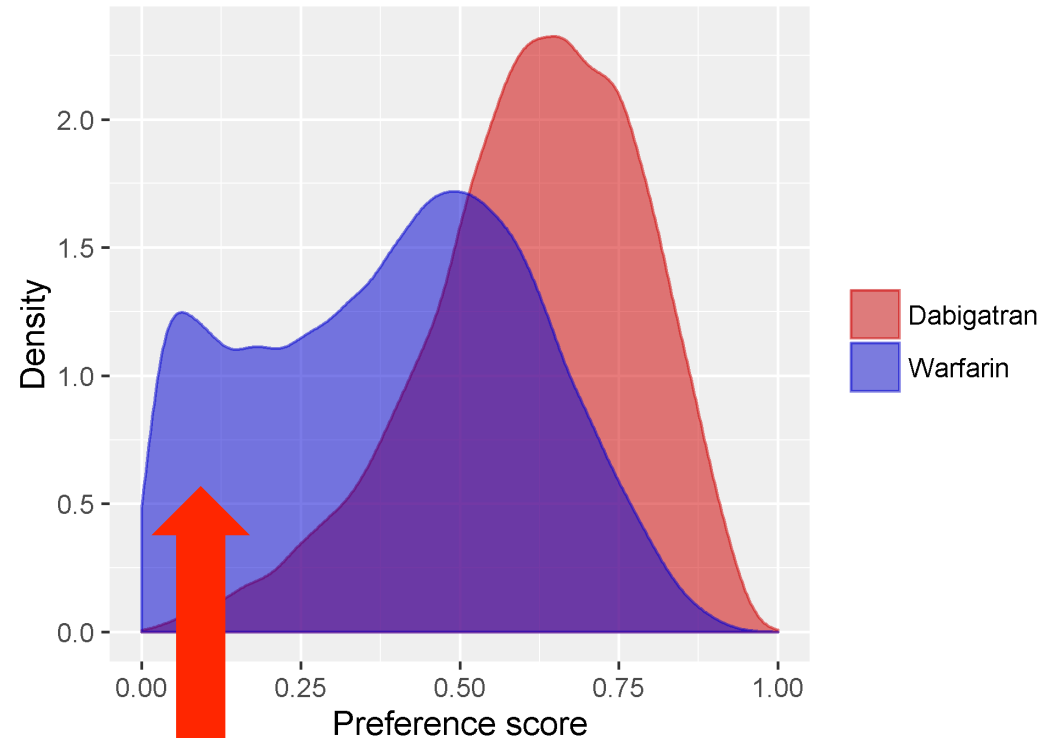


Effect on propensity score distribution

Using hand-picked covariates



Using all covariates

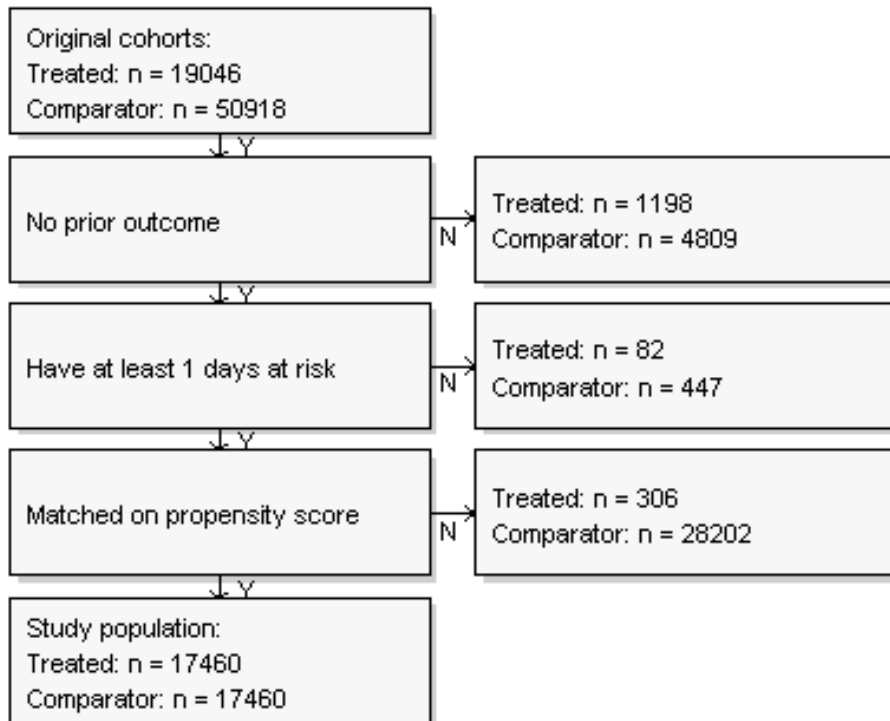


Whole subgroup of people likely to get warfarin was not identified by Graham

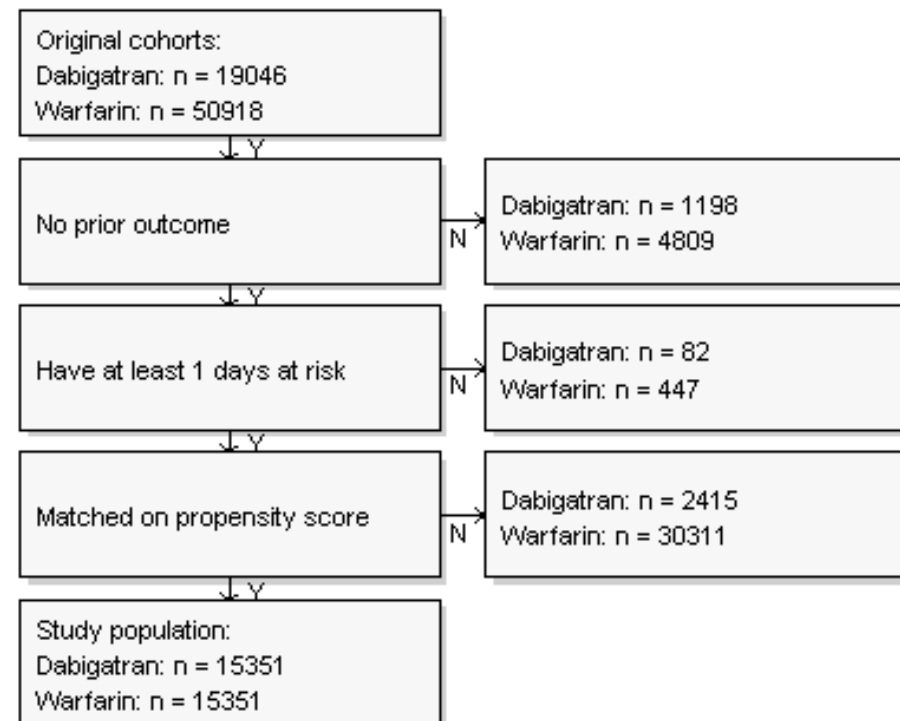


Effect on matching

Using hand-picked covariates



Using all covariates



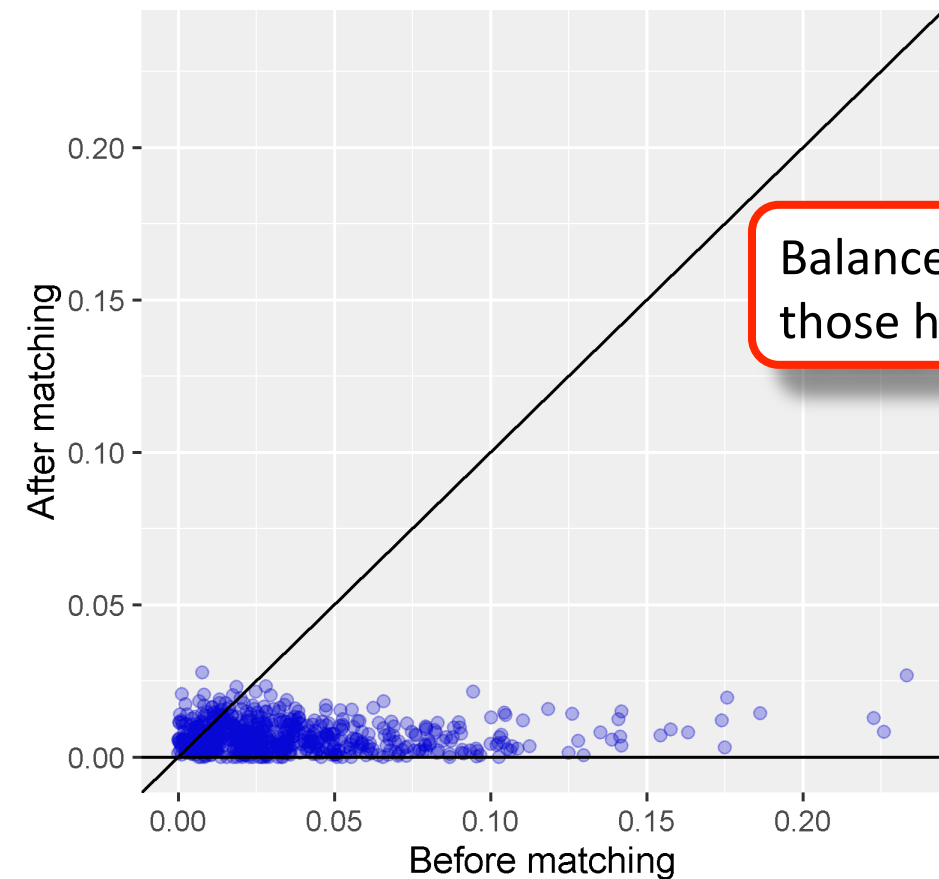
Fewer people left after matching



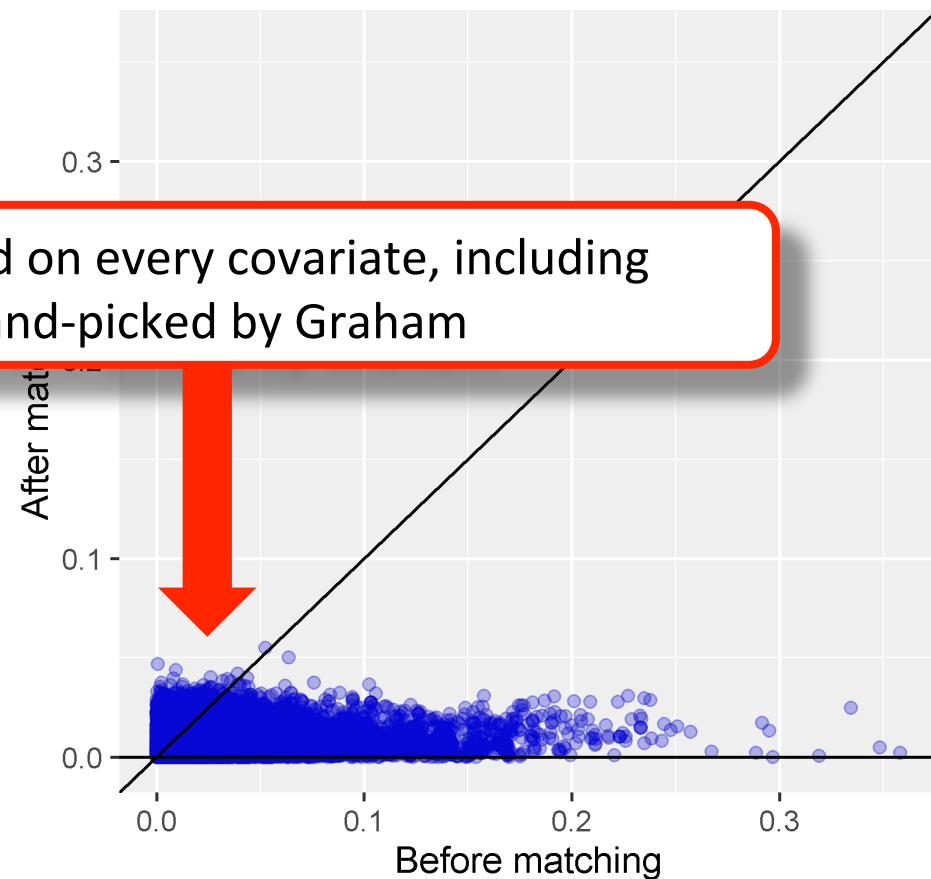
Effect on balance

Using hand-picked covariates

Using all covariates



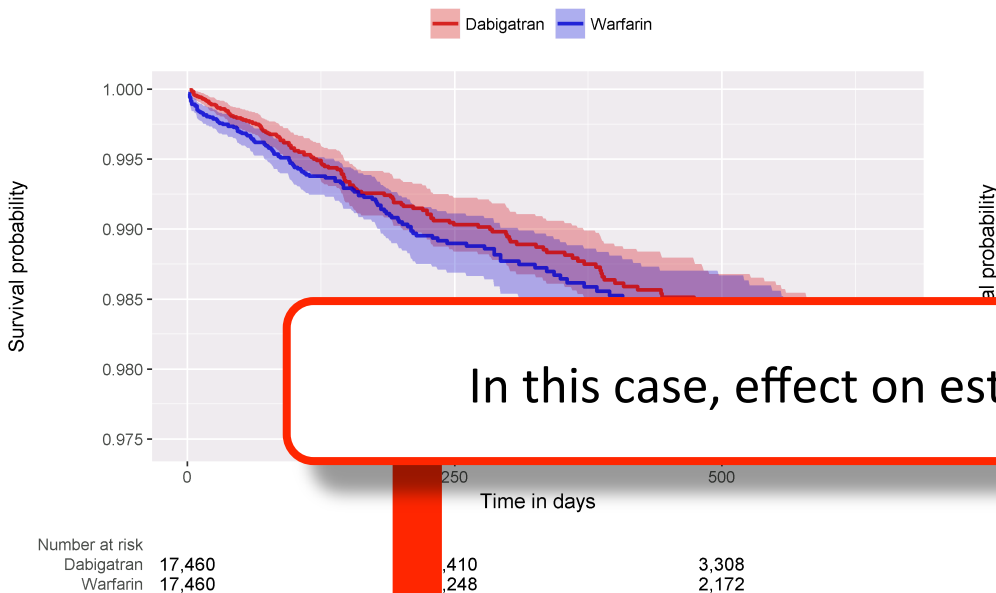
Balanced on every covariate, including those hand-picked by Graham





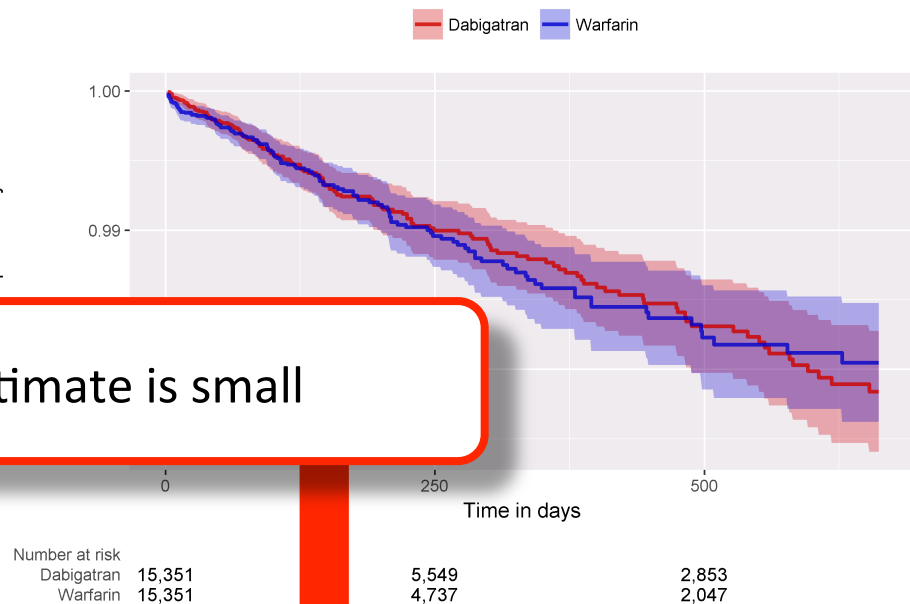
Effect on hazard ratio

Using hand-picked covariates



HR = 0.90 (0.72 – 1.12)

Using all covariates



HR = 0.95 (0.75 - 1.21)

In this case, effect on estimate is small



In conclusion

- OHDSI tools can replicate the Graham study
 - Diagnostics are an important part of both design and execution
 - Show impact of adjustments: lots of imbalance before matching
 - Lot of attrition: how generalizable are our results?
 - Hand picking covariates is not recommended
-