



Review of R code generated by ATLAS

Martijn Schuemie



Purpose of R code

- Starting point (to be adapted)
- Perform analyses as specified
- Generate diagnostics



Creating cohorts

- Not included in the code is the creation of the required cohorts
 - Target cohort(s)
 - Comparator cohort(s)
 - Outcome cohort(s)
- Cohorts go in a table with the same structure as the *cohort* table in the CDM (cohort_definition_id, subject_id, cohort_start_date, cohort_end_date)
- Cohorts can be created
 - By ATLAS
 - By running SQL created by ATLAS



Data locations and connections

```
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms = "pdw",  
                                                                server = "my_server.com",  
                                                                user = "martijn",  
                                                                password = "secret")  
  
cdmDatabaseSchema <- "cdm_truven_mdcd.dbo"  
oracleTempSchema <- NULL  
exposureDatabaseSchema <- cdmDatabaseSchema  
outcomeDatabaseSchema <- cdmDatabaseSchema  
exposureTable <- "cohort"  
outcomeTable <- "cohort"  
cdmVersion <- "5"  
outputFolder <- "c:/temp/Garbe_mdcd"  
maxCores <- 32
```



Data locations and connections

```
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms = "pdw",  
                                                                server = "my_server.com",  
                                                                user = "martijn",  
                                                                password = "secret")
```

```
cdmDatabaseSchema <- "cdm_truven_mdcd.dbo"  
oracleTempSchema <- NULL  
exposureDatabaseSchema <- cdmDatabaseSchema  
outcomeDatabaseSchema <- cdmDatabaseSchema  
exposureTable <- "cohort"  
outcomeTable <- "cohort"  
cdmVersion <- "5"  
outputFolder <- "c:/temp/Garbe_mdcd"  
maxCores <- 32
```

How to connect to connect to the
server



Data locations and connections

```
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms = "pdw",  
                                                                server = "my_server.com",  
                                                                user = "martijn",  
                                                                password = "secret")
```

```
cdmDatabaseSchema <- "cdm_truven_mdcd.dbo"  
oracleTempSchema <- NULL  
exposureDatabaseSchema <- cdmDatabaseSchema  
outcomeDatabaseSchema <- cdmDatabaseSchema  
exposureTable <- "cohort"  
outcomeTable <- "cohort"  
cdmVersion <- "5"
```

```
outputFolder <- "c:/temp/Garbe_mdcd"  
maxCores <- 32
```

Where is the data?



Data locations and connections

```
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms = "pdw",  
                                                                server = "my_server.com",  
                                                                user = "martijn",  
                                                                password = "secret")
```

```
cdmDatabaseSchema <- "cdm_truven_mdcd.dbo"
```

```
oracleTempSchema <- NULL
```

```
exposureDatabaseSchema <- cdmDatabaseSchema
```

```
outcomeDatabaseSchema <- cdmDatabaseSchema
```


```
exposureTable <- "cohort"
```

```
outcomeTable <- "cohort"
```

```
cdmVersion <- "5"
```

```
outputFolder <- "c:/temp/Garbe_mdcd"
```

```
maxCores <- 32
```



Output folder in local file
system



Concept sets

```
sql <- paste("select distinct l.concept_id FROM  
  (  
    select concept_id from @cdm_database_schema.CONCEPT where concept_id in  
(21603933,1118084,21603991,1124300)and invalid_reason is null  
    UNION select c.concept_id  
    from @cdm_database_schema.CONCEPT c  
    join @cdm_database_schema.CONCEPT_ANCESTOR ca on c.concept_id = ca.descendant_concept_id  
    and ca.ancestor_concept_id in (1118084,1124300)  
    and c.invalid_reason is null  
  ) l  
  ")  
sql <- SqlRender::renderSql(sql, cdm_database_schema = cdmDatabaseSchema)$sql  
sql <- SqlRender::translateSql(sql, targetDialect = connectionDetails$dbms)$sql  
connection <- DatabaseConnector::connect(connectionDetails)  
excludedConcepts <- DatabaseConnector::querySql(connection, sql)  
excludedConcepts <- excludedConcepts$CONCEPT_ID
```

Concept sets are created by
executing SQL against the vocab

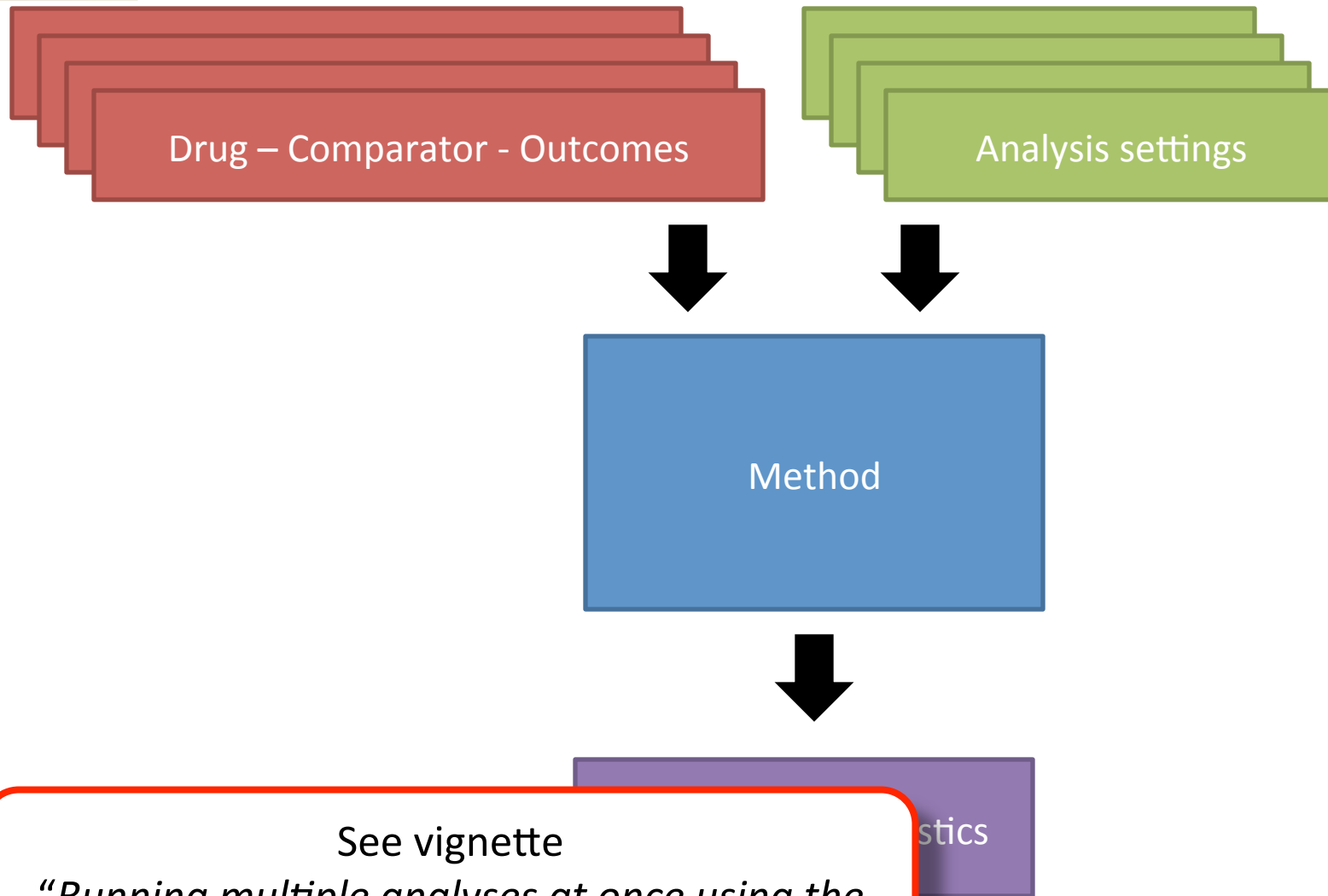


Execution automation

- Previously we discussed how to perform 1 analysis
- May want to run multiple analyses, for example
 - Main + sensitivity analyses
 - Negative control outcomes
 - Multiple comparisons
- CohortMethod can execute in efficient manner



Running multiple analyses



See vignette
*“Running multiple analyses at once using the
CohortMethod package”*



TCO definition

```
dcos <- CohortMethod::createDrugComparatorOutcomes(  
  targetId = targetCohortId,  
  comparatorId = comparatorCohortId,  
  excludedCovariateConceptIds = excludedConcepts,  
  includedCovariateConceptIds = includedConcepts,  
  outcomeIds = c(outcomeList, negativeControlConcepts))  
drugComparatorOutcomesList <- list(dcos)
```



Creating argument objects

```
matchOnPsArgs1 <- CohortMethod::createMatchOnPsArgs(  
  caliper = 0.25,  
  caliperScale = "standardized",  
  maxRatio = 1)
```



Analysis definition

...

```
cmAnalysis1 <- CohortMethod::createCmAnalysis(  
  analysisId = 1,  
  description = "OHDSI estimation tutorial: Garbe replication: celecoxi...",  
  getDbCohortMethodDataArgs = getDbCmDataArgs,  
  createStudyPopArgs = createStudyPopArgs,  
  createPs = TRUE,  
  createPsArgs = createPsArgs1,  
  matchOnPs = TRUE,  
  matchOnPsArgs = matchOnPsArgs1,  
  computeCovariateBalance = TRUE,  
  fitOutcomeModel = TRUE,  
  fitOutcomeModelArgs = fitOutcomeModelArgs1)  
cmAnalysisList <- list(cmAnalysis1)
```



Running all analyses

```
result <- CohortMethod::runCmAnalyses(  
  connectionDetails = connectionDetails,  
  cdmDatabaseSchema = cdmDatabaseSchema,  
  exposureDatabaseSchema = exposureDatabaseSchema,  
  exposureTable = exposureTable,  
  outcomeDatabaseSchema = outcomeDatabaseSchema,  
  outcomeTable = outcomeTable,  
  cdmVersion = cdmVersion,  
  outputFolder = outputFolder,  
  cmAnalysisList = cmAnalysisList,  
  drugComparatorOutcomesList = drugComparatorOutcomesList,  
  getDbCohortMethodDataThreads = 1,  
  createPsThreads = 1,  
  psCvThreads = min(16, maxCores),  
  computeCovarBalThreads = min(3, maxCores),  
  createStudyPopThreads = min(3, maxCores),  
  trimMatchStratifyThreads = min(10, maxCores),  
  fitOutcomeModelThreads = max(1, round(maxCores/4)),  
  outcomeCvThreads = min(4, maxCores),  
  refitPsForEveryOutcome = FALSE)
```



Running all analyses

```
result <- CohortMethod::runCmAnalyses(  
  connectionDetails = connectionDetails,  
  cdmDatabaseSchema = cdmDatabaseSchema,  
  exposureDatabaseSchema = exposureDatabaseSchema,  
  exposureTable = exposureTable,  
  outcomeDatabaseSchema = outcomeDatabaseSchema,  
  outcomeTable = outcomeTable,  
  cdmVersion = cdmVersion,  
  outputFolder = outputFolder,  
  cmAnalysisList = cmAnalysisList,  
  drugComparatorOutcomesList = drugComparatorOutcomesList,  
  getDbCohortMethodDataThreads = 1,  
  createPsThreads = 1,  
  psCvThreads = min(16, maxCores),  
  computeCovarBalThreads = min(3, maxCores),  
  createStudyPopThreads = min(3, maxCores),  
  trimMatchStratifyThreads = min(10, maxCores),  
  fitOutcomeModelThreads = max(1, maxCores),  
  outcomeCvThreads = min(4, maxCores),  
  refitPsForEveryOutcome = FALSE)
```

The locations we specified earlier



Running all analyses

```
result <- CohortMethod::runCmAnalyses(  
  connectionDetails = connectionDetails,  
  cdmDatabaseSchema = cdmDatabaseSchema,  
  exposureDatabaseSchema = exposureDatabaseSchema,  
  exposureTable = exposureTable,  
  outcomeDatabaseSchema = outcomeDatabaseSchema,  
  outcomeTable = outcomeTable,  
  cdmVersion = cdmVersion,  
  outputFolder = outputFolder,  
  cmAnalysisList = cmAnalysisList,  
  drugComparatorOutcomesList = drugComparatorOutcomesList,  
  getDbCohortMethodDataThreads = 1,  
  createPsThreads = 1,  
  psCvThreads = min(16, maxCores),  
  computeCovarBalThreads = min(3, maxCores),  
  createStudyPopThreads = min(3, maxCores),  
  trimMatchStratifyThreads = min(10, maxCores),  
  fitOutcomeModelThreads = max(1, maxCores),  
  outcomeCvThreads = min(4, maxCores),  
  refitPsForEveryOutcome = FALSE)
```

The TCOs and analyses we
specified earlier



Running all analyses

```
result <- CohortMethod::runCmAnalyses(  
  connectionDetails = connectionDetails,  
  cdmDatabaseSchema = cdmDatabaseSchema,  
  exposureDatabaseSchema = exposureDatabaseSchema,  
  exposureTable = exposureTable,  
  outcomeDatabaseSchema = outcomeDatabaseSchema,  
  outcomeTable = outcomeTable,  
  cdmVersion = cdmVersion,  
  outputFolder = outputFolder,  
  cmAnalysisList = cmAnalysisList,  
  drugComparatorOutcomesList = drugComparatorOutcomesList,  
  getDbCohortMethodDataThreads = 1,  
  createPsThreads = 1,  
  psCvThreads = min(16, maxCores),  
  computeCovarBalThreads = min(3, maxCores),  
  createStudyPopThreads = min(3, maxCores),  
  trimMatchStratifyThreads = min(10, maxCores),  
  fitOutcomeModelThreads = max(1, round(maxCores/4)),  
  outcomeCvThreads = min(4, maxCores),  
  refitPsForEveryOutcome = FALSE)
```

For the various analyses steps the number of parallel threads to use



Running all analyses

result

```
runCmAnalyses(  
    connectionDetails = connectionDetails,  
    cdmDatabaseSchema = cdmDatabaseSchema,  
    exposureDatabaseSchema = exposureDatabaseSchema,  
    outcomeDatabaseSchema = outcomeDatabaseSchema,  
    cmAnalysisList = cmAnalysisList,  
    drugComparatorOutcomesList = drugComparatorOutcomesList,  
    getDbCohortMethodDataThreads = 1,  
    createPsThreads = 1,  
    psCvThreads = min(16, maxCores),  
    computeCovarBalThreads = min(3, maxCores),  
    createStudyPopThreads = min(3, maxCores),  
    trimMatchStratifyThreads = min(10, maxCores),  
    fitOutcomeModelThreads = max(1, round(maxCores/4)),  
    outcomeCvThreads = min(4, maxCores),  
    refitPsForEveryOutcome = FALSE)
```

What is this?



The result object

Contains for every TCO – analysis ID combination, path to

- CohortMethod data file
- Study population file
- Propensity score file
- Matched population file
- Outcome model file

If you lost it, it is saved automatically as
outcomeModelReference.rds



Summarizing

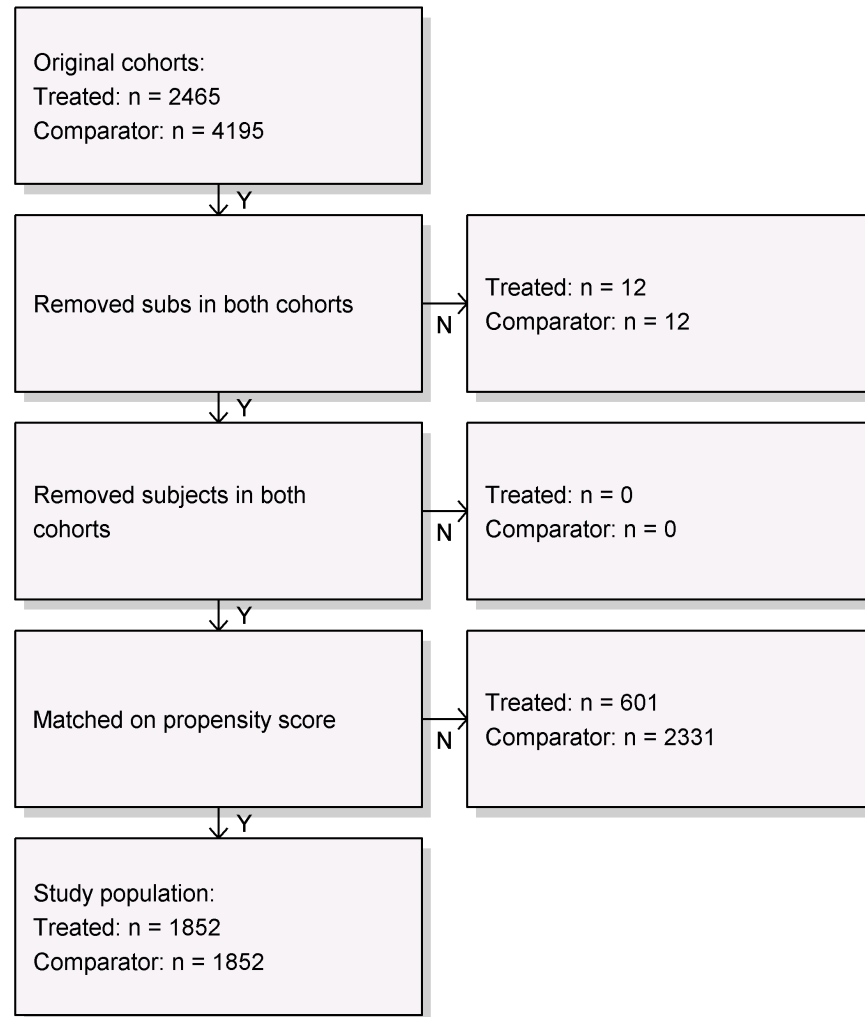
```
analysisSummary <- CohortMethod::summarizeAnalyses(result)
```

Contains for every TCO – analysis ID combination:

- Effect size estimate + 95% confidence interval
- Number of subjects in T and C (after matching)
- Number of subjects with O (in T and C)

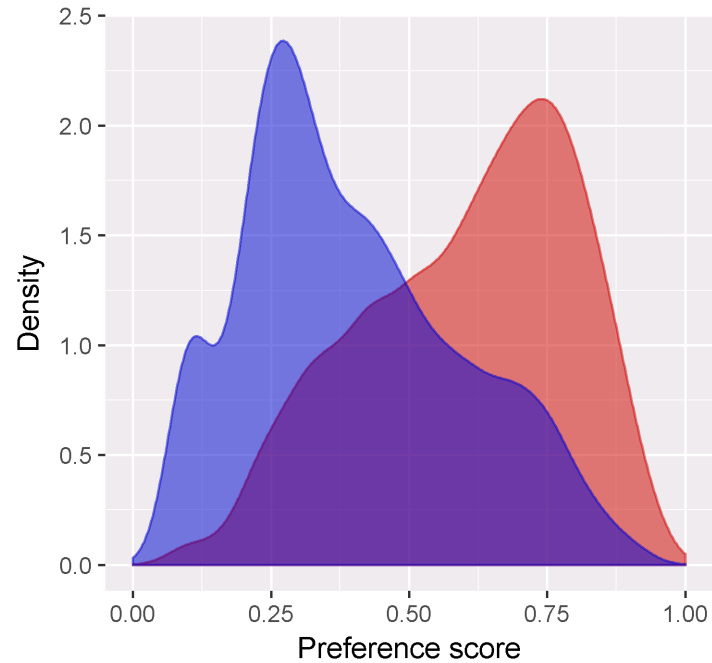


Diagnostics: attrition diagram

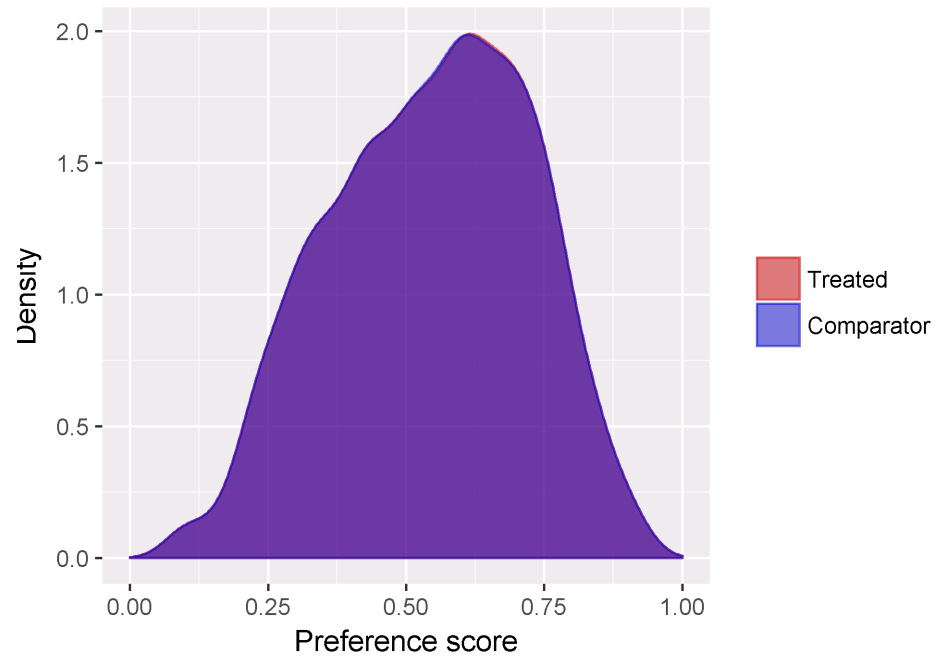




Diagnostics: PS plot



Before matching



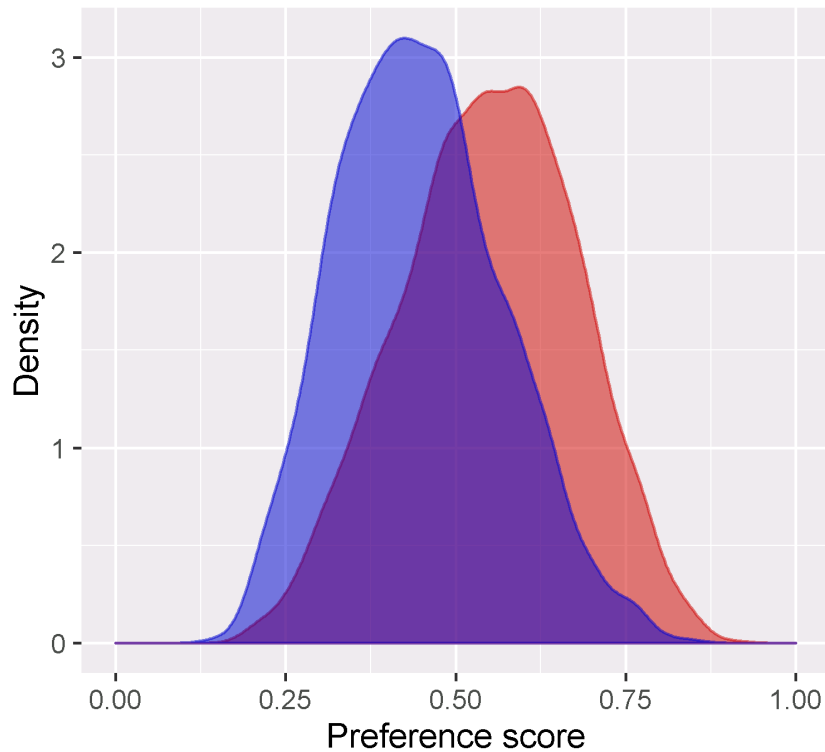
After matching



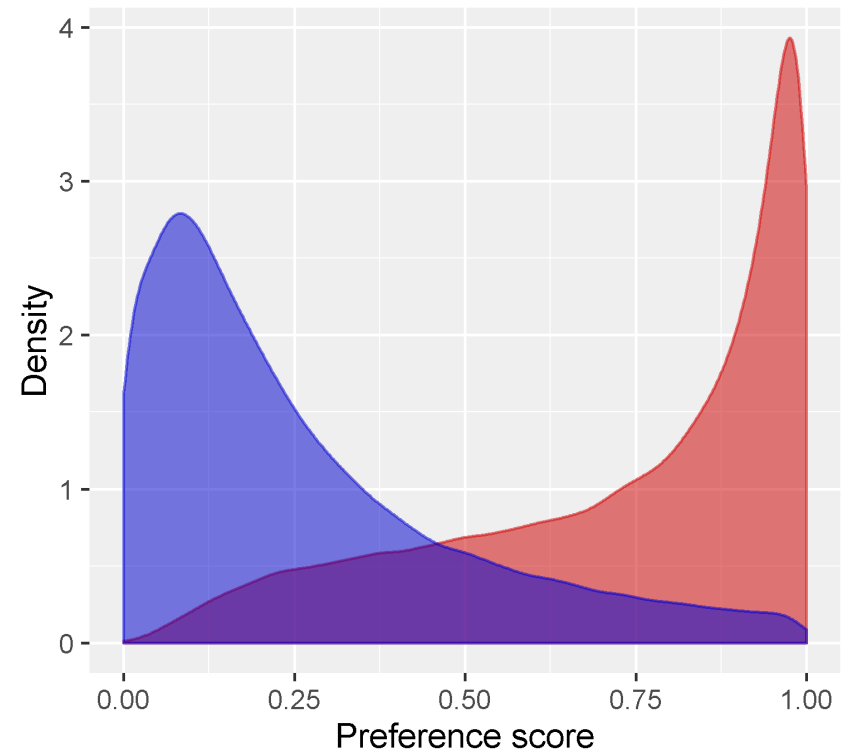
Diagnostics: PS plot

Is there overlap between the cohorts?

Good:

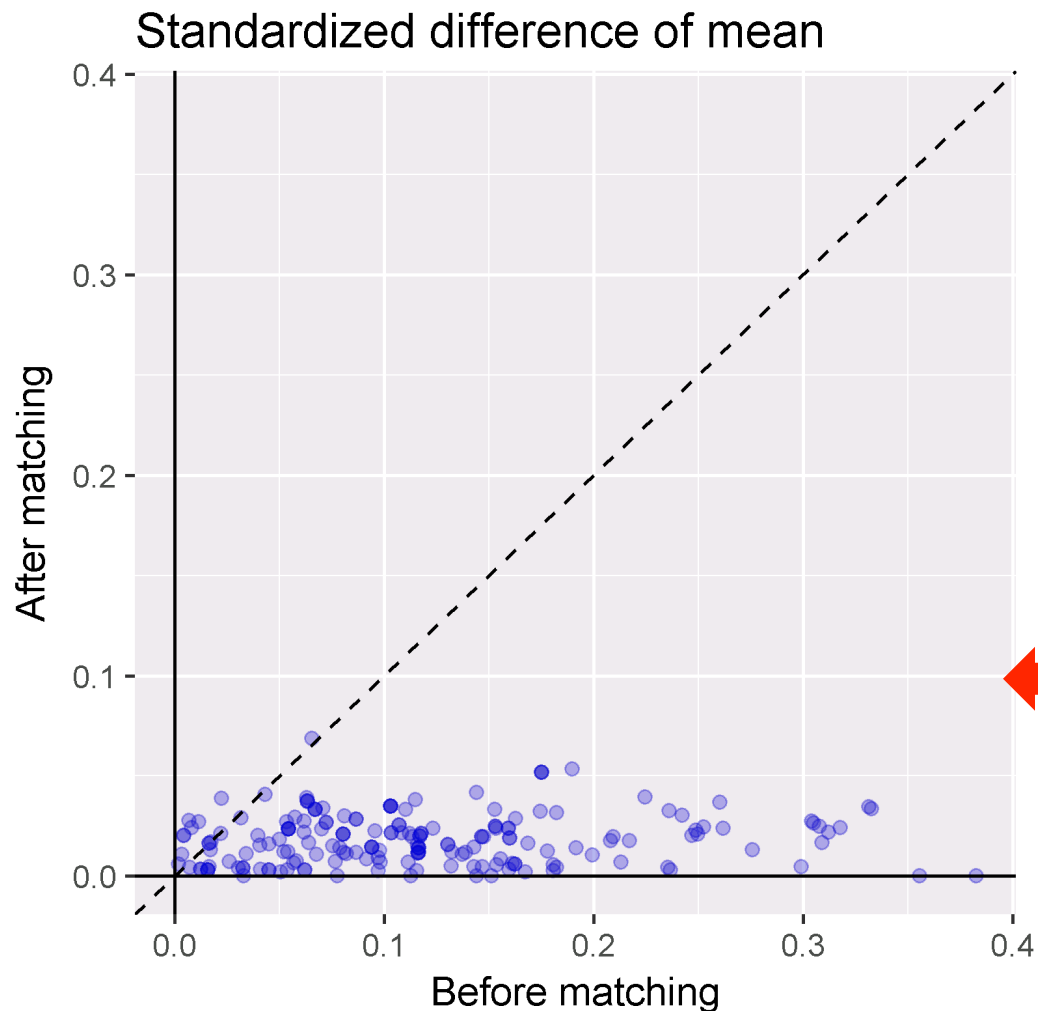


Bad:





Diagnostics: covariate balance

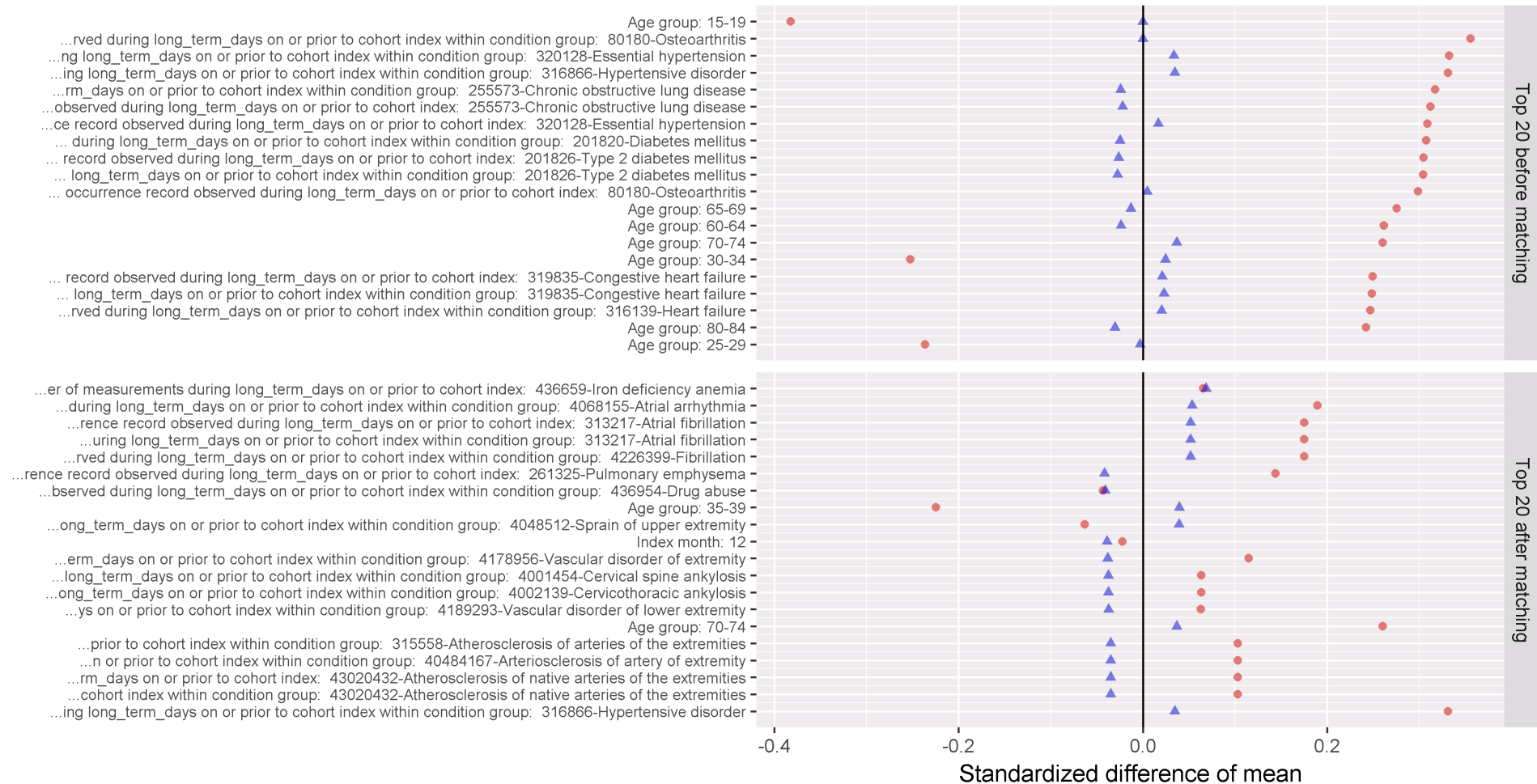


Rule of thumb: all covariates
below 0.1 after matching?



Diagnostics: covariate balance

• before matching
▲ after matching





Diagnostic: residual bias

