# Causal inference tools for pharmacovigilance

Michele Fusaroli Joseph Mitchell Annette Rudolph
Elena Rocca Riccardo Fusaroli

## Running Code

Note that this is a supplementary material of an article and the descriptions of the DAGs and the analyses are retrieved from it.

### Install DiAna

This project uses FAERS cleaned data and functions as provided by the DiAna package. To install it follow the instruction at the https://github.com/fusarolimichele/DiAna_package github.

Moreover, this project uses the PVdagger package to draw the DAGs. To install it follow the instruction at the https://github.com/PVverse/pv_dagger github.

### Set Up Project

From an R project in the main directory of DiAna we create a subdirectory to store all the scripts and the outputs of the analysis.

```
dir.create("projects/DAGs")
```

```
Warning in dir.create("projects/DAGs"): cannot create dir 'projects/DAGs',
reason 'No such file or directory'
```

Then we retrieve the functions from the package and we set up the last FAERS quarterly update we want to include in the analysis (note that all the quarters previous to it will also be included).

```
library("DiAna")
```

```
Loading required package: data.table
```

```
Loading required package: ggplot2
```

```
Warning: package 'ggplot2' was built under R version 4.4.3
```

```
library("PVdagger")
```

```
Loading required package: DiagrammeR
```

```
FAERS_version = "23Q1"
```

Then we import FAERS data and other dictionaries we are going to use to standardize drugs and events definition (MedDRA and ATC)

```
import("DRUG")
```

```
          primaryid   drug_seq       substance role_cod
              <num>      <int>          <fctr>    <ord>
       1:   4261825 1004486228        atropine        C
       2:   4261825 1004486228   diphenoxylate        C
       3:   4262057 1004487206     amoxicillin       PS
       4:   4262057 1004487206 clavulanic acid       PS
       5:   4262057 1004487210        macrogol        C
      ---
64409055:  99974963          6       ibuprofen        I
64409056:  99974963          7       lorazepam        I
64409057:  99974963          8            <NA>        I
64409058:  99974963          9     paracetamol        I
64409059:  99974963         10    zolmitriptan       SS
```

```
import("REAC")
```

```
          primaryid                   pt drug_rec_act
              <num>               <ord>        <ord>
       1:   4204616       abdominal pain         <NA>
       2:   4204616 heart rate increased         <NA>
       3:   4204616               nausea         <NA>
       4:   4204616              pyrexia         <NA>
       5:   4204616   uterine tenderness         <NA>
      ---
48356242: 998834879          joint noise         <NA>
48356243:  99953304     drug interaction         <NA>
48356244:  99953304        breast cancer         <NA>
48356245:  99974963        breast cancer         <NA>
48356246:  99974963     drug interaction         <NA>
```

```
import("DEMO")
```

```
          primaryid    sex age_in_days wt_in_kgs             occr_country
              <num> <fctr>       <num>     <num>                   <fctr>
       1:  45217461      M        9855        NA                     <NA>
       2:  57910401   <NA>          NA        NA                     <NA>
       3:  56962401   <NA>          NA        NA                     <NA>
       4:  54662121      F       22630        NA                     <NA>
       5:  31231172      F       21535        68                     <NA>
      ---
16265820:  89418103      M        4015        NA  France, French Republic
16265821:  89516744      M        4015        NA  France, French Republic
16265822:  89516903      M        2920        NA  France, French Republic
16265823:  90224005      M          NA        NA                   Canada
16265824:  91610302      F        7665        NA United States of America
          event_dt occp_cod          reporter_country rept_cod init_fda_dt
             <int>   <fctr>                    <fctr>   <fctr>       <int>
       1: 19861106       MD  United States of America      DIR    19861224
       2:       NA     <NA>  United States of America      DIR    19920917
       3: 19960123     <NA>  United States of America      DIR    19960126
       4: 19960919       CN South Africa, Republic of      EXP    19961022
       5: 19970927       MD  United States of America      EXP    19971203
      ---
16265820:       NA       HP   France, French Republic      EXP    20121203
16265821:       NA       HP   France, French Republic      EXP    20121204
16265822:       NA       HP   France, French Republic      EXP    20121204
16265823: 20020201       MD                    Canada      EXP    20130118
```

```
16265824:       NA       HP  United States of America      PER    20130313
            fda_dt premarketing literature RB_duplicates
             <int>       <lgcl>     <lgcl>        <lgcl>
       1: 19861224        FALSE      FALSE         FALSE
       2: 19920917        FALSE      FALSE         FALSE
       3: 19960126        FALSE      FALSE         FALSE
       4: 19961022        FALSE      FALSE         FALSE
       5: 19971212        FALSE      FALSE         FALSE
      ---
16265820: 20230331        FALSE       TRUE         FALSE
16265821: 20230331        FALSE       TRUE         FALSE
16265822: 20230331        FALSE       TRUE         FALSE
16265823: 20230331        FALSE      FALSE         FALSE
16265824: 20230331        FALSE       TRUE         FALSE
          RB_duplicates_only_susp
                            <lgcl>
       1:                   FALSE
       2:                   FALSE
       3:                   FALSE
       4:                   FALSE
       5:                   FALSE
      ---
16265820:                   FALSE
16265821:                   FALSE
16265822:                   FALSE
16265823:                   FALSE
16265824:                   FALSE
```

```
import("INDI")
```

```
          primaryid   drug_seq                          indi_pt
              <num>      <int>                            <ord>
       1:   4204616 1004278786                 abortion induced
       2:   4223542 1004334703                  crohn's disease
       3:   4250482 1004434717                  crohn's disease
       4:   4261823 1004486217              bronchial carcinoma
       5:   4261823 1004486218              bronchial carcinoma
      ---
38150726:  99974963          6 product used for unknown indication
38150727:  99974963          7 product used for unknown indication
38150728:  99974963          8 product used for unknown indication
```

```
38150729:  99974963          9 product used for unknown indication
38150730:  99974963         10                              migraine
```

```
import("OUTC")
```

```
          primaryid outc_cod
              <num>    <ord>
       1:   4204616       HO
       2:   4204616       RI
       3:   4214534       DE
       4:   4223542       OT
       5:   4250482       OT
      ---
11950778:  99563723       OT
11950779: 998834879       HO
11950780: 998834879       OT
11950781:  99953304       OT
11950782:  99974963       OT
```

```
import_ATC()
```

```
                                    substance    code primary_code   Lvl4
                                       <char>  <char>       <char> <char>
   1:                     iodine apamistamab V10XA04      V10XA04  V10XA
   2:                          flotufolastat V09IX18      V09IX18  V09IX
   3:                     avacincaptad pegol S01XA32      S01XA32  S01AX
   4:                              lotilaner S01AX25      S01AX25  S01AX
   5:                            vanzacaftor R07AX33      R07AX33  R07AX
  ---
3367:                   radium-223 dichloride V10XX03      V10XX03  V10XX
3368:               lutetium (177lu) dotatate V10XX04      V10XX04  V10XX
3369: lutetium (177lu) vipivotide tetraxetan V10XX05      V10XX05  V10XX
3370:                               tenofovir J05AR03      J05AR03  J05AR
3371:                            brimonidine S01EA55      S01EA55  S01EA
                                                         Class4   Lvl3
                                                         <char> <char>
   1:                                    Iodine (131I) compounds   V10X
   2: Other diagnostic radiopharmaceuticals for tumour detection   V09I
   3:                                       Other antiinfectives   S01A
   4:                                       Other antiinfectives   S01A
```

```
   5:                          Other respiratory system products   R07A
  ---
3367:                   Various therapeutic radiopharmaceuticals   V10X
3368:                   Various therapeutic radiopharmaceuticals   V10X
3369:                   Various therapeutic radiopharmaceuticals   V10X
3370:   Antivirals for treatment of HIV infections, combinations   J05A
3371:                       Sympathomimetics in glaucoma therapy   S01E
                                  Class3   Lvl2
                                  <char> <char>
   1: Other diagnostic radiopharmaceutical    V10
   2:                      Tumor detection    V09
   3:                       Antiinfectives    S01
   4:                       Antiinfectives    S01
   5:                    Other respiratory    R07
  ---
3367: Other diagnostic radiopharmaceutical    V10
3368: Other diagnostic radiopharmaceutical    V10
3369: Other diagnostic radiopharmaceutical    V10
3370:             Direct acting antivirals   J05A
3371:                          Antiglaucoma    S01
                               Class2   Lvl1         Class1
                               <char> <char>         <char>
   1: Therapeutic radiopharmaceuticals      V        Various
   2:  Diagnostic radiopharmaceuticals      V        Various
   3:                 Ophtalmologicals      S        Sensory
   4:                 Ophtalmologicals      S        Sensory
   5:                Other respiratory      R    Respiratory
  ---
3367: Therapeutic radiopharmaceuticals      V        Various
3368: Therapeutic radiopharmaceuticals      V        Various
3369: Therapeutic radiopharmaceuticals      V        Various
3370:                       Antivirals      J Antiinfectives
3371:                 Ophtalmologicals      S        Sensory
```

```
import_MedDRA() ##This database is only available to MedDRA subscribers.
```

```
       def                                        soc
    <char>                                     <char>
1:   cong     congenital, familial and genetic disorders
2:    inv                                investigations
3:    inv                                investigations
```

```
    4:    inv                                    investigations
    5:    inv                                    investigations
   ---
25408:  inj&p injury, poisoning and procedural complications
25409:   preg pregnancy, puerperium and perinatal conditions
25410:   surg                surgical and medical procedures
25411:   surg                surgical and medical procedures
25412:  blood           blood and lymphatic system disorders
                                                         hlgt
                                                       <char>
    1:        metabolic and nutritional disorders congenital
    2:          endocrine investigations (incl sex hormones)
    3:          endocrine investigations (incl sex hormones)
    4:          endocrine investigations (incl sex hormones)
    5:          endocrine investigations (incl sex hormones)
   ---
25408:     procedural related injuries and complications nec
25409: pregnancy, labour, delivery and postpartum conditions
25410:   obstetric and gynaecological therapeutic procedures
25411:             male genital tract therapeutic procedures
25412:                           white blood cell disorders
                                                    hlt
                                                  <char>
    1:            inborn errors of steroid synthesis
    2:                 reproductive hormone analyses
    3:                 reproductive hormone analyses
    4:                 reproductive hormone analyses
    5:                 reproductive hormone analyses
   ---
25408:    non-site specific procedural complications
25409:         normal pregnancy, labour and delivery
25410:                 cervix therapeutic procedures
25411: male genital tract therapeutic procedures nec
25412:                       eosinophilic disorders
                                  pt
                                <char>
    1:   11-beta-hydroxylase deficiency
    2:            17 ketosteroids urine
    3:  17 ketosteroids urine decreased
    4:  17 ketosteroids urine increased
    5:     17 ketosteroids urine normal
   ---
25408:     incision site skin puckering
```

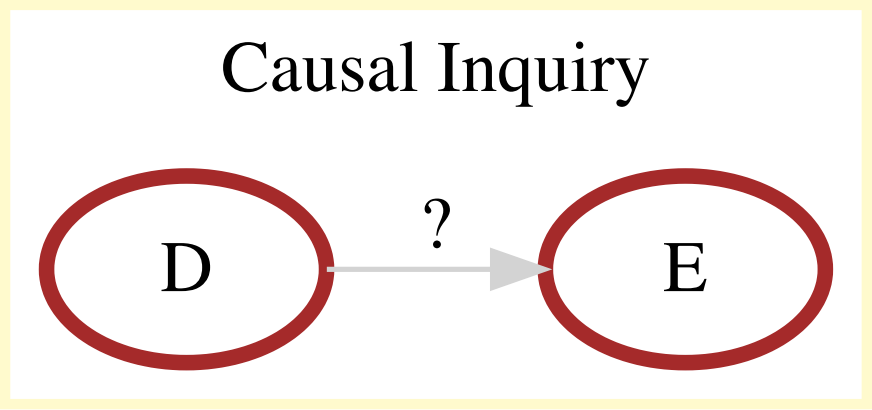
```
25409: spontaneous rupture of membranes
25410:           cervix stent placement
25411:           sperm cryopreservation
25412:      paraneoplastic eosinophilia
```

```
##If you own a subscription and want to access the MedDRA,
##you have to follow instructions in the
##https://github.com/fusarolimichele/DiAna github
##to shape it into the right format.
```

### Statistical and causal dependence

```
create_dag("D","E")
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfYwJP8/file63f72fa9d24/

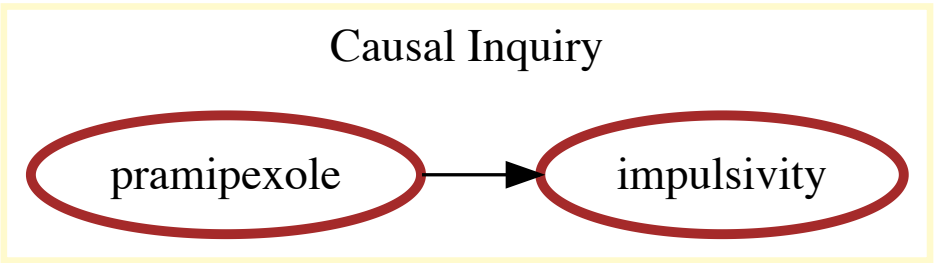

Causal Inquiry
D
?
E

Our inquiry, or causal estimand (grey arrow with a question mark), concerns the existence of a direct causal relationship between drug and event. Assuming, for the sake of the example, a simplified scenario with no other factor involved, a sample size sufficient to reliably exclude chance association, and no reporting bias, there are only two possible scenarios: a statistical independence suggesting causal independence or a statistical dependence suggesting causal dependence. DPA helps assess the statistical dependence (or association).

### Statistical dependence

### Direct causality

```
create_dag("pramipexole","impulsivity", scenario = "causal")
```


Causal Inquiry
pramipexole
impulsivity

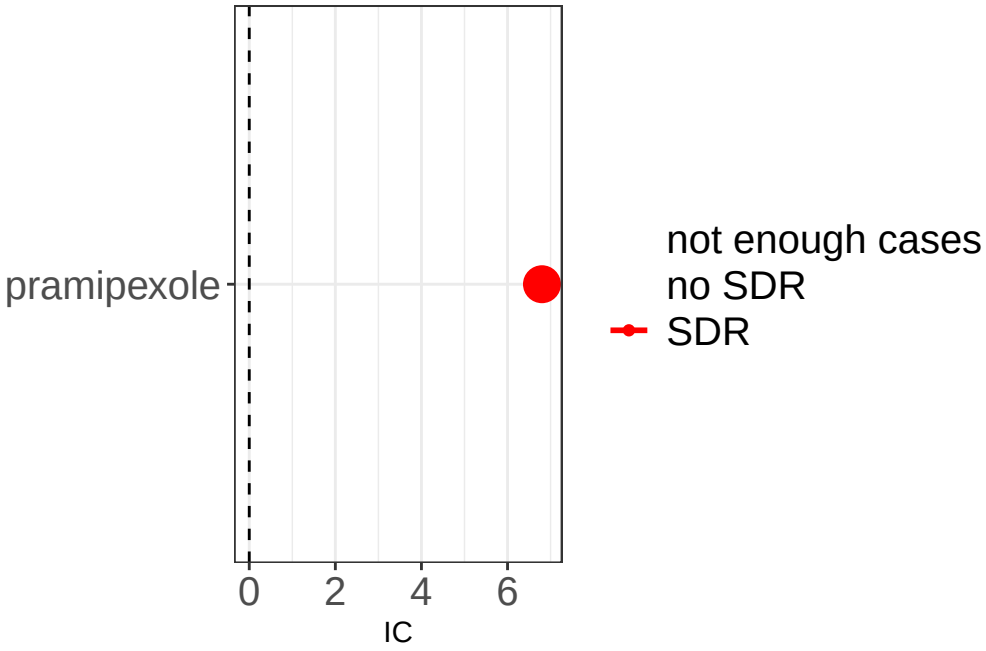
A well known example of direct causality (an actual adverse drug reaction) is pramipexole-induced impulse-control disorder. Here below we find a strong association between the reporting of pramipexole and the reporting of impulse-control disorders.

```
temp <- disproportionality_analysis("pramipexole","impulse-control disorder")
paste0("IC = ",temp$label_IC)
```

```
[1] "IC = 6.8 (6.63-6.92) [403]"
```

```
render_forest(temp)
```

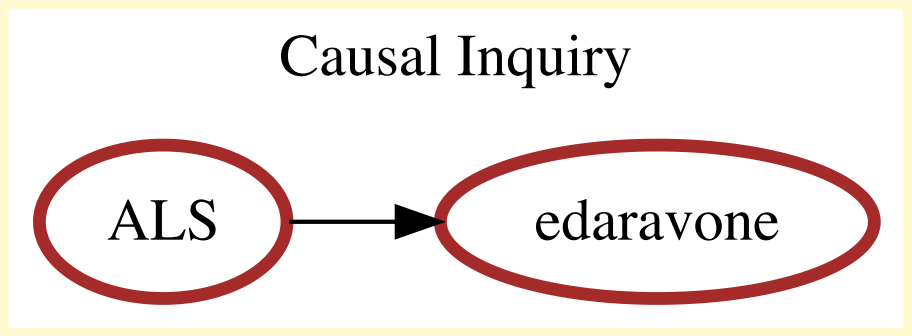
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Reverse causality**

```r
create_dag("ALS","edaravone",scenario = "causal")
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfYwJP8/file63f7400724d/


Causal Inquiry
ALS
edaravone

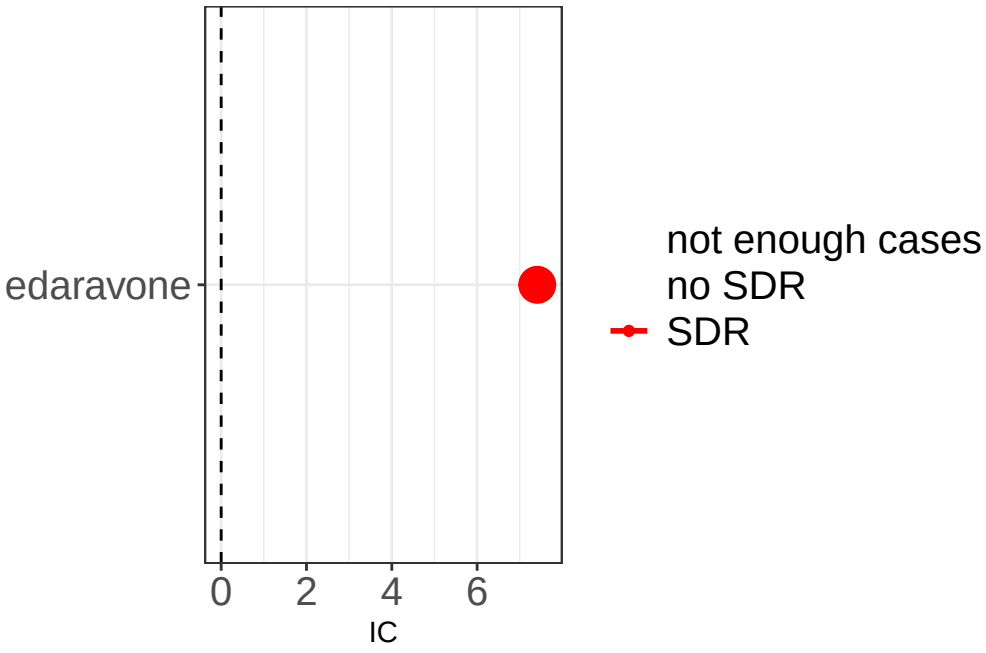
The occurrence of an event may justify the administration of a drug, as in the case where the event is the indication for using the drug. For example, edaravone is used to treat amyotrophic lateral sclerosis, and we will expect to find an association between the two (DPA cannot discriminate between direct and reverse causality). In these cases, clinical reasoning is crucial to correctly assess the association.

```
temp <- disproportionality_analysis("edaravone","amyotrophic lateral sclerosis")
paste0("IC = ",temp$label_IC)
```

```
[1] "IC = 7.41 (7.13-7.61) [144]"
```

```
render_forest(temp)
```

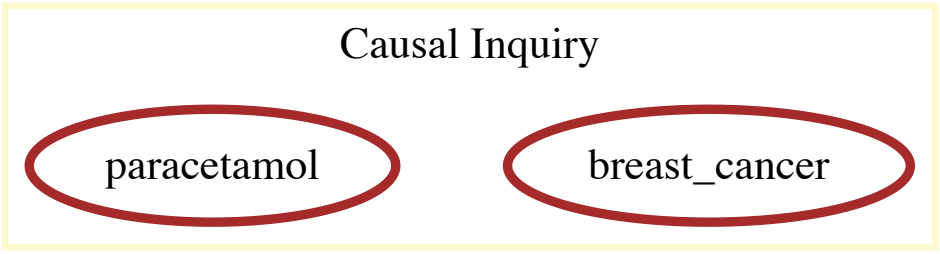
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```



### Statistical independence

```
create_dag("paracetamol","breast_cancer",scenario="non-causal")
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfYwJP8/file63f71c0e934e,


Causal Inquiry
paracetamol
breast_cancer

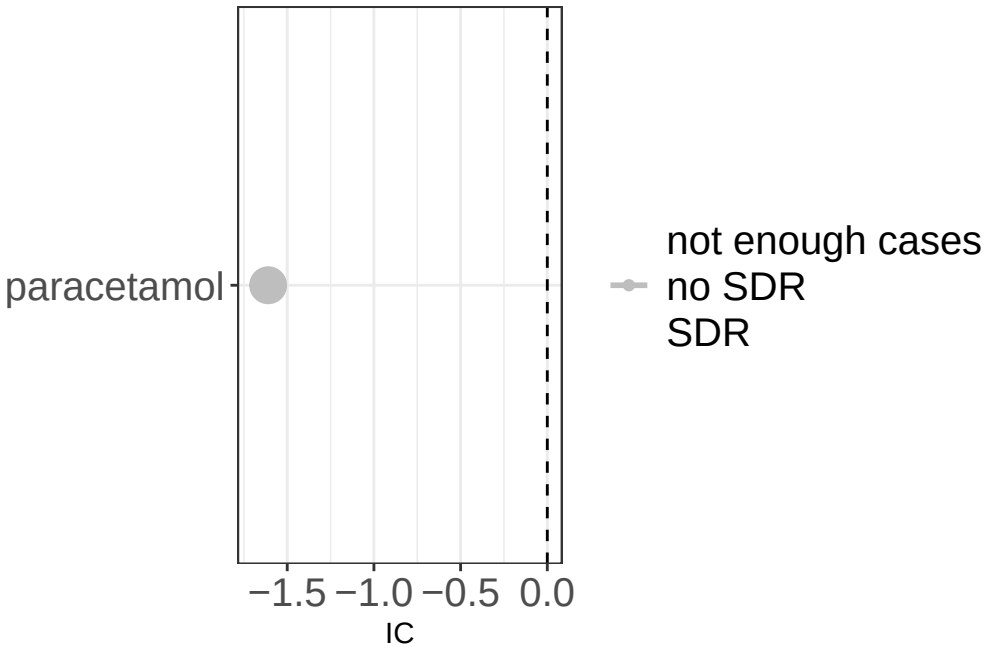
Based on what we know we expect that paracetamol has no causal dependence with breast cancer. For this reason we expect to find no association between the two. Note that the negative association that we find should not be interpreted as the existence of a protective effect of paracetamol agains breast cancer. First, DPA has amongst its goals to identify early warnings of ADRs, and not unexpected positive effects. Second, DPA relies on reports that are likely not representative of patients with the same susceptibility to the event but not being administered the drug. We could for example imagine that patients with breast cancer are less likely to record paracetamol in favour of anticancer drugs relative to patients with other diseases. Whether a causal inference informed pharmacovigilance could make better use of negative associations is outside the scope of this article.

```
temp <- disproportionality_analysis("paracetamol","breast cancer")
paste0("IC = ",temp$label_IC)
```

```
[1] "IC = -1.61 (-1.7--1.55) [1448]"
```

```
render_forest(temp)
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```

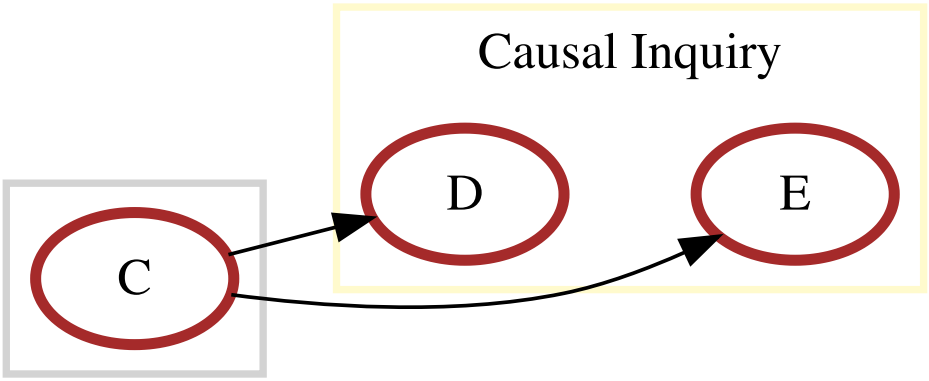



### Confounding bias

Many pharmacovigilance biases can be reframed as confounding biases, that is, open backdoor paths through which information flows even in absence of causal dependencies, thus potentially resulting in a spurious association. In all these cases the solution is the same: closing the backdoor, that is conditioning directly on this element - the confounder C -, or on any of the other nodes on the backdoor path.

```
create_dag("D","E",scenario = "non-causal",
           confounder_path = list(nodes=list("C"),
                                  signs=list("","")))
```

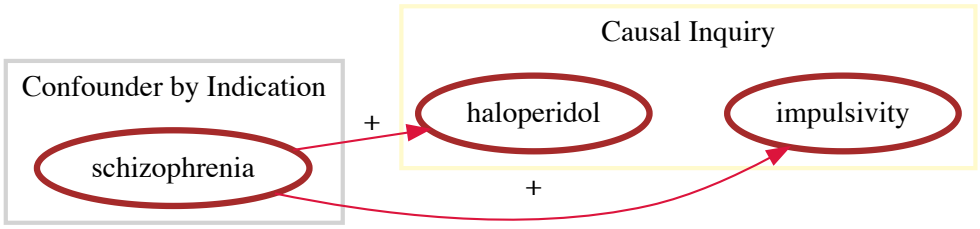
file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfecX71/filea9464d8db579/


Causal Inquiry
D
E
C

#### Confounding by indication #### Confounding by indication

```
create_dag("haloperidol","impulsivity",scenario = "non-causal",
           confounder_path = list(nodes=list("schizophrenia"),
                                  signs=list("+","+"),
                                  label="Confounder by Indication"))
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfecX71/filea94653769847/


Causal Inquiry
Confounder by Indication
haloperidol
impulsivity
schizophrenia
+
+

Let us consider a case of indication bias (an event associated with a drug is in fact caused by the drug indication). If we investigate the relation between haloperidol and gambling, we do indeed find a positive statistical association in the data (grey estimate). However, we need to consider that schizophrenia is likely a common cause of both: schizophrenia is both an approved indication for haloperidol, and often associated with impulsivity. Since patients prescribed haloperidol are likely to have schizophrenia, and patients with schizophrenia tend to be more impulsive, we might observe an association between haloperidol and impulsivity even without any causal effect of D on E. We should thus condition on schizophrenia, for instance by restricting our analysis to only reports recording schizophrenia among indications. As we do so, the association between haloperidol and impulsivity disappears (gold estimate), suggesting a lack of causation. Note, however, that restricting the analysis to reports that do not mention schizophrenia would not be as effective at closing the backdoor: many of these reports might still pertain to patients with schizophrenia without specifying that, and indeed in those reports we observe the same positive association (blue estimate) that we also observe without conditioning. In this scenario, appropriately conditioning on the confounder allowed us to discard a spurious signal of disproportionate reporting during signal refinement.

The negative association we get could suggest a protective effect of haloperidol against impulsivity. That seems indeed to be the case: by treating (some aspects of) schizophrenia, which also causes impulsivity, haloperidol can reduce impulsivity. However, in pharmacovigilance negative associations are typically not considered. This caution is again driven by the specific features of disproportionality analysis. First, DPA has amongst its goals to identify early warnings of ADRs, and not unexpected positive effects. Second, DPA relies on reports that are likely not representative of patients with the same susceptibility to the event but not being administered the drug. Whether a causal inference informed pharmacovigilance could make better use of negative associations is outside the scope of this article.

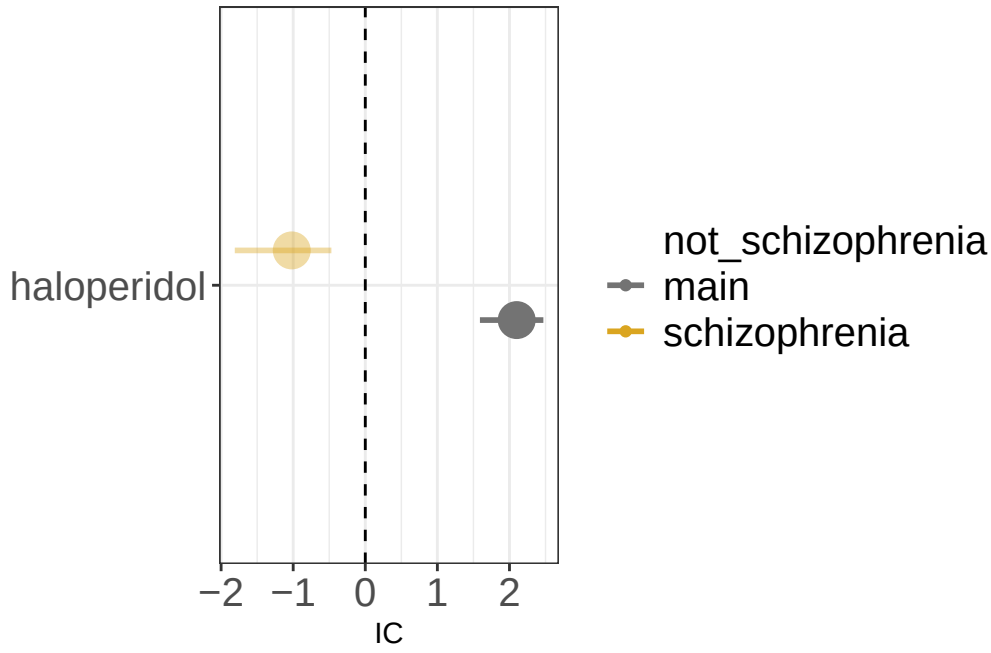
```
pids_schizophrenia <- unique(Indi[
  indi_pt%in%MedDRA[
    hlgt=="schizophrenia and other psychotic disorders"]$pt]$primaryid)

restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "schizophrenia"=list(pids_schizophrenia))
#,"not_schizophrenia"=list(setdiff(unique(Demo$primaryid),pids_schizophrenia)))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="haloperidol",
                                    reac_selected = "gambling disorder",
                                    meddra_level = "pt",
```

```
                                    restriction = t_pids)[,nested:=t_name]
    df_results <- rbindlist(list(df_results,df),fill = TRUE)
  }


  df_results <- df_results[,nested:=factor(nested,
                                           levels=c("not_schizophrenia",
                                                    "main","schizophrenia"),
                                           ordered=TRUE)]
  df_results[,.(nested,label_IC)]
```

```
          nested                 label_IC
           <ord>                   <char>
1:          main     2.1 (1.59-2.47) [42]
2: schizophrenia -1.02 (-1.81--0.47) [18]
```

```
  render_forest(df_results,nested = "nested",
                nested_colors=c("blue","gray45","goldenrod"))
```

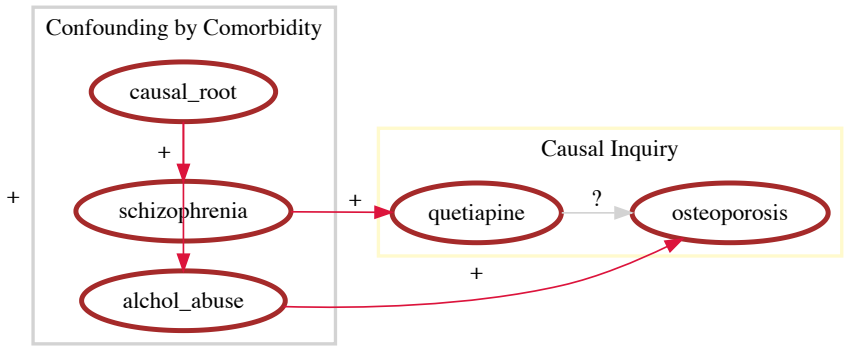
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Confounding by comorbidity**

```
create_dag("quetiapine", "osteoporosis", scenario = "inquiry",
           confounder_path = list(nodes = c("schizophrenia","causal_root",
                                            "alchol_abuse"),
                                  signs = c("+","+","+","+"),
                                  direction=c("","back"),
                                  label="Confounding by Comorbidity"))
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfecX71/filea94645319c72,


Confounding by Comorbidity
causal_root
+
+
schizophrenia
+
alchol_abuse
Causal Inquiry
quetiapine
?
osteoporosis
+

Another instance involves the issue of confounding by comorbidity (a comorbidity of the indication for using the drug is the real cause of the event). Here, our focus might be on investigating whether quetiapine leads to osteoporosis. However, since quetiapine is often prescribed for schizophrenia, which frequently co-occurs with alcoholism—a known risk factor for osteoporosis—we encounter a positive confounding effect. This means we could find a statistical dependence (grey estimate) even in the absence of a causal dependence. When we condition on reports of alcoholism, the association tends to vanish (gold estimate), although we cannot determine whether this is simply due to increased uncertainty, since only a small number of reports mentions alcoholism-related terms. Alternatively, we could adjust for schizophrenia, which would mitigate the positive confounding effect. Note, however, this adjustment would also address a negative confounding factor not depicted in the Directed Acyclic Graph (DAG): the association between schizophrenia and younger age, which correlates with a reduced risk of osteoporosis. For this reason it is always important to remember that DAG help causal inference, but that these inferences are true as long as the assumptions drawn in the DAG are verified (and may be distorted by an incomplete or wrong DAG). Thus, the impact of conditioning is less straightforward to predict. Another possibility would be to restrict to all reports without mentions of alcoholism (blue estimate), however, alcoholism is drastically under-reported in ICSRs, and the conditioning unlikely to be effective in closing the backdoor path - information would still flow through the reports from alcoholic patients, not reporting alcoholism.

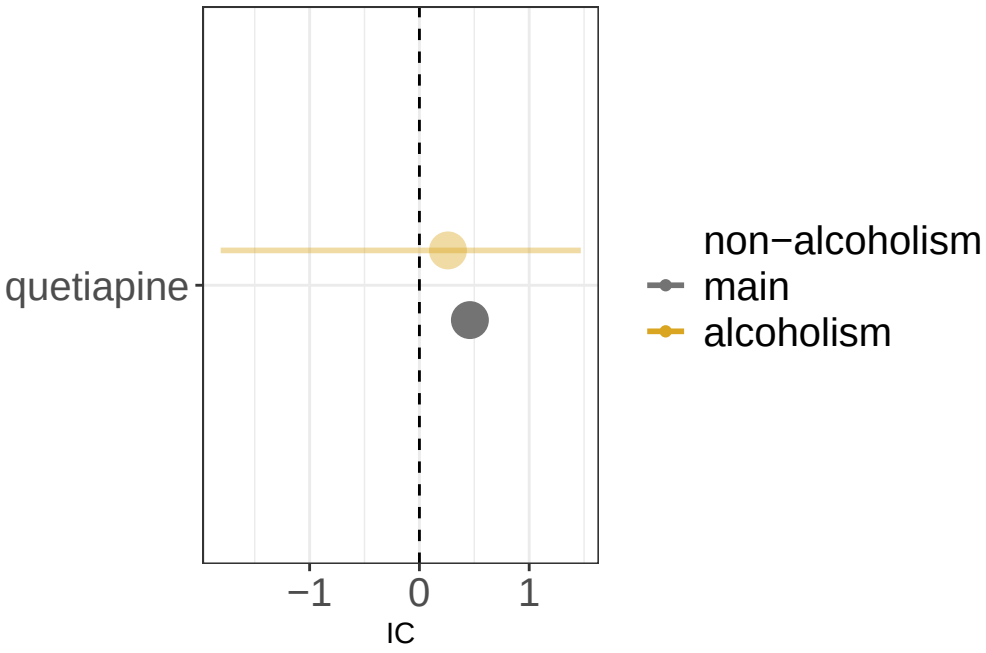
```
alcoholism <- c("alcoholism","alcohol problem","alcoholic encephalopathy",
                "alcoholic coma",
                "alcohol use disorder","alcoholic ketoacidosis",
                "ex-alcohol user",
                "alcoholic pancreopathy","ex-alcoholic",
                "alcoholic pancreatitis",
                "prophylaxis against alcoholic withdrawal syndrome",
                "alcoholic seizure","alcohol rehabilitation",
                "alcohol withdrawal syndrome",
                "alcohol detoxification","alcohol use","social alcohol drinker",
                "polyneuropathy alcoholic","hepatitis alcoholic",
                "gastritis alcoholic haemorrhagic",
                "gastritis alcoholic","fatty liver alcoholic",
                "cirrhosis alcoholic",
                "cardiomyopathy alcoholic","amblyopia alcohol",
                "alcoholic psychosis",
                "alcoholic pseudo cushing's syndrome","alcoholic liver disease",
                "alcoholic hangover","alcohol poisoning",
                "alcohol induced persisting dementia",
                "alcohol abuse")
pids_alcoholism <- union(Reac[pt%in%alcoholism]$primaryid,
```

```
                          Indi[indi_pt%in%alcoholism]$primaryid)
  restrictions <- list("main"=list(unique(Demo$primaryid)),
                       "alcoholism"=list(pids_alcoholism))#,
              #   "non-alcoholism"=list(setdiff(Demo$primaryid,pids_alcoholism)))

  df_results <- data.table()
  for (n in 1:length(restrictions)){
    t <- restrictions[n]
    t_name <- names(t)
    t_pids <- unlist(t)
    df <- disproportionality_analysis(drug_selected="quetiapine",
                                      reac_selected = "osteoporosis",
                                      meddra_level = "pt",
                                      restriction = t_pids)[,nested:=t_name]
    df_results <- rbindlist(list(df_results,df),fill = TRUE)
  }

  df_results <- df_results[
    ,nested:=factor(nested, levels=c("non-alcoholism","main","alcoholism"),
                    ordered=TRUE)]
  df_results[,.(nested,label_IC)]
```

```
       nested              label_IC
        <ord>                 <char>
1:       main 0.46 (0.3-0.57) [461]
2: alcoholism 0.26 (-1.81-1.47) [3]
```

```
  render_forest(df_results,nested = "nested",
                nested_colors=c("blue","gray45","goldenrod"))
```

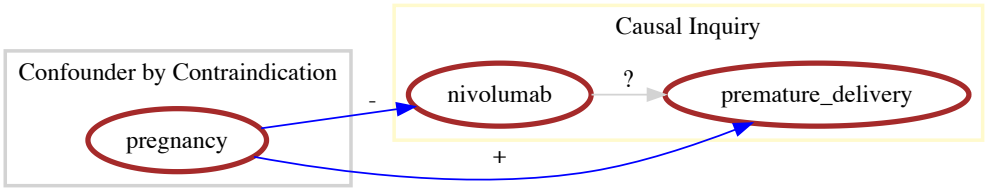
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Confounding by differential prescription**

```
create_dag("nivolumab","premature_delivery",
           confounder_path = list(nodes=c("pregnancy"),signs=c("-","+"),
                                  label="Confounder by Contraindication"))
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpRzIHCY/file80da1763f2fe,


Causal Inquiry
Confounder by Contraindication
nivolumab
?
premature_delivery
pregnancy
-
+

A third example concerns confounding by differential prescription (allocation bias/channelling bias). The statistical dependence is here generated by the existence of differential prescriptive habits in subgroups with different susceptibility to the event. If we investigate the relation between nivolumab and risk of preterm birth, we do find a negative statistical association (grey estimate), that is, no evidence of a positive association, according to pharmacovigilance standards (see footnote 2). However, pregnancy is associated with both the risk of the event (no pregnancy, no preterm delivery), and a lower use of nivolumab. The drug is indeed unlikely to be used during pregnancy due to lower probability of cancer in fertile age and lack of safety evidence for this population (partial contraindication). Drawing the DAG, we can infer that confounding would push towards a negative association (odd number of minus signs): nivolumab is less used in pregnancy and therefore is less co-reported with preterm birth. Restricting to cases of pregnancy, a positive association between nivolumab and preterm birth emerges (gold estimate). In this instance, not conditioning on the confounder would have led to missing a potential safety signal during signal detection.

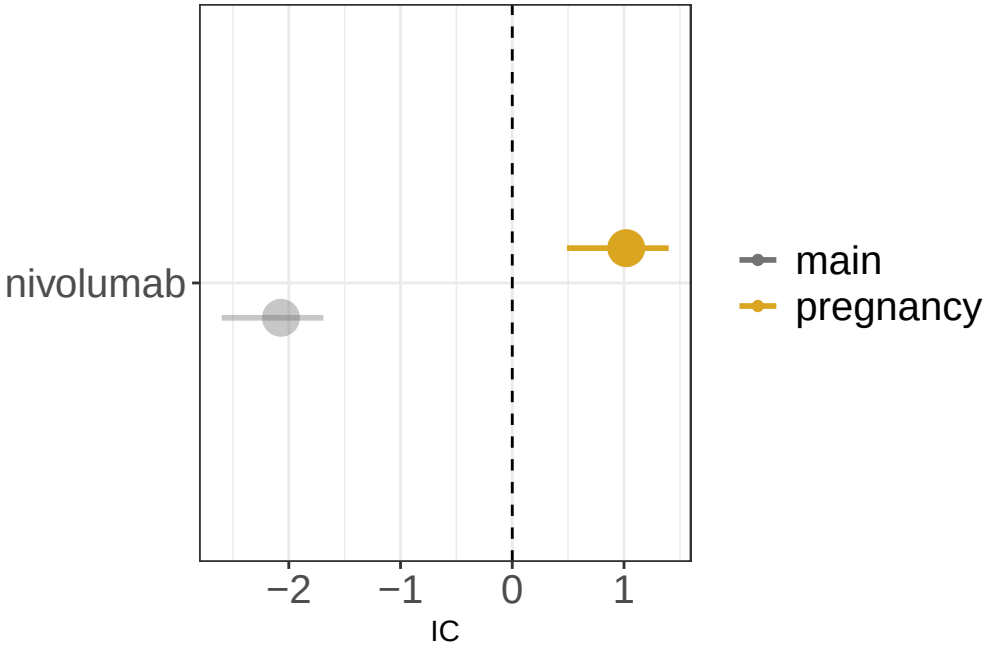
```
pids_pregnancy_indi <- unique(Indi[
  indi_pt%in%MedDRA[
    soc=="pregnancy, puerperium and perinatal conditions"]$pt]$primaryid)
pids_pregnancy_reac <- unique(Reac[
  pt%in%MedDRA[
    soc=="pregnancy, puerperium and perinatal conditions"]$pt]$primaryid)
pids_pregnancy <- union(pids_pregnancy_indi,pids_pregnancy_reac)

restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "pregnancy"=list(pids_pregnancy))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="nivolumab",
                                    reac_selected = list("preterm birth"=list(
                                      "premature baby","premature delivery")),
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}
df_results <- df_results[,nested:=factor(nested, levels=c("main","pregnancy"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
   nested                label_IC
    <ord>                  <char>
```

```
1:      main -2.07 (-2.6--1.69) [39]
2: pregnancy    1.02 (0.49-1.4) [39]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("gray45","goldenrod"))
```

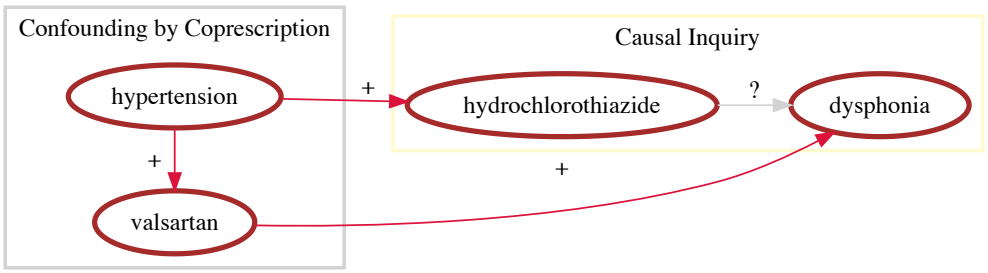
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Confounding by coprescription**

```
create_dag("hydrochlorothiazide", "dysphonia",
           label_inquiry = "Causal Inquiry", scenario = "inquiry",
           confounder_path = list(nodes = c("hypertension","valsartan"),
                                  signs = c("+","+","+"),
                                  label="Confounding by Coprescription"))
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpRzIHCY/file80da79cd07ea/


Confounding by Coprescription
Causal Inquiry
hypertension
+
hydrochlorothiazide
?
dysphonia
+
+
valsartan

Confounding due to co-prescription is generated by the frequent coprescription of the drug of interest with a drug that is the real cause of the event. When examining the potential link between hydrochlorothiazide and dysphonia, it is important to consider that hydrochlorothiazide is commonly co-prescribed with valsartan, which causes dysphonia through coughing. By adjusting for reports mentioning valsartan (gold estimate), we can infer that the observed association between the drug and the event (grey estimate) is likely solely due to the co-prescription with valsartan. In both confounding by comorbidity and by co-prescription we see again that conditioning on reports not recording the value (blue estimate) does not seem to affect the IC estimate, likely due to residual unaccounted confounding.

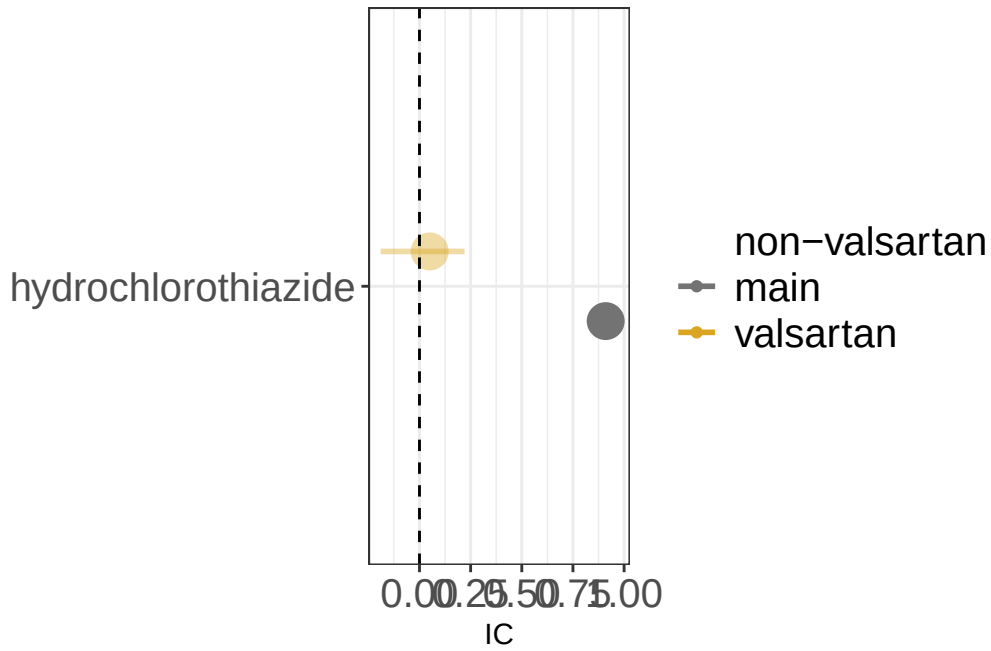
```
pids_valsartan <- unique(Drug[substance=="valsartan"]$primaryid)

restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "valsartan"=list(pids_valsartan))#,
                   #"non-valsartan"=list(setdiff(Demo$primaryid,pids_valsartan)))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="hydrochlorothiazide",
                                    reac_selected = "dysphonia",
                                    meddra_level = "pt",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}

df_results <- df_results[,nested:=factor(nested, levels=c("non-valsartan",
                                                          "main","valsartan"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
      nested                label_IC
       <ord>                  <char>
1:      main 0.91 (0.83-0.97) [1606]
2: valsartan 0.05 (-0.19-0.22) [186]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","gray45","goldenrod"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
```

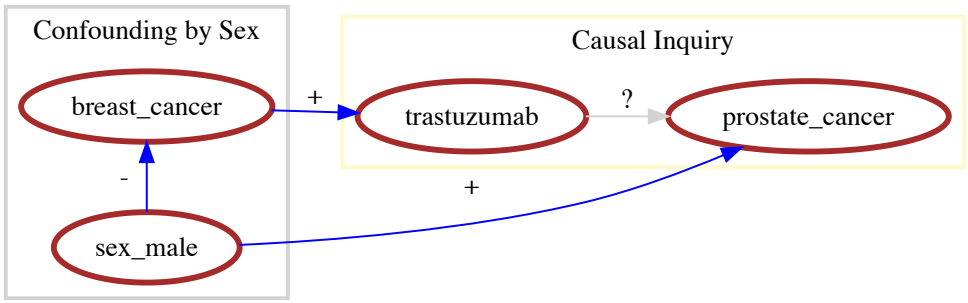
```
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Confounding by demographic factor**

```
create_dag("trastuzumab", "prostate_cancer",
           label_inquiry = "Causal Inquiry", scenario = "inquiry",
           confounder_path = list(nodes = c("breast_cancer","sex_male"),
                                  signs = c("+","-","+"),
                                  direction=c("","back"),
                                  label="Confounding by Sex"))
```

```
file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfecX71/filea946511e0840,
```


Confounding by Sex
Causal Inquiry
breast_cancer
+
trastuzumab
?
prostate_cancer
-
+
sex_male

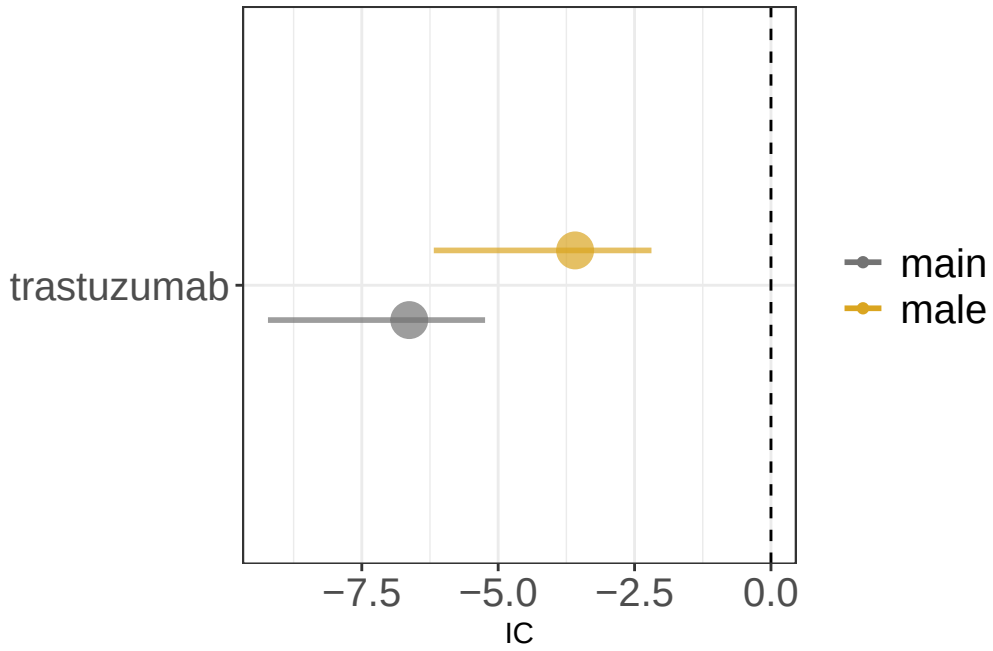
Lastly, demographic factors can also introduce confounding. For instance, when exploring the potential link between trastuzumab and prostate cancer, we must consider that among all individuals receiving trastuzumab (primarily for breast cancer), only a small subset is male and thus at risk of prostate cancer. This demographic confounding can decrease the statistical dependence (grey estimate), although it remains statistically not significant when focusing solely on males (gold estimate).

```
pids_male <- Demo[sex=="M"]$primaryid

restrictions <- list("main"=list(unique(Demo$primaryid)),"male"=list(pids_male))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="trastuzumab",
                                    reac_selected = "prostate cancer",
                                    meddra_level = "pt",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}
df_results <- df_results[,nested:=factor(nested, levels=c("main","male"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
   nested                label_IC
    <ord>                  <char>
1:   main -6.63 (-9.22--5.24) [2]
2:   male -3.59 (-6.18--2.19) [2]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("gray45","goldenrod"))
```

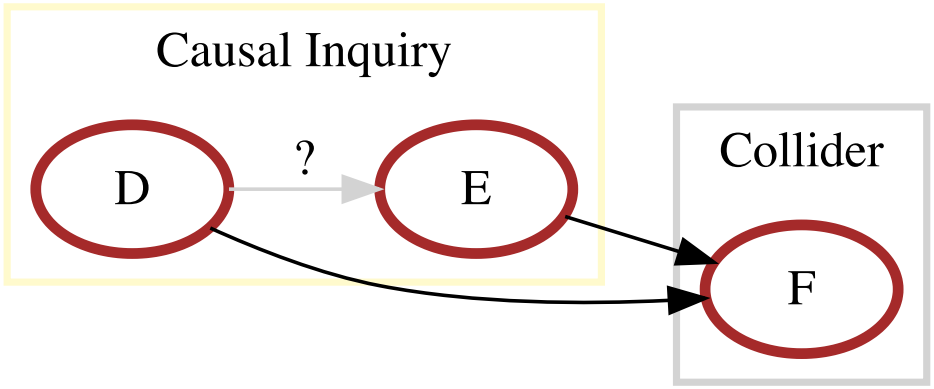
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Collider bias**

```
create_dag("D", "E", label_inquiry = "Causal Inquiry", scenario = "inquiry",
           collider_path = list(nodes = c("F"), signs = c("",""),
                                label="Collider"))
```

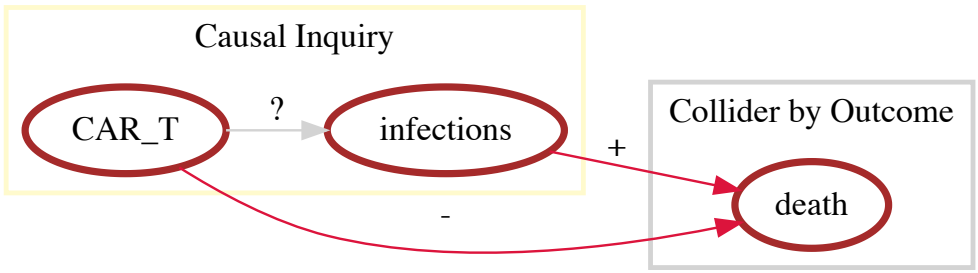
file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfYwJP8/file63f7328c416a/


Causal Inquiry
D
?
E
Collider
F

**Collider bias by outcome**

```
create_dag("CAR_T", "infections", label_inquiry = "Causal Inquiry",
           scenario = "inquiry",
           collider_path = list(nodes = c("death"), signs = c("-","+"),
                                label="Collider by Outcome"))
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpRzIHCY/file80da2fe72324/


Causal Inquiry
CAR_T
?
infections
+
-
Collider by Outcome
death

If we investigate the causal dependence between Chimeric Antigen Receptor T (CAR-T) therapy and infections, we find a positive statistical association (grey estimate). Were we to limit our analysis to reports that mention death as one of the outcomes we would find an even stronger association (red estimate). This is because CAR-T therapies reduce mortality, and infections themselves may lead to mortality, thus restricting to reports mentioning death opens a path and likely introduces a positive distortion into the data. Again, were we to restrict to reports not mentioning death, we would produce a noisy inference (blue estimate).

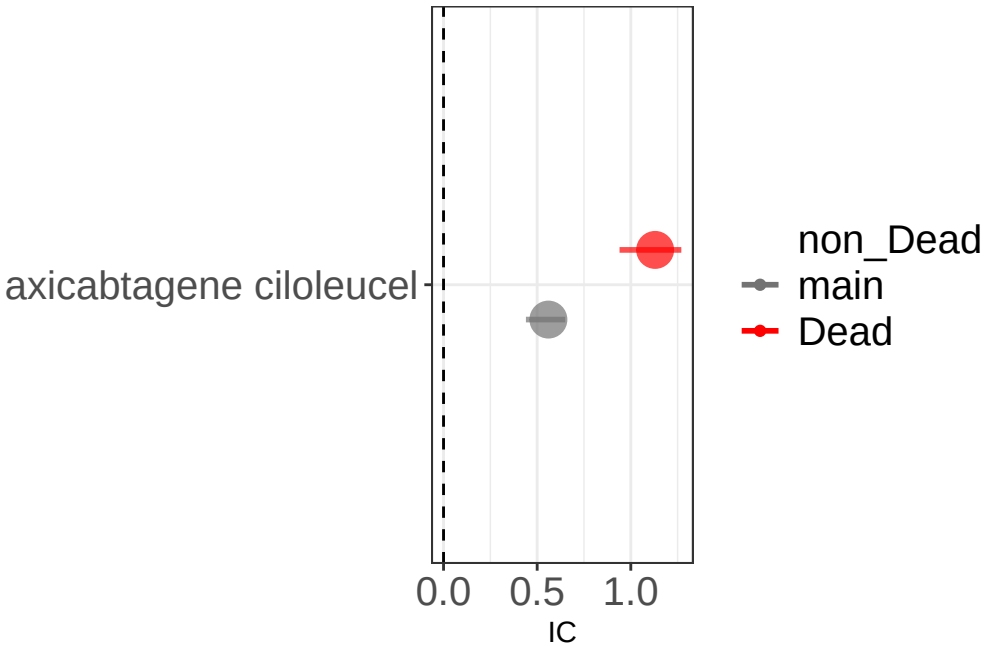
```
pids_death <- unique(Outc[outc_cod=="DE"]$primaryid)
restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "Dead"=list(pids_death))
#,"non_Dead"=list(setdiff(unique(Demo$primaryid),pids_death)))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="axicabtagene ciloleucel",
                                    reac_selected = "infections and infestations",
                                    meddra_level = "soc",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}
df_results <- df_results[,nested:=factor(nested,
                                         levels=c("non_Dead","main","Dead"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
   nested               label_IC
    <ord>                  <char>
1:   main 0.56 (0.44-0.65) [775]
2:   Dead 1.13 (0.94-1.27) [308]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","gray45","red"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
```

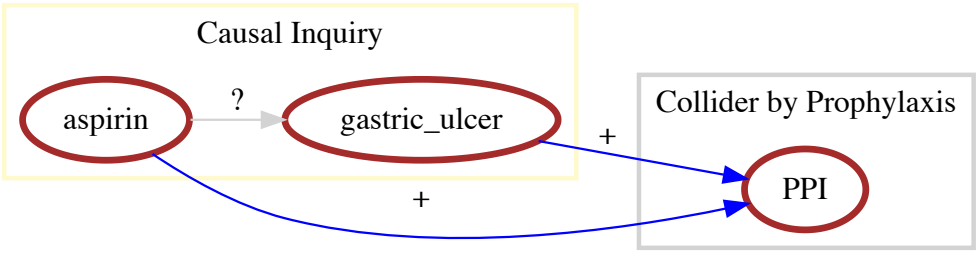
```
i Did you misspell an argument name?
```




**Collider bias by prophylaxis**

```
create_dag("aspirin", "gastric_ulcer", label_inquiry = "Causal Inquiry",
           scenario = "inquiry",
           collider_path = list(nodes = c("PPI"), signs = c("+","+"),
                                label="Collider by Prophylaxis"))
```

```
file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpRzIHCY/file80da4f4a2496/
```


Causal Inquiry
aspirin
?
gastric_ulcer
+
+
Collider by Prophylaxis
PPI

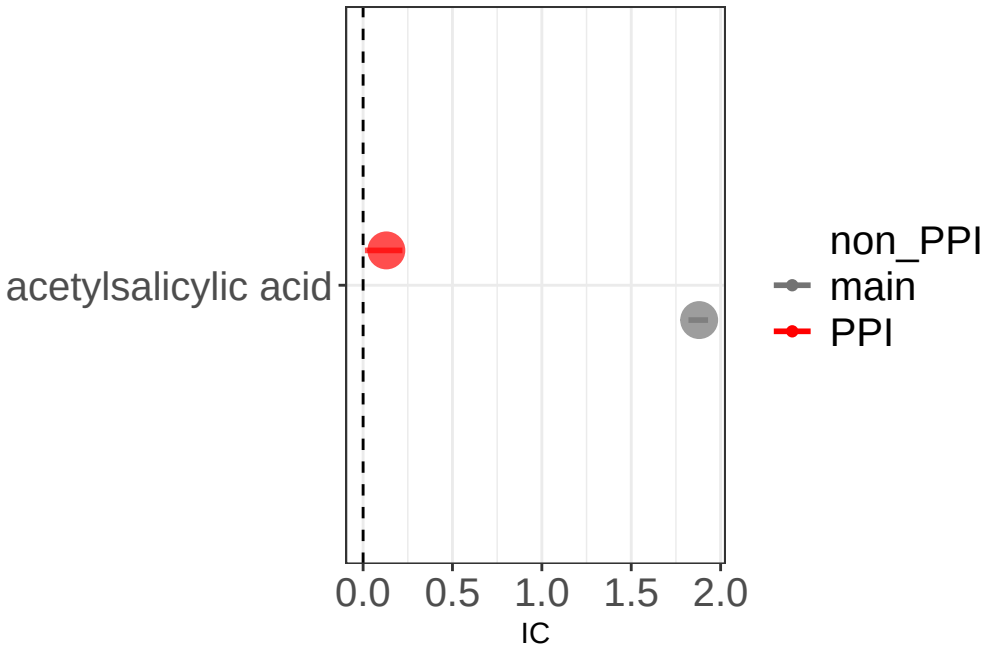
If we investigate the relation between aspirin and gastric ulcer, we find a positive statistical association (grey estimate). However, we should be wary of conditioning on whether the report also mentions proton pump inhibitors (PPIs). PPIs can indeed be used to treat gastric ulcer or – if administered together with aspirin - to prevent it. In this case, if we condition on the use of proton pump inhibitors (red estimate), the expected association between aspirin and gastric ulcer is going to be biased towards the null: among patients taking PPIs, those not taking aspirin are more likely to not be in prophylaxis and therefore to already have gastric ulcer. Therefore, by conditioning on a collider we risk losing a potential signal. Once more, we observe that adjusting for reports that do not mention PPIs (blue estimate) has minimal impact on the IC estimate. This is likely because the reporting rate of PPIs when they are not the suspected drug is presumed to be very low.

```
pids_PPI <- unique(Drug[
  substance%in%ATC[Class4=="Proton pump inhibitors"]$substance]$primaryid)

restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "PPI"=list(pids_PPI))
#,"non_PPI"=list(setdiff(unique(Demo$primaryid),pids_PPI)))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="acetylsalicylic acid",
                                    reac_selected = "gastric ulcer",
                                    meddra_level = "pt",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}
df_results <- df_results[,nested:=factor(nested,
                                         levels=c("non_PPI","main","PPI"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
   nested                label_IC
    <ord>                  <char>
1:   main 1.88 (1.82-1.93) [2757]
2:    PPI  0.13 (0.01-0.22) [709]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","gray45","red"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




### Collider bias by double mechanism

Another scenario arises when we are aware that a drug induces an event, and the suspected reaction itself may contribute to that event (i.e., a potential indirect effect of the drug). For example, if we are interested in the causal dependence between Glucagone Like Peptide GLP 1-agonists and anxiety, their statistical association is negative (grey estimate). However, we have to be wary that both GLP 1-agonists and anxiety cause weight-loss. Therefore, if we were to condition on weight-loss we would open a path and introduce a negative selection bias (red estimate). In this case, the direction of the statistical association does not change, but we can anticipate and then confirm the shift towards a more negative estimate. Once again, adjusting for reports that do not mention weight loss does not change the outcome (blue estimate).

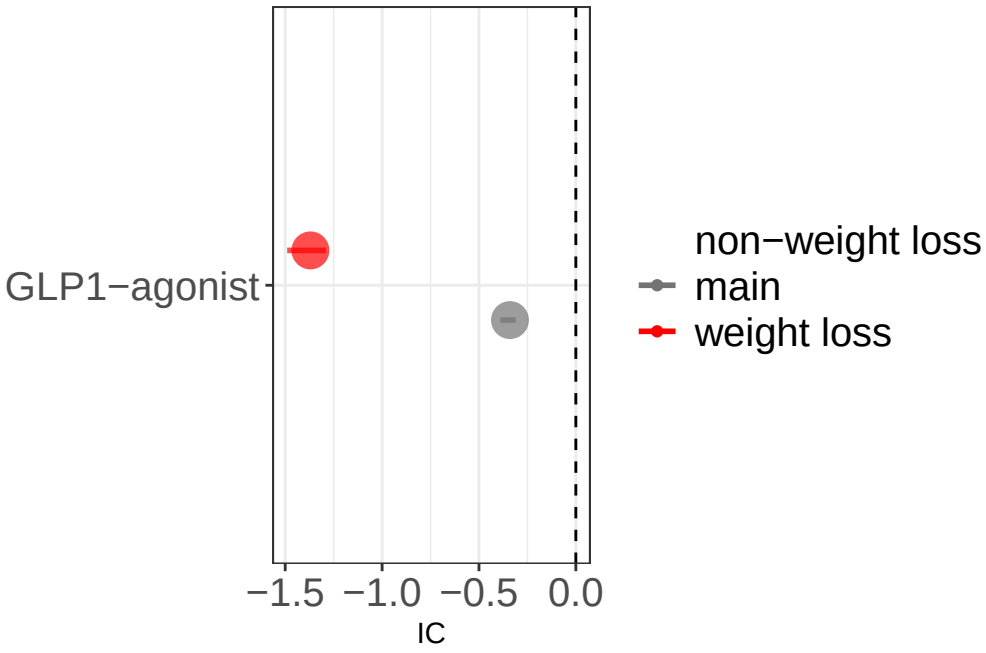
```
pids_weight <- union(Reac[pt%in%c("weight decreased")]$primaryid,
                     Indi[indi_pt%in%c("weight decreased")]$primaryid)
restrictions <- list("main"=list(unique(Demo$primaryid)),"weight loss"=list(pids_weight))
```

```
  ##,"non-weight loss"=list(setdiff(Demo$primaryid,pids_weight)))

  df_results <- data.table()
  for (n in 1:length(restrictions)){
    t <- restrictions[n]
    t_name <- names(t)
    t_pids <- unlist(t)
    df <- disproportionality_analysis(drug_selected=list(
      "GLP1-agonist"=list("liraglutide","exenatide","semaglutide","dulaglutide")),
      reac_selected = "anxiety disorders and symptoms",
      meddra_level = "hlgt",
      restriction = t_pids)[,nested:=t_name]
    df_results <- rbindlist(list(df_results,df),fill = TRUE)
  }

  df_results <- df_results[,nested:=factor(nested,
                                           levels=c("non-weight loss",
                                                    "main",
                                                    "weight loss"),ordered=TRUE)]
  df_results[,.(nested,label_IC)]
```

```
       nested                    label_IC
        <ord>                      <char>
1:       main -0.34 (-0.39--0.31) [4412]
2: weight loss  -1.37 (-1.49--1.29) [802]
```

```
  render_forest(df_results,nested = "nested",
                nested_colors=c("blue","gray45","red"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```



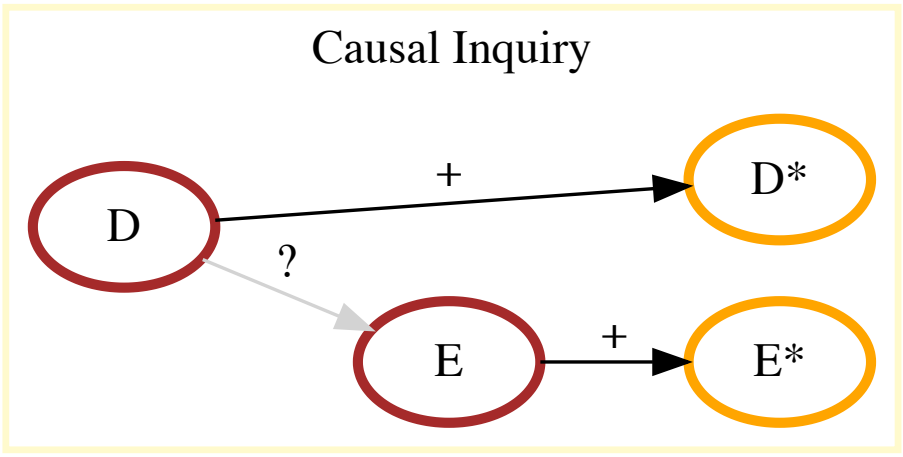

**Measurement errors**

The problem is that we do not have direct access to constructs (i.e., real exposure, outcome, and other variables/characteristics of realities), but only to their measurements (marked with an asterisk *). These measurements can be conceptualized as surrogate variables (or proxies) of their own construct, which, while not accurate, can still be appropriate and useful to answer causal questions if they share their relevant variance with their underlying construct. In other words, a measure, and any conditioning performed on it, is appropriate only if we can rely that it mirrors with a negligible error the value of the underlying construct. This is often not a given in ICSRs because of incompleteness: while we can rely in general that if someone measured (or recorded, or reported) something that something actually occurred, we often cannot say the opposite as, for example, the indication is often left unspecified, and the concomitants (which may have for example a role as confounders or colliders for the causal inquiry at hand) are often not reported. In fact the disproportionality is itself not a measure of association between construct (that is, D and E), but between measures (that is, D* and E*), and this is part of what is meant when we say that disproportionality is a measure of association between the reporting of a drug and the reporting of an event. D-criteria still work, as what we want is that any association between D* and E* can be explained only by a direct causal relation between D and E. Particular attention has therefore to be granted to those situations in which the lack of a construct is derived from the lack of a value in the report (i.e., the measurement) but may in fact be related only with the pervasive incompleteness of ICSRs. In particular, subgrouping on the presence of the value, as done above, is in general safe. In the following

paragraphs we show how subgrouping on the lack of a value is instead often so unreliable that we cannot use it as a way to address a bias.

```
create_dag("D", "E", scenario = "inquiry", add_measurements = TRUE,
           label_inquiry ="Causal Inquiry")
```

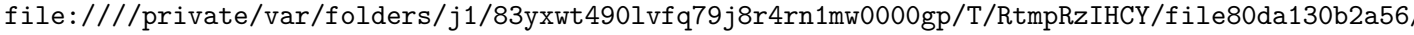
file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfYwJP8/file63f73b39eb15/

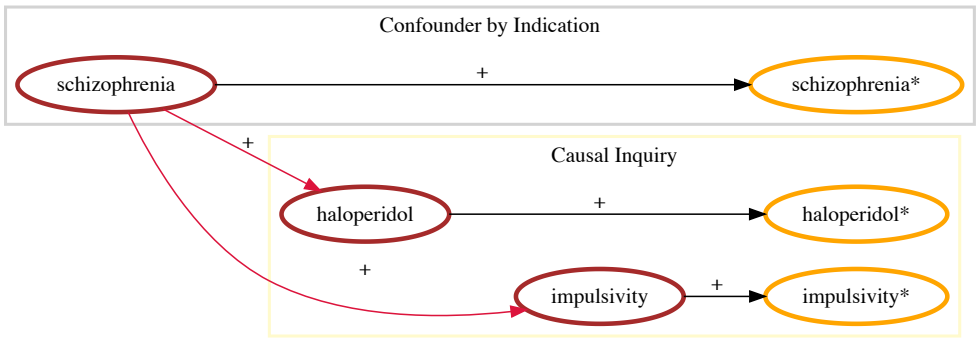

Causal Inquiry
D
+
D*
?
E
+
E*

### Noise from conditioning on inaccurate confounder measure

Let's take again into account the confounders presented above, excluding the pregnancy and the sex related biases as it does not make sense in them to condition on the lack of the value (as they are necessary conditions for the event to manifest). This time let's introduce measurements and assess the differences of conditioning on the presence vs on the lack of the value.

```
create_dag("haloperidol","impulsivity",scenario = "non-causal",
           confounder_path = list(nodes=list("schizophrenia"),
                                  signs=list("+","+"),
                                  label="Confounder by Indication"),
           add_measurements = TRUE)
```




Confounder by Indication
schizophrenia
+
schizophrenia*
+
Causal Inquiry
haloperidol
+
haloperidol*
+
impulsivity
+
impulsivity*

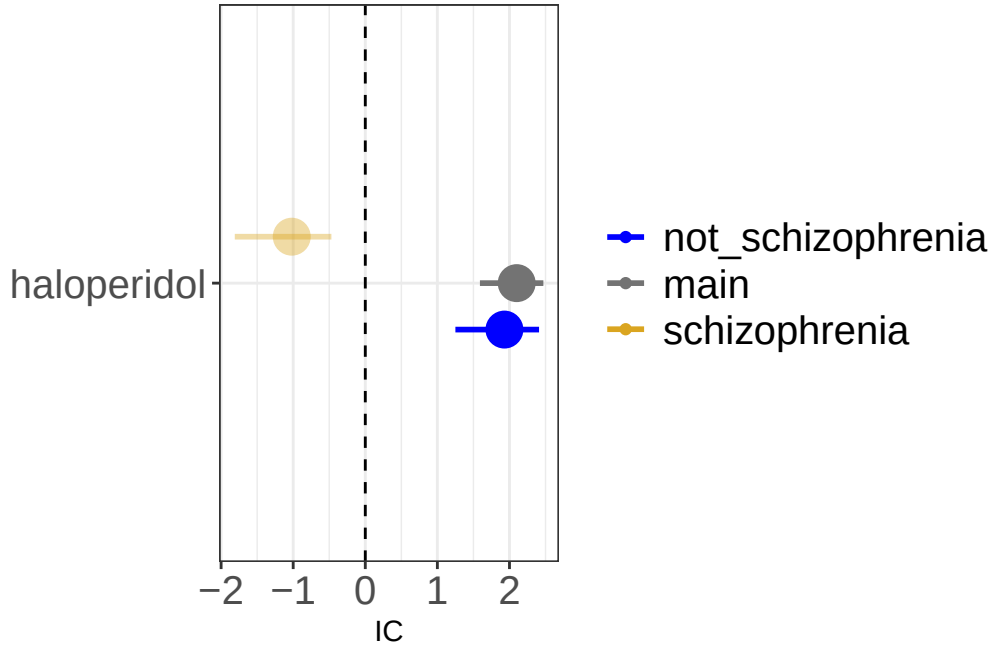
```
pids_schizophrenia <- unique(Indi[
  indi_pt%in%MedDRA[hlgt=="schizophrenia and other psychotic disorders"]$pt
  ]$primaryid)

restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "schizophrenia"=list(pids_schizophrenia),
                     "not_schizophrenia"=list(setdiff(
                       unique(Demo$primaryid),
                       pids_schizophrenia)))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="haloperidol",
                                    reac_selected = "gambling disorder",
                                    meddra_level = "pt",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}


df_results <- df_results[,nested:=factor(nested,
                                         levels=c("not_schizophrenia",
                                                  "main",
                                                  "schizophrenia"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
             nested                 label_IC
              <ord>                   <char>
1:              main     2.1 (1.59-2.47) [42]
2:     schizophrenia -1.02 (-1.81--0.47) [18]
3: not_schizophrenia    1.93 (1.25-2.41) [24]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","gray45","goldenrod"))
```

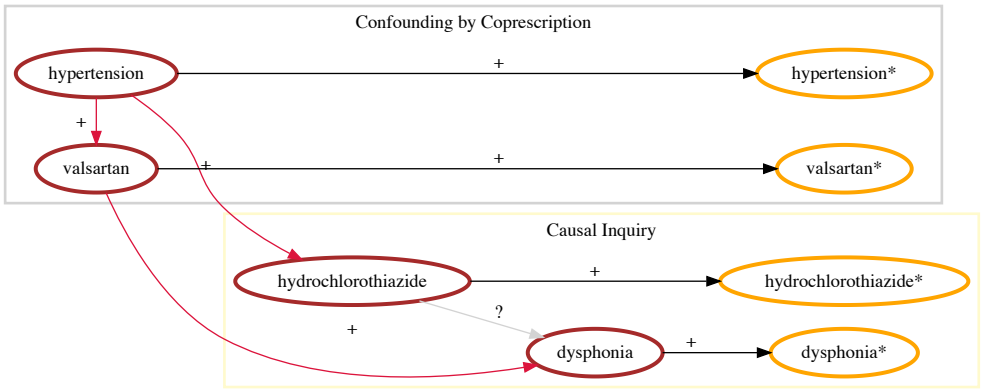
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
```

```
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




```
create_dag("hydrochlorothiazide", "dysphonia",
           label_inquiry = "Causal Inquiry", scenario = "inquiry",
           confounder_path = list(nodes = c("hypertension","valsartan"),
                                  signs = c("+","+","+"),
                                  label="Confounding by Coprescription"),
           add_measurements = TRUE)
```

```
file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpRzIHCY/file80da196c9b3c/
```

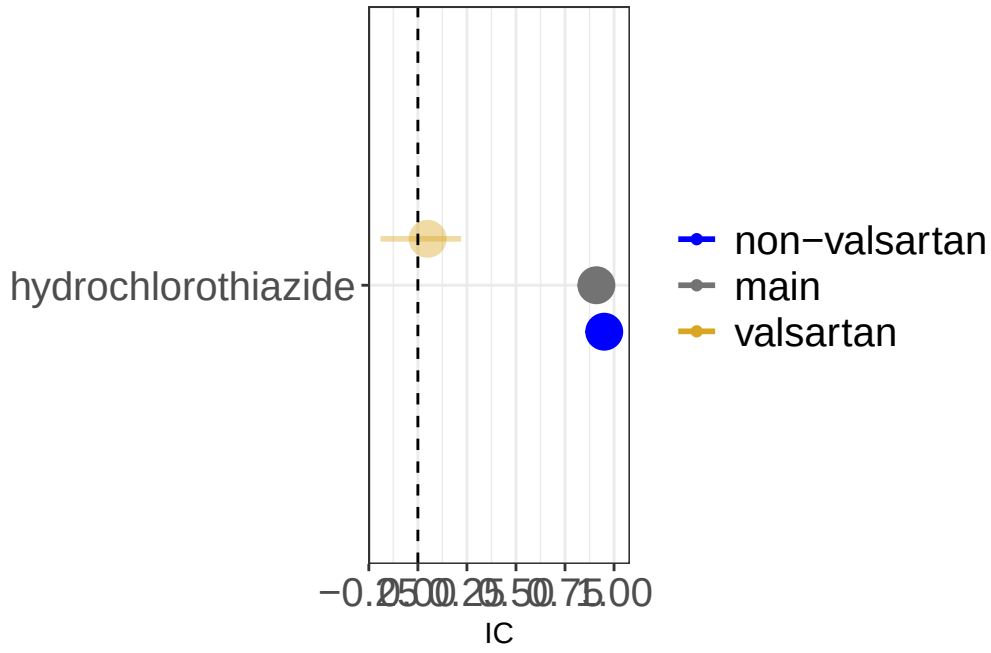

Confounding by Coprescription
hypertension
+
hypertension*
+
valsartan
+
+
valsartan*
Causal Inquiry
hydrochlorothiazide
+
hydrochlorothiazide*
?
+
dysphonia
+
dysphonia*

```
pids_valsartan <- unique(Drug[substance=="valsartan"]$primaryid)

restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "valsartan"=list(pids_valsartan),
                     "non-valsartan"=list(setdiff(Demo$primaryid,
                                                  pids_valsartan)))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="hydrochlorothiazide",
                                    reac_selected = "dysphonia",
                                    meddra_level = "pt",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}

df_results <- df_results[,nested:=factor(nested,
                                         levels=c("non-valsartan",
                                                  "main",
                                                  "valsartan"),ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
          nested                label_IC
           <ord>                  <char>
1:          main 0.91 (0.83-0.97) [1606]
2:     valsartan 0.05 (-0.19-0.22) [186]
3: non-valsartan 0.95 (0.86-1.02) [1420]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","gray45","goldenrod"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```

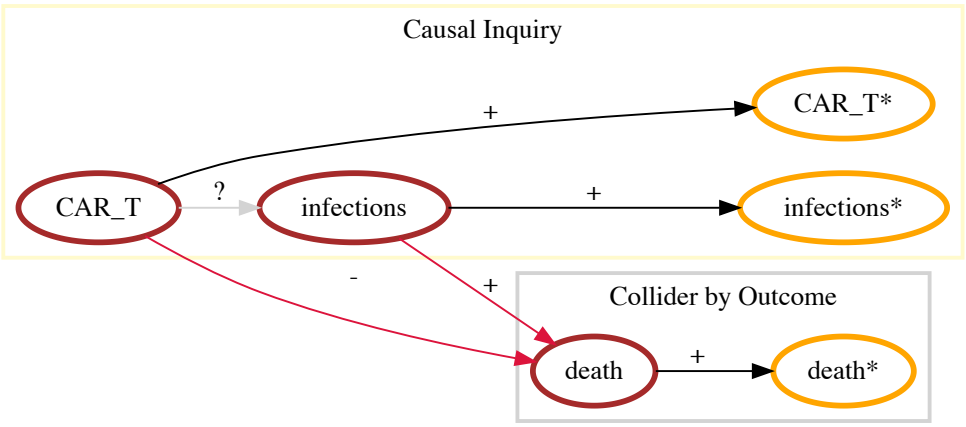



**Noise from conditioning on inaccurate collider measure**

Let's take again into account the colliders presented above. This time let's introduce measurements and assess the differences of conditioning on the presence vs on the lack of the value.

```
create_dag("CAR_T", "infections", label_inquiry = "Causal Inquiry",
           scenario = "inquiry",
           collider_path = list(nodes = c("death"),
                                signs = c("-","+"),
                                label="Collider by Outcome"),
           add_measurements = TRUE)
```

```
file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpRzIHCY/file80da2125981e/
```

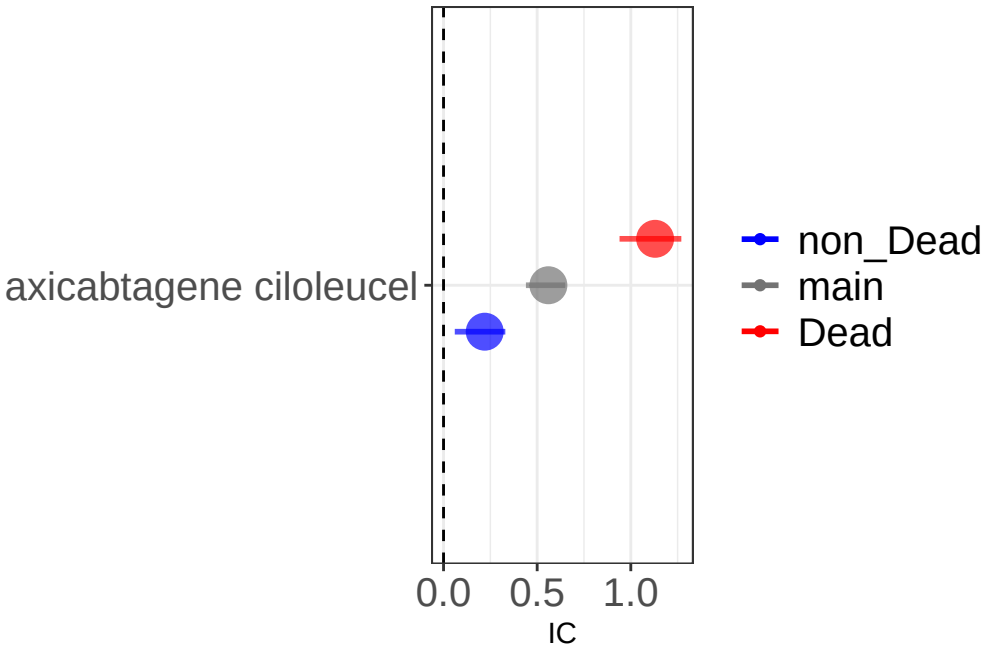

Causal Inquiry
CAR_T
?
infections
+
CAR_T*
+
infections*
-
+
Collider by Outcome
death
+
death*

```
pids_death <- unique(Outc[outc_cod=="DE"]$primaryid)
restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "Dead"=list(pids_death),
                     "non_Dead"=list(setdiff(unique(Demo$primaryid),
                                             pids_death)))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="axicabtagene ciloleucel",
                                    reac_selected = "infections and infestations",
                                    meddra_level = "soc",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}
df_results <- df_results[,nested:=factor(nested,
                                         levels=c("non_Dead","main","Dead"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
    nested               label_IC
     <ord>                  <char>
1:    main 0.56 (0.44-0.65) [775]
2:    Dead 1.13 (0.94-1.27) [308]
3: non_Dead 0.22 (0.06-0.33) [467]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","gray45","red"))
```

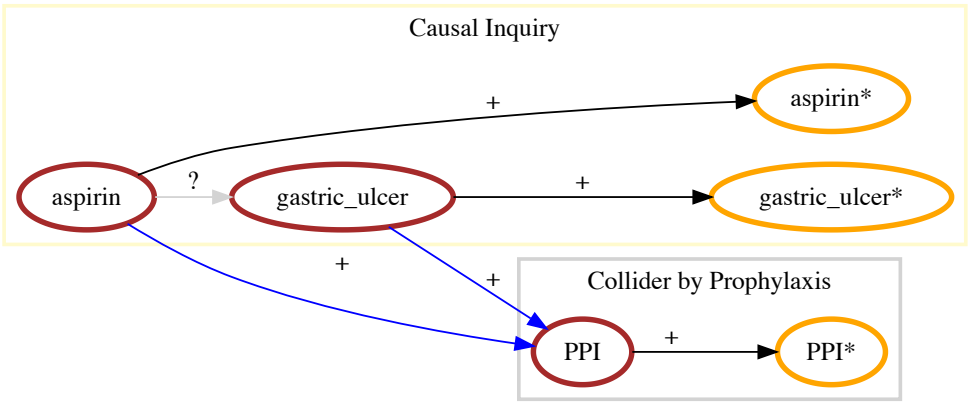
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Collider bias by prophylaxis**

```
create_dag("aspirin", "gastric_ulcer", label_inquiry = "Causal Inquiry",
           scenario = "inquiry",
           collider_path = list(nodes = c("PPI"), signs = c("+","+"),
                                label="Collider by Prophylaxis"),
           add_measurements = TRUE)
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpRzIHCY/file80da72fc8dd0/

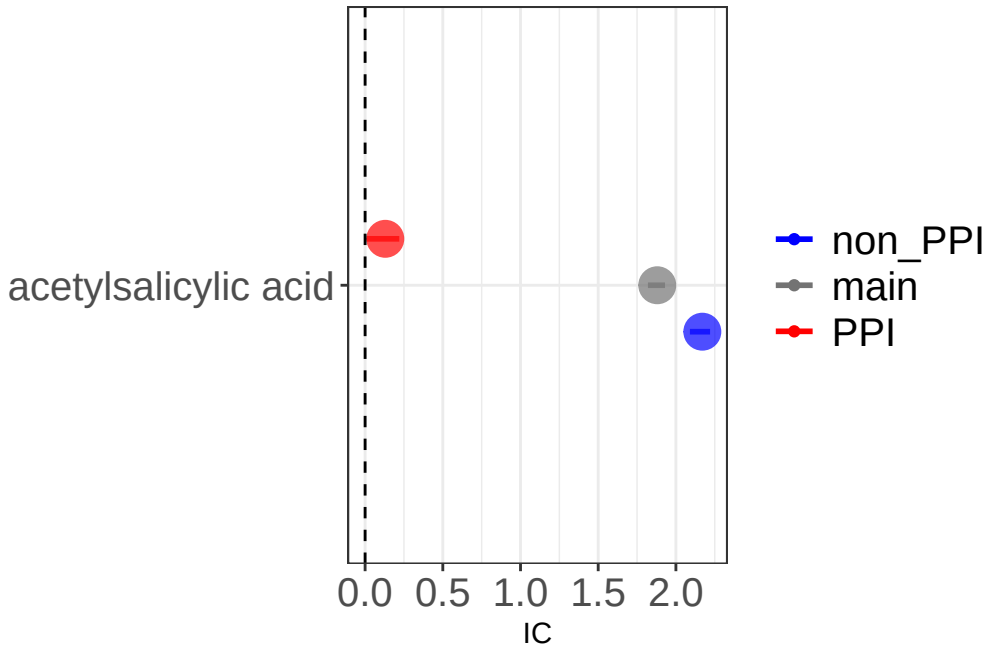

Causal Inquiry
aspirin*
+
aspirin
?
gastric_ulcer
+
gastric_ulcer*
+
+
Collider by Prophylaxis
PPI
+
PPI*

```
pids_PPI <- unique(Drug[
  substance%in%ATC[Class4=="Proton pump inhibitors"]$substance]$primaryid)

restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "PPI"=list(pids_PPI),
                     "non_PPI"=list(setdiff(unique(Demo$primaryid),pids_PPI)))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="acetylsalicylic acid",
                                    reac_selected = "gastric ulcer",
                                    meddra_level = "pt",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}
df_results <- df_results[,nested:=factor(nested,
                                         levels=c("non_PPI","main","PPI"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
    nested                label_IC
     <ord>                  <char>
1:    main 1.88 (1.82-1.93) [2757]
2:     PPI  0.13 (0.01-0.22) [709]
3: non_PPI 2.17 (2.09-2.22) [2048]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","gray45","red"))
```

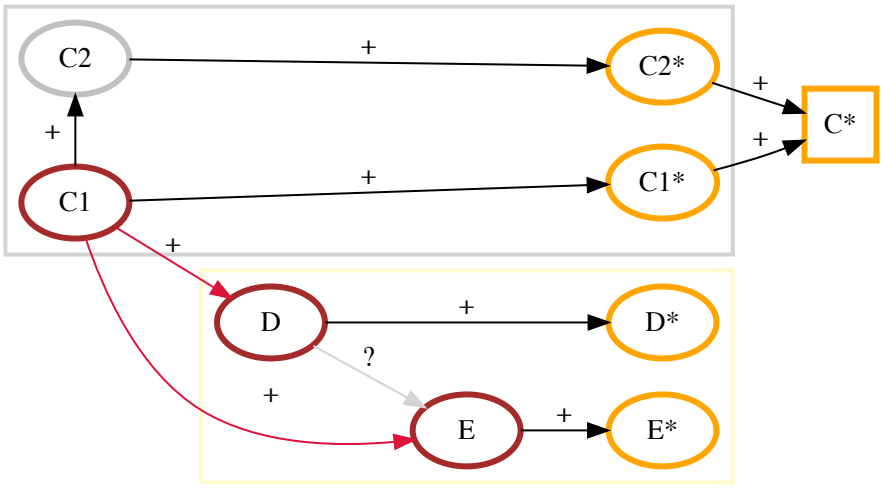
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Surrogate confounder and missing data**

```
create_dag("D", "E", label_inquiry = "", scenario = "inquiry",
           surrogate_confounder = list(surrogate="C2", root="C1"),
           confounder_path = list(nodes = c("C1"), signs = c("+","+"),
                                  label=""),add_measurements = TRUE,
           additional_rows = "
           'C*' [shape = square, style = filled, fillcolor = white, penwidth=3,color=orang
           'C1*' -> 'C*' [label= '+', color = 'black']
           'C2*' -> 'C*' [label= '+', color = 'black']
           "
           #,additional_rows = "cancer -> E [label= '+', color = 'red']")
)
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpRzIHCY/file80dab07478/wi


C2
+
C2*
+
C*
+
+
C1
+
C1*
+
D
+
D*
?
+
E
+
E*

Finally, missing data is a common issue of ICSRs because only the suspected drug and reaction are mandatory fields in the reporting system. For instance, when investigating whether cisplatin causes thrombosis, we find an association (grey estimate) but we recognize that cancer is a likely common cause for both exposure and outcome. We may condition on the confounder either restricting to reports recording cancer among the indications or to reports recording any antineoplastic agent (blue estimates), but this may lead to a small sample and reduced accuracy because of missing information. To address this challenge, we might restrict to reports satisfying any of the two inclusion criteria (gold estimate).

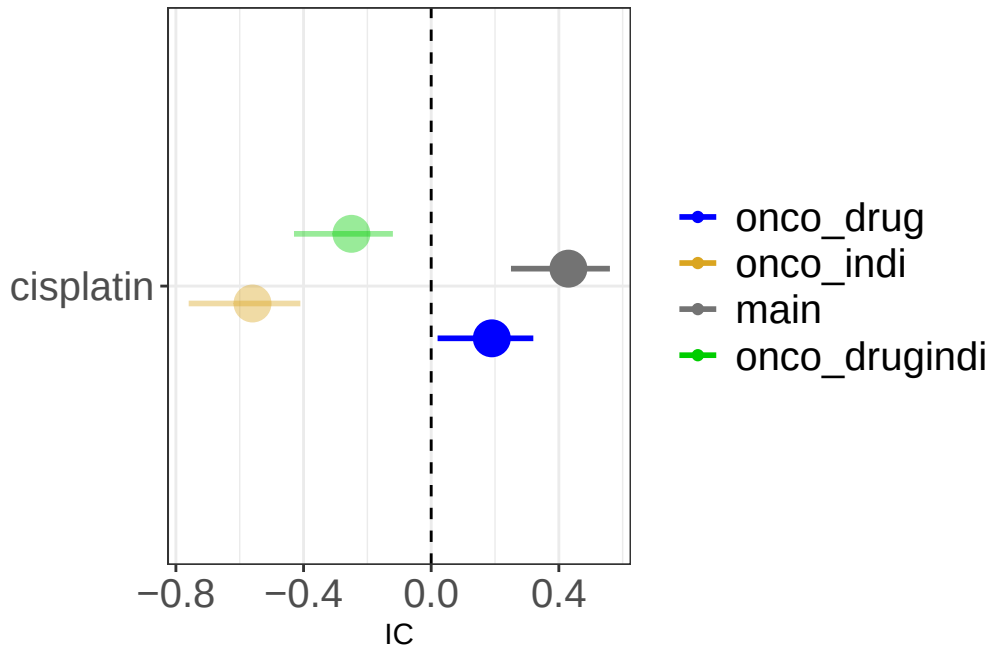
```
pids_onco_indi <- Indi[
  indi_pt%in%MedDRA[
    soc=="neoplasms benign, malignant and unspecified (incl cysts and polyps)"
    ]$pt
  ]$primaryid
pids_onco_drug <- Drug[substance%in%c(ATC[Class2=="Antineoplastic"]$substance,
                                      "navitoclax")]$primaryid
pids_onco_drugindi <- union(pids_onco_drug,pids_onco_indi)
restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "onco_drug"=list(pids_onco_drug),
                     "onco_indi"=list(pids_onco_indi),
                     "onco_drugindi"=list(pids_onco_drugindi))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="cisplatin",
                                    reac_selected = "thrombosis",
                                    meddra_level = "pt",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}

df_results <- df_results[,nested:=factor(nested,
                                         levels=c("onco_drug",
                                                  "onco_indi",
                                                  "main",
                                                  "onco_drugindi"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
       nested                    label_IC
```

```
           <ord>                     <char>
1:          main    0.43 (0.25-0.56) [348]
2:     onco_drug    0.19 (0.02-0.32) [348]
3:     onco_indi -0.56 (-0.76--0.41) [272]
4: onco_drugindi -0.25 (-0.43--0.12) [348]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","goldenrod","gray45","green3"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```

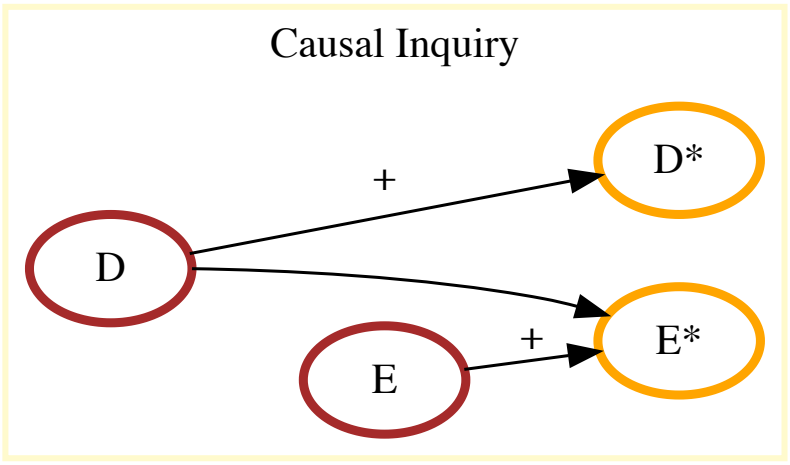



**Ascertainment bias**

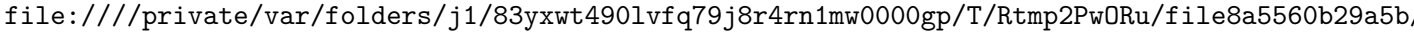
```
create_dag("D", "E", label_inquiry = "Causal Inquiry",
           scenario = "non-causal",add_measurements = TRUE,
```

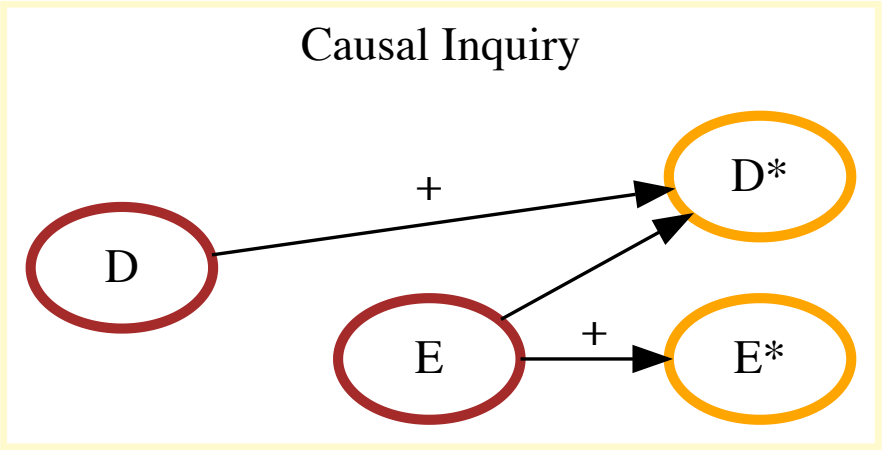
```
            ascertainment_drug = list(""))
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/Rtmp2PwORu/file8a556086dc33,

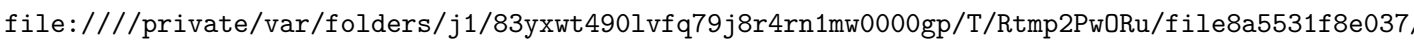

Causal Inquiry
D*
+
D
E*
+
E

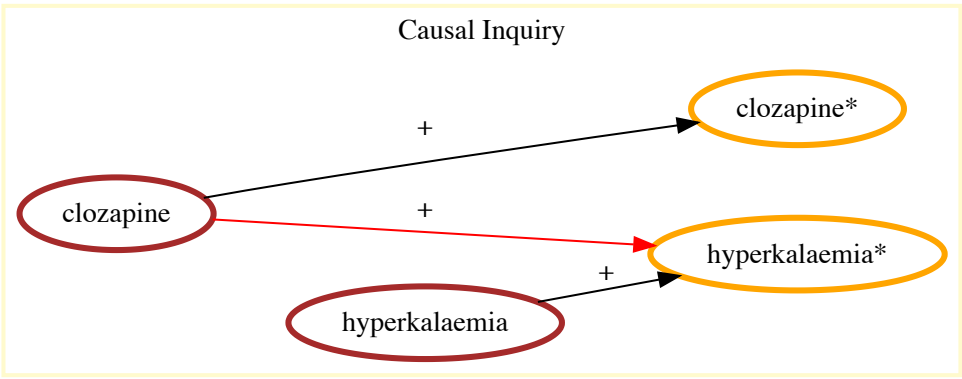
```
create_dag("D", "E", label_inquiry = "Causal Inquiry",
           scenario = "non-causal",add_measurements = TRUE,
           ascertainment_event = list(""))
```




Causal Inquiry
D
+
D*
E
+
E*

```r
create_dag("clozapine", "hyperkalaemia", label_inquiry = "Causal Inquiry",
           scenario = "non-causal",add_measurements = TRUE,
           ascertainment_drug = list("+"))
```




Causal Inquiry
clozapine*
+
clozapine
+
hyperkalaemia*
+
hyperkalaemia

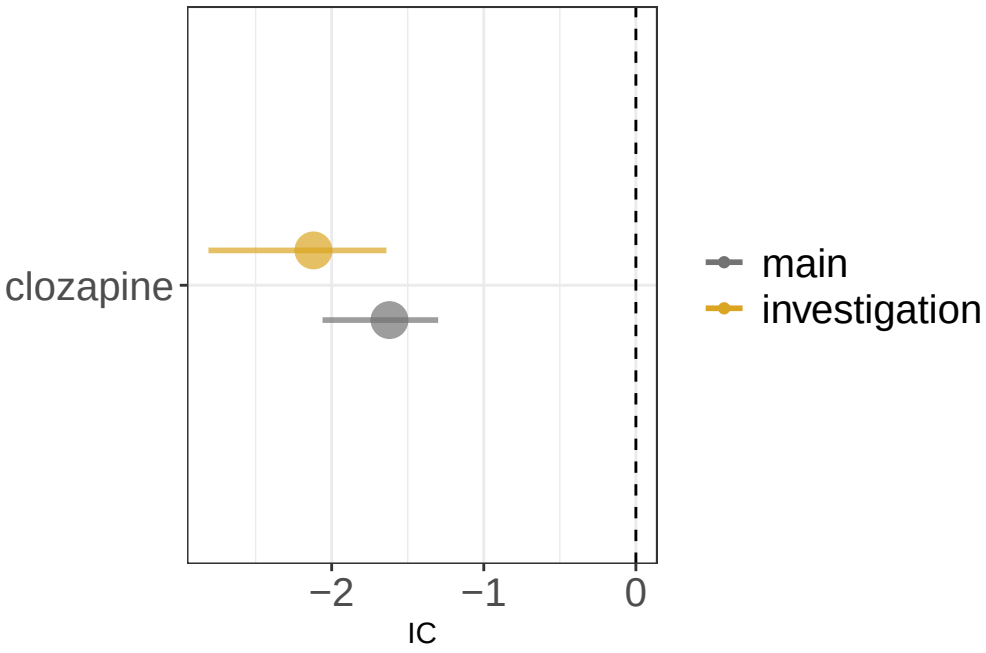
Ascertainment bias occurs when the exposure to a drug determines an increased attention and reporting rate for a specific event. Taking clozapine, because of the risk of agranulocytosis, results in frequent blood tests and higher probability of detecting and reporting hyperkalaemia. For this reason, the analysis on the entire population (grey estimate), gives a higher (but still not significant) IC than when we restrict to reports recording any investigation (gold estimate).

```
pids_investigations <- Reac[pt%in%MedDRA[soc=="investigations"]$pt]$primaryid
restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "investigation"=list(pids_investigations))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="clozapine",
                                    reac_selected = "hyperkalaemia",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}
df_results <- df_results[,nested:=factor(nested,
                                         levels=c("main","investigation"),
                                         ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
          nested                 label_IC
           <ord>                   <char>
1:          main  -1.62 (-2.06--1.3) [57]
2: investigation -2.12 (-2.81--1.64) [24]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("gray45","goldenrod"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```



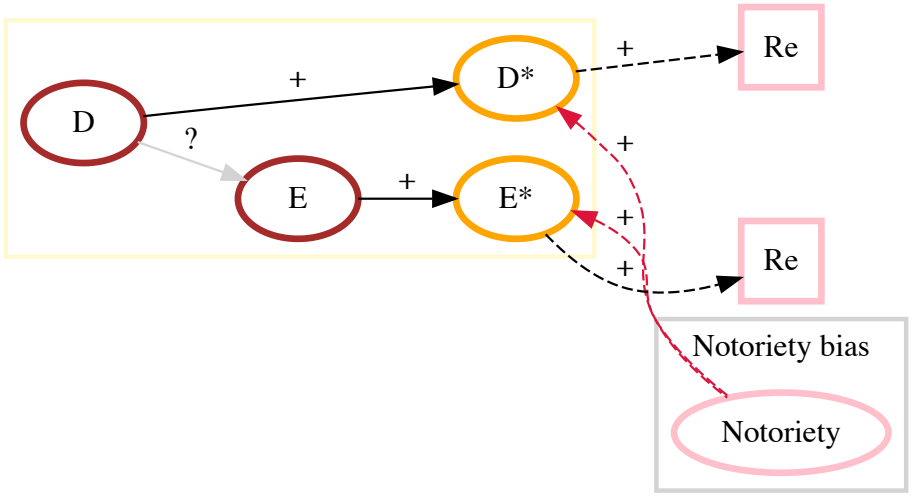

**Reporting bias**

Up to this point, we did not consider that many drug-event co-occurrences are never reported (underreporting) due, for instance, to lack of awareness or motivation in the patient and/or clinician, or to the complexity and time demands in submitting a report. Perhaps even more worrisome, drug exposure without adverse events is not reported, nor are events occurring without drug administration. To account for such scenarios, DAGs have to incorporate biases that differentially affect baseline reported occurrences and cases of interest. In particular, disproportionality can be directly assessed if we know how often drug D is used – with and without the occurrence of the event E - and how often event E occurs – with and without the administration of D. However, we do not have access to such data. Rather, disproportionality analyses calculate measures of drug use (D) relying on the number of reports with the drug and event of interest (DE) and the number of reports with the drug of interest and other events (DnE, rather than directly counting patients taking the drug but not experiencing the event), compared to reports (of other drug-event occurrences) not mentioning the drug (nD). Analogously, we do not have access to the occurrence of event E, but we rely on the number of reports with the drug and event of interest (DE) and reports of the same event co-occurring with other drugs but D (nDE), compared to reports not mentioning the event (nE). These measurements are used to calculate the IC as a measure of association between D and E* but, if they present measurement errors (reporting biases, R), they can introduce inaccuracies and imprecision into the inference. The utility of the IC as a surrogate measure of association between the drug and the event is based on the key assumption that any reporting

bias will equally affect the reporting of all drugs and events (independent reporting bias). The problem arises when there is dependent reporting bias, differentially affecting the four different measures (DE, nDE, DnE, nDnE). We can distinguish positive and negative reporting biases based on whether they amplify (i.e., increase DE* and/or nDnE) or dampen the IC (i.e., increase DnE* and/or nDE). We can tackle reporting biases by restricting our analyses to the reports not affected by the biased measurement error, or in principle – although not done yet in practice – explicitly include estimates of measurement error in the model as mitigation factors. Reporting rates can vary depending on report type (e.g., spontaneous, solicited, literature case report), reporter's expertise (misclassification bias, e.g., use of ambiguous terms to refer to speech impairment) and diverse reporting habits among different reporter types (e.g., medical doctors, patients, lawyers), regulatory systems, and settings of use. The variability in the way pharmacovigilance officers code verbatim events recorded by patients to MedDRA might further affect the downstream statistical analyses. When anticipating such differential effects, it may be appropriate to consider restrictions, even if predicting the exact direction of the distortion is often challenging: e.g., the severity of an event might increase its reporting rate, and therefore both DE (which inflates the IC) and nDE* (which deflates the IC). For the better investigated reporting biases, the exact direction of the distortion can be predicted.

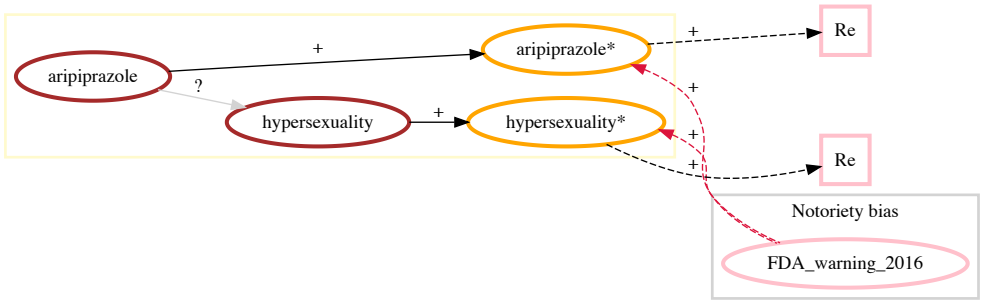
### Notoriety bias

```
create_dag("D", "E", scenario = "inquiry", notoriety_bias = "Notoriety",
           add_measurements = TRUE,
           label_inquiry ="")
```


D
+
D*
+
Re
?
+
+
E
+
E*
+
+
Re
Notoriety bias
Notoriety

```r
create_dag("aripiprazole", "hypersexuality", scenario = "inquiry",
           notoriety_bias = "FDA_warning_2016", add_measurements = TRUE,
           label_inquiry ="")
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/Rtmp2PwORu/file8a55375febd2/


aripiprazole
+
aripiprazole*
+
Re
?
hypersexuality
+
hypersexuality*
+
+
Re
Notoriety bias
FDA_warning_2016

Let us take the example of the notoriety bias, which may affect both healthcare professionals and patients. When regulatory warnings are issued, healthcare professionals may become more aware of and inclined to report specific drug-event combinations. Similarly, a few highly publicized cases can attract media attention, leading the public to higher awareness and inclination to report. This inflates the DE* reporting rate compared to reporting just the drug or the event alone, and therefore inflates the IC (grey estimate). To address this bias, we can, for instance, restrict the DPA to the reports collected before the regulatory warning, if enough cases are available, thereby conditioning on the source of the bias (gold estimate). As a proof of concept, we can restrict to the cases following the regulatory warning and obtain an inflated statistical dependence (blue estimate).

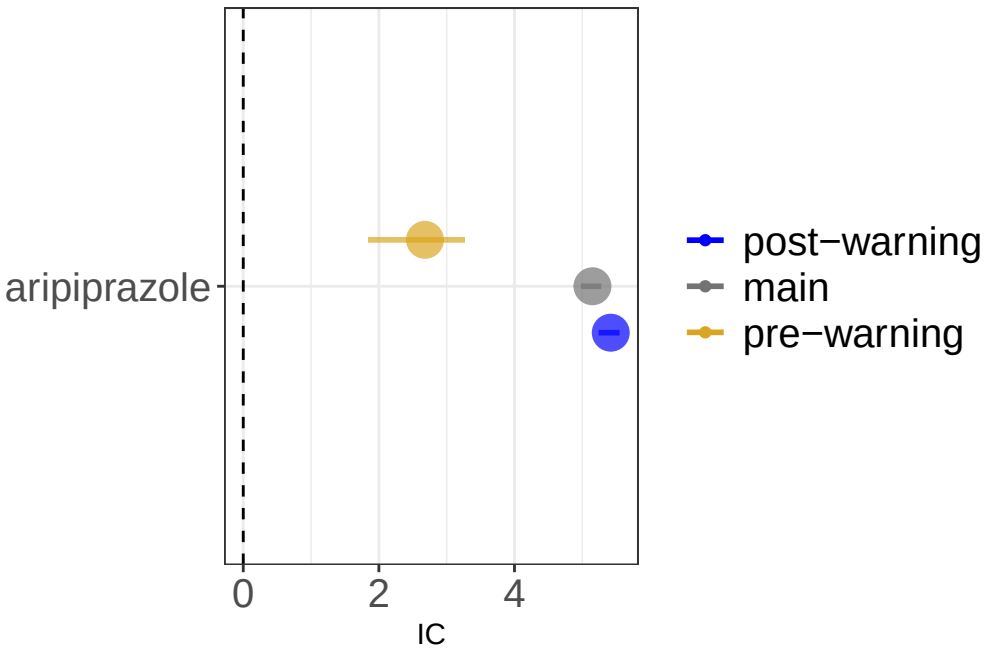
```
pids_warning <- Demo[init_fda_dt<20160503]$primaryid
restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "pre-warning"=list(pids_warning),
                     "post-warning"=setdiff(Demo$primaryid,pids_warning))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="aripiprazole",
                                    reac_selected = "hypersexuality",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}
df_results <- df_results[,nested:=factor(nested,
                                         levels=c("post-warning","main",
                                                  "pre-warning"),ordered=TRUE)]
df_results[,.(nested,label_IC)]
```

```
        nested               label_IC
         <ord>                  <char>
1:         main 5.15 (4.98-5.28) [354]
2:  pre-warning  2.68 (1.84-3.27) [16]
3: post-warning 5.42 (5.24-5.55) [338]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","gray45","goldenrod"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
```

```
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Lawyers reports**

The notoriety bias and the similarly acting precautionary bias (e.g., reports from patient support programs, with increased reporting rate) could be characterized as differential underreporting: more DE are reported – i.e., less underreported - than background cases. Conversely, reports may even occur in the lack of an actual event – as in the case of superficial diagnostic verification (e.g., reporting drug-induced liver injury without verifying all diagnostic criteria) and of malicious reporting (as speculated for court litigations) – or may proliferate from a single report into many duplicates. This differential reporting of DE* can be dealt with conditioning on the source of the bias: e.g., removing reports by lawyers and duplicates. For instance, when investigating the causal dependence between aripiprazole and gambling, we find a positive association (grey estimate). However, due to potential malicious reporting by lawyers within court litigation, this result may not be reliable. To inspect this potential bias, we restrict to reports not produced by lawyers and we find a smaller but still significant association (gold estimate). As proof of concept, we can restrict to reports produced by lawyers and there we find a bigger association (blue estimates). It's important to note that all reports

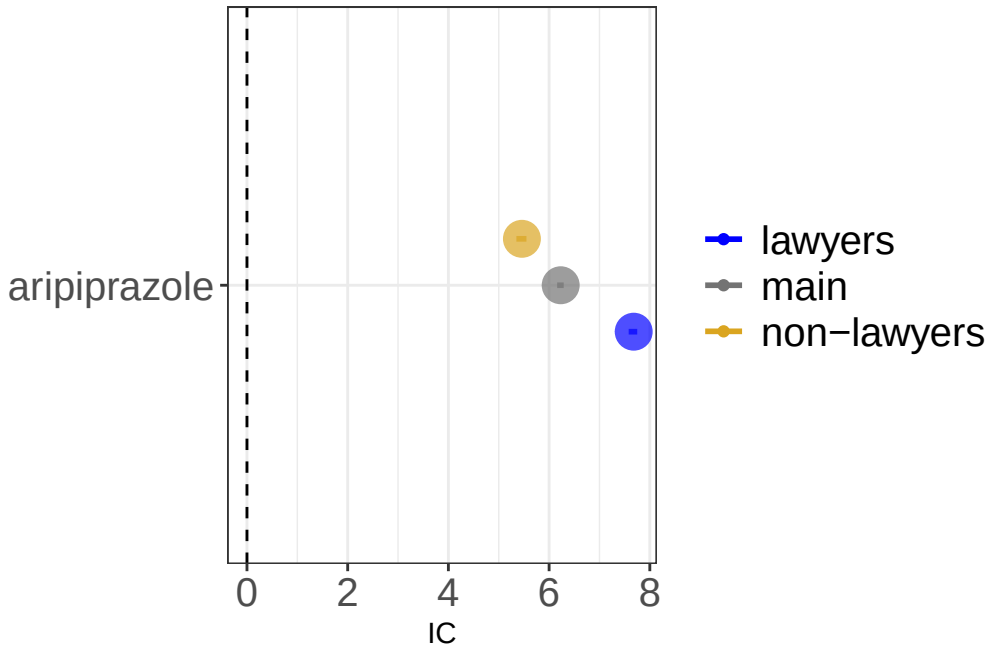
are valuable, as illustrated by the FDA warning for aripiprazole-induced impulse control disorders in 2016, which relied on reports by patients and lawyers (REF DAA partial). Hence, all reports should be encouraged, and considerations on the potential bias should be limited to the analysis and not to data collection. Based on the DAG, sensitivity quantitative and qualitative analyses can be performed by restricting to specific subpopulations to account for expected biases. When conducting such sensitivity analyses, both the raw disproportionality result and the result of the sensitivity analysis should be reported and discussed together.

```
pids_lawyers <- Demo[occp_cod=="LW"]$primaryid
restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "lawyers"=list(pids_lawyers),
                     "non-lawyers"=setdiff(Demo$primaryid,pids_lawyers))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="aripiprazole",
                                    reac_selected = "gambling disorder",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}

df_results[,.(nested,label_IC)]
```

```
       nested                label_IC
       <char>                  <char>
1:       main 6.23 (6.16-6.29) [1937]
2:    lawyers 7.68 (7.58-7.75) [1118]
3: non-lawyers  5.46 (5.35-5.55) [819]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("blue","gray45","goldenrod"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```



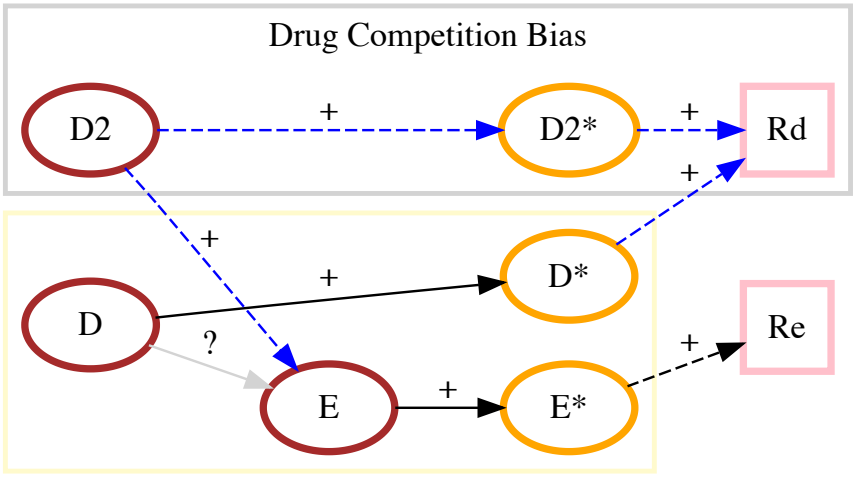

### Masking

Masking/cloaking is an additional example of reporting bias and can take the form of a drug-competition bias (increased reporting rate for the same event with other drugs nDE*) or an event competition bias (increased reporting rate for the same drug with other events DnE*), which may also derive from notoriety biases sharing with DE exclusively the drug or the event. This bias likely dampens the IC and can be tackled by removing from the database reports of the competing drug-event pair. The fact that these distortions induced by masking are not so evident compared to the effect of other biases presented in this manuscript is in agreement with previous studies that question its importance.

### Drug competition bias

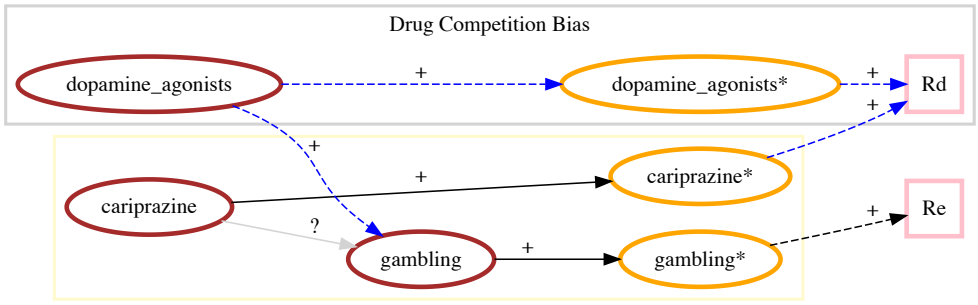
```
create_dag("D", "E", scenario = "inquiry", drug_competition_bias = "D2",
           add_measurements = TRUE,
           label_inquiry ="")
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/Rtmp2PwORu/file8a5574e7e4bb/


Drug Competition Bias
D2
+
D2*
+
Rd
+
+
D*
D
+
Re
?
+
E
+
E*

```r
create_dag("cariprazine", "gambling", scenario = "inquiry",
           drug_competition_bias = "dopamine_agonists", add_measurements = TRUE,
           label_inquiry ="")
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/Rtmp2PwORu/file8a5554576dd9/


Drug Competition Bias
dopamine_agonists
+
dopamine_agonists*
+
Rd
+
+
cariprazine
+
cariprazine*
?
gambling
+
gambling*
+
Re

Investigating the causal dependence between cariprazine and gambling, we find a statistical dependence (grey estimate) which nonetheless increases even more when we remove reports of gambling with the competing dopamine agonists (gold estimate).

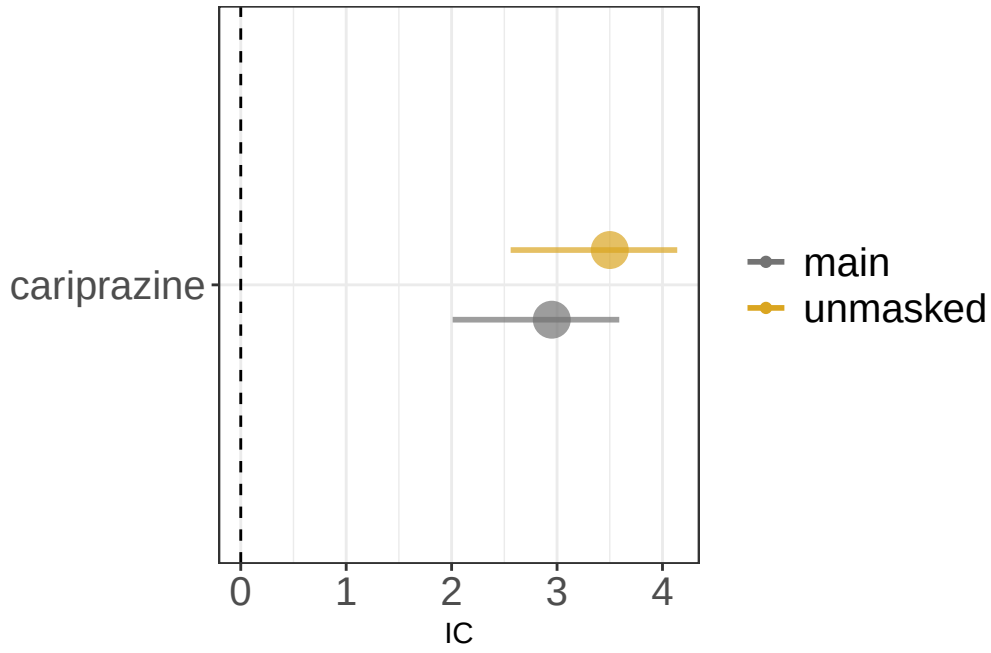
```
pids_unmasked <- setdiff(Demo$primaryid,
                         Drug[substance%in%ATC[
                           Class4=="Dopamine agonists"]$substance
                              ]$primaryid)
pids_unmasked <- setdiff(Demo$primaryid,
                         intersect(Drug[
                           substance%in%ATC[
                             Class4=="Dopamine agonists"]$substance
                           ]$primaryid,
                                   Reac[pt=="gambling disorder"]$primaryid))
restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "unmasked"=list(pids_unmasked))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="cariprazine",
                                    reac_selected = "gambling disorder",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}

df_results[,.(nested,label_IC)]
```

```
     nested              label_IC
     <char>                 <char>
1:     main 2.95 (2.01-3.59) [13]
2: unmasked  3.5 (2.56-4.14) [13]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("gray45","goldenrod"))
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
```

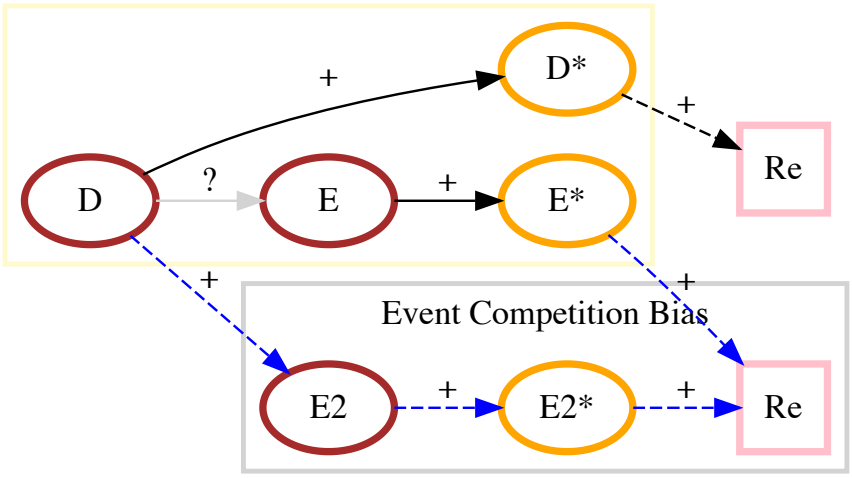
```
* alpha = 0.7
i Did you misspell an argument name?
```



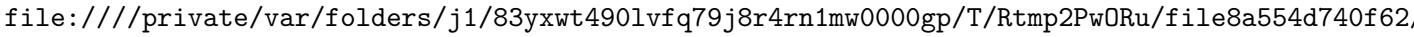

**Event competition bias**

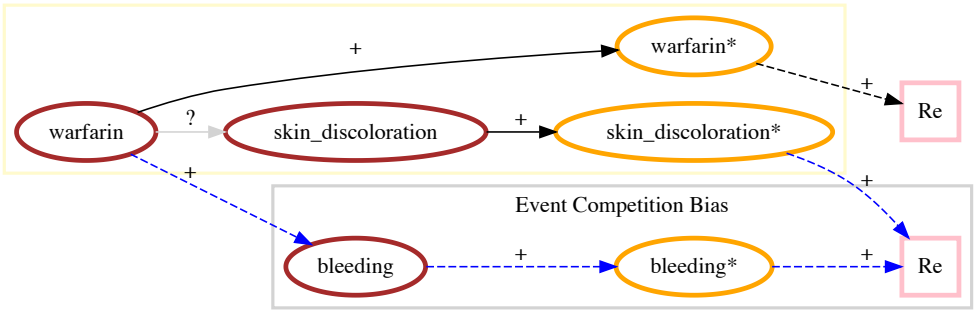
```
create_dag("D", "E", scenario = "inquiry", event_competition_bias = "E2",
           add_measurements = TRUE,
           label_inquiry ="")
```

```
file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/Rtmp2PwORu/file8a5523fa471c/
```


D*
+
+
Re
D
?
E
+
E*
+
+
Event Competition Bias
E2
+
E2*
+
Re

```
create_dag("warfarin", "skin_discoloration", scenario = "inquiry",
           event_competition_bias = "bleeding", add_measurements = TRUE,
           label_inquiry ="")
```




warfarin*
+
+
Re
warfarin
?
skin_discoloration
+
skin_discoloration*
+
+
Event Competition Bias
bleeding
+
bleeding*
+
Re

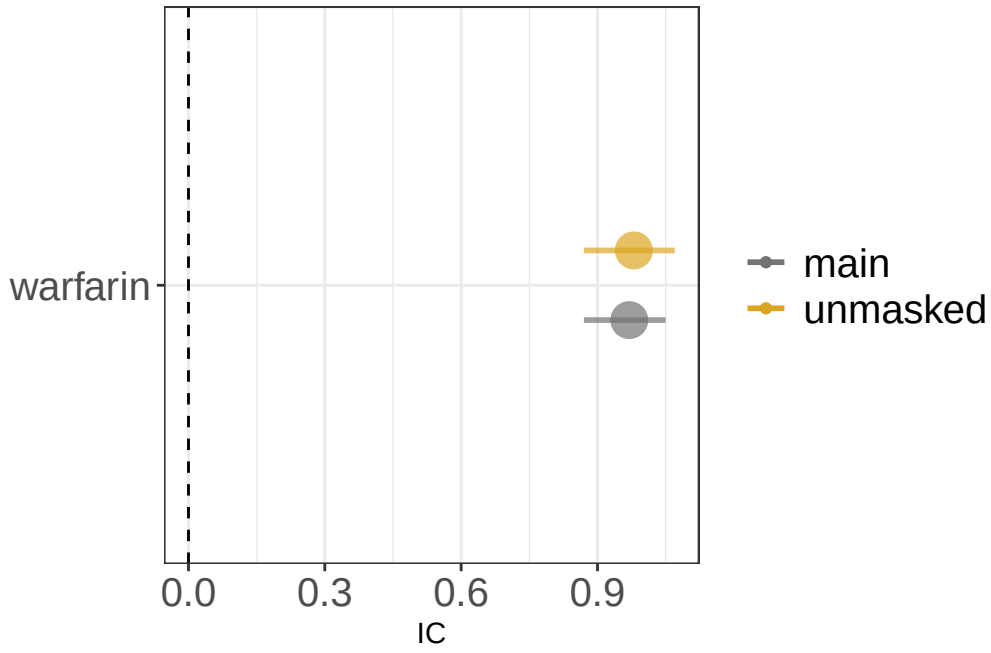
Similarly, when investigating the causal dependence between warfarin and skin discoloration we observe a statistical dependence (grey estimate) which even increase when we remove reports of warfarin with the competing event bleeding (gold estimate).

```
pids_unmasked <- setdiff(Demo$primaryid,
                         intersect(
                           Reac[pt%in%MedDRA[grepl("bleeding|haemorrhage",
                                                   pt)]$pt]$primaryid,
                                   Drug[substance=="warfarin"]$primaryid))
restrictions <- list("main"=list(unique(Demo$primaryid)),
                     "unmasked"=list(pids_unmasked))
df_results <- data.table()
for (n in 1:length(restrictions)){
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(drug_selected="warfarin",
                                    reac_selected = "skin discolouration",
                                    restriction = t_pids)[,nested:=t_name]
  df_results <- rbindlist(list(df_results,df),fill = TRUE)
}

df_results[,.(nested,label_IC)]
```

```
     nested               label_IC
     <char>                  <char>
1:     main 0.97 (0.87-1.05) [951]
2: unmasked 0.98 (0.87-1.07) [822]
```

```
render_forest(df_results,nested = "nested",
              nested_colors=c("gray45","goldenrod"))
```

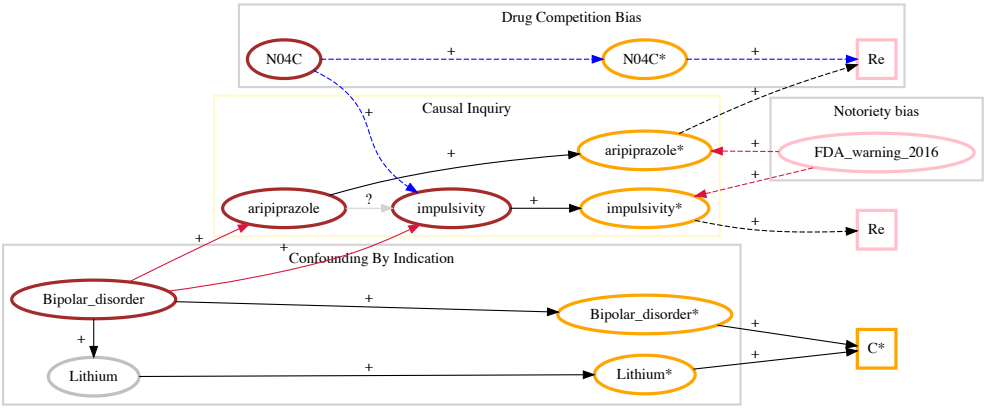
```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```




**Complex example: wrapping up**

```
create_dag("aripiprazole", "impulsivity",
           add_measurements = TRUE,
           drug_competition_bias = "N04C",
           confounder_path = list(nodes = c("Bipolar_disorder"),
                                  signs = c("+", "+"),
                                  label = "Confounding By Indication"),
           notoriety_bias = "FDA_warning_2016",
           surrogate_confounder = list(root="Bipolar_disorder",
                                       surrogate="Lithium"),
           additional_rows = "
           'C*' [shape = square, style = filled, fillcolor = white, penwidth=3,color=orang
           'Lithium*' -> 'C*' [label= '+', color = 'black']
           'Bipolar_disorder*' -> 'C*' [label= '+', color = 'black']
           ")
```

file:////private/var/folders/j1/83yxwt490lvfq79j8r4rn1mw0000gp/T/RtmpfecX71/filea946323eef02


Drug Competition Bias
N04C
N04C*
Re
Causal Inquiry
Notoriety bias
aripiprazole*
FDA_warning_2016
aripiprazole
impulsivity
impulsivity*
Re
Confounding By Indication
Bipolar_disorder
Bipolar_disorder*
C*
Lithium
Lithium*

```r
drug_selected <- "aripiprazole"

ICDs <- list(
  "gambling disorder" = list("gambling disorder", "gambling"),
  "hypersexuality" = list(
    "compulsive sexual behaviour",
    "hypersexuality",
    "excessive masturbation",
    "excessive sexual fantasies",
    "libido increased",
    "sexual activity increased",
    "kluver-bucy syndrome"
  ),
  "paraphilia" = list(
    #"erotophonophilia", "frotteurism",# absent in the current database
    "exhibitionism","fetishism",
    "masochism", "paraphilia", "paedophilia",
    "sadism", "transvestism", "voyeurism",
    "sexually inappropriate behaviour"
  ),
  "compulsive shopping" = list("compulsive shopping"),
  "hyperphagia" = list(
    "binge eating", "food craving", "hyperphagia",
    "increased appetite"
  ),
  "gaming disorder" = list("gaming disorder"),
  "pyromania" = list("pyromania"),
  "kleptomania" = list("kleptomania", "shoplifting"),
  "compulsive hoarding" = list("compulsive hoarding"),
  "excessive exercise" = list("excessive exercise"),
  "overwork" = list("overwork"),
  "poriomania" = list("poriomania"),
  "body-focused disorder" = list(
    "compulsive cheek biting",
    "compulsive lip biting",
    "dermatillomania",
    "dermatophagia", "nail picking",
    "compulsive handwashing",
    "trichotemnomania",
    "trichotillomania", "onychophagia",
    "thumb sucking"
```

```
  ),
  "stereotypy" = list("automatism", "stereotypy"),
  "impulsivity" = list(
    "impulse-control disorder",
    "impulsive behaviour",
    "disinhibition", "behavioural addiction",
    "abnormal behaviour"
  )
)
reac_selected <- list("ICD" = as.list(unlist(purrr::flatten(ICDs))))


raw_pids <- unique(Demo$primaryid)
bipo_pids <- union(Indi[
  indi_pt %in% MedDRA[hlgt=="manic and bipolar mood disorders and disturbances"
                      ]$pt]$primaryid,
                   Drug[substance=="lithium"]$primaryid)
preW_pids <- Demo[ifelse(is.na(init_fda_dt), fda_dt < 20160305,
                         init_fda_dt < 20160305)]$primaryid
no_comp_pids <- setdiff(Demo$primaryid,
                        Drug[substance%in%ATC[Lvl3=="N04C"]$substance
                             ]$primaryid)
complete <- intersect(preW_pids,intersect(bipo_pids, no_comp_pids))
restrictions <- list(
  "raw" = list(raw_pids),
  "bipolar_disorder" = list(bipo_pids),
  "pre_warning" = list(preW_pids),
  "no_competition" = list(no_comp_pids),
  "complete" = list(complete)
)

disproportionality_df <- data.table()
for (n in 1:length(restrictions)) {
  t <- restrictions[n]
  t_name <- names(t)
  t_pids <- unlist(t)
  df <- disproportionality_analysis(
    drug_selected = drug_selected,
    reac_selected = reac_selected,
    restriction = t_pids
  )[, nested := t_name]
```

```
  disproportionality_df <- rbindlist(list(disproportionality_df, df),
                                     fill = TRUE)
}

render_forest(disproportionality_df, row = "event", facet_v = "substance",
              nested = "nested")
```

```
Warning in fortify(data, ...): Arguments in `...` must be used.
x Problematic arguments:
* position = position_dodge(dodge)
* show.legend = show_legend
* alpha = 0.7
i Did you misspell an argument name?
```