vigicaen: : CHEAT SHEET

Optionally, set local paths to files as R objects

```
path_main  <- '/your/path/to/vigibase/main/'
path_sub   <- '/your/path/to/vigibase/sub/'
path_who   <- '/your/path/to/vigibase/who/'
path_meddra <- '/your/path/to/meddra/'
```

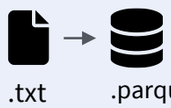
Build Tables

Do it **once** per database version

To get started, you will need Vigibase ECL and MedDRA licenses.

! Remember to unzip source files.

VigiBase and WHO Drug tables



tb_vigibase(path_main, path_sub)
tb_who(path_who)

MedDRA tables

tb_meddra(path_meddra)

Load and assign Tables

Do it **every time** you work on VigiBase®

```
demo  <- dt_parquet(path_main, 'demo')
drug  <- dt_parquet(path_main, 'drug')
(same for adr, link, out, srce, suspdup)
mp    <- dt_parquet(path_who, 'mp')
(same for thg)
meddra <- dt_parquet(path_meddra,
                     'meddra_hierarchy')
smq_list <- dt_parquet(path_meddra, 'smq_list')
smq_content <- dt_parquet(path_meddra,
                          'smq_content')
```

Hint: Leave tables OUT of memory with arg in_memory = FALSE

Set up dictionary

Create named **lists** for drugs and adverse drug reactions (adrs).

drug list

```
d_sel <-
 list(analgesics = c('paracetamol', 'tramadol'))
a_sel <-
 list(hepatitis = c('Hepatitis'))
```

adr list — Both are **named** lists

With Standardized MedDRA queries

```
smq_sel <-
 list(ihd = c('Ischaemic heart disease 'SMQ)'))
```

With Anatomical and Therapeutical Classes (ATC)

```
atc_sel <- list(cardiac_drugs = c('C01'))
```

Syntax for multiple drugs or adrs is the same

```
a_sel_many <-
 list(analgesics = c('paracetamol', 'tramadol'),
      nivolumab = 'nivolumab',
      ici = c('nivolumab', 'ipilimumab'))
```

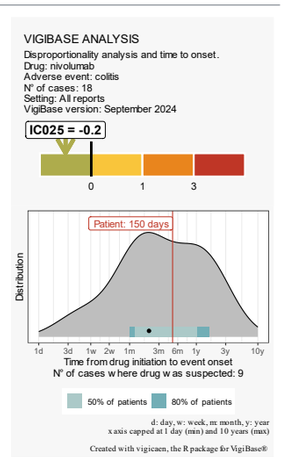
PV Routine

Plot graph

You must use lists with a single item for drugs and adrs – like d_drecno and a_llt here.

```
vigi_routine(
  demo, drug, adr, link,
  d_code = d_drecno,
  a_code = a_llt,
  case_tto = 50,
  vigibase_version = 'Sep 2024')
```

Export with export_to = 'graph.svg'



Collect IDs

IDs

Ids are numbers that identify drugs and adrs

ID	Description
DrecNo	Drug Record Number, a single drug
LLT	Low level term ID of an adr

(also MedicinalProd_Id, see ?get_atc_code)

get_* functions collect IDs

get_drecno()

d_list — drug list → list(analgesics = c(586321, 186542)) — IDs

DrecNo

```
d_drecno      <- get_drecno(d_sel , mp)
d_drecno_atc <- get_atc_code(atc_list, mp, thg)
```

LLT

```
a_llt <- get_llt_soc(a_list, term_level = "pt",
                     meddra)
a_llt_smq <- get_llt_smq(smq_sel,
    smq_list      = smq_list,
    smq_content = smq_content)
```

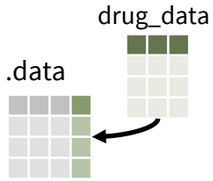
PV Advanced

Add drug & adr columns

Once you've collected drugs and adrs **IDs**, Add columns to the desired dataset

Table	Purpose
demo	Study drugs and adrs at CASE level
drug	Screen co-reported drugs
adr	Study adrs outcome
link	Study at DRUG-ADR PAIR level

Add drugs

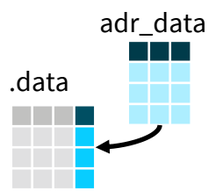



```
demo <-
 add_drug(demo, d_drecno, drug_data = drug)
```

Arg repbasis to select suspected ('s'), concomitant ('c') and/or interacting ('i') drugs

```
demo <-
 add_drug(demo, d_drecno_atc, drug_data = drug)
```

Add adrs




```
demo <-
 add_adr(demo, a_llt, adr_data = adr)
```

Replace demo by other tables… link <- add_drug(link…)

Check data management

Wrapper around sum(.x == 1)

check_dm(demo, c('analgesics', 'hepatitis'))

vigicaen: : CHEAT SHEET

PV Advanced

Disproportionality

Univariate Reporting Odds-Ratio and Information Component (IC)

```
compute_dispro(
  demo, 'hepatitis', 'analgesics')
```

Multivariate Reporting Odds-Ratio

```
mod <- glm(hepatitis ~ analgesics + AgeGroup +
Gender, family = "binomial", data = demo)
summary(mod)$coefficients |>
compute_or_mod(
  estimate = Estimate, std_er = Std..Error)
```

Hint: be sure to clean AgeGroup and Gender variables before using them

Interactions

```
compute_interaction(
  demo, 'hepatitis', 'analgesics', 'cardiac_drugs')
```

Subset Tables

Create subsets of VigiBase, based on drug or adr IDs, or AgeGroup

Subset tables

path_main

```
tb_subset(wd_in,
  wd_out,
  subset_var = "age",
  sv_selection = 7))
```

e.g. path_main/subset

Subset can be performed on drug or adr
Arg subset_var 'drecno' or 'meddra_id'

IDs

PV Advanced

Description

desc_* functions will synthetize data

Use the right table to find your data

Descriptive variables	Type of analysis	Table
Drugs, adrs, age, sex	case level	demo
Time to onset	drug-adr pair level	link
Dechallenge	drug-adr pair level	link
Rechallenge	drug-adr pair level	link
Adr outcome	adr level	adr
Drugs/adrs screening	case level	drug/adr

demo table

Categorical variables

```
desc_facvar(demo, vf = c('hepatitis', 'Gender'))
```

Arg format let you change output according to your preference. **n_** is number of cases at this level, **N_** is total number of cases with available data, **pc_** is percentage.
e.g. format = "**n_** out of **N_**, **pc_**%" → "**7** out of **50**, **14**%"

Continuous variables

```
desc_cont(demo, vc = 'FirstDateDatabase')
```

Arg format: **median**, **q1** and **q3** for interquartile range, **min** and **max** for range.

link table

```
link <-
  add_drug(link, d_drecno, drug_data = drug) |>
  add_adr(a_llt, adr_data = adr)
```

Remember to add_drug(), add_adr() to link

Time to onset


```
desc_tto(link, 'analgesics',
  'hepatitis')
```

Extract and draw time to onset

```
ttos <- extract_tto(link,
  'analgesics',
  'hepatitis')
hist(ttos$tto_max)
```

Dechallenge


```
desc_dch(link,
  'analgesics',
  'hepatitis')
```

Rechallenge


```
desc_rch(link, 'analgesics', 'hepatitis')
```

adr table

```
adr <-
  add_drug(adr, d_drecno, drug_data = drug) |>
  add_adr(a_llt, adr_data = adr)
```

Outcome **adr** → **Outcome?**

```
desc_outcome(adr, 'analgesics', 'hepatitis')
```

Screen most reported drugs & adrs

Identify top reported drugs/adrs (counting *cases*).

```
screen_drug(drug, mp, 0.10)
screen_adr(adr, meddra, 'hlgt', 0.10)
```

Built-in data

Datasets

You can test the package with built-in datasets

Add "_" to any of demo, adr, drug, link, srce, out, link, mp, thg, meddra, smq_list, smq_content, to use the example table

```
demo <- demo_
drug  <- drug_
…
```

name + '_'