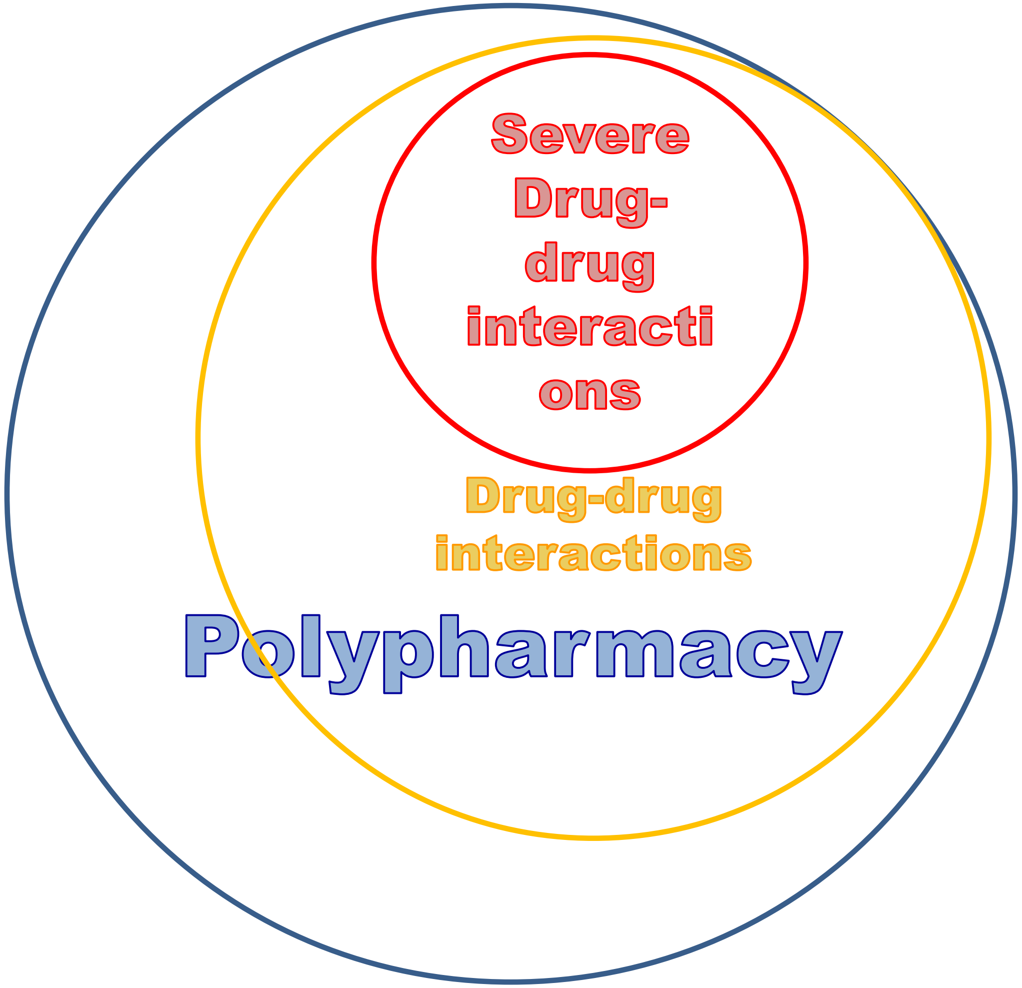
Calculating Drug-drug Interactions in Administrative Databases

Using R coding

Polypharmacy and DDIs

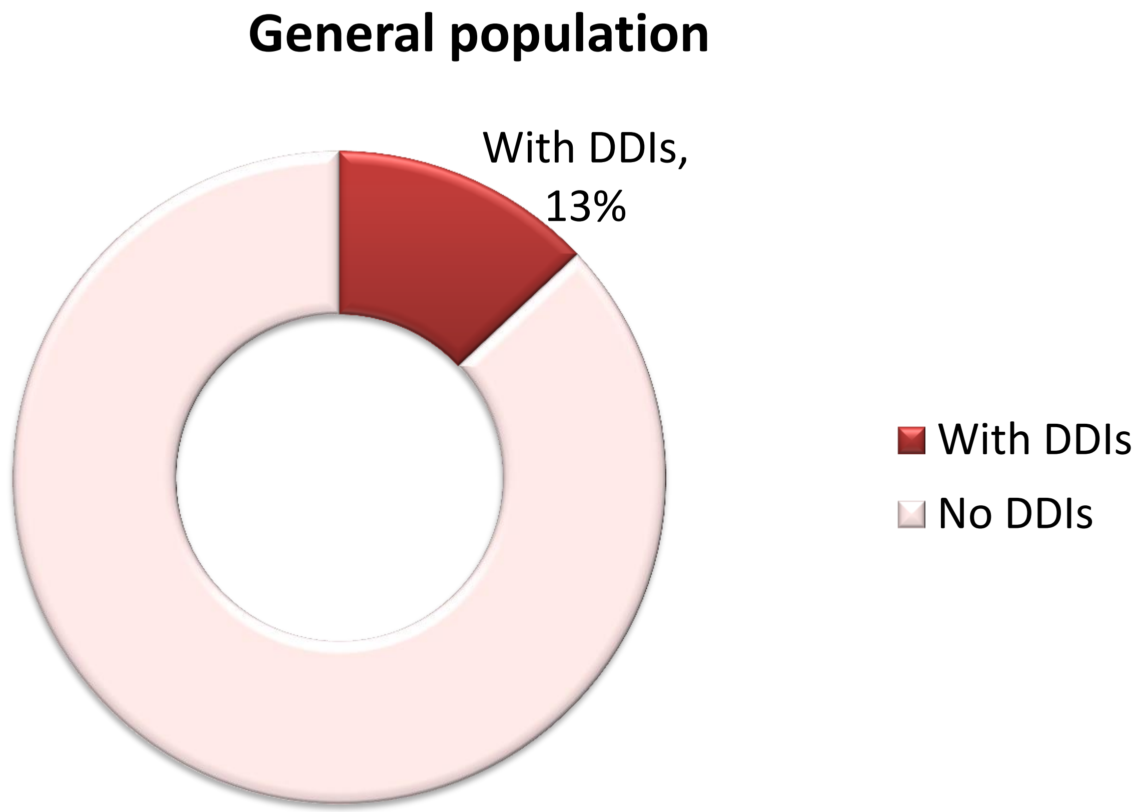


Severe Drug-drug interacti ons

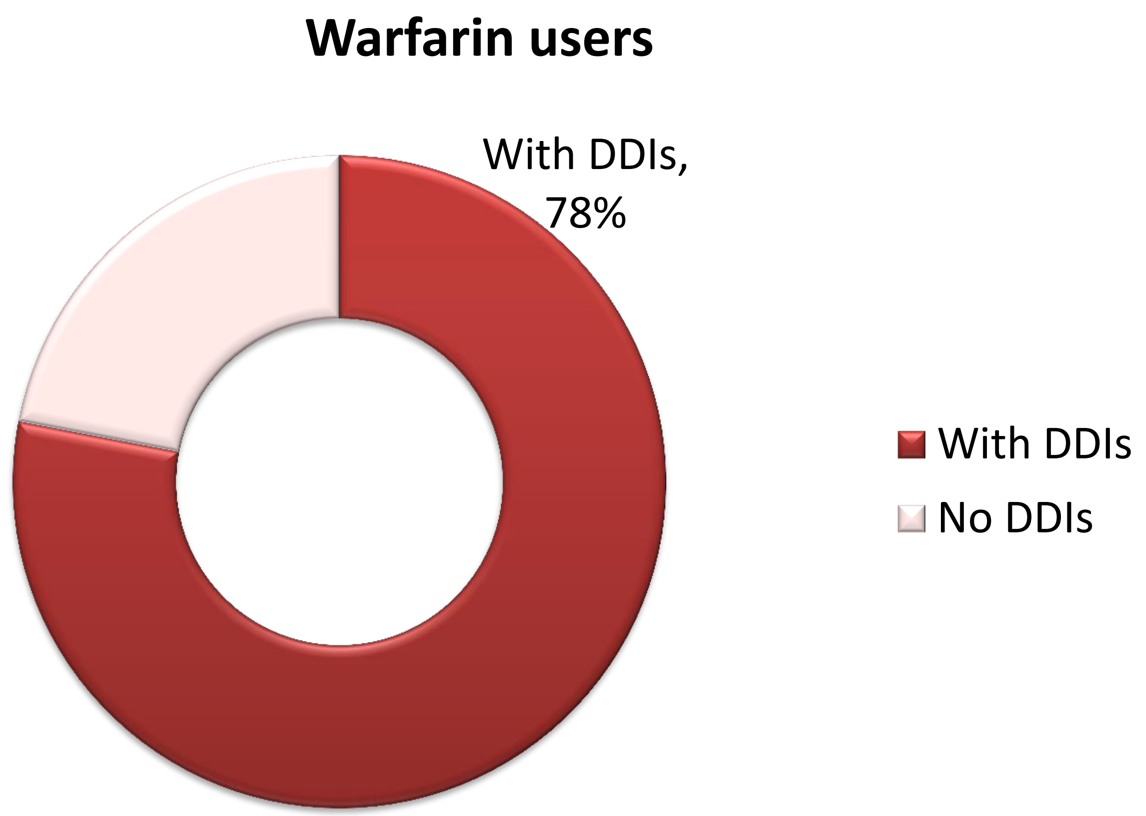
Drug-drug interactions

Polypharmacy

Prevalence of DDIs



Prevalence of DDIs

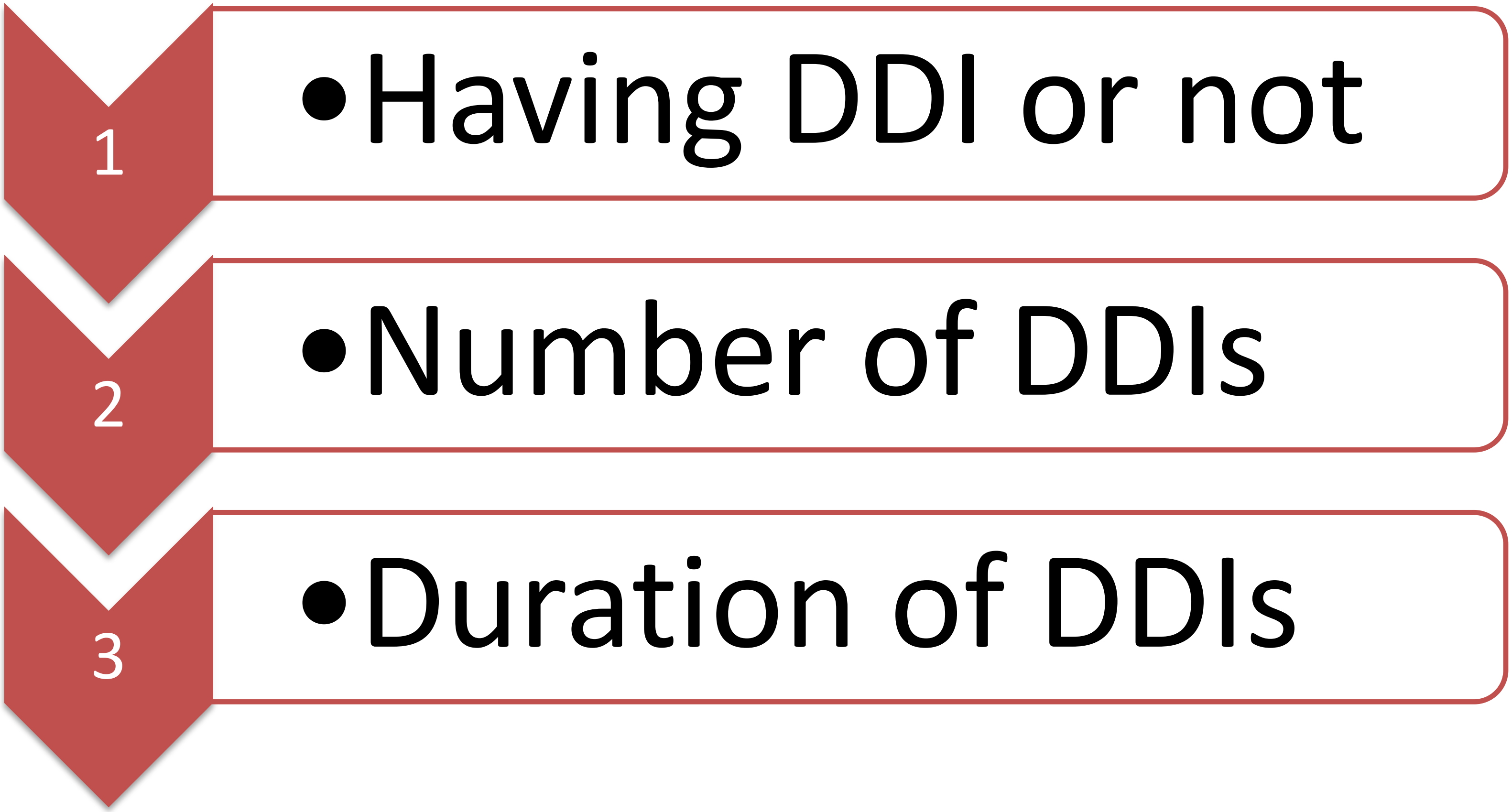


Why DDIs is important

Exposure to DDIs is associated with

- Adverse events
- Ineffectiveness of medications
- Increasing health utilizations & costs

To measure DDIs



1

- Having DDI or not

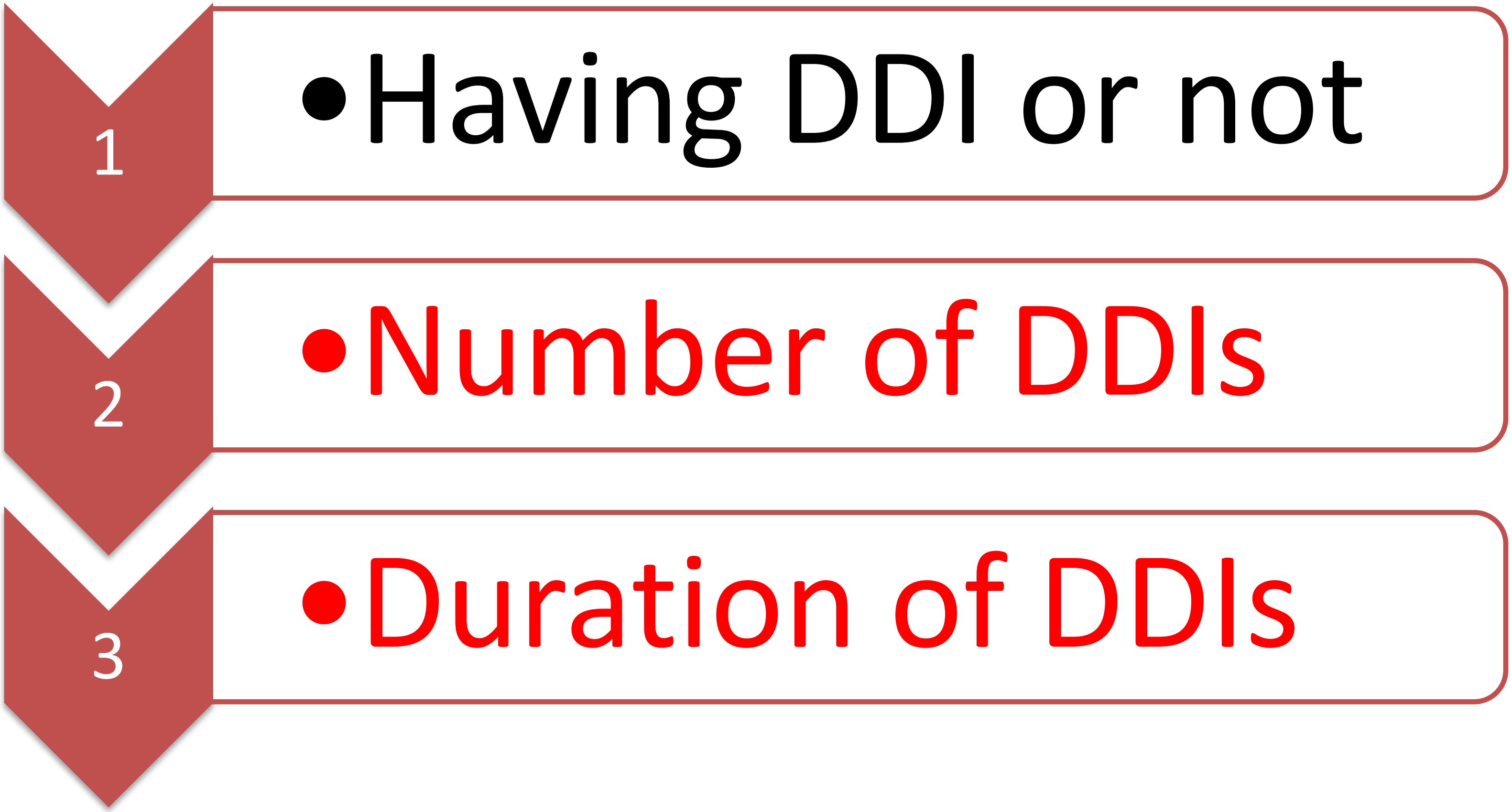
2

- Number of DDIs

3

- Duration of DDIs

To measure DDIs



1. Having DDI or not
2. Number of DDIs
3. Duration of DDIs

Outline

- Objectives
- Datasets
- R functions and inputs
- R coding

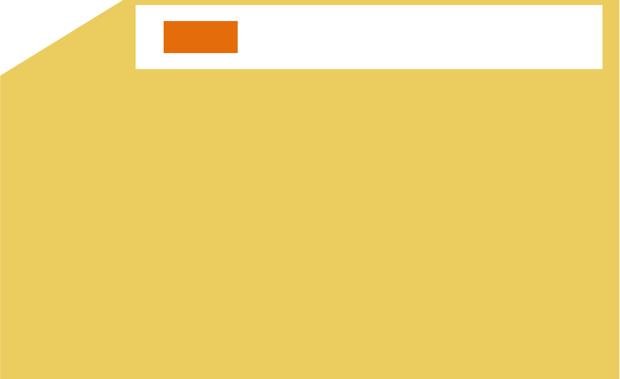
Objective

- Calculating proportion of days with potential drug-drug interactions using R

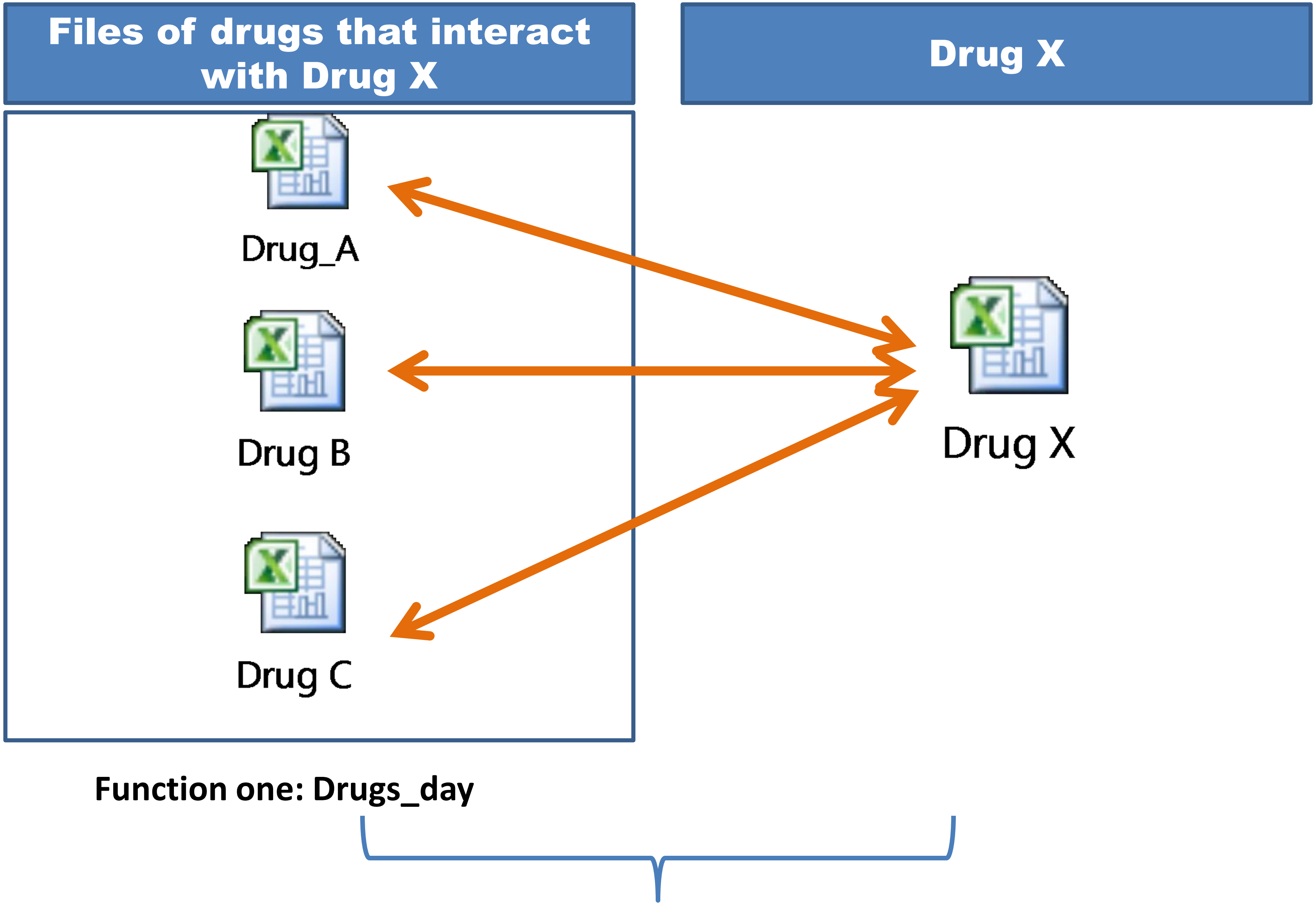
Calculating potential DDIs with “study drug”

Files of drugs that interact with Drug X

Drug X



Drug X



Files of drugs that interact with Drug X

Drug X

Drug_A

Drug B

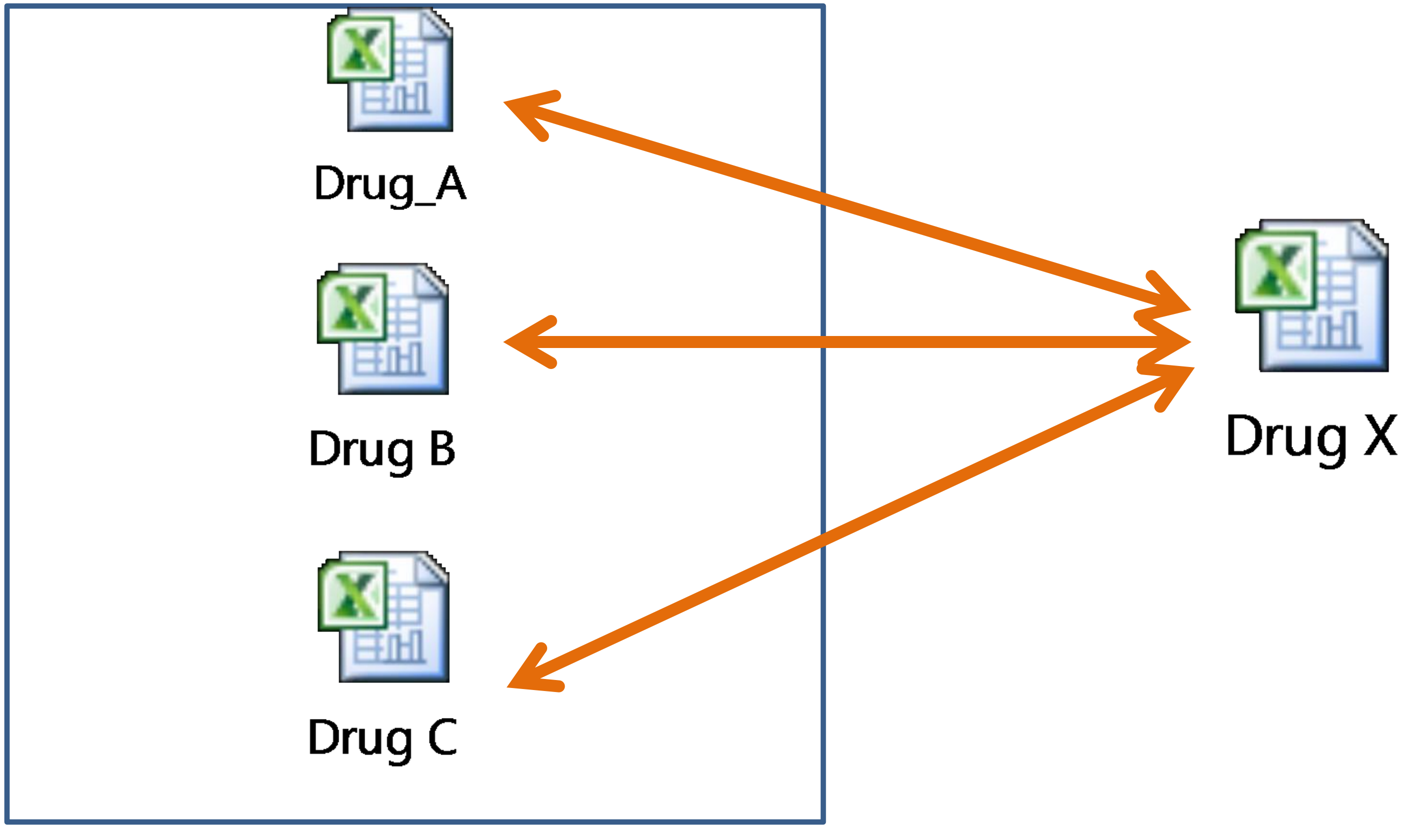
Drug C

Drug X

Function one: Drugs_day

Function two: Drugs_day2

Files of drugs that interact with Drug X	Drug X



Drug_A

Drug B

Drug C

Drug X

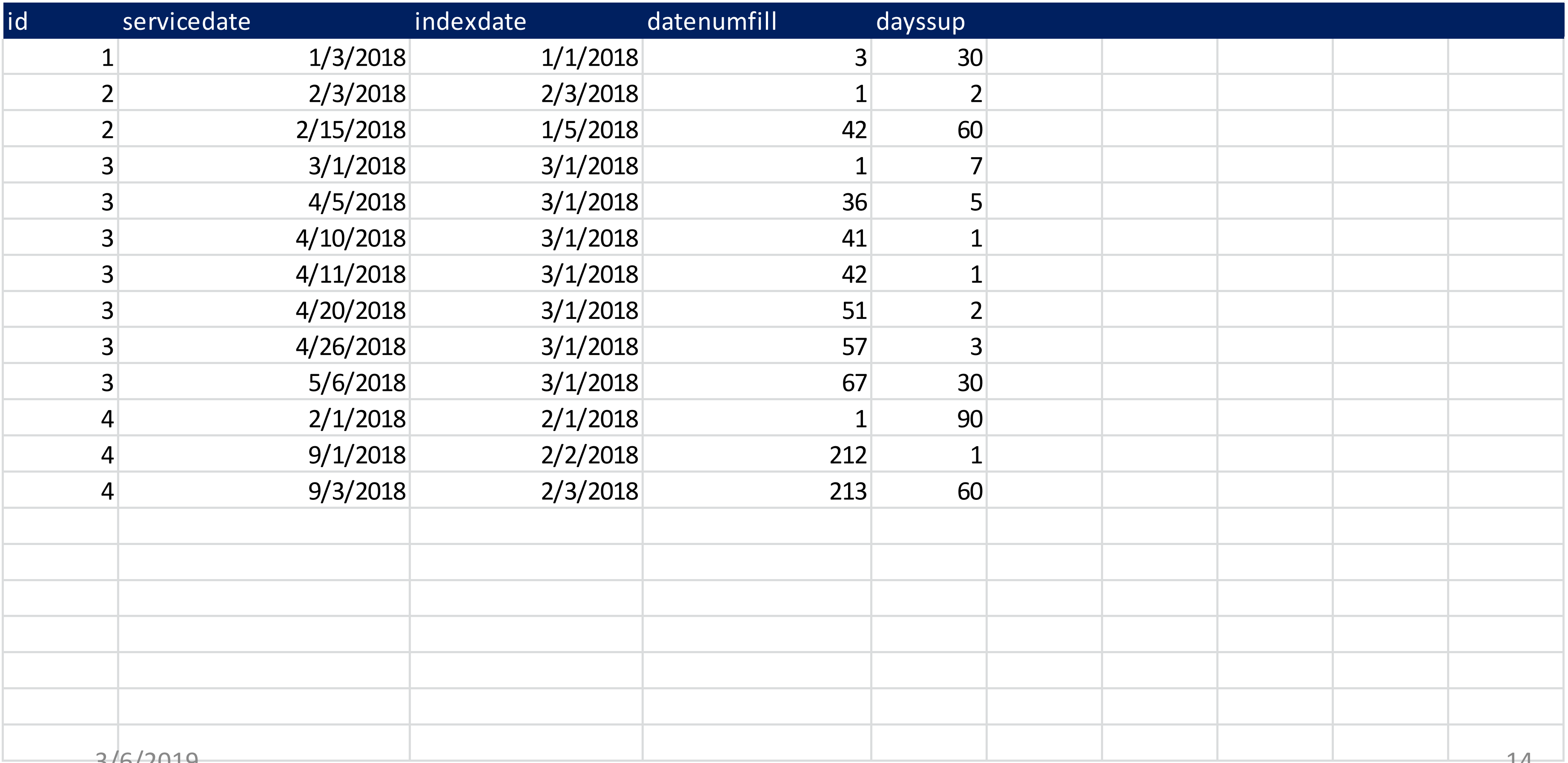
Patient ID	Patient ID
Refill day	Refill day
Days of supply	Days of supply

Rx refill date

1st date of assessment

Drug X

Patient ID **Service date** - **Index date** = **Refill day** **Days of supply**



id	servicedate	indexdate	datenumfill	dayssup
1	1/3/2018	1/1/2018	3	30
2	2/3/2018	2/3/2018	1	2
2	2/15/2018	1/5/2018	42	60
3	3/1/2018	3/1/2018	1	7
3	4/5/2018	3/1/2018	36	5
3	4/10/2018	3/1/2018	41	1
3	4/11/2018	3/1/2018	42	1
3	4/20/2018	3/1/2018	51	2
3	4/26/2018	3/1/2018	57	3
3	5/6/2018	3/1/2018	67	30
4	2/1/2018	2/1/2018	1	90
4	9/1/2018	2/2/2018	212	1
4	9/3/2018	2/3/2018	213	60

Calculating DDIs using R

- 2 functions
- 6 inputs

```
drugs_day2(input1,input2,input5,drugs_day(
input1,input2,input3,input4,input6)
```

drugs_day2(input1,input2,input5,drugs_day(input1,input2,input3,input4),input6)

- #input 1: number of patients;
- #input 2: number of day;
- #input 3: number of all drug-drug interactions;
- #input 4: drugs files directory: all other drugs profiles except study drug (cvs. files)
- #input 5: study drug file directory: study drug profile (cvs. file)
- #input 6: assessment time frame

Function: drugs_day

```
drugs_day <- function(n_pat, n_day, n_drug, files){
  n_drug_day <- matrix(0, ncol = n_day, nrow = n_pat)
  for (k in 1: n_drug){
    day_on_drug <- matrix(0, ncol = n_day, nrow = n_pat)
    drug_supply_refill <- read.csv(files[k])
    for (i in 1:n_pat) {
      # identify individual patient #
      drug_supply_refill_pat <- drug_supply_refill[drug_supply_refill$id==i,]
      n_record <- nrow(drug_supply_refill_pat)
      for (j in 1:n_record){
        subdata <- drug_supply_refill_pat[drug_supply_refill_pat$seq_id==j,]
        first <- subdata$datenumfill
        last <- subdata$datenumfill+subdata$dayssup-1
        day_on_drug[i,first:last] <- 1
      }
    }
    n_drug_day <- n_drug_day + day_on_drug
    return(day_on_drug)
  }
}
```

Drug level

Person level

Claim level

Function: drugs_day

```
drugs_day <- function(n_pat, n_day, n_drug, files){
 n_drug_day <- matrix(0, ncol = n_day, nrow = n_pat)
 for (k in 1: n_drug){
  day_on_drug <- matrix(0, ncol = n_day, nrow = n_pat)
  drug_supply_refill <- read.csv(files[k])
  for (i in 1:n_pat) {
   # identify individual patient #
   drug_supply_refill_pat <- drug_supply_refill[drug_supply_refill$id==i,]
   n_record <- nrow(drug_supply_refill_pat)
   for (j in 1:n_record){
    subdata <- drug_supply_refill_pat[drug_supply_refill_pat$seq_id==j,]
    first <- subdata$datenumfill
    last <- subdata$datenumfill+subdata$dayssup-1
    day_on_drug[i,first:last] <- 1
   }
  }
  n_drug_day <- n_drug_day + day_on_drug
  return(day_on_drug)
 }
}
```

Start from an empty matrix
X:number of patients
Y: number of day

X: number of day

Y: number of patients

	1	2	3	4	5	6	7	8	…
1									
2									
3									
4									
5									
6									
…									

Final output:
Summarized drugs profile

Function: drugs_day

```
drugs_day <- function(n_pat, n_day, n_drug, files){
 n_drug_day <- matrix(0, ncol = n_day, nrow = n_pat)
 for (k in 1: n_drug){
  day_on_drug <- matrix(0, ncol = n_day, nrow = n_pat)
  drug_supply_refill <- read.csv(files[k])
  for (i in 1:n_pat) {
   # identify individual patient #
   drug_supply_refill_pat <- drug_supply_refill[drug_supply_refill$id==i,]
   n_record <- nrow(drug_supply_refill_pat)
   for (j in 1:n_record){
    subdata <- drug_supply_refill_pat[drug_supply_refill_pat$seq_id==j,]
    first <- subdata$datenumfill
    last <- subdata$datenumfill+subdata$dayssup-1
    day_on_drug[i,first:last] <- 1
   }
  }
  n_drug_day <- n_drug_day + day_on_drug
  return(day_on_drug)
 }
}
```

For each of drug profile

Function: drugs_day

```
drugs_day <- function(n_pat, n_day, n_drug, files){
  n_drug_day <- matrix(0, ncol = n_day, nrow = n_pat)
  for (k in 1: n_drug){
    day_on_drug <- matrix(0, ncol = n_day, nrow = n_pat)
    drug_supply_refill <- read.csv(files[k])
    for (i in 1:n_pat) {
      # identify individual patient #
      drug_supply_refill_pat <- drug_supply_refill[drug_supply_refill$id==i,]
      n_record <- nrow(drug_supply_refill_pat)
      for (j in 1:n_record){
        subdata <- drug_supply_refill_pat[drug_supply_refill_pat$seq_id==j,]
        first <- subdata$datenumfill
        last <- subdata$datenumfill+subdata$dayssup-1
        day_on_drug[i,first:last] <- 1
      }
    }
    n_drug_day <- n_drug_day + day_on_drug
    return(day_on_drug)
  }
}
```

An empty matrix: drug A (B, C)
X:number of patients
Y: number of day

X: number of day

Y: number of patients

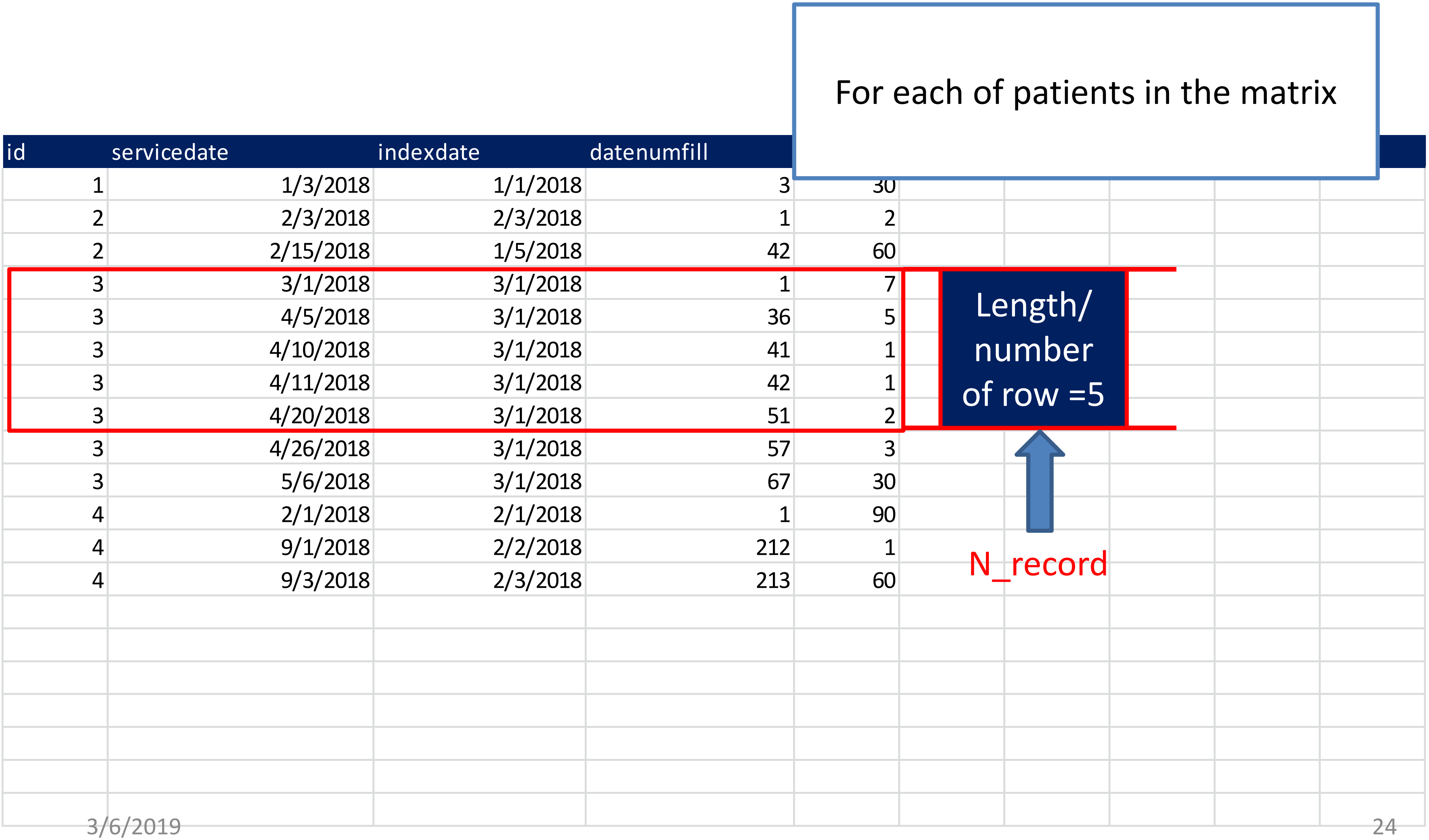
	1	2	3	4	5	6	7	8	…
1									
2									
3									
4									
5									
6									
…									

Drug A (B,C) profile

Function: drugs_day

```
drugs_day <- function(n_pat, n_day, n_drug, files){
  n_drug_day <- matrix(0, ncol = n_day, nrow = n_pat)
  for (k in 1: n_drug){
    day_on_drug <- matrix(0, ncol = n_day, nrow = n_pat)
    drug_supply_refill <- read.csv(files[k])
    for (i in 1:n_pat) {
      # identify individual patient #
      drug_supply_refill_pat <- drug_supply_refill[drug_supply_refill$id==i,]
      n_record <- nrow(drug_supply_refill_pat)
      for (j in 1:n_record){
        subdata <- drug_supply_refill_pat[drug_supply_refill_pat$seq_id==j,]
        first <- subdata$datenumfill
        last <- subdata$datenumfill+subdata$dayssup-1
        day_on_drug[i,first:last] <- 1
      }
    }
    n_drug_day <- n_drug_day + day_on_drug
    return(day_on_drug)
  }
}
```

For each of patients in the matrix



For each of patients in the matrix

id	servicedate	indexdate	datenumfill	
1	1/3/2018	1/1/2018	3	30
2	2/3/2018	2/3/2018	1	2
2	2/15/2018	1/5/2018	42	60
3	3/1/2018	3/1/2018	1	7
3	4/5/2018	3/1/2018	36	5
3	4/10/2018	3/1/2018	41	1
3	4/11/2018	3/1/2018	42	1
3	4/20/2018	3/1/2018	51	2
3	4/26/2018	3/1/2018	57	3
3	5/6/2018	3/1/2018	67	30
4	2/1/2018	2/1/2018	1	90
4	9/1/2018	2/2/2018	212	1
4	9/3/2018	2/3/2018	213	60

Length/ number of row =5

N_record

Function: drugs_day

```
drugs_day <- function(n_pat, n_day, n_drug, files){
  n_drug_day <- matrix(0, ncol = n_day, nrow = n_pat)
  for (k in 1: n_drug){
    day_on_drug <- matrix(0, ncol = n_day, nrow = n_pat)
    drug_supply_refill <- read.csv(files[k])
    for (i in 1:n_pat) {
      # identify individual patient #
      drug_supply_refill_pat <- drug_supply_refill[drug_supply_refill$id==i,]
      n_record <- nrow(drug_supply_refill_pat)
      for (j in 1:n_record){
        subdata <- drug_supply_refill_pat[drug_supply_refill_pat$seq_id==j,]
        first <- subdata$datenumfill
        last <- subdata$datenumfill+subdata$dayssup-1
        day_on_drug[i,first:last] <- 1
      }
    }
    n_drug_day <- n_drug_day + day_on_drug
    return(day_on_drug)
  }
}
```

Identify the days with prescription of A (B, C)

X: number of day

Y: number of patients

	1	2	3	4	5	6	7	8	…
1									
2									
3									
4									
5									
6									
…									

Drug A profile

X: number of day

Y: number of patients

	1	2	3	4	5	6	7	8	…
1		1	1	1	1	1	1		
2									
3				1	1	1	1	1	1
4		1	1	1	1				
5									
6	1	1	1	1	1	1	1	1	
…									

Drug A profile

X: number of day

Y: number of patients

	1	2	3	4	5	6	7	8	…
1									
2		1	1	1	1	1	1	1	
3									
4									
5				1	1	1	1		
6									
…									

Drug B profile

Function: drugs_day

```
drugs_day <- function(n_pat, n_day, n_drug, files){
  n_drug_day <- matrix(0, ncol = n_day, nrow = n_pat)
  for (k in 1: n_drug){
    day_on_drug <- matrix(0, ncol = n_day, nrow = n_pat)
    drug_supply_refill <- read.csv(files[k])
    for (i in 1:n_pat) {
      # identify individual patient #
      drug_supply_refill_pat <- drug_supply_refill[drug_supply_refill$id==i,]
      n_record <- nrow(drug_supply_refill_pat)
      for (j in 1:n_record){
        subdata <- drug_supply_refill_pat[drug_supply_refill_pat$seq_id==j,]
        first <- subdata$datenumfill
        last <- subdata$datenumfill+subdata$dayssup-1
        day_on_drug[i,first:last] <- 1
      }
    }
    n_drug_day <- n_drug_day + day_on_drug
    return(day_on_drug)
  }
}
```

Function 1 output:

X: number of day

Y: number of patients

	1	2	3	4	5	6	7	8	…
1	0	1	1	2	2	2	0	0	0
2	0	1	1	1	1	1	1	1	0
3	0	0	0	1	1	1	1	1	1
4	0	1	1	1	1	0	0	0	0
5	0	0	0	1	2	2	2	1	1
6	1	1	1	1	1	1	1	1	0
…									

Matrix [A] + Matrix [B] + Matrix [C]

Function: drugs_day2

```
drugs_day2 <- function(n_pat,n_day,drugtaget,n_drug_day, time){
drug<-data.frame(matrix(0, ncol =n_day, nrow = n_pat))
for (j in 1:n_pat) {
  subkd<-drugtaget[drugtaget$id==j,]
  leng<-length(subkd$id)*1
  for (i in 1:leng){
   subdata <-  subkd[ subkd$seq_id==i,]
   firsta <- subdata$datenumfill
   lasta <- subdata$datenumfill+subdata$dayssup-1
   drug[j,firsta:lasta]<-1
  }
 }
 n_drug_day[n_drug_day < 1] <-0
 n_drug_day[n_drug_day > 0] <-1
 ddi_have<-n_drug_day + drug
 ddi_have[ddi_have < 2] <-0
 ddi_have[ddi_have > 1] <-1
 ddi_have_time <-ddi_have[, 1:time]
 ddi_have_timea<-as.data.frame(t(ddi_have_time ))
 colsum <- function(data) apply(data, 2, sum)
 ddi_proportion<-colsum(ddi_have_timea)
 ddi_proportion<-as.matrix(ddi_proportion)
 proportion<-mean(ddi_proportion)/time
```

Similar steps as function: drugs_day

Function: drugs_day2

```
drugs_day2 <- function(n_pat,n_day,drugtaget,n_drug_day, time){
drug<-data.frame(matrix(0, ncol =n_day, nrow = n_pat))
for (j in 1:n_pat) {
   subkd<-drugtaget[drugtaget$id==j,]
   leng<-length(subkd$id)*1
   for (i in 1:leng){
     subdata <-  subkd[ subkd$seq_id==i,]
     firsta <- subdata$datenumfill
     lasta <- subdata$datenumfill+subdata$dayssup-1
     drug[j,firsta:lasta]<-1
   }
  }
 n_drug_day[n_drug_day < 1] <-0
 n_drug_day[n_drug_day > 0] <-1
 ddi_have<-n_drug_day + drug
 ddi_have[ddi_have < 2] <-0
 ddi_have[ddi_have > 1] <-1
 ddi_have_time <-ddi_have[, 1:time]
 ddi_have_timea<-as.data.frame(t(ddi_have_time ))
 colsum <- function(data) apply(data, 2, sum)
 ddi_proportion<-colsum(ddi_have_timea)
 ddi_proportion<-as.matrix(ddi_proportion)
 proportion<-mean(ddi_proportion)/time
```

Similar steps as function: drugs_day

Function 1 output:

X: number of day

Y: number of patients

	1	2	3	4	5	6	7	8	…
1	0	1	1	2	2	2	0	0	0
2	0	1	1	1	1	1	1	1	0
3	0	0	0	1	1	1	1	1	1
4	0	1	1	1	1	0	0	0	0
5	0	0	0	1	2	2	2	1	1
6	1	1	1	1	1	1	1	1	0
…									

Matrix [A] + Matrix [B] + Matrix [C]

Function 1 output:

X: number of day

Y: number of patients

	1	2	3	4	5	6	7	8	…
1	0	1	1	1	1	1	0	0	0
2	0	1	1	1	1	1	1	1	0
3	0	0	0	1	1	1	1	1	1
4	0	1	1	1	1	0	0	0	0
5	0	0	0	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	0
…									

Assign “1” if patients took ≥ 1 prescription

	1	2	3	4	5	6	7	8	…
1	0	1	1	1	1	1	0	0	0
2	0	1	1	1	1	1	1	1	0
3	0	0	0	1	1	1	1	1	1
4	0	1	1	1	1	0	0	0	0
5	0	0	0	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	0
…									

Summarized drug matrix

+

	1	2	3	4	5	6	7	8	…
1	0	0	0	0	1	1	1	1	1
2	0	1	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	1
4	1	1	1	1	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	1	1	1	1
…									

Drug X matrix

=

	1	2	3	4	5	6	7	8	…
1	0	1	1	1	2	2	2	1	1
2	0	2	2	1	1	1	1	1	0
3	0	0	0	1	1	1	1	2	2
4	1	2	2	2	2	0	0	0	0
5	0	0	0	1	1	1	1	1	1
6	2	2	2	2	2	2	2	2	1
…									

	1	2	3	4	5	6	7	8	…
1	0	1	1	1	1	1	0	0	0
2	0	1	1	1	1	1	1	1	0
3	0	0	0	1	1	1	1	1	1
4	0	1	1	1	1	0	0	0	0
5	0	0	0	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	0
…									

Summarized drug matrix

+

	1	2	3	4	5	6	7	8	…
1	0	0	0	0	1	1	1	1	1
2	0	1	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	1
4	1	1	1	1	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	1	1	1	1
…									

Drug X matrix

=

	1	2	3	4	5	6	7	8	…
1	0	0	0	0	1	1	1	0	0
2	0	1	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	1
4	0	1	1	1	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	1	1	1	0
…									

	1	2	3	4	5	6	7	8	…
1	0	1	1	1	1	1	0	0	0
2	0	1	1	1	1	1	1	1	0
3	0	0	0	1	1	1	1	1	1
4	0	1	1	1	1	0	0	0	0
5	0	0	0	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	0
…									

Summarized drug matrix

+

	1	2	3	4	5	6	7	8	…
1	0	0	0	0	1	1	1	1	1
2	0	1	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	1
4	1	1	1	1	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	1	1	1	1
…									

Drug X matrix

=

1	3
2	2
3	2
4	4
5	0
6	8
…	

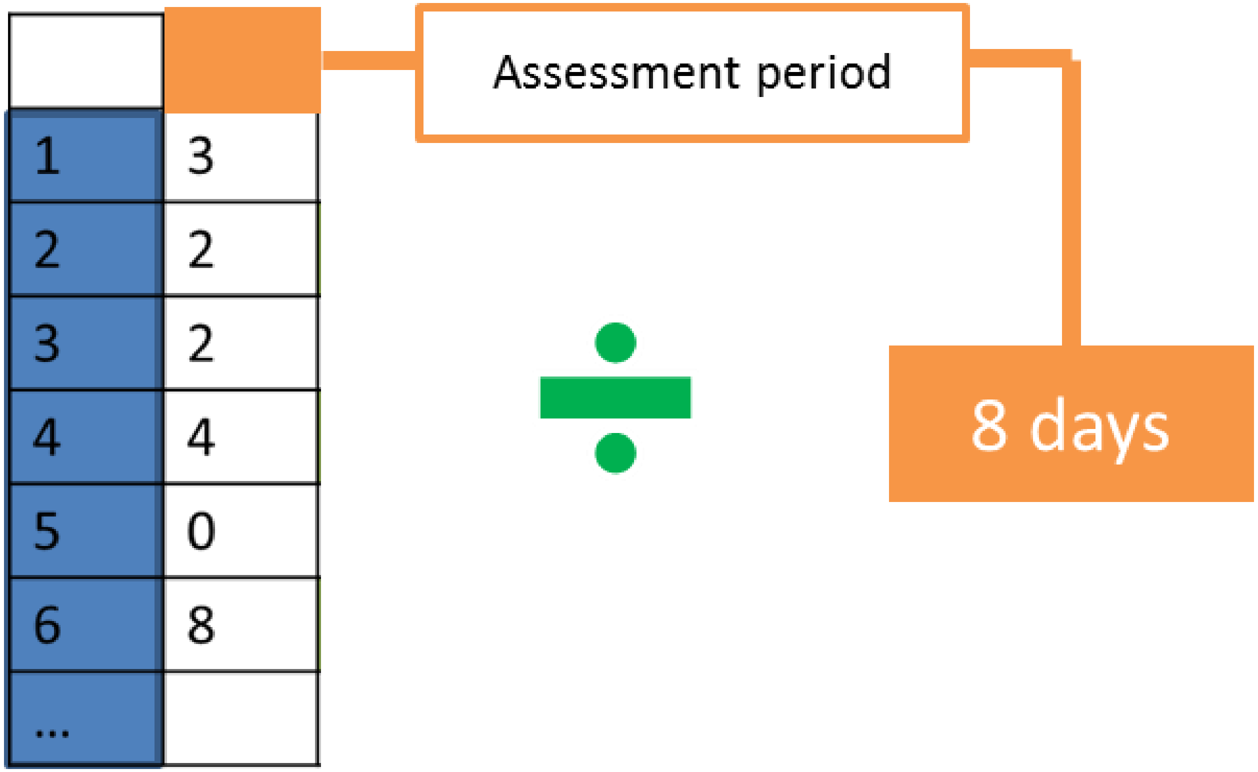
Sum the number in each row

Function: drugs_day2

```
drugs_day2 <- function(n_pat,n_day,drugtaget,n_drug_day, time){
drug<-data.frame(matrix(0, ncol =n_day, nrow = n_pat))
for (j in 1:n_pat) {
  subkd<-drugtaget[drugtaget$id==j,]
  leng<-length(subkd$id)*1
  for (i in 1:leng){
   subdata <-  subkd[ subkd$seq_id==i,]
   firsta <- subdata$datenumfill
   lasta <- subdata$datenumfill+subdata$dayssup-1
   drug[j,firsta:lasta]<-1
  }
 }
n_drug_day[n_drug_day < 1] <-0
n_drug_day[n_drug_day > 0] <-1
ddi_have<-n_drug_day + drug
ddi_have[ddi_have < 2] <-0
ddi_have[ddi_have > 1] <-1
ddi_have_time <-ddi_have[, 1:time]
ddi_have_timea<-as.data.frame(t(ddi_have_time ))
colsum <- function(data) apply(data, 2, sum)
ddi_proportion<-colsum(ddi_have_timea)
ddi_proportion<-as.matrix(ddi_proportion)
proportion<-mean(ddi_proportion)/time
```

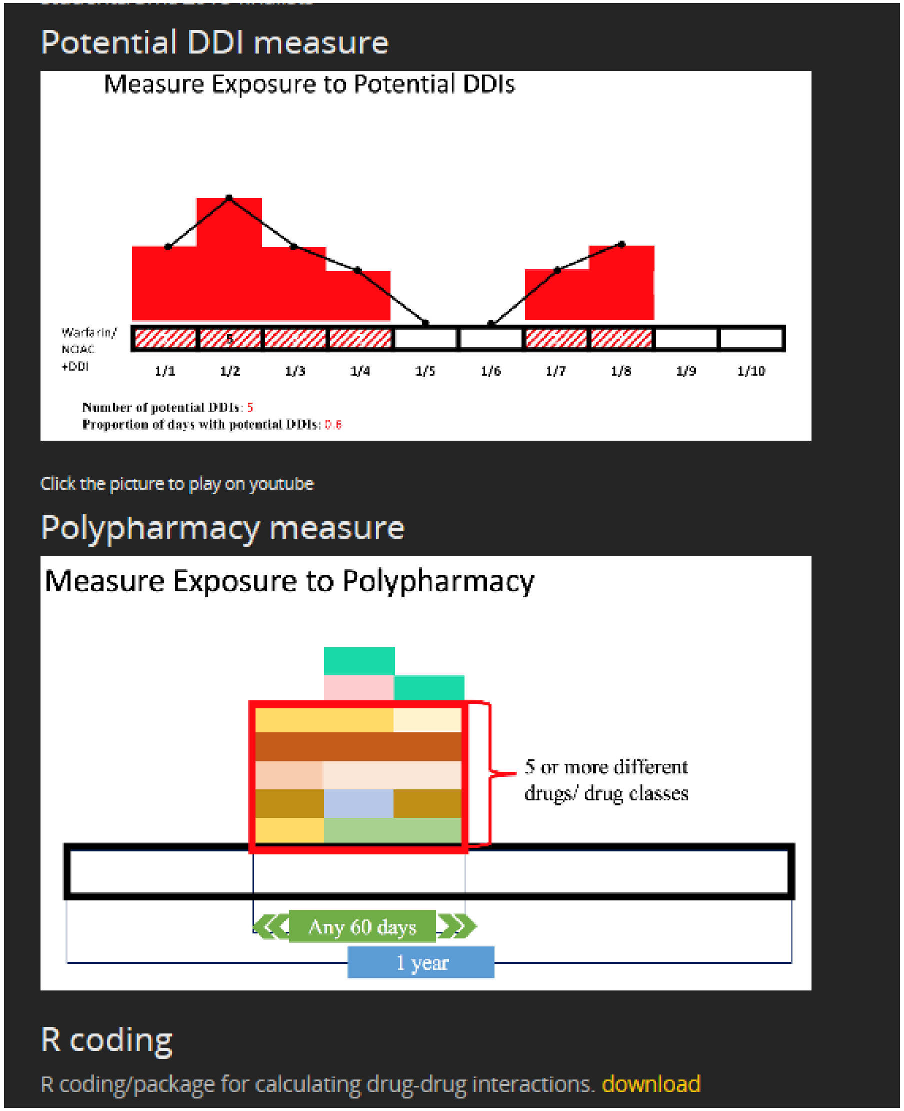
Add the matrix together and identify the days with concomitant medication use.

Proportion of days with DDIs



1	3
2	2
3	2
4	4
5	0
6	8
…	

https://medicationmanagement.github.io/



Potential DDI measure

Measure Exposure to Potential DDIs

Number of potential DDIs: 5
Proportion of days with potential DDIs: 0.6

Click the picture to play on youtube

Polypharmacy measure

Measure Exposure to Polypharmacy

R coding

R coding/package for calculating drug-drug interactions. download

Snow Feng. PhD
xfeng@tuftsmedicalcenter.org