

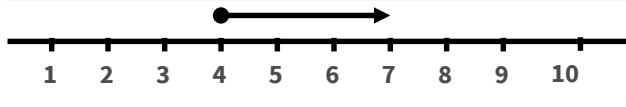
Multi-stage deterministic linkages and case definitions with diyar: : CHEAT SHEET

number_line objects

Series of real numbers on a number line.

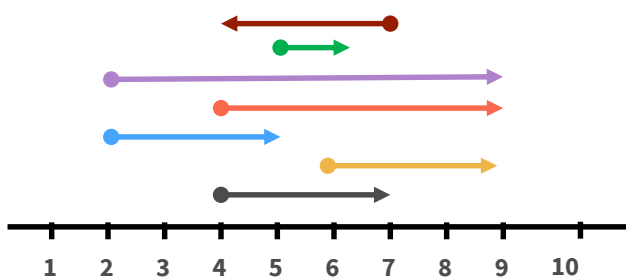
Stand alone S4 objects but also used in record and episode grouping

```
number_line(4, 7)
```



MANIPULATE NUMBER LINE OBJECTS

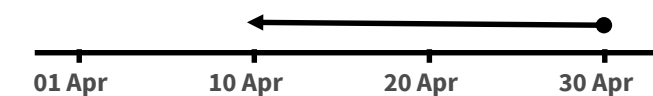
```
n1 <- number_line(4, 7)
shift_number_line(n1, 2)
shift_number_line(n1, -2)
expand_number_line(n1, 2, "end")
expand_number_line(n1, 2)
expand_number_line(n1, -1)
reverse_number_line(n1)
```



```
> number_line_width(n1)
[1] 3
> number_line_sequence(n1, 1)
[1] 4 5 6 7
> number_line_sequence(n1, .5)
[1] 4.0 4.5 5.0 5.5 6.0 6.5 7.0
> left_point(reverse_number_line(n1))
[1] 7
> start_point(reverse_number_line(n1))
[1] 4
> right_point(reverse_number_line(n1))
[1] 4
> end_point(reverse_number_line(n1))
[1] 7
```

Also supports objects that can be coerced to numeric values

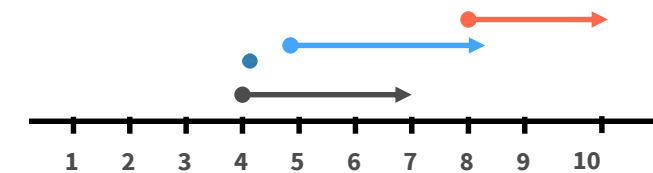
```
dates <- c("30/04/2019", "10/04/2019")
dates <- as.Date(dates, "%d/%m/%Y")
number_line(dates[1], dates[2])
```



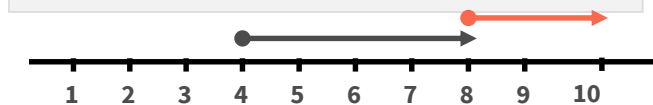
COMBINE NUMBER LINE OBJECTS

Overlapping number_line objects can be merged

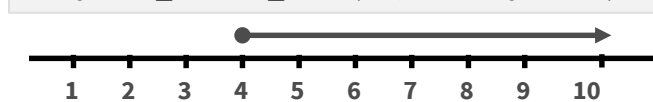
```
n1 <- number_line(4, 7)
n2 <- as.number_line(4)
n3 <- number_line(5, 8)
n4 <- number_line(8, 10)
n1 <- c(n1, n2, n3, n4)
```



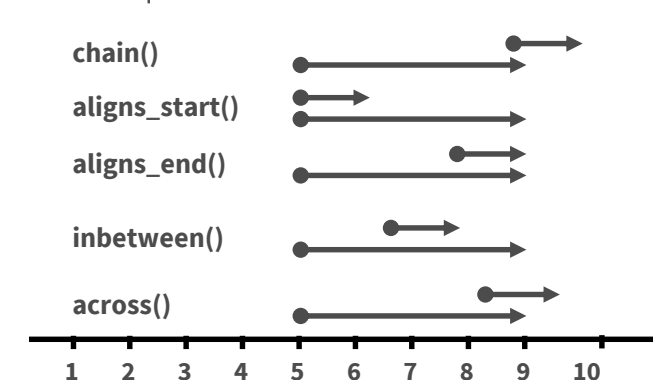
```
compress_number_line(n1)
```



```
compress_number_line(n1, collapse = T)
```



• overlap methods



Episode grouping

EPISODES FROM POINTS IN TIME

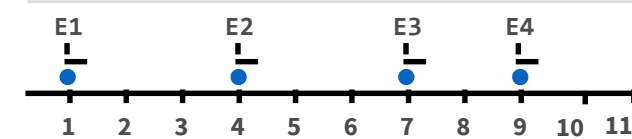
```
dates <- c("01", "04", "07", "09")
dates <- paste(dates, "04/2019", sep = "/")
dates <- as.Date(dates, "%d/%m/%Y")
fixed_episodes(dates,
case_length = 5, to_s4 = T)
```



```
fixed_episodes(dates,
case_length = 5, to_s4 = T, from_last = T)
```

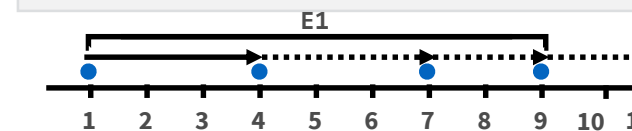


```
fixed_episodes(dates, case_length = 5, to_s4 = T, episode_unit = "hours")
```

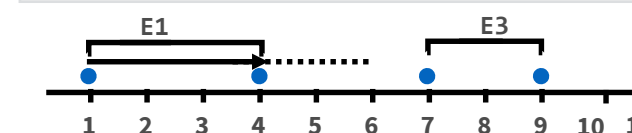


ROLLING EPISODES

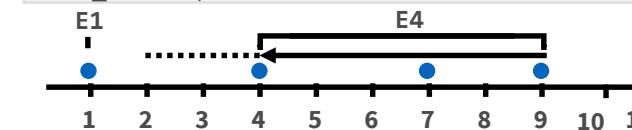
```
rolling_episodes(dates,
case_length = 5, to_s4 = T)
```



```
rolling_episodes(dates, case_length = 5,
recurrence_length = 2, to_s4 = T)
```

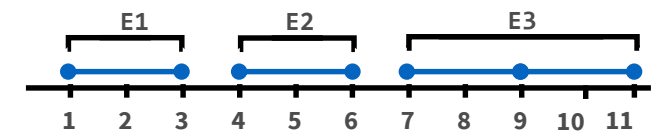


```
rolling_episodes(dates, case_length = 5,
recurrence_length = 2, from_last = T,
to_s4 = T)
```

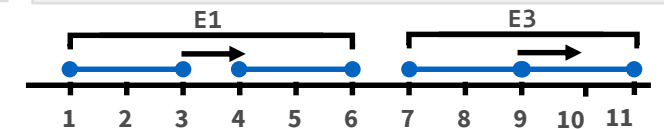


EPISODES FROM PERIODS IN TIME

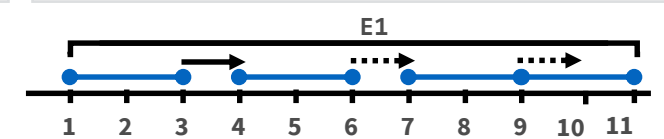
```
p <- as.number_line(dates)
p <- expand_number_line(p, 2, "end")
periods <- p
fixed_episodes(periods,
case_length = 0, to_s4 = T)
```



```
fixed_episodes(periods,
case_length = 1, to_s4 = T)
```

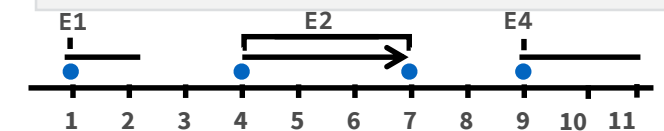


```
rolling_episodes(periods,
case_length = 1, to_s4 = T)
```

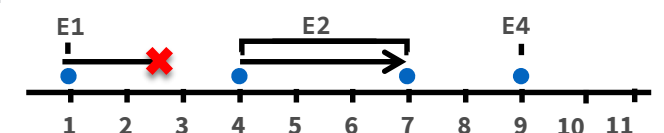


CONTROL CASE ASSIGNMENT

```
fixed_episodes(dates,
case_length = c(1, 3, 2, 2), to_s4 = T)
```



```
fixed_episodes(dates, case_length = 5,
custom_sort = c(1, 0, 1, 1), to_s4 = T)
```



dashed lines – recurrence periods
solid – initial case period
round end – start of an episode
arrow head – end of an episode

Multi-stage deterministic linkages and case definitions with diyar: : CHEAT SHEET

Record grouping

- Multi-stage deterministic linkage

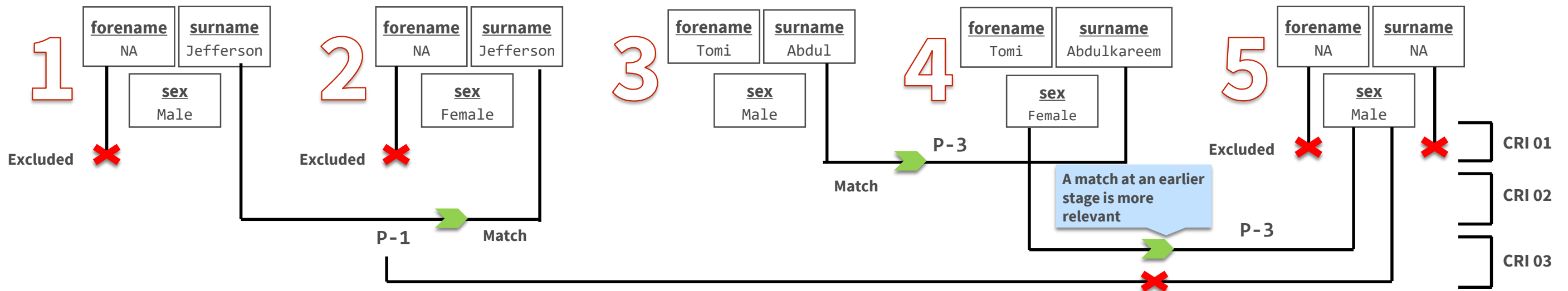
- Relevance of each stage controlled by `criteria`

- Use `sub_criteria` for additional matching conditions

- Missing data handled with alternative matching `criteria`

- Record grouping separately for subsets of the dataset using `strata`

```
record_group(diyar::patient_list_2, rd_id, c(forename, surname, sex), to_s4 = TRUE)
```



```
df <- Opes[c("department", "date_of_birth", "db_pt1", "db_pt2", "db_pt3")]
df$age <- dmy("02/11/2019") - dmy(df$date_of_birth)
df$age <- round(as.numeric(df$age)/365)
```

```
df$age_range <- number_line(df$age, df$age +5, gid=df$age)
record_group(df, criteria = c(department, department),
sub_criteria = list("s1"=c("age_range"), "s2"=c("db_pt1", "db_pt2", "db_pt3")),
display = FALSE, to_s4 = TRUE)
```

