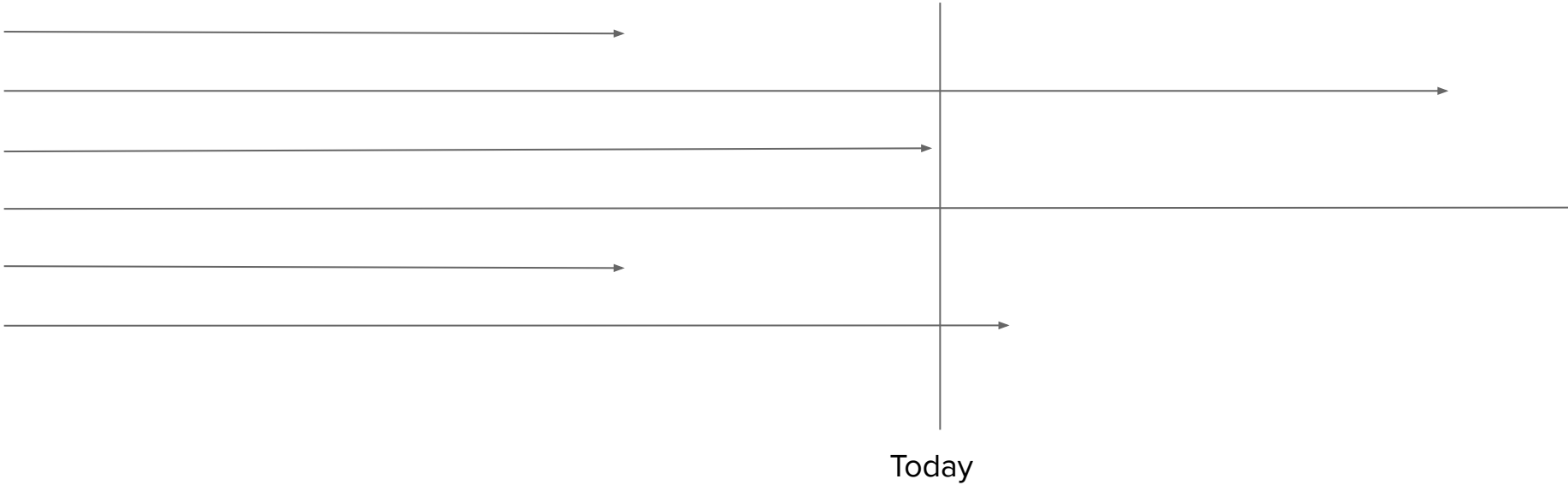
Time-to-Event Analysis

For Non-Medical Applications

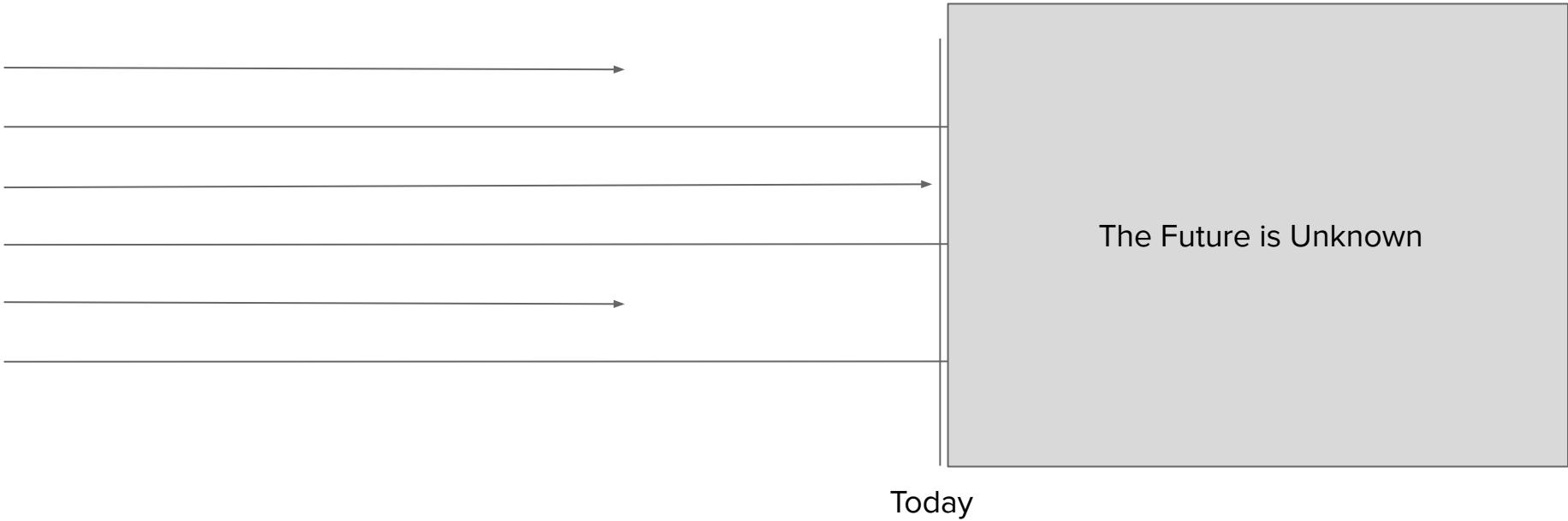
Outline

- Right-censored data
- Estimating the Survival Function
- Real world examples
- SQL for Kaplan-Meier curves

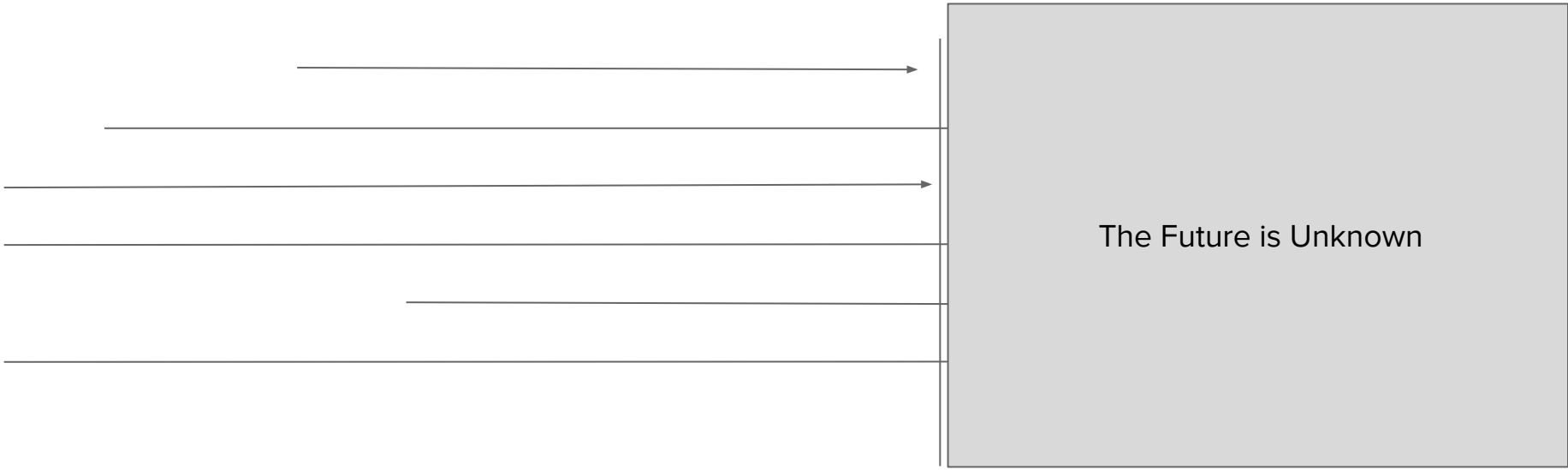
Right-Censored Data



Right-Censored Data



Right-Censored Data

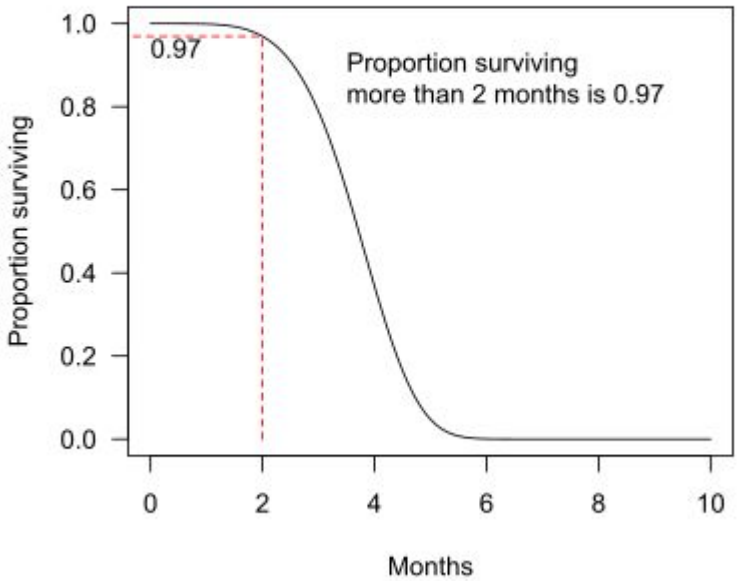


How do you find the midpoint when many points are unknown?

Kaplan-Meier Curves

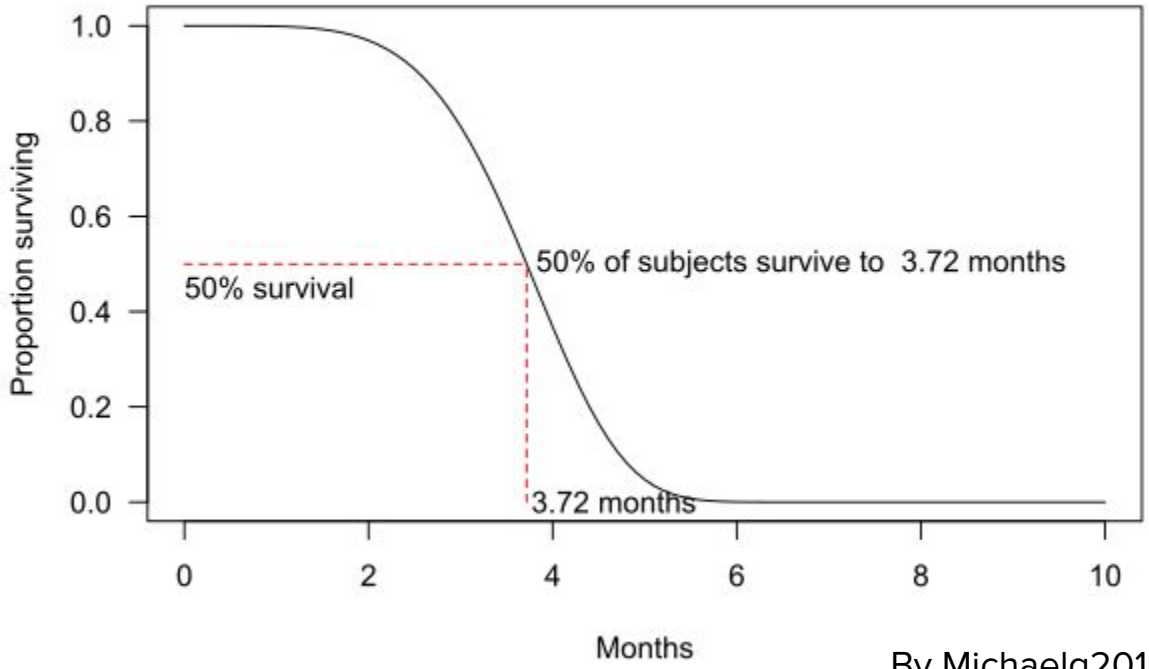
- Uses all the data available for each time point
- Approaches the true survival function
- No distributional assumptions

$$\hat{S} = \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$



https://en.wikipedia.org/wiki/Survival_function

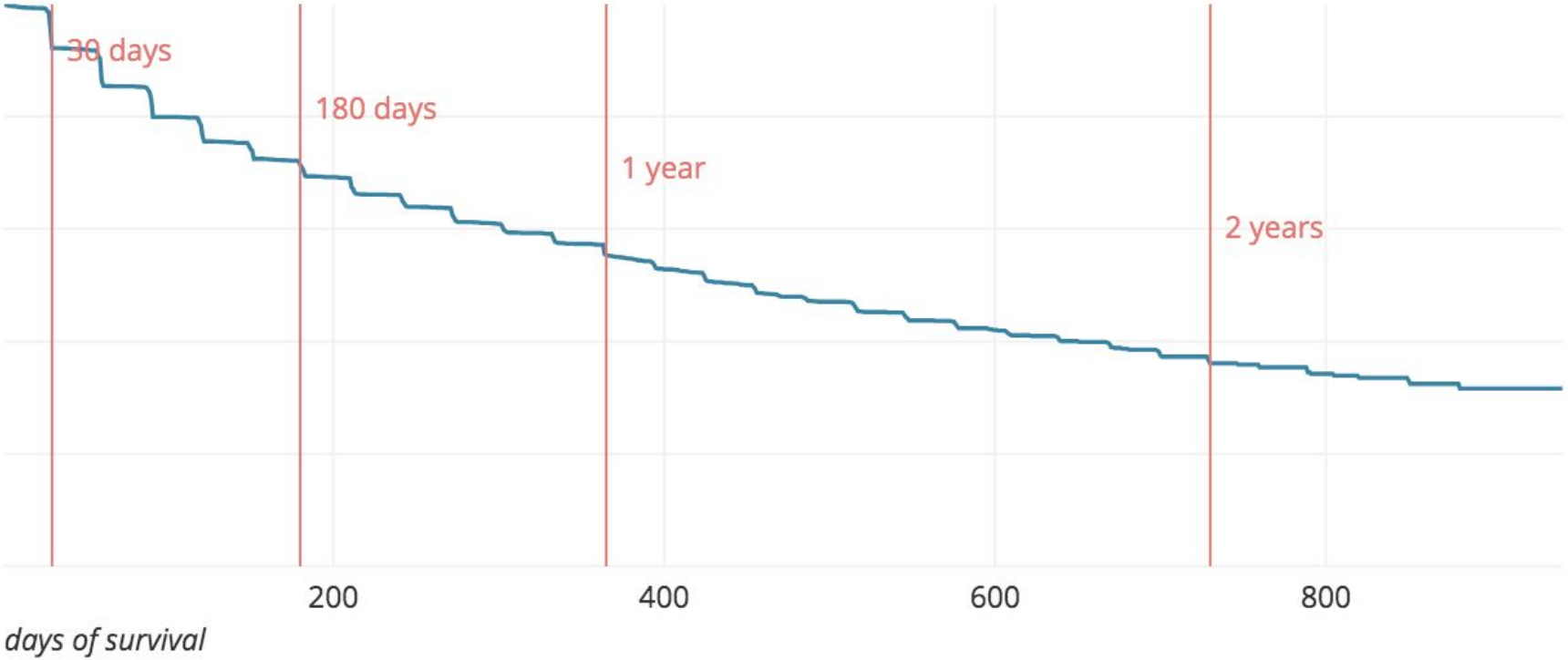
Finding the Median



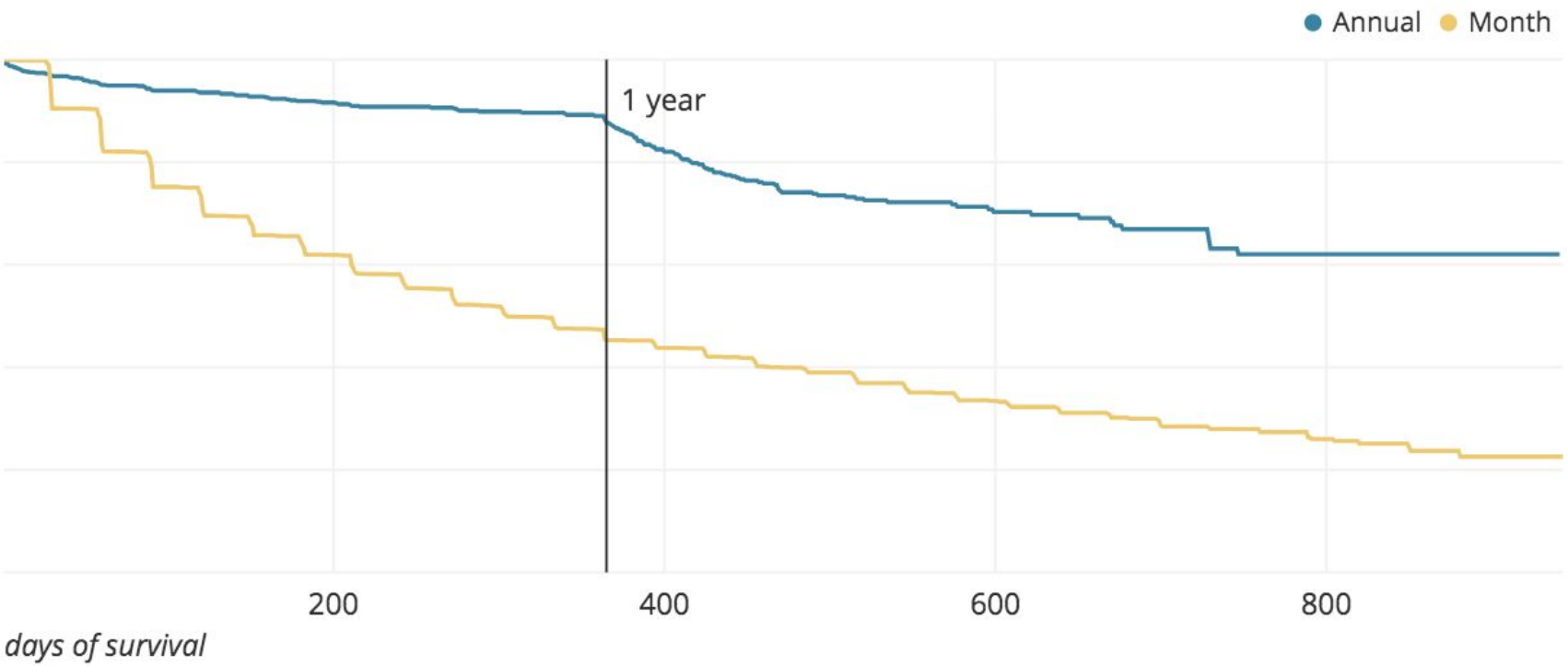
By Michaelg2015 / CC BY-SA

https://en.wikipedia.org/wiki/Survival_function

Time before Customer Ends Subscription



Time before Customer Ends Subscription



SQL for Kaplan-Meier Curves

Tricks we will use:

- Generate a list of numbers using row_number()
- Summing binary results of a case statement to count things
- The log of a product is equal to the sum of the logs

$$\hat{S} = \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) = \exp\left(\sum_{i:t_i \leq t} \log\left(1 - \frac{d_i}{n_i}\right)\right)$$

Heavily inspired by https://daynebatten.com/2015/10/sql-survival-curve-redshift-periscope/

SQL for Kaplan-Meier Curves

Format your main table as:

```
Id, event_time, time_seen
```

In my case, time_seen is in days, but you could use weeks, months, years, whatever makes sense for your application

```
day_shift as (
  select row_number() over () as day_num from any_big_table limit 3000 )
```

SQL for Kaplan-Meier Curves

```
daily_survival as (
  select day_num,
  (1-
sum(case when time_seen = day_num and event_time is not null then 1 else 0 end )::float
/sum(1)::float) as inside
  from day_shift d
  left outer join my_table
  on (time_seen >= day_num)
  group by 1
    having sum(case when time_seen >= day_num then 1 else 0 end) > 0

  order by 1)
```

SQL for Kaplan-Meier Curves

```
select
  day_num,
  exp(sum(ln(inside)) over(order by day_num rows between unbounded preceding and current row))
     as survival
from
  daily_survival
order by
  day_num;
```

day_num	survival
1	.9999
2	.9998
3	.9994

SQL for Kaplan-Meier Curves

```
select
  day_num, billing_period,
  exp(sum(ln(survival)) over ( partition by billing_period
                order by day_num rows between unbounded preceding and current row)) as survival
from
  daily_survival
order by
  day_num;
```

day_num	billing_period	survival
1	annual	.9999
1	monthly	.9996
2	annual	.9994
2	monthly	.9994

Time-to-Event Analysis can help!

- You actually need to know the average time until an event
- Examining the curve helps you understand the process
- Comparing curves helps you see different groups over time

How to find me

- #survival-analysis on the CSV,Conf slack
- @ansate on twitter
- @ansate@weirder.earth on mastodon