

Edith Heintjes



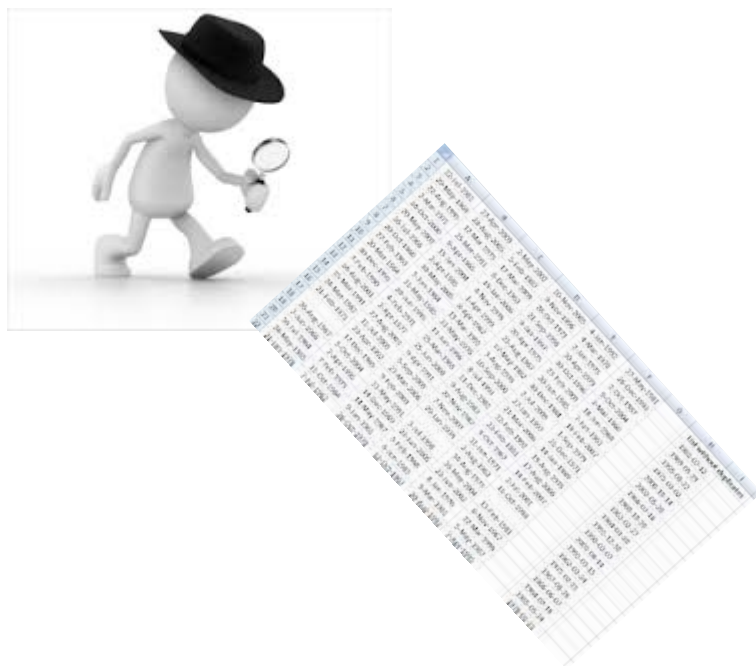
Simple and Efficient Matching Algorithms for Case-Control Matching

Contents

- Observational studies
- Basic technique
- Different matching options
- Conclusions

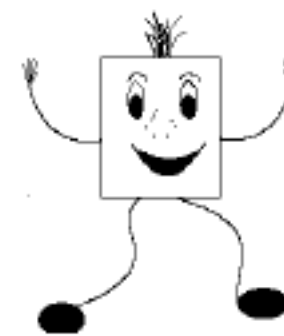
Observational studies

- (Retrospective) cohort
- Case-Control



Case

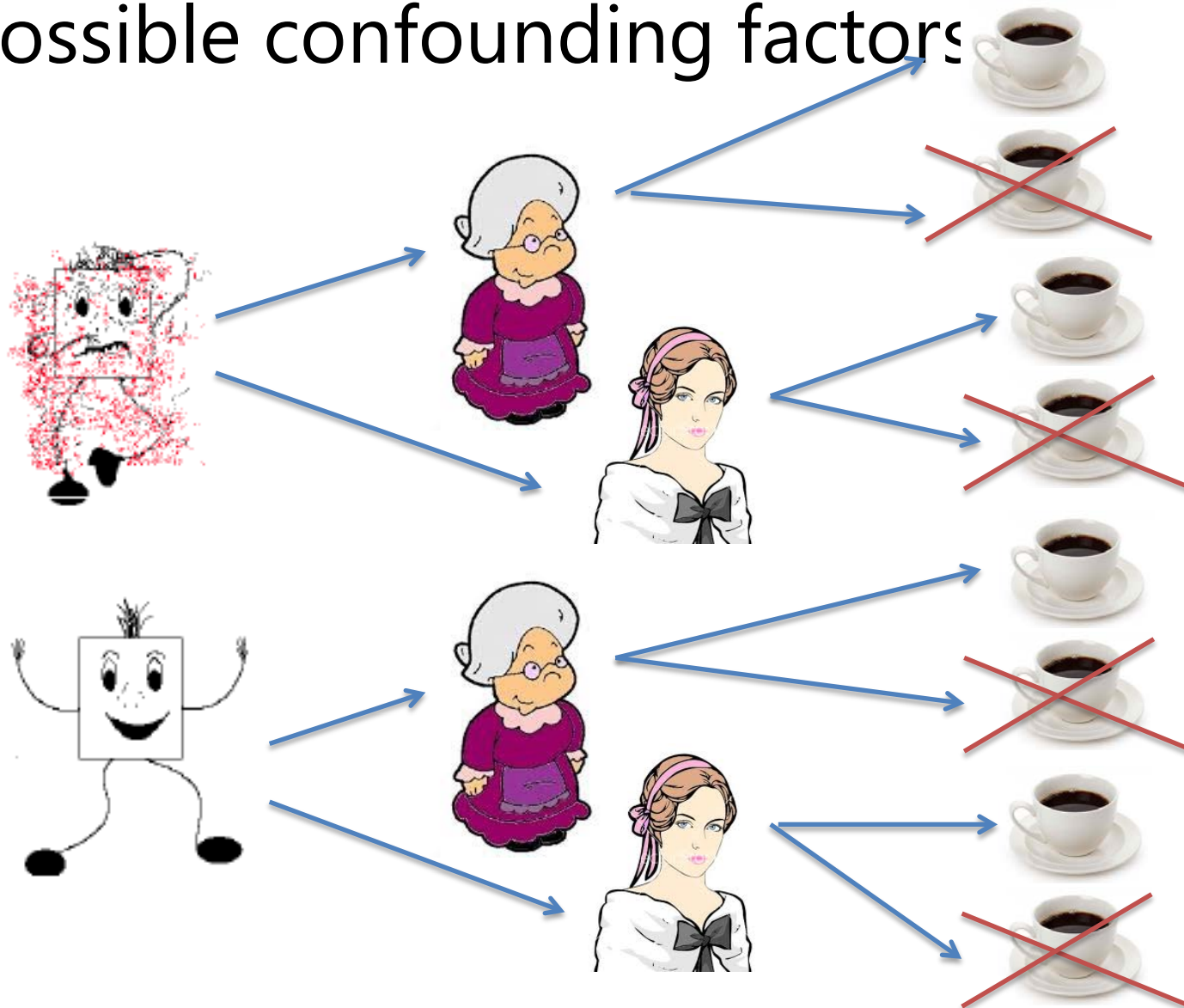
VS



Control

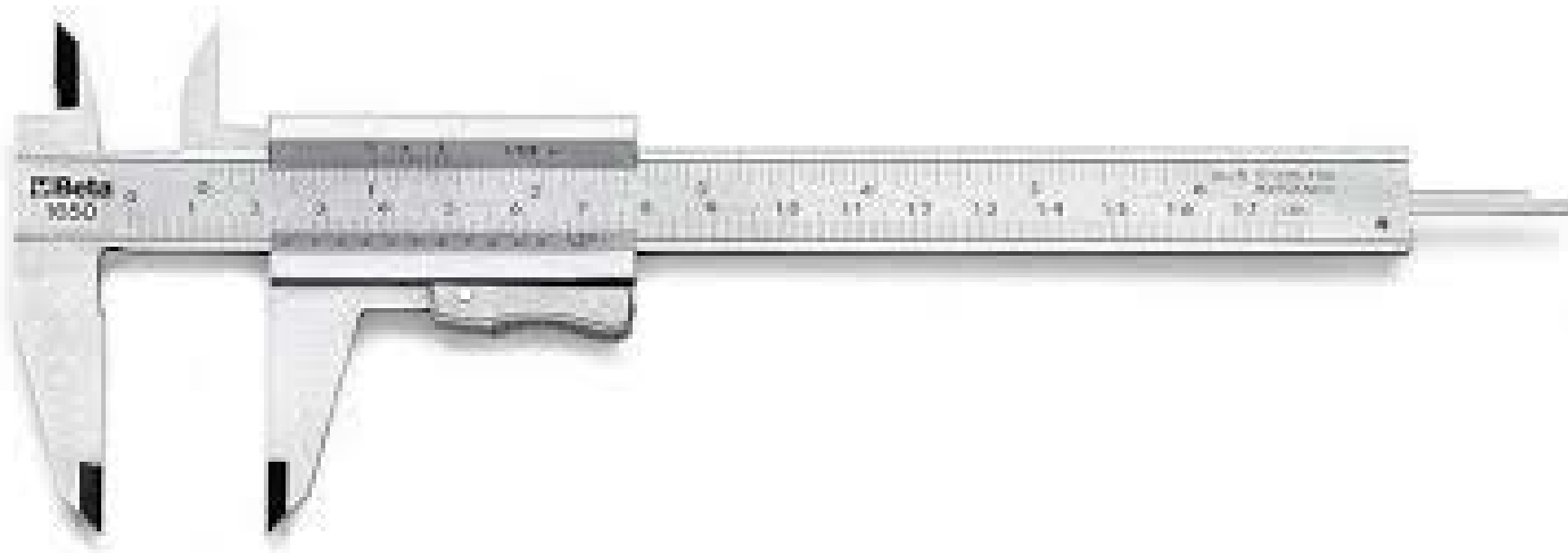
Case-control studies

Limit possible confounding factors

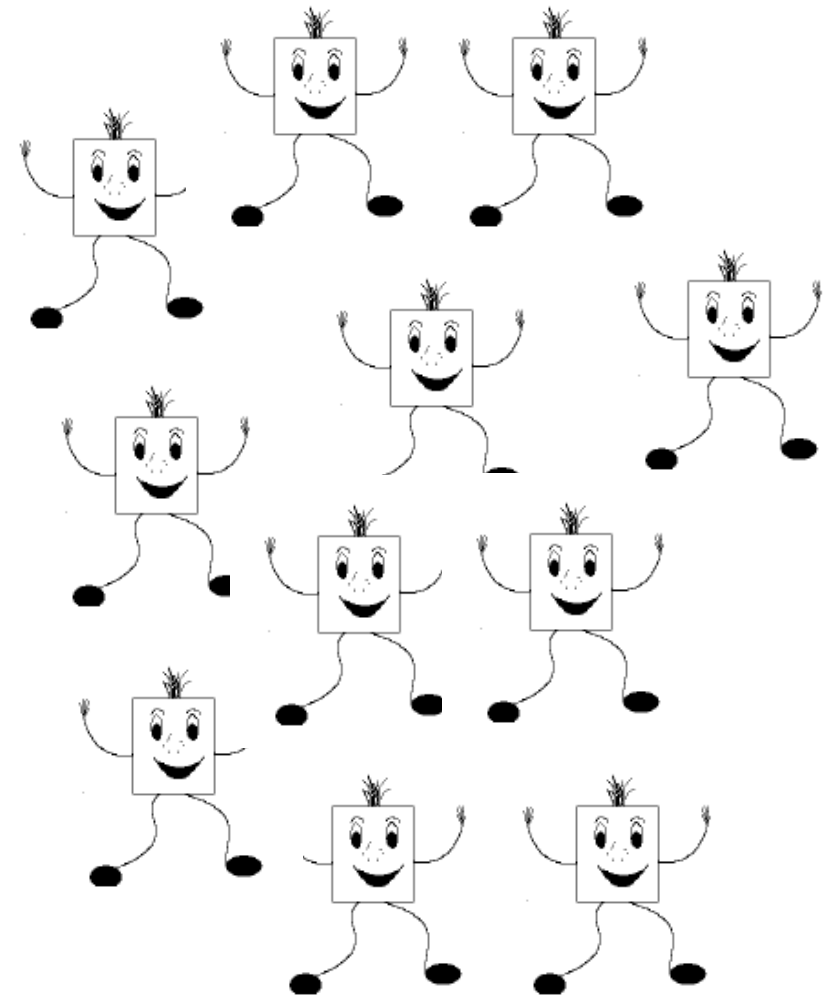
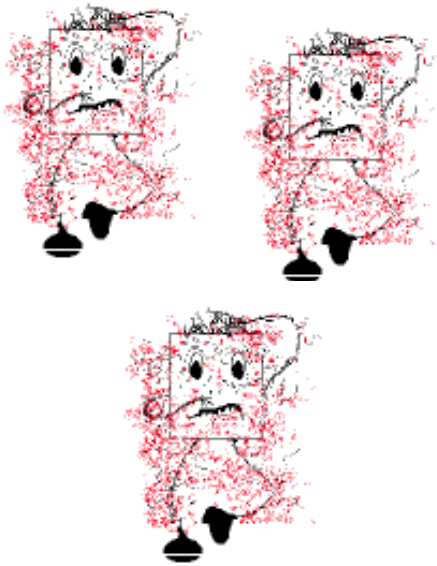


Case-control studies

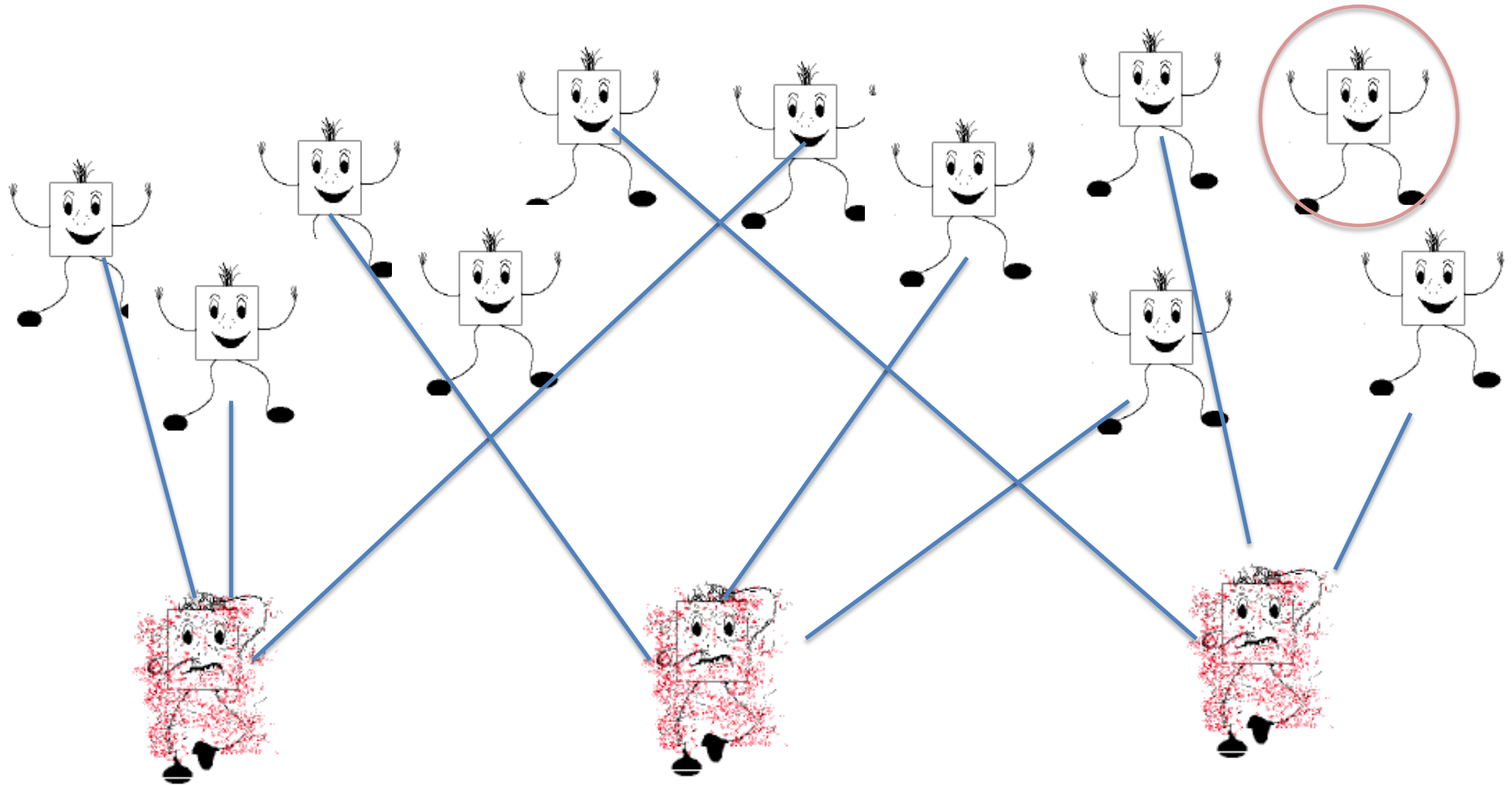
- Exact and caliper matching



Case-control studies



Expected result



Matching

Optir



ers



can

Efficient programming

- Limit number of data steps

```
PROC sql;
    CREATE table Myagbs AS
        SELECT Distinct agb FROM data.fi_medijnen_20145
quit;

data _null_;
    input POSTCODE MONIERS PLAAIS FIF3 NAAMFIF3 ;
run ;

proc SQL;
    create table xar3 as
    SELECT f.fif3, f.naamfif3, oapo_artcd, month(oapo_afldat) as month, year(oapo_afldat) as year ,
    ORDER BY fif3, oapo_artcd, year, month ;
QUIT;

data _null_;
    fif3 (RENAME=(var1=agb var2=fif3 ));
    format var1 var2 ;
    var1-var2 zindex periode verpak ;
run ;

proc sql ;
    create table data.fi_medijnen_fif3
    as select a.agb, a.zindex, a.fif3, a.verpak as aantalstuks, a.djm format=ddmmyy10.,
    from inkoop_fif3 a left join data.fi_knmp as b
    on a.zindex = left(b.knmp_artcd);

quit;

Proc SQL;
    CREATE TABLE XXXAS
    SELECT zindex, djm, fif3, knmp_prccd, knmp_atccd, knmp_inkhoev, SUM(aantalstuks) as aantalstuks
    FROM data.fi_medijnen_fif3
    GROUP BY zindex, djm, fif3, knmp_prccd, knmp_atccd, knmp_inkhoev;

QUIT;

PROC SQL;
    CREATE TABLE Xar4 AS
    SELECT
    FROM xar3 as a FULL OUTER JOIN Yarelto as b ON a.oapo_artcd=b.zindex ;
QUIT;
```

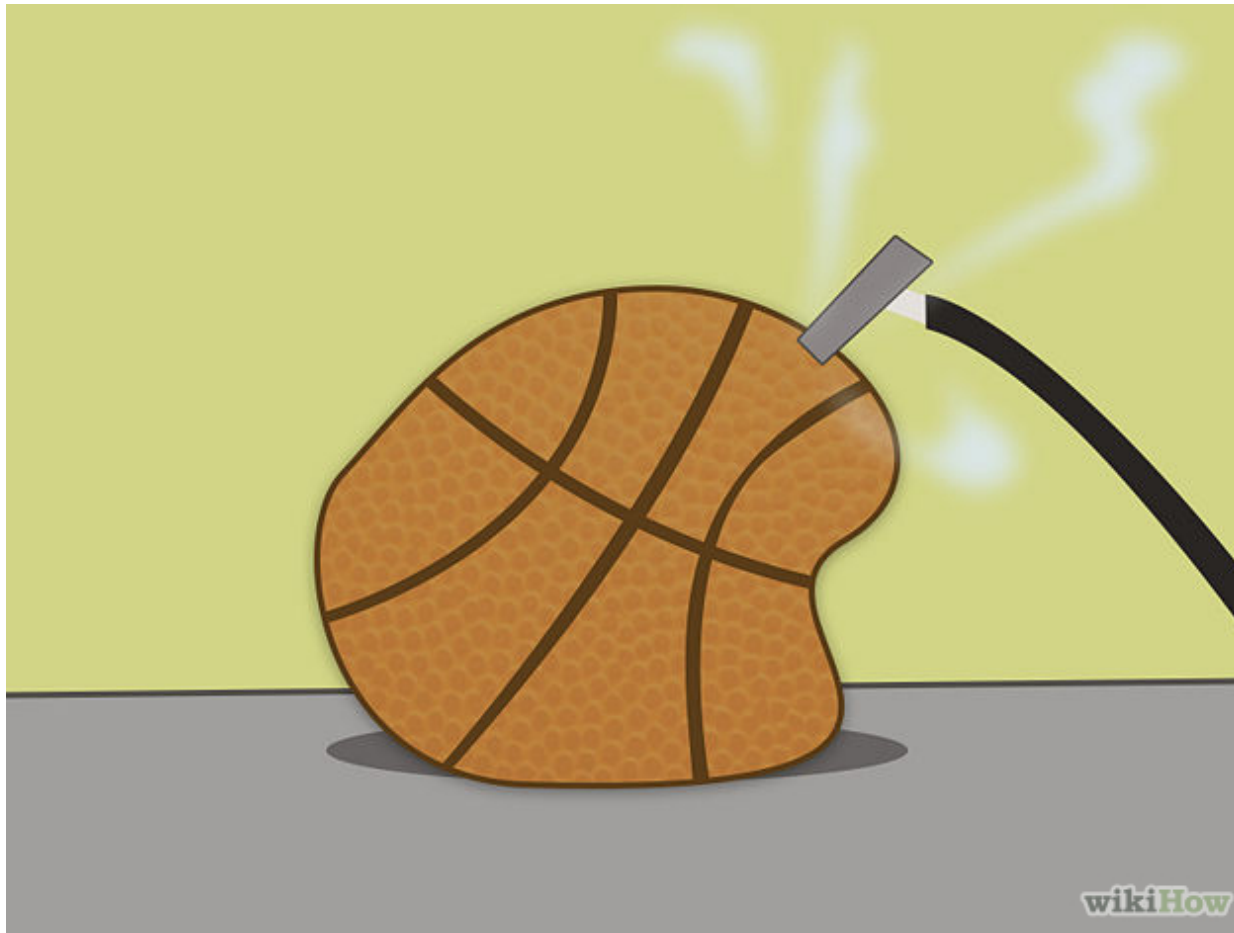
Efficient programming

- Limit sorting



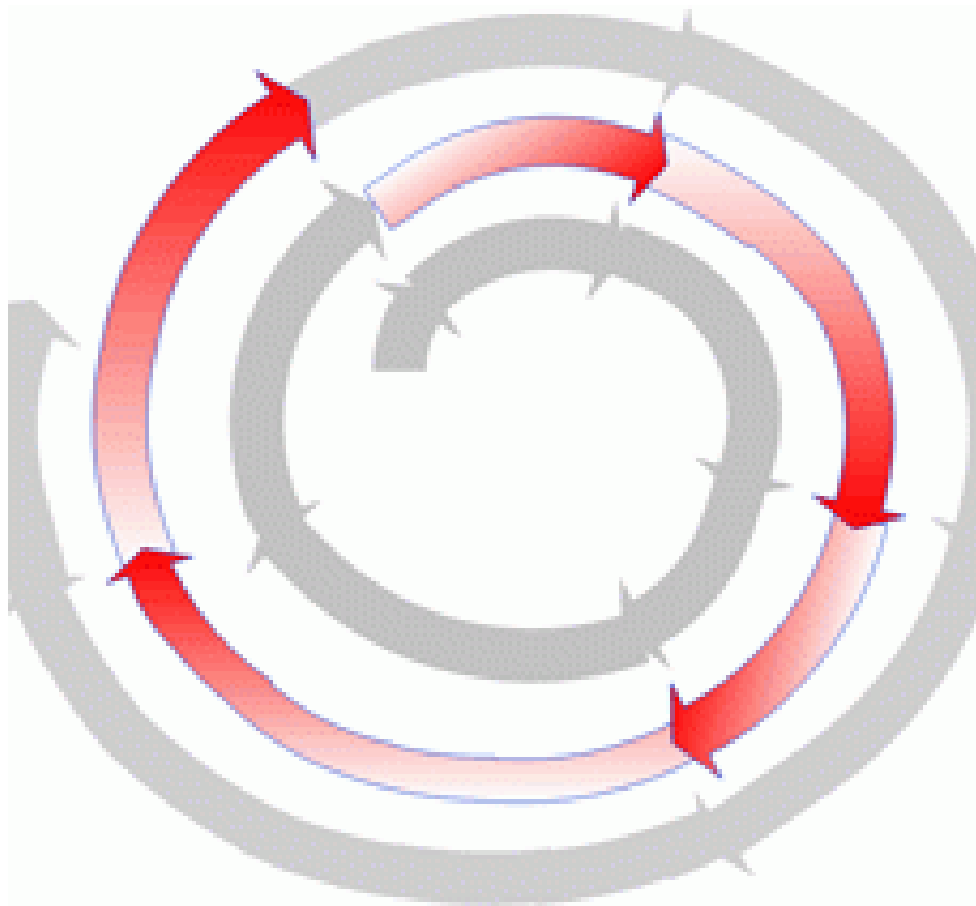
Efficient programming

- Decrease size of datasets



Efficient programming

- Limit number of iterations



Basic technique

1. Construct all possible pairs
2. Add a random number to each combination
3. Sort by control and random number

```
PROC SQL;
```

```
    CREATE _Input AS
```

```
        SELECT a.*, b.*,
```

```
            ranuni(&Seed) as randomnum
```

```
        FROM Cases as a
```

```
            INNER JOIN Controls as b ON ...
```

```
                (all exact and caliper criteria)
```

```
            ORDER BY Pt_control, randomnum;
```

```
QUIT;
```

Basic technique

4. Pick the first case for each control

```
data _Result1;
```

```
  set _Input2;
```

```
  by Pt_control;
```

```
  if first.pt_control then output;
```

```
run;
```

5. Sort by case

```
proc sort data = _Result1;
```

```
  by Pt_case randomnum;
```

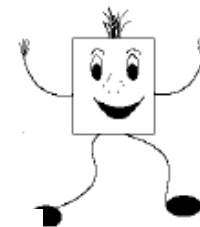
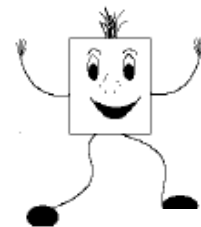
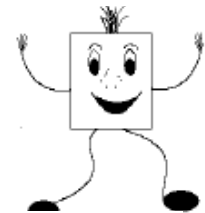
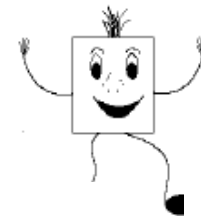
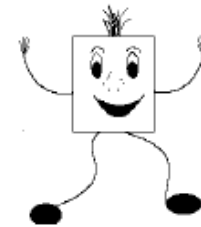
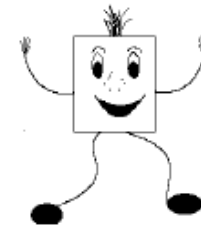
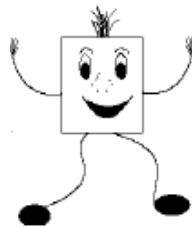
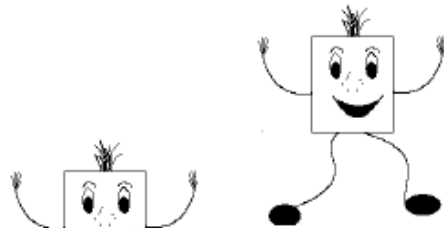
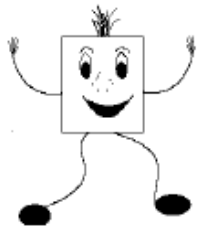
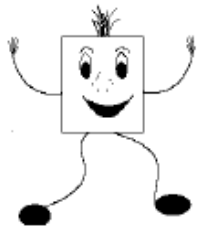
```
run;
```

Basic technique

6. Pick the controls up to the maximum number of controls you desire

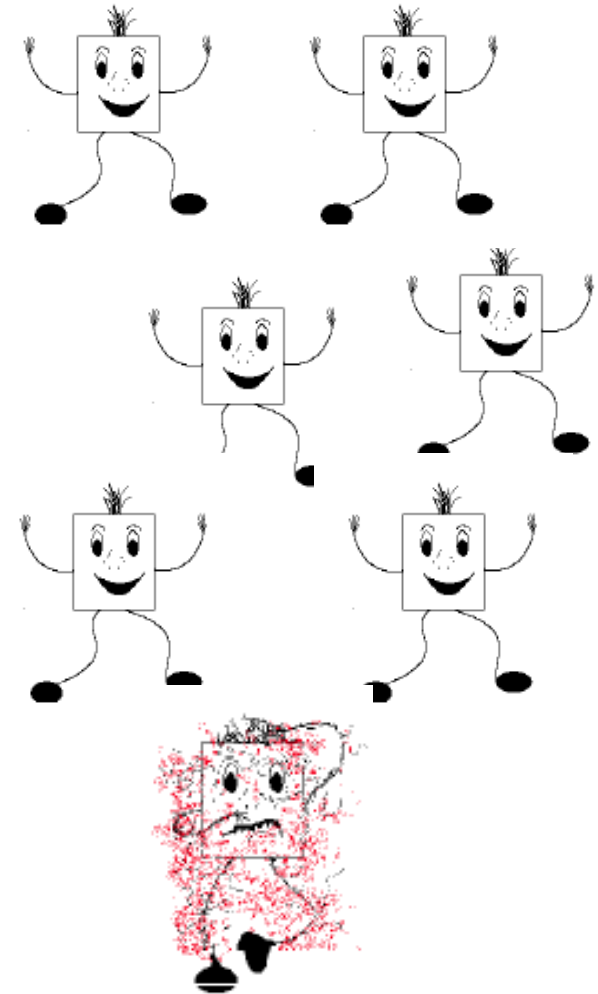
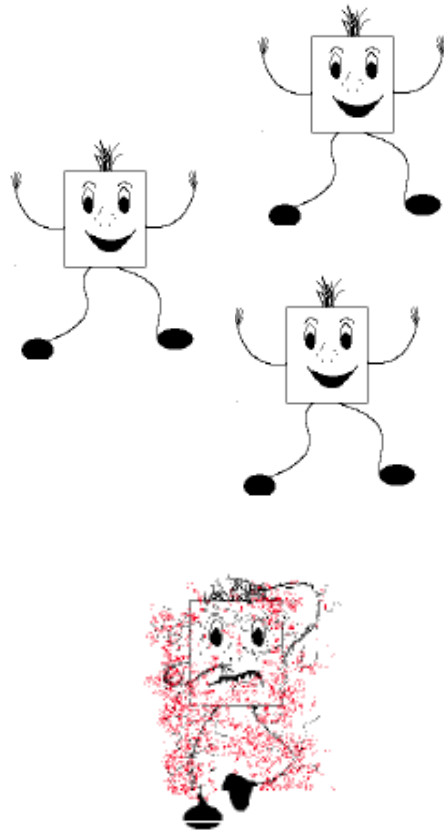
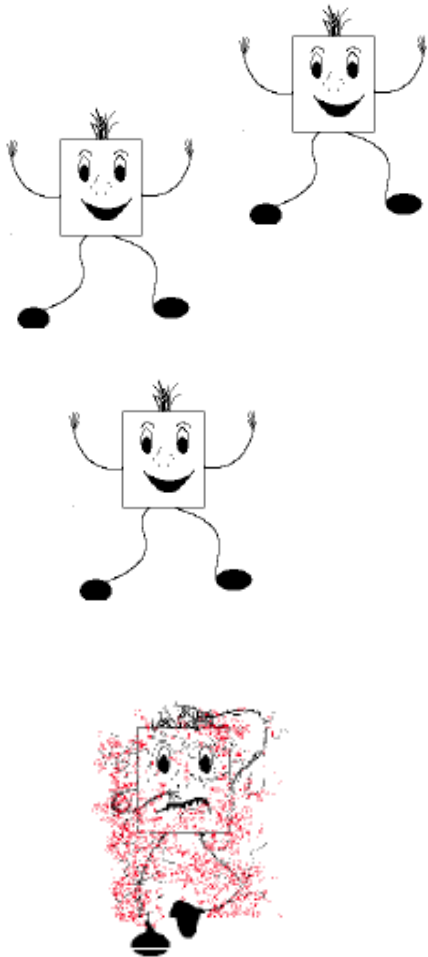
```
data _result2;  
  set _result1;  
  retain Matchno;  
  by Pt_case;  
  if first.pt_case then Matchno=1;  
    ELSE MatchNo=MatchNo+1;  
  if Matchno<=&MaxMatch then output _result2;  
run;
```

Basic technique



By round

Round 3, iteration 2



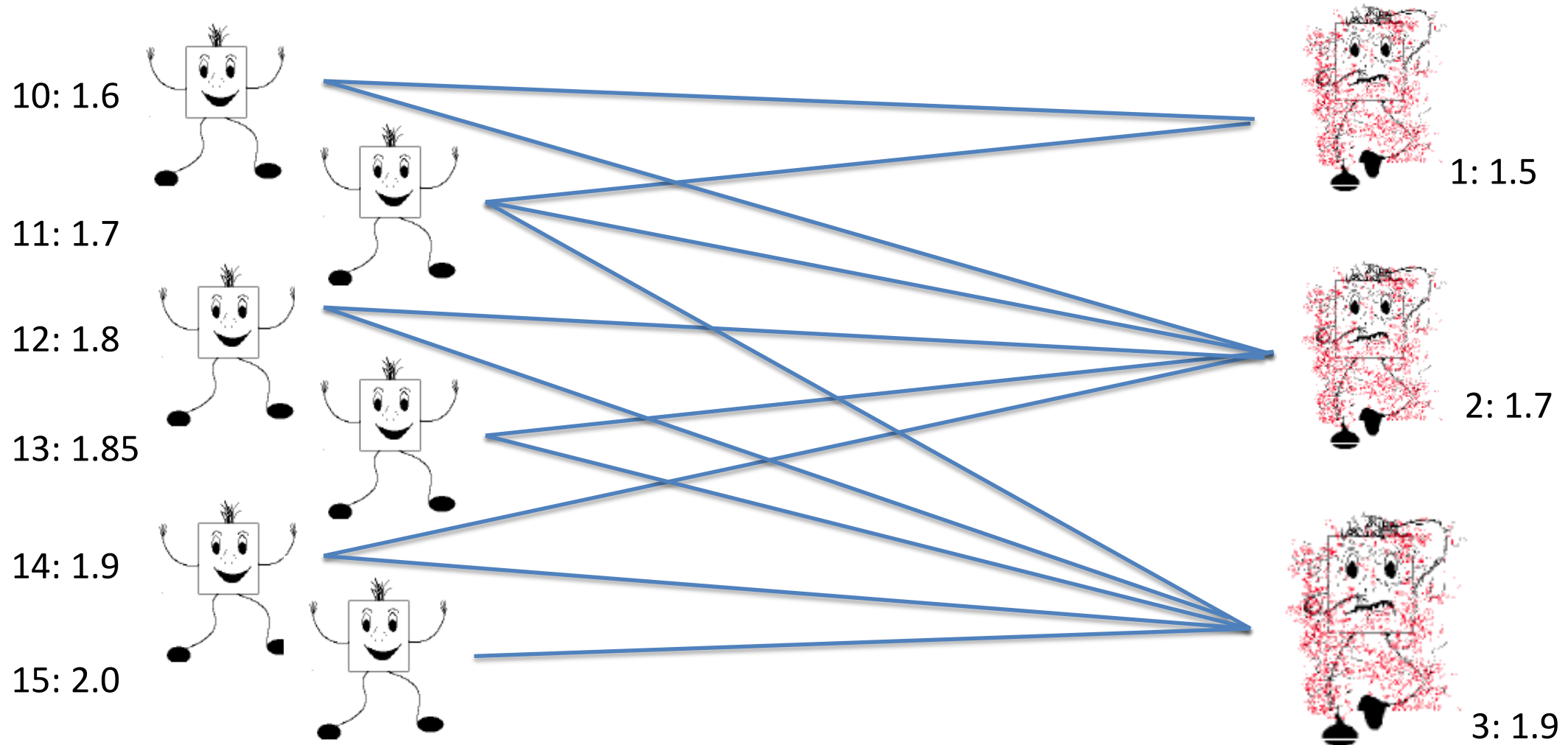
Closest match

Calculate all absolute differences between the case and controls.

Sort by absolute difference and then closest distance.

```
PROC SQL;  
  CREATE _Input AS  
    SELECT a.*, b.* ,  
           ranuni(&Seed) as randomnum,  
           Abs(CaseVal-RefVal) as AbsDif  
    FROM Cases as a  
      INNER JOIN Controls as b ON ...  
        (all exact and caliper criteria)  
    ORDER BY Pt_control, AbsDif, randomnum;  
QUIT;
```

Closest match – plaatje omdraaien



Tests

Match 1 control by round	Distance Rank Priority	Least number of matches priority	Run Time	Total number of matched cases	Total number of matched Pairs	Number of iterations
No	No	No	1 min, 4 sec	1670	8025	25
No	Yes	No	1 min, 0 sec	1924	8781	6
No	No	Yes	1 min, 19 sec	1715	9831	7
No	Yes	Yes	1 min, 57 sec	1685	9828	9
Yes	No	No	4 min, 41 sec	2223	8441	74
Yes	Yes	No	4 min, 37 sec	2290	8859	32
Yes	No	Yes	5 min, 29 sec	2338	9190	39
Yes	Yes	Yes	9 min, 37 sec	2308	9171	45

2500 cases, 25000 possible matches, maximum of 8 controls per case

Least number of matches method

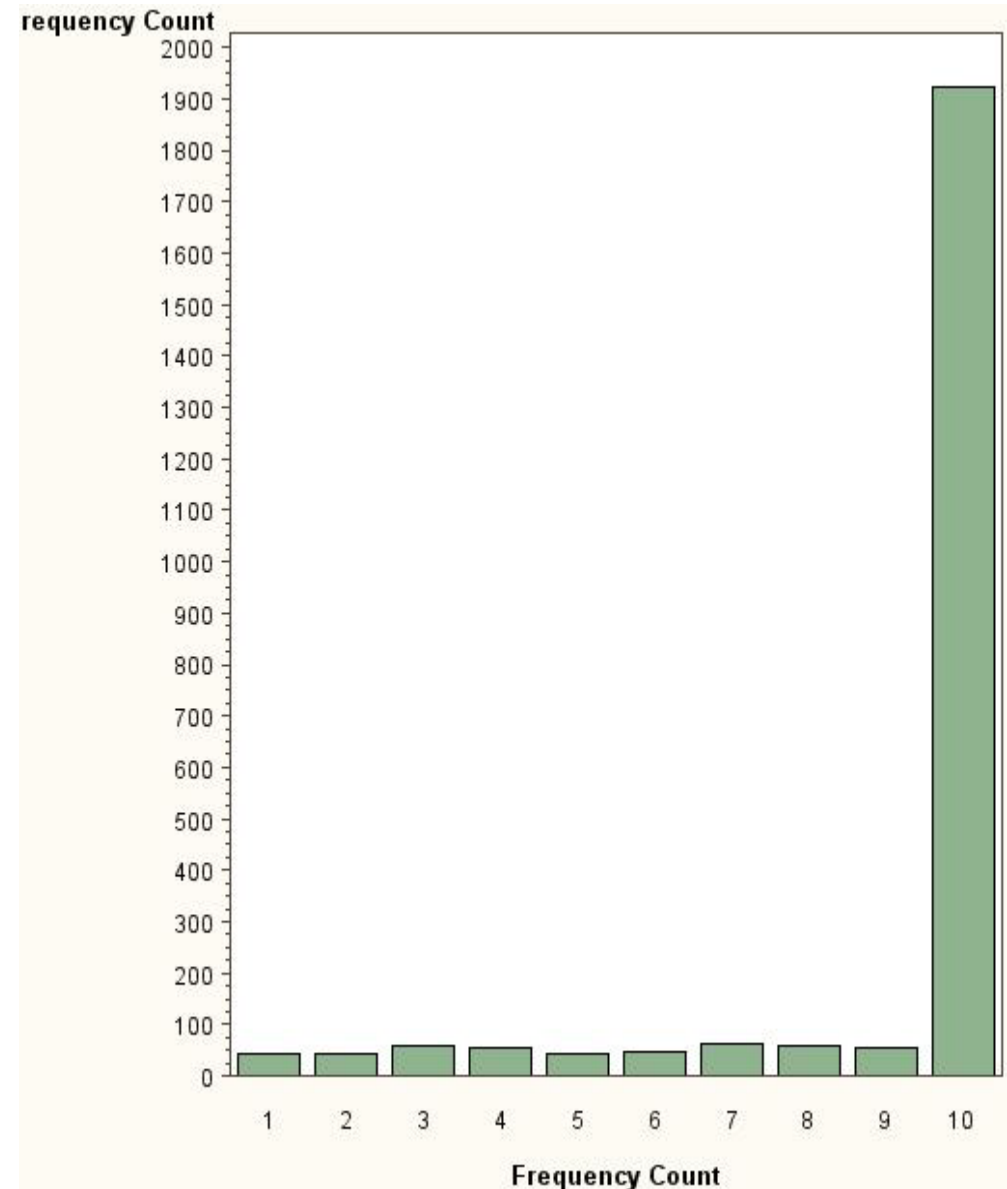
```
Proc SQL;  
  Create table _input2 as  
    select *,   ranuni(&Seed) AS randomnum,  
              Count(*) as Nmatches  
    from _InputMe  
    group by pt_case  
    order by pt_control, Nmatches, randomnum ;  
Quit;  
  
data _Result1;  
  set _Input2;  
  by Pt_control;  
  if first.pt_control then output;  
run;
```

Least number of matches method (2)

```
Proc SQL;
  Create table _input as
  select *,  ranuni(Seed) AS randomnum,
  case when (Count(*) <= 10) Then count(*)
        when (Count(*) <= 100) Then ROUND(count(*), 10.)
        when (count(*) <= 1000) then round(Count(*), 100.)
        when (count(*) <= 10000) then
            round(count(*), 1000.)
        else 10000 end as Nmatches
  from _InputMe
  group by pt_case
  order by pt_control, Nmatches, AbsDif, randomnum ;
Quit;
```

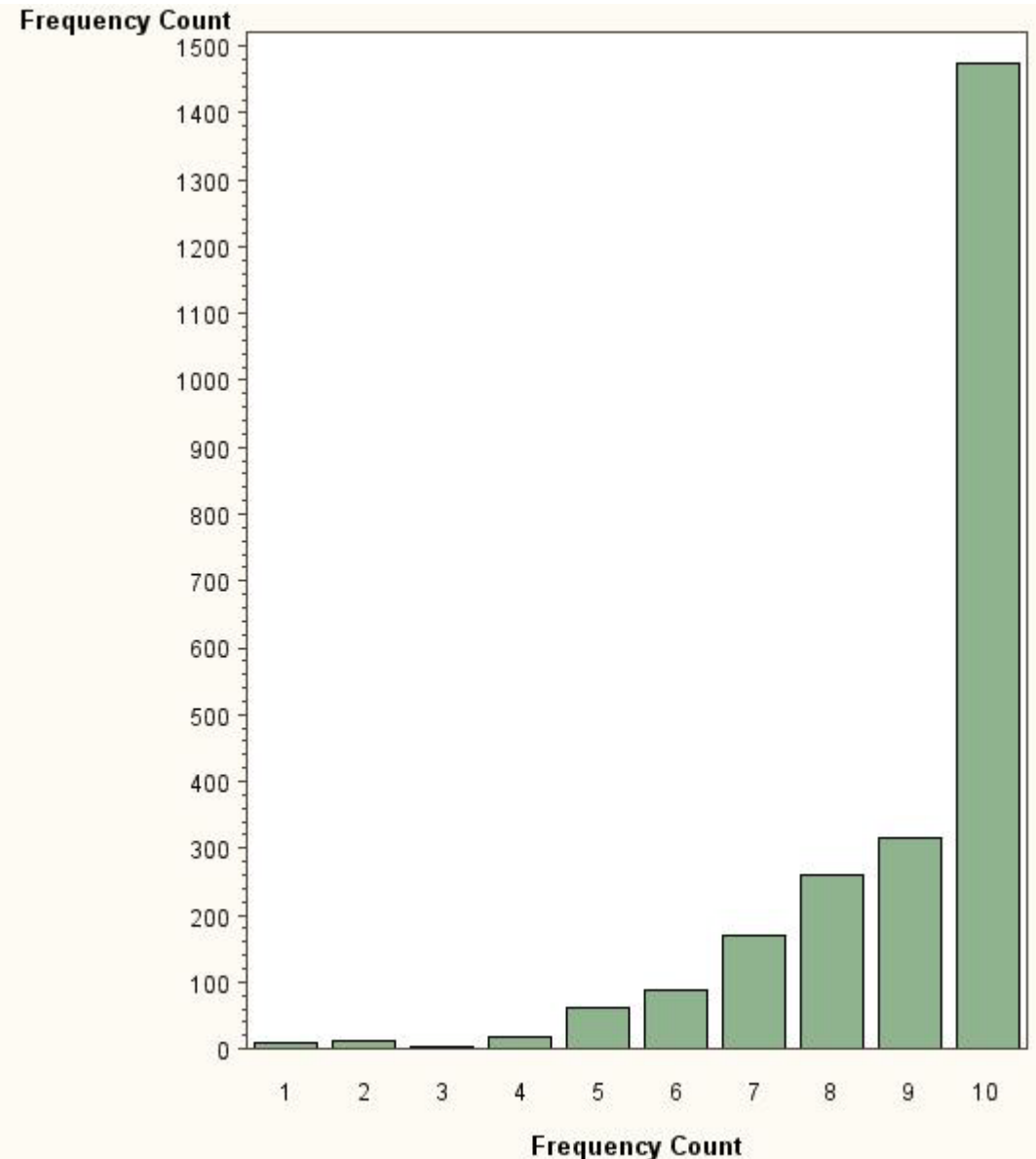
Example

- 2415 cases
- 22140 possible matches
- Match on
 - gender
 - age range (+/- 2.5 year)
- Max 10 matches per case
- No replacement
- All at once
- 7 rounds
- 47 seconds



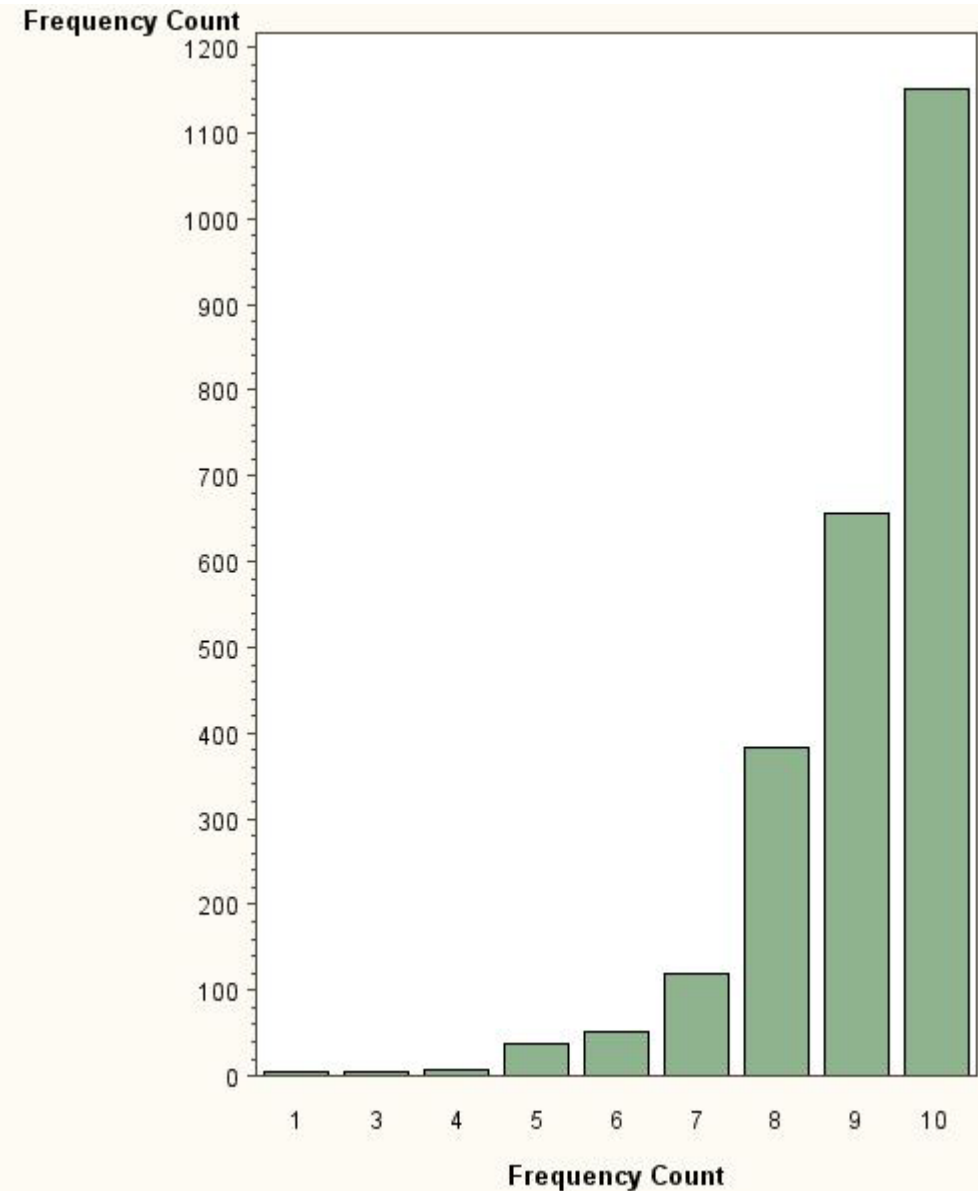
Example

- 2415 cases
- 22140 possible matches
- Match on
 - gender
 - age range (+/- 2.5 year)
- Max 10 matches per case
- No replacement
- Round by round, 10% saturation
- 16 rounds
- 1 min 50 seconds



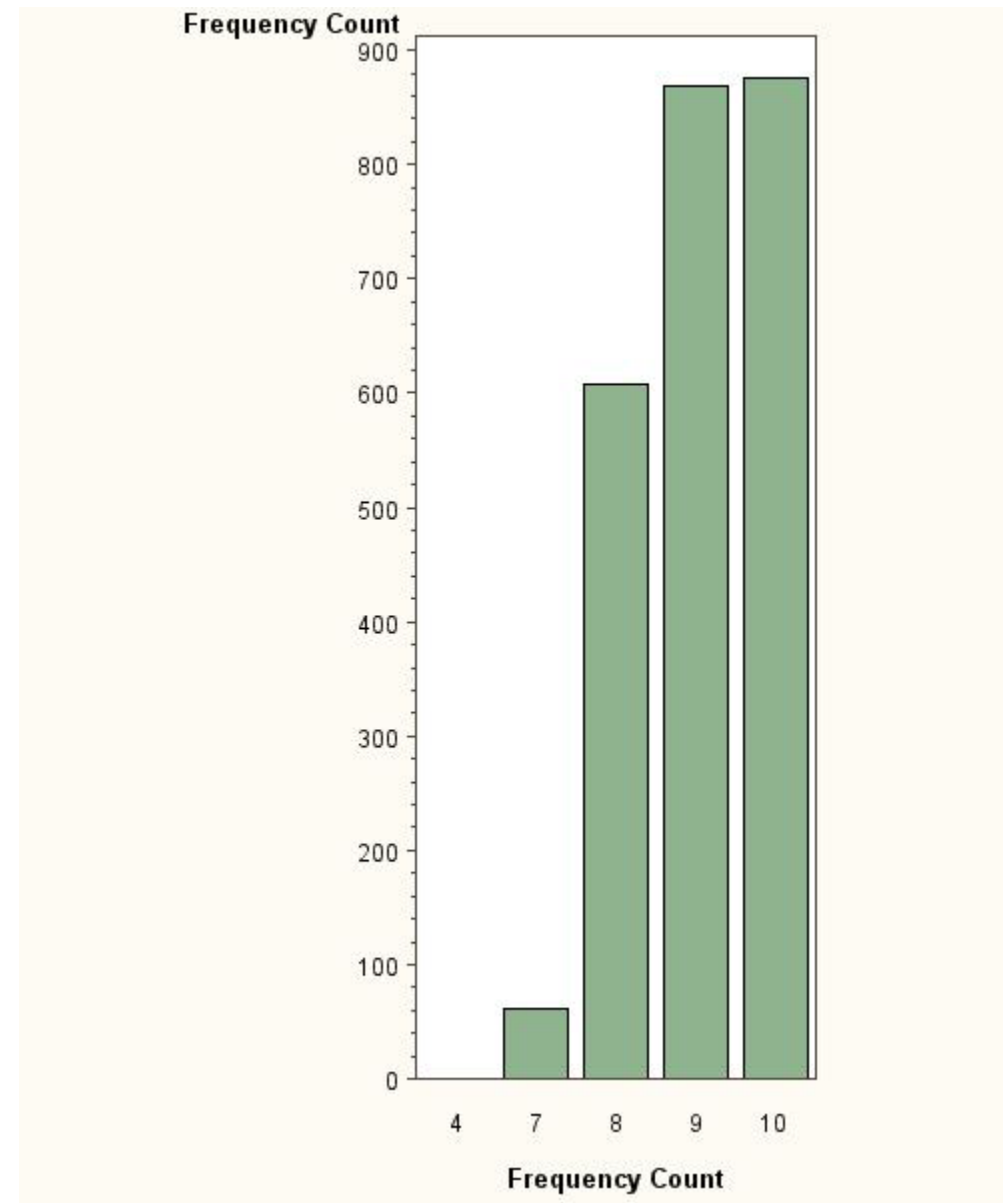
Example

- 2415 cases
- 22140 possible matches
- Match on
 - gender
 - age range (+/- 2.5 year)
- Max 10 matches per case
- No replacement
- Round by round, 60% saturation
- 19 rounds
- 1 min 58 seconds



Example

- 2415 cases
- 22140 possible matches
- Match on
 - gender
 - age range (+/- 2.5 year)
- Max 10 matches per case
- No replacement
- Round by round, full saturation
- 41 rounds
- 2 min 21 seconds



Conclusions

- Efficient and fast
- Useful with Big data
- Optimal
- Can handle any combination of exact and caliper variables
- Can handle any number of matches to controls
- Final distribution can be examined and best options can be chosen

- Questions?

