Decomposition Methods in the Social Sciences

GESIS Training Course

January 29 – February 1, 2024, Cologne

Johannes Giesecke (Humboldt University Berlin)

Ben Jann (University of Bern)

6. Reweighting

Beyond the mean

- The discussed Oaxaca-Blinder procedures and their extensions to non-linear models focus on the decomposition of differences in the expected value (mean) of an outcome variable.
- In many cases, however, one is interested in other distributional statistics, say the Gini coefficient or the D9/D1 quantile ratio, or even in whole distributions (density curves, Lorenz curves).
- The basic setup is the same; an estimate of $F_{Y^g|G\neq g}$ is needed to be able to compute a decomposition such as

$$\begin{aligned}\Delta^\nu &= \nu(F_{Y|G=0}) - \nu(F_{Y|G=1}) \\ &= \{\nu(F_{Y|G=0}) - \nu(F_{Y^0|G=1})\} + \{\nu(F_{Y^0|G=1}) - \nu(F_{Y|G=1})\} \\ &= \Delta^\nu_X + \Delta^\nu_S\end{aligned}$$

where

$$F_{Y^g|G\neq g}(y) = \int F_{Y|X,G=g}(y|x) f_{X|G\neq g}(x)\, dx$$

Beyond the mean

- Several approaches have been proposed in the literature:
  - Estimating $F_{Y^g|G\neq g}$ by reweighting (DiNardo et al. 1996).
  - Estimating $\nu(F_{Y^g|G\neq g})$ via recentered influence function regression (Firpo et al. 2007, 2009)
  - Imputing values for Y^g in group $G \neq g$
    - based on regression residuals (Juhn et al. 1993)
    - based on quantile regression (Machado and Mata 2005, Melly 2005, 2006)
  - Estimating $F_{Y^g|G\neq g}$ by distribution regression (Chernozhukov et al. 2013)
- We will now look at reweighting.

Contents

General procedure

- DiNardo, Fortin, and Lemieux (DFL) (1996) proposed a simple reweighting procedure to obtain an estimate of $F_{Y^g|G\neq g}$ or any functional $\nu()$ of $F_{Y^g|G\neq g}$.
- Let $F_{Y|X^g}$ stand for $F_{Y|X,G=g}$ and F_{X^g} for $F_{X|G=1}$. Multiplying

$$F_{Y^0|G=1}(y) = \int F_{Y|X^0}(y|x)\, dF_{X^1}(x)$$

by $dF_{X^0}/\, dF_{X^0}$ leads to

$$\begin{aligned} F_{Y^0|G=1}(y) &= \int F_{Y|X^0}(y|x) \frac{dF_{X^1}(x)}{dF_{X^0}(x)}\, dF_{X^0}(x) \\ &= \int F_{Y|X^0}(y|x)\Psi(x)\, dF_{X^0}(x) \end{aligned}$$

where

$$\Psi(x) = \frac{dF_{X^1}(x)}{dF_{X^0}(x)} = \frac{\Pr(x|G=1)}{\Pr(x|G=0)}$$

General procedure

- Based on Bayes' rule $\Pr(A|B) = \Pr(B|A)\Pr(A)/\Pr(B)$ we can rewrite $\Pr(X|G = g)$ as

$$\Pr(X|G = g) = \frac{\Pr(G = g|X)\Pr(X)}{\Pr(G = g)}$$

such that

$$\Psi(X) = \frac{\Pr(X|G = 1)}{\Pr(X|G = 0)} = \frac{\Pr(G = 1|X)\Pr(x)/\Pr(G = 1)}{\Pr(G = 0|X)\Pr(X)/\Pr(G = 0)}$$
$$= \frac{\Pr(G = 1|X)/\Pr(G = 1)}{\Pr(G = 0|X)/\Pr(G = 0)} = \frac{\Pr(G = 1|X)}{\Pr(G = 0|X)} \times \frac{\Pr(G = 0)}{\Pr(G = 1)}$$

- $\Psi(X)$ is easy to estimate.
- An estimate for $\Pr(G = 1) = 1 - \Pr(G = 0)$ is simply the proportion of group 1 in the sample.
- $\Pr(G = 1|X) = 1 - \Pr(G = 0|X)$, the "propensity score", can be estimated by regressing G on X using logit or similar.

General procedure

- As soon as we have $\widehat{\Psi}(X)$, the counterfactual distribution $F_{Y^0|G=1}$, or any functional of the distribution, can be estimated from the $G = 0$ sample by weighting the observations by $\widehat{\Psi}(X)$.
- In this way we can easily get
  - a counterfactual kernel density estimate
  - an estimate of the counterfactual mean
  - an estimate of the counterfactual variance
  - estimates of counterfactual quantiles
  - an estimate of the counterfactual D9/D1 ratio
  - an estimate of the counterfactual Gini
  - ...
- A commands called `dfl` exists for Stata, but is limited to comparing kernel density estimates.
- In practice, therefore, one has to compute $\widehat{\Psi}(X)$ and the resulting decomposition manually (which is fairly easy to do).

How to estimate the weights

- As said, the propensity score $\Pr(G = 1|X)$ can be estimated by regressing G on X using logit or probit or similar.
- The model specification should be flexible enough to capture possible non-linearities and interaction effects. If data permits, you can also try nonparametric estimators such as `npregress` (official Stata) or `krls` (Hainmueller and Hazlett 2014).
- Furthermore, note that $\frac{\Pr(G=0)}{\Pr(G=1)}$ in $\Psi(X)$ does not depend on X. It is the same for all observations and can be omitted from the weights.
- This also clarifies that weighting by $\Psi(X)$ is equivalent to inverse probability weighting (IPW) known in the causal inference literature.
- That is, you can also obtain the weights by other causal inference procedures such as matching (e.g. `kmatch` by Jann 2017) or entropy balancing (`ebalance` by Hainmueller 2012, `ebalfit` by Jann 2021).

Limitations

- If the sample is small, flexible estimation of the propensity score will not be possible and the performance of the reweighting procedure may be poor.
- A related problem is that in small samples common support problems are likely (observations for which the estimated propensity score is close to zero or one); this can make the estimates unreliably (large variance in the weights).
- The effect of the weights is that they balance X between the groups, i.e. the distribution of X in one group is adjusted to the distribution of X in the other group. If the groups are very different with respect to X, this is hard to achieve. One consequence is again that the weights will have a large variance (making estimates imprecise). Furthermore, the desired balancing of X may be very poor in such cases.
- It is thus always a good idea to check the balancing, like you would do in a matching analysis.

Data preparation

```
. use gsoep-extract, clear
(Example data based on the German Socio-Economic Panel)

. keep if wave==2015
(29,970 observations deleted)

. keep if inrange(age, 25, 55)
(5,671 observations deleted)

. generate lnwage = ln(wage)
(1,709 missing values generated)

. generate expft2 = expft^2
(35 missing values generated)

. svyset psu [pw=weight], strata(strata)

Sampling weights: weight
             VCE: linearized
     Single unit: missing
        Strata 1: strata
 Sampling unit 1: psu
           FPC 1: <zero>

. summarize wage lnwage yeduc expft expft2 public

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
        wage |      5,600    17.57278    9.858855       3.03     121.42
      lnwage |      5,600    2.736721    .5062968   1.108563   4.799255
       yeduc |      7,121    12.28823    2.783974          7         18
       expft |      7,274    11.63359    9.556508          0       39.5
      expft2 |      7,274    226.6548    293.3739          0    1560.25
-------------+---------------------------------------------------------
      public |      5,770    .2353553    .4242574          0          1

. drop if missing(lnwage, yeduc, expft, public) // remove unused observation
(1,851 observations deleted)
```

Observed statistics in private sector

```
. sum lnwage if public==0 [aw=weight], detail
                           lnwage
-------------------------------------------------------------
      Percentiles      Smallest
 1%     1.510722       1.108563
 5%     1.950187       1.115142
10%     2.136531       1.115142       Obs               4,184
25%     2.388763        1.12493       Sum of wgt.   9,231,939

50%      2.72589                      Mean           2.732109
                        Largest       Std. dev.      .5008582
75%     3.065258       4.659848
90%     3.378952       4.766694       Variance       .2508589
95%     3.570096       4.781641       Skewness       .0484253
99%     3.874321       4.799255       Kurtosis       3.258202

. local prAVG  = r(mean)

. local prD9D1 = r(p90)-r(p10)

. local prD9D5 = r(p90)-r(p50)

. local prD5D1 = r(p50)-r(p10)

. local prVar  = r(Var)

. display "prD9D1 =" %7.0g `prD9D1' " (ratio =" %7.0g exp(`prD9D1') ")" _n ///
>         "prD9D5 =" %7.0g `prD9D5' " (ratio =" %7.0g exp(`prD9D5') ")" _n ///
>         "prD5D1 =" %7.0g `prD5D1' " (ratio =" %7.0g exp(`prD5D1') ")"
prD9D1 = 1.2424 (ratio =  3.464)
prD9D5 = .65306 (ratio = 1.9214)
prD5D1 = .58936 (ratio = 1.8028)
```

Observed statistics in public sector

```
. sum lnwage if public==1 [aw=weight], detail
                            lnwage
-------------------------------------------------------------
      Percentiles      Smallest
 1%      1.413423      1.115142
 5%      2.032088      1.181727
10%      2.302585       1.18479       Obs               1,274
25%       2.65956      1.217876       Sum of wgt.   2,914,832

50%      2.901422                     Mean           2.866068
                        Largest       Std. dev.      .4438737
75%      3.145875      4.163404
90%      3.363496      4.239455       Variance       .1970238
95%      3.526066       4.24219       Skewness      -.8049336
99%      3.697839      4.356068       Kurtosis        4.64433
. local puAVG  = r(mean)
. local puD9D1 = r(p90)-r(p10)
. local puD9D5 = r(p90)-r(p50)
. local puD5D1 = r(p50)-r(p10)
. local puVar  = r(Var)
. display "puD9D1 =" %7.0g `puD9D1' " (ratio =" %7.0g exp(`puD9D1') ")" _n ///
>         "puD9D5 =" %7.0g `puD9D5' " (ratio =" %7.0g exp(`puD9D5') ")" _n ///
>         "puD5D1 =" %7.0g `puD5D1' " (ratio =" %7.0g exp(`puD5D1') ")"
puD9D1 = 1.0609 (ratio =  2.889)
puD9D5 = .46207 (ratio = 1.5874)
puD5D1 = .59884 (ratio =   1.82)
```

Private–public differences in observed statistics

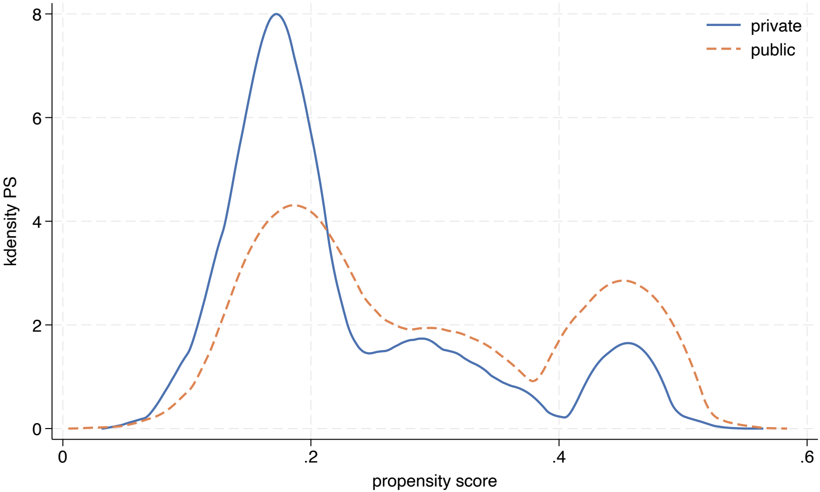
```
. display `prAVG'  - `puAVG'
-.13395921

. display `prD9D1' - `puD9D1'
.18151069

. display `prD9D5' - `puD9D5'
.19098759

. display `prD5D1' - `puD5D1'
-.0094769

. display `prVar'  - `puVar'
.0538351
```

Propensity-score model

```
. svy: logit public c.yeduc##c.expft##c.expft, vsquish
(running logit on estimation sample)

Survey: Logistic regression

Number of strata =     15                         Number of obs   =      5,458
Number of PSUs   = 2,036                          Population size = 12,146,771
                                                  Design df       =      2,021
                                                  F(5, 2017)      =      20.67
                                                  Prob > F        =     0.0000

------------------------------------------------------------------------------------------
                         |             Linearized
                  public | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------------------+----------------------------------------------------------------
                   yeduc |   .1950069    .043749     4.46   0.000      .109209    .2808047
                   expft |  -.0366746   .0925297    -0.40   0.692    -.2181382    .1447891
          c.yeduc#c.expft|   .0014953   .0070145     0.21   0.831    -.0122612    .0152517
          c.expft#c.expft|   .0009309   .0029449     0.32   0.752    -.0048445    .0067064
 c.yeduc#c.expft#c.expft |  -.0000218   .0002303    -0.09   0.925    -.0004734    .0004298
                   _cons |  -3.679054   .5935616    -6.20   0.000     -4.84311   -2.514997
------------------------------------------------------------------------------------------

. predict PS if e(sample), pr
```

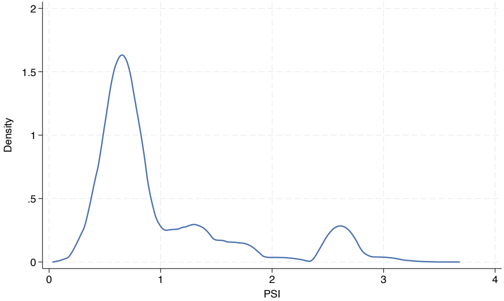
Distribution of propensity-score by sector

```
. quietly two (kdens PS if public==0 [pw=weight]) (kdens PS if public==1 [pw=weight]), ///
>     xti("propensity score") legend(order(1 "private" 2 "public"))
```



Generate the weights from the propensity score

```
. summarize public [aw=weight]

    Variable |     Obs      Weight        Mean   Std. dev.       Min        Max
-------------+-------------------------------------------------------------------
      public |   5,458  12146770.6    .2399677    .4271026         0          1

. local P_public = r(mean)

. generate PSI = (PS / `P_public') / ((1-PS) / (1 - `P_public')) if public==0
(1,274 missing values generated)

. replace PSI = 1 if public==1
(1,274 real changes made)

. summarize PSI [aw=weight] if public==0

    Variable |     Obs      Weight        Mean   Std. dev.       Min        Max
-------------+-------------------------------------------------------------------
         PSI |   4,184   9231938.6    1.003234    .6637265   .2511509   3.463624

. kdens PSI [pw=weight] if public==0
(bandwidth = .22094267)
```



Balancing: Raw mean differences in covariates

```
. tabstat PS yeduc expft expft2 [aw=weight], by(public) nototal ///
>     stat(mean var p10 p50 p90) columns(statistics)

Summary for variables: PS yeduc expft expft2
Group variable: public (public service)

public |      Mean  Variance       p10       p50       p90
-------+--------------------------------------------------
    no |  .2246685  .0101674  .1339219  .1865309  .4469216
       |  12.42709  6.958165        10      11.5        18
       |  14.32145  101.5838      2.25      12.5     29.25
       |  306.6634  116024.7    5.0625    156.25  855.5625
-------+--------------------------------------------------
   yes |  .2884234  .0140197  .1597596   .259925  .4545366
       |  14.07113  8.474033      10.5      13.5        18
       |  13.46011  98.61003       1.5      11.5      28.5
       |  279.7071  105614.7      2.25    132.25    812.25
----------------------------------------------------------
```

Balancing: Mean differences in reweighted sample

```
. tabstat PS yeduc expft expft2 [aw=PSI*weight], by(public) ///
>     nototal stat(mean var p10 p50 p90) columns(statistics)

Summary for variables: PS yeduc expft expft2
Group variable: public (public service)

public |      Mean  Variance       p10       p50       p90
-------+--------------------------------------------------
    no |  .2907174    .01476  .1520262  .2803326  .4550497
       |  14.09968  8.976983      10.5        14        18
       |  13.49386  98.65649         2      11.5      28.5
       |  280.7173  107385.7         4    132.25    812.25
-------+--------------------------------------------------
   yes |  .2884234  .0140197  .1597596   .259925  .4545366
       |  14.07113  8.474033      10.5      13.5        18
       |  13.46011  98.61003       1.5      11.5      28.5
       |  279.7071  105614.7      2.25    132.25    812.25
----------------------------------------------------------
```

Counterfactual statistics in reweighted private sector

```
. sum lnwage [aw=PSI*weight] if public==0, detail
                           lnwage
-------------------------------------------------------------
      Percentiles      Smallest
 1%     1.581038       1.108563
 5%     2.036012       1.115142
10%     2.204972       1.115142       Obs               4,184
25%     2.481568        1.12493       Sum of wgt.   9,261,797

50%     2.852439                      Mean            2.85921
                        Largest       Std. dev.      .5175626
75%     3.236323       4.659848
90%     3.530763       4.766694       Variance       .2678711
95%     3.688379       4.781641       Skewness      -.0480414
99%     3.985088       4.799255       Kurtosis       2.992935
. local cAVG  = r(mean)
. local cD9D1 = r(p90)-r(p10)
. local cD9D5 = r(p90)-r(p50)
. local cD5D1 = r(p50)-r(p10)
. local cVar  = r(Var)
. display "cD9D1 =" %7.0g `cD9D1' " (ratio =" %7.0g exp(`cD9D1') ")" _n ///
>         "cD9D5 =" %7.0g `cD9D5' " (ratio =" %7.0g exp(`cD9D5') ")" _n ///
>         "cD5D1 =" %7.0g `cD5D1' " (ratio =" %7.0g exp(`cD5D1') ")"
cD9D1 = 1.3258 (ratio = 3.7652)
cD9D5 = .67832 (ratio = 1.9706)
cD5D1 = .64747 (ratio = 1.9107)
```

Results of decomposition

```
. foreach s in AVG D9D1 D9D5 D5D1 Var {
  2.     display %6s "`s': " ///
>         "total difference = " %9.0g `pr`s'' - `pu`s'' ///
>            "    explained = " %9.0g `pr`s'' - `c`s''
  3. }
 AVG: total difference = -.1339592    explained = -.1271007
D9D1: total difference =  .1815107    explained = -.0833693
D9D5: total difference =  .1909876    explained = -.0252619
D5D1: total difference = -.0094769    explained = -.0581074
 Var: total difference =  .0538351    explained = -.0170121
```

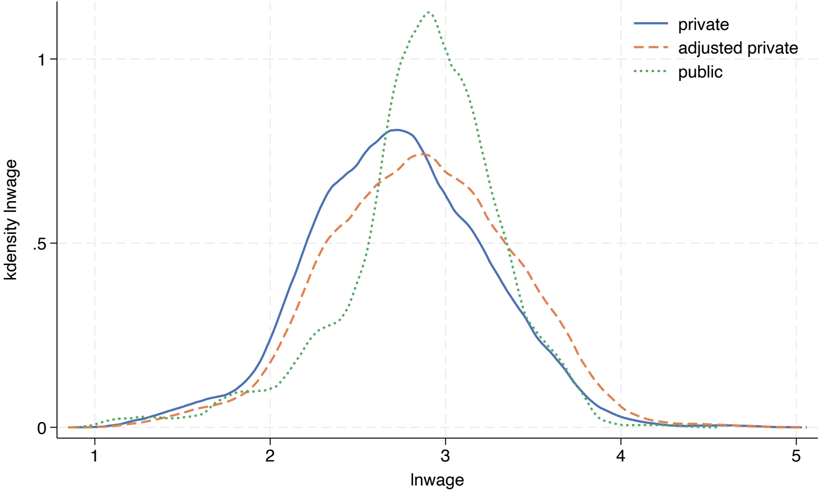
For comparison: results from oaxaca for the mean

```
. oaxaca lnwage yeduc expft expft2, by(public) weight(1) nodetail svy
Blinder-Oaxaca decomposition
Number of strata =     15                        Number of obs     =      5,458
Number of PSUs   = 2,036                         Population size   = 12,146,771
                                                 Design df         =      2,021
                                                 Model             =     linear
Group 1: public = 0                              N of obs 1        =      4,184
Group 2: public = 1                              N of obs 2        =      1,274
    explained: (X1 - X2) * b1
  unexplained: X2 * (b1 - b2)
```

lnwage	Coefficient	Linearized std. err.	t	P>|t|	[95% conf.	interval]
overall						
group_1	2.732109	.0137664	198.46	0.000	2.705111	2.759107
group_2	2.866068	.0213224	134.42	0.000	2.824252	2.907885
difference	-.1339592	.0249495	-5.37	0.000	-.1828886	-.0850298
explained	-.1262644	.0170609	-7.40	0.000	-.1597232	-.0928056
unexplained	-.0076948	.022508	-0.34	0.732	-.0518361	.0364464

How the reweighting affects the wage distribution in the private sector:

```
. gen PSI_weight = PSI*weight // [pw=PSI*weight] does not work with every command
. quietly two (kdens lnwage [pw=weight]     if public==0) ///
>             (kdens lnwage [pw=PSI_weight] if public==0) ///
>             (kdens lnwage [pw=weight]     if public==1) ///
>     , legend(order(1 "private" 2 "adjusted private" 3 "public")) xti(lnwage)
```



Matching and entropy balancing

- As mentioned above, alternative approaches can be used to compute the weights for the reweighting decomposition.
- The default approach using predictions from a logit or probit model is equivalent to IPW in the causal inference literature.
- In general, all matching estimators can be expressed as weighting estimators. Hence, we can use any matching technique to obtain $\Psi(X)$.
- We now repeat the above analysis using (1) entropy balancing and (2) nearest-neighbor Mahalanobis distance matching (as implemented in command `kmatch`).

Example analysis: entropy balancing

```
. kmatch eb public yeduc expft [pw=weight], att wgen(PSIeb) ///
>     targets(2) covariances
(fitting balancing weights ... done)
Entropy balancing                               Number of obs = 5,458
                                           Balance tolerance =     .00001

Treatment   : public = 1
Targets     : 2 + covariances
Covariates  : yeduc expft
Matching statistics
```

	Matched			Controls			Balance
	Yes	No	Total	Used	Unused	Total	loss
Treated	1274	0	1274	4184	0	4184	1.48e-15

```
Stored variables
Variable       Storage   Display    Value
    name          type    format    label       Variable label
PSIeb           double  %10.0g                  Matching weights for ATT
. kmatch summarize yeduc expft expft2, meanonly
(refitting the model using the generate() option)
```

	Raw			Matched(ATT)		
Means	Treated	Untreated	StdDif	Treated	Untreated	StdDif
yeduc	14.07113	12.42709	.5918535	14.07113	14.07113	1.92e-15
expft	13.46011	14.32145	-.0860921	13.46011	13.46011	7.10e-16
expft2	279.7071	306.6634	-.080975	279.7071	279.7071	6.83e-16

Example analysis: entropy balancing

```
. sum lnwage [aw=PSIeb] if public==0, detail
                              lnwage
-------------------------------------------------------------
      Percentiles      Smallest
 1%     1.581038       1.108563
 5%      2.03862       1.115142
10%     2.217027       1.115142       Obs               4,184
25%     2.483238        1.12493       Sum of wgt.   2,914,832
50%     2.847232                      Mean           2.856837
                        Largest       Std. dev.        .51236
75%     3.218076       4.659848
90%     3.518388       4.766694       Variance       .2625128
95%     3.658163       4.781641       Skewness      -.0501883
99%     3.975936       4.799255       Kurtosis       3.014726
. local ebAVG  = r(mean)
. local ebD9D1 = r(p90)-r(p10)
. local ebD9D5 = r(p90)-r(p50)
. local ebD5D1 = r(p50)-r(p10)
. local ebVar  = r(Var)
. foreach s in AVG D9D1 D9D5 D5D1 Var {
  2.     display %6s "`s': " "total difference = " %9.0g `pr`s'' - `pu`s'' ///
>          "    explained = " %9.0g `pr`s'' - `eb`s''
  3. }
 AVG: total difference = -.1339592    explained = -.1247281
D9D1: total difference =  .1815107    explained = -.0589392
D9D5: total difference =  .1909876    explained = -.0180936
D5D1: total difference = -.0094769    explained = -.0408456
 Var: total difference =  .0538351    explained = -.0116538
```

Example analysis: nearest-neighbor matching

```
. kmatch md public yeduc expft [pw=weight], att nn(5) wgen(PSInn)
Multivariate-distance nearest-neighbor matching
                                              Number of obs = 5,458
                                         Neighbors:     min =         5
Treatment   : public = 1                                max =        50
Metric      : mahalanobis
Covariates  : yeduc expft
Matching statistics
```

	Matched			Controls		
	Yes	No	Total	Used	Unused	Total
Treated	1274	0	1274	3446	738	4184

```
Stored variables
Variable      Storage   Display    Value
    name         type    format    label      Variable label

PSInn           double  %10.0g               Matching weights for ATT
. kmatch summarize yeduc expft expft2, meanonly
(refitting the model using the generate() option)
```

Means	Raw Treated	Raw Untreated	Raw StdDif	Matched(ATT) Treated	Matched(ATT) Untreated	Matched(ATT) StdDif
yeduc	14.07113	12.42709	.5918535	14.07113	14.07365	-.000908
expft	13.46011	14.32145	-.0860921	13.46011	13.45335	.0006755
expft2	279.7071	306.6634	-.080975	279.7071	278.9917	.002149

Example analysis: nearest-neighbor matching

```
. sum lnwage [aw=PSInn] if public==0, detail
                             lnwage
-------------------------------------------------------------
      Percentiles      Smallest
 1%     1.578979       1.108563
 5%     2.022871       1.115142
10%     2.198335       1.115142       Obs               3,446
25%     2.490723        1.12493       Sum of wgt.   2,914,832

50%     2.841415                      Mean           2.853928
                        Largest       Std. dev.      .5195719
75%     3.217274       4.659848
90%     3.554776       4.766694       Variance        .269955
95%     3.679082       4.781641       Skewness      -.0563432
99%     3.977249       4.799255       Kurtosis        2.94303

. local nnAVG  = r(mean)

. local nnD9D1 = r(p90)-r(p10)

. local nnD9D5 = r(p90)-r(p50)

. local nnD5D1 = r(p50)-r(p10)

. local nnVar  = r(Var)

. foreach s in AVG D9D1 D9D5 D5D1 Var {
  2.     display %6s "`s': " "total difference = " %9.0g `pr`s'' - `pu`s'' ///
>          "    explained = " %9.0g `pr`s'' - `nn`s''
  3. }
 AVG: total difference = -.1339592    explained = -.1218186
D9D1: total difference =  .1815107    explained = -.1140201
D9D5: total difference =  .1909876    explained = -.0602999
D5D1: total difference = -.0094769    explained = -.0537202
 Var: total difference =  .0538351    explained =  -.019096
```

Example analysis: IPW

- By the way: you can also use `kmatch ipw` to compute the default reweighting $\Psi(X)$; only the scaling will be different because factor $\frac{\Pr(G=0)}{\Pr(G=1)}$ is ignored

```
. kmatch ipw public c.yeduc##c.expft##c.expft [pw=weight], ///
>     att wgen(PSIipw)
  (output omitted)
. summarize PSI_weight PSIipw if public==0

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
  PSI_weight |      4,184    2213.623    4121.928   1.932269   64862.96
      PSIipw |      4,184    696.6616    1297.235   .6081152   20413.38
. corr PSI_weight PSIipw if public==0
(obs=4,184)

             | PSI_we~t   PSIipw
-------------+------------------
  PSI_weight |   1.0000
      PSIipw |   1.0000   1.0000
```

Automatic reweighting using command dstat

- Command dstat (Jann 2020) has built-in options for IPW and entropy balancing and supports a variety of distributional statistics.
- It does not directly provide decompositions, but it can be used to compute the counterfactuals using IPW or entropy balancing; see option balance().

```
. dstat (Var) lnwage, over(public) vce(svy) ///
>     balance(ipw:c.yeduc##c.expft##c.expft, reference(1))
(running dstat_svyr on estimation sample)

Survey: Var

Number of strata =    15          Number of obs   =      5,458
Number of PSUs   = 2,036          Population size = 12,146,771
                                  Design df       =      2,021
                                  Balancing:
                                           method =        ipw
                                        reference =   1.public
                                         controls = e(balance)

------------------------------------------------------------------
             |             Linearized
      lnwage | Coefficient  std. err.      [95% conf. interval]
-------------+----------------------------------------------------
      public |
         no  |   .2678711    .011242       .245824    .2899181
        yes  |   .1970238   .0179054      .1619089    .2321387
------------------------------------------------------------------
```

Automatic reweighting using command dstat

- Decomposition using generated RIFs:

```
. dstat (Var) lnwage, over(public) vce(svy) rif(RIF0c) ///
>     balance(ipw:c.yeduc##c.expft##c.expft, reference(1))
  (output omitted)
. dstat (Var) lnwage if e(sample), over(public) vce(svy) rif(RIF0 RIF1)
  (output omitted)
. generate difference  = RIF0  - RIF1
. generate explained   = RIF0  - RIF0c
. generate unexplained = RIF0c - RIF1
. svy: mean difference explained unexplained
(running mean on estimation sample)
Survey: Mean estimation
Number of strata =    15          Number of obs   =      5,458
Number of PSUs   = 2,036          Population size = 12,146,771
                                  Design df       =      2,021
```

	Mean	Linearized std. err.	[95% conf.	interval]
difference	.0538351	.0203621	.0139021	.0937681
explained	-.0170121	.007549	-.0318167	-.0022076
unexplained	.0708472	.0199062	.0318083	.1098861

Detailed decomposition

- For binary covariates, a detailed decomposition of the contribution to the explained part can be obtained as follows.
- Let X_1 be a binary and X_2 be the vector of all other covariates. A counterfactual distribution of Y in group 0, where the conditional distribution of X_1 given the other covariates is changed to the conditional distribution of X_1 in group 1, can be written as

$$\begin{aligned} F_{Y^0|X_1^1}(y) &= \int\int F_{Y|X^0}(y|X_1, X_2)\, dF_{X^1}(X_1|X_2)\, dF_{X^0}(X_2) \\ &= \int\int F_{Y|X^0}(y|X_1, X_2)\Psi_1(X_1, X_2)\, dF_{X^0}(X_1|X_2)\, dF_{X^0}(X_2) \\ &= \int\int F_{Y|X^0}(y|X_1, X_2)\Psi_1(X_1, X_2)\, dF_{X^0}(X_1, X_2) \end{aligned}$$

where

$$\Psi_1(X_1, X_2) = \frac{dF_{X^1}(X_1|X_2)}{dF_{X^0}(X_1|X_2)} = X_1 \frac{\Pr^1(X_1 = 1|X_2)}{\Pr^0(X_1 = 1|X_2)} + (1 - X_1)\frac{\Pr^1(X_1 = 0|X_2)}{\Pr^0(X_1 = 0|X_2)}$$

Detailed decomposition

- To compute $\Psi_1(X_1, X_2)$, regress X_1 on X_2 separately in group 0 and in group 1 using logistic regression or similar. Then replace $\Pr^0(X_1 = 1|X_2)$, $\Pr^0(X_1 = 0|X_2)$, $\Pr^1(X_1 = 1|X_2)$ and $\Pr^1(X_1 = 0|X_2)$ by predictions from these models.
- A similar approach can also be used to determine the contribution of a binary covariate to the structure component (see Fortin et al. 2011).
- For continuous covariates, things are less clear. One approach followed in the literature is to compute a series of reweighting decompositions where the covariates are introduced one after the other. The problem with this approach is that results will be path dependent.
- A better approach is, for each covariate, to compute the contribution of the covariate while controlling for all other covariates.

Detailed decomposition

- Let $X_{\bar{k}}$ be all covariates except X_k. Based on a similar derivation as above, Fortin et al. (2001) suggest using reweighting factor

$$\Psi_{X_k|X_{\bar{k}}}(X_{\bar{k}}) = \Psi(X)/\Psi(X_{\bar{k}})$$

  where $\Psi(X_{\bar{k}})$ is computed in the same way as the overall reweighting factor $\Psi(X)$, only that variable X_k is omitted from the logit model.
- Using this reweighting factor we can get the counterfactual distribution of Y in group 0, if the conditional distribution of X_k given the other covariates is changed to the conditional distribution of X_k in group 1.
- That procedure is as follows:
  1. Compute $\Psi(X)$ using all covariates.
  2. For each k, compute $\Psi(X_{\bar{k}})$.
  3. For each k, compute the counterfactual statistic using weights $\Psi(X)/\Psi(X_{\bar{k}})$ and compare the result to the unweighted statistic. The difference is the contribution of X_k to the composition effect.
- Note that the single contributions do not add up to the total composition effect.

Detailed decomposition: Contribution of education

```
. drop PS
. quietly logit public c.expft##c.expft [pw=weight], vsquish
. predict PS if e(sample), pr
. generate PSI_yeduc = (PS / `P_public') / ((1-PS) / (1 - `P_public')) if public==0
(1,274 missing values generated)
. quietly sum lnwage [aw=(PSI/PSI_yeduc)*weight] if public==0, detail
. local  cAVGx = r(mean)
. local cD9D1x = r(p90)-r(p10)
. local cD9D5x = r(p90)-r(p50)
. local cD5D1x = r(p50)-r(p10)
. local  cVarx = r(Var)
. foreach s in AVG D9D1 D9D5 D5D1 Var {
  2.     display %6s "`s': " "explained by education = " %9.0g `pr`s'' - `c`s'x'
  3. }
 AVG: explained by education = -.1376105
D9D1: explained by education = -.0826552
D9D5: explained by education = -.0311887
D5D1: explained by education = -.0514665
 Var: explained by education = -.0150466
```

Detailed decomposition: Contribution of experience

```
. drop PS
. quietly logit public yeduc [pw=weight], vsquish
. predict PS if e(sample), pr
. generate PSI_experience = (PS / `P_public') / ((1-PS) / (1 - `P_public')) if public==0
(1,274 missing values generated)
. quietly sum lnwage [aw=(PSI_weight/PSI_experience)*weight] if public==0, detail
. local  cAVGx = r(mean)
. local cD9D1x = r(p90)-r(p10)
. local cD9D5x = r(p90)-r(p50)
. local cD5D1x = r(p50)-r(p10)
. local  cVarx = r(Var)
. foreach s in AVG D9D1 D9D5 D5D1 Var {
  2.     display %6s "`s': " "explained by experience = " %9.0g `pr`s'' - `c`s'x'
  3. }
 AVG: explained by experience = -.0401236
D9D1: explained by experience =  .0269032
D9D5: explained by experience =   .022445
D5D1: explained by experience =  .0044582
 Var: explained by experience =  .0038217
```

For comparison: Results from oaxaca for the mean

```
. oaxaca lnwage yeduc (experience: expft expft2), by(public) weight(1) svy
Blinder-Oaxaca decomposition

Number of strata =    15                      Number of obs     =      5,458
Number of PSUs   = 2,036                      Population size   = 12,146,771
                                              Design df         =      2,021
                                              Model             =     linear
Group 1: public = 0                           N of obs 1        =      4,184
Group 2: public = 1                           N of obs 2        =      1,274

    explained: (X1 - X2) * b1
  unexplained: X2 * (b1 - b2)
```

lnwage	Coefficient	Linearized std. err.	t	P>|t|	[95% conf.	interval]
overall						
group_1	2.732109	.0137664	198.46	0.000	2.705111	2.759107
group_2	2.866068	.0213224	134.42	0.000	2.824252	2.907885
difference	-.1339592	.0249495	-5.37	0.000	-.1828886	-.0850298
explained	-.1262644	.0170609	-7.40	0.000	-.1597232	-.0928056
unexplained	-.0076948	.022508	-0.34	0.732	-.0518361	.0364464
explained						
yeduc	-.1413666	.0166136	-8.51	0.000	-.1739481	-.108785
experience	.0151022	.0099863	1.51	0.131	-.0044824	.0346867
unexplained						
yeduc	.2833853	.1068306	2.65	0.008	.0738756	.492895
experience	-.0650598	.0501303	-1.30	0.194	-.1633722	.0332526
_cons	-.2260203	.116197	-1.95	0.052	-.4538987	.0018581

```
experience: expft expft2
```

The Ñopo decomposition

- As mentioned above, the variance of the weights used in the reweighting decomposition can get large if the compared groups are very different in terms of the distributions of the X variables.
- More fundamentally, there might be a common support problem in the sense that some of the observations cannot be "matched". In this case, the groups cannot be made comparable based on matching or reweighting.
- Ñopo (2008) proposed a decomposition in which the usual decomposition into an explained and an unexplained part is performed only within the "common support". In addition, for each group, a term is computed that captures the difference between observations inside and outside the common support.
- We illustrate the procedure for the mean using exact matching. However, the procedure can also be used with other matching algorithms and it can, in principle, be generalized to other summary measures.

Run the matching and store helper variables

```
. kmatch em public yeduc expft [pw=weight], att generate
Exact matching                                         Number of obs = 5,458
                                              Neighbors:     min =         1
Treatment    : public = 1                                    max =        50
Covariates   : yeduc expft
Matching statistics
```

	Matched			Controls		
	Yes	No	Total	Used	Unused	Total
Treated	1114	160	1274	2476	1708	4184

```
Stored variables
Variable      Storage   Display    Value
    name         type    format    label      Variable label

_KM_treat       byte    %8.0g                 Treatment indicator
_KM_nc          byte    %10.0g                Number of matched controls
_KM_nm          byte    %10.0g                Number of times used as a match
_KM_mw          double  %10.0g                Matching weight
_KM_strata      int     %8.0g                 Matching stratum
```

Computation of results required for the decomposition

- private sector: overall

```
. summarize lnwage [aw=weight] if public==0
    Variable |     Obs      Weight        Mean    Std. dev.       Min        Max
-------------+------------------------------------------------------------------
      lnwage |   4,184   9231938.6    2.732109    .5008582   1.108563   4.799255
. local priv = r(mean)
```

- private sector: out of support (just for information)

```
. summarize lnwage [aw=weight] if public==0 & _KM_nm==0
    Variable |     Obs      Weight        Mean    Std. dev.       Min        Max
-------------+------------------------------------------------------------------
      lnwage |   1,708   3848941.8    2.714669     .498534   1.217876   4.799255
```

- private sector: within of support

```
. summarize lnwage [aw=weight] if public==0 & _KM_nm!=0
    Variable |     Obs      Weight        Mean    Std. dev.       Min        Max
-------------+------------------------------------------------------------------
      lnwage |   2,476   5382996.8    2.744579    .5022448   1.108563   4.781641
. local priv_in = r(mean)
```

Computation of results required for the decomposition

- private sector: within support; reweighted

```
. summarize lnwage [aw = _KM_mw] if public==0
    Variable |        Obs      Weight        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------------------
      lnwage |      2,476   2469852.8    2.845155    .5299685   1.108563   4.781641
. local priv_adj = r(mean)
```

- public sector: within support

```
. summarize lnwage [aw=weight] if public==1 & _KM_nc!=0
    Variable |        Obs      Weight        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------------------
      lnwage |      1,114   2469852.8    2.856464    .4379313   1.115142   4.356068
. local pub_in = r(mean)
```

- public sector: out of support (just for information)

```
. summarize lnwage [aw=weight] if public==1 & _KM_nc==0
    Variable |        Obs      Weight        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------------------
      lnwage |        160  444979.202    2.919378    .4733894    1.18479   4.239455
```

Computation of results required for the decomposition

- public sector: overall

```
. summarize lnwage [aw=weight] if public==1
    Variable |     Obs      Weight        Mean   Std. dev.       Min        Max
-------------+------------------------------------------------------------------
      lnwage |   1,274     2914832    2.866068   .4438737   1.115142   4.356068
. local pub = r(mean)
```

Compute the terms of the decomposition

```
. local A = `priv_in'  - `priv'
. local B = `priv_adj' - `priv_in'
. local C = `pub_in'   - `priv_adj'
. local D = `pub'      - `pub_in'
. di as txt "Overall difference            = " as res `pub'-`priv' ///
> _n as txt "A: private out of support     = " as res `A' ///
> _n as txt "B: explained within support   = " as res `B' ///
> _n as txt "C: unexplained within support = " as res `C' ///
> _n as txt "D: public out of support      = " as res `D' ///
> _n as txt "Total (A+B+C+D)               = " as res `A' + `B' + `C' + `D'
Overall difference            = .13395921
A: private out of support     = .01247014
B: explained within support   = .10057561
C: unexplained within support = .01130901
D: public out of support      = .00960445
Total (A+B+C+D)               = .13395921
```

Reweighted OB decomposition

- Focusing on the mean, yet another decomposition is proposed by Fortin et al. (2011).
- The argument is that the coefficients in the linear regressions used by the standard OB decompositions might be biased if there are nonlinear effects.
- By combining reweighting with OB, the decomposition can be made more robust against such specification errors.

Reweighted OB decomposition

- Recall the standard OB decomposition

$$\begin{aligned}\widehat{\Delta}^{\mu} = \widehat{\Delta}^{\mu}_X + \widehat{\Delta}^{\mu}_S &= (\bar{X}^0 - \bar{X}^1)\widehat{\beta}^0 + \bar{X}^1(\widehat{\beta}^0 - \widehat{\beta}^1) \\ &= (\bar{X}^0\widehat{\beta}^0 - \bar{X}^1\widehat{\beta}^0) + (\bar{X}^1\widehat{\beta}^0 - \bar{X}^1\widehat{\beta}^1)\end{aligned}$$

  where, in the current example, group 0 is private sector and group 1 is public sector.
- The suggestion now is to replace $\bar{X}^1\widehat{\beta}^0$ by $\bar{X}^0_C\widehat{\beta}^0_C$ where $\bar{X}^0_C$ is the average of X in reweighted group 0 and $\widehat{\beta}^0_C$ are coefficient estimates from reweighted group 0.
- Note that $\bar{X}^0_C$ will approximate $\bar{X}^1$ if the reweighting is successful. Hence, a deviation between $\bar{X}^0_C$ and $\bar{X}^1$ points to a "reweighting error".
- Furthermore, if there is no specification error, $\widehat{\beta}^0_C$ will be the same as $\widehat{\beta}^0$ (i.e. reweighting has no effect on the coefficients if the model is correctly specified).

Reweighted OB decomposition

- Inserting $\bar{X}^0_C \widehat{\beta}^0_C$, we get the following decomposition:

$$\widehat{\Delta}^\mu = \widehat{\Delta}^\mu_X + \widehat{\Delta}^\mu_S = (\bar{X}^0 \widehat{\beta}^0 - \bar{X}^0_C \widehat{\beta}^0_C) + (\bar{X}^0_C \widehat{\beta}^0_C - \bar{X}^1 \widehat{\beta}^1)$$

- The two components can be rewritten as

$$\widehat{\Delta}^\mu_X = \underbrace{(\bar{X}^0 - \bar{X}^0_C)\widehat{\beta}^0}_{\text{explained}} + \underbrace{\bar{X}^0_C(\widehat{\beta}^0 - \widehat{\beta}^0_C)}_{\text{specification error}}$$

$$\widehat{\Delta}^\mu_S = \underbrace{\bar{X}^1(\widehat{\beta}^0_C - \widehat{\beta}^1)}_{\text{unexplained}} + \underbrace{(\bar{X}^0_C - \bar{X}^1)\widehat{\beta}^0_C}_{\text{reweighting error}}$$

Reweighted OB decomposition

- obtain coefficients and means in private sector

```
svy: regress lnwage yeduc expft expft2 if public==0
mat b_priv = e(b)
mean yeduc expft expft2 [pw=weight] if public==0
mat X_priv = (e(b),1)
```

- compute weights and obtain counterfactual coefficients and means in private sector

```
kmatch ipw public yeduc c.yeduc##c.expft##c.expft ///
    [pw=weight], att wgen(IPW)
regress lnwage yeduc expft expft2 [pw=IPW] if public==0
mat b_priv_C = e(b)
mean yeduc expft expft2 [pw=IPW] if public==0
mat X_priv_C = (e(b),1)
```

- obtain coefficients and means in public sector

```
svy: regress lnwage yeduc expft expft2 if public==1
mat b_pub = e(b)
svy: mean yeduc expft expft2 if public==1
mat X_pub = (e(b),1)
```

Compute the terms of the decomposition

```
. mat D = (X_priv * b_priv' - X_pub * b_pub') ///
>       \ (X_priv - X_priv_C) * b_priv'       ///
>       \  X_priv_C * (b_priv - b_priv_C)'    ///
>       \  X_pub * (b_priv_C - b_pub)'        ///
>       \ (X_priv_C - X_pub) * b_priv_C'
. mat rown D = "Overall difference" "Explained" "Specification error" ///
>              "Unexplained" "Reweighting error"
. matlist D, twidth(20)

                     |        y1
---------------------+-----------
  Overall difference |  -.1339592
           Explained |  -.1293328
 Specification error |   .0022321
         Unexplained |  -.0098497
   Reweighting error |   .0029912
```

Using entropy balancing

- The “reweighting error” will be zero if we use weights that perfectly balance the data:

```
. kmatch eb public c.yeduc##c.expft##c.expft [pw=weight], ///
>     att wgen(EB)
```

(*output omitted*)

```
. regress lnwage yeduc expft expft2 [pw=EB] if public==0
```

(*output omitted*)

```
. mat b_priv_C = e(b)
. mean yeduc expft expft2 [pw=EB] if public==0
```

(*output omitted*)

```
. mat X_priv_C = (e(b),1)
. mat D[1,1] = (X_priv * b_priv' - X_pub * b_pub') ///
>            \ (X_priv - X_priv_C) * b_priv'       ///
>            \  X_priv_C * (b_priv - b_priv_C)'    ///
>            \  X_pub * (b_priv_C - b_pub)'        ///
>            \ (X_priv_C - X_pub) * b_priv_C'
. matlist D, twidth(20)
```

	y1
Overall difference	-.1339592
Explained	-.1262644
Specification error	.0019249
Unexplained	-.0096197
Reweighting error	4.94e-16

Exercise 6

References

Chernozhukov, Victor, Iván Fernández-Val, Blaise Melly (2013). Inference on Counterfactual Distributions. Econometrica 81(6):2205–2268.

DiNardo, John E., Nicole Fortin, Thomas Lemieux (1996). Labour Market Institutions and the Distribution of Wages, 1973-1992: A Semiparametric Approach. Econometrica 64(5):1001–1046.

Firpo, Sergio, Nicole Fortin, Thomas Lemieux (2007). Decomposing Wage Distributions using Recentered Influence Function Regressions. Working paper.

Firpo, Sergio, Nicole M. Fortin, Thomas Lemieux (2009). Unconditional Quantile Regressions. Econometrica 77:953–973.

Fortin, Nicole, Thomas Lemieux, Sergio Firpo (2011). Decomposition Methods in Economics. Pp. 1–102 in: O. Ashenfelter and D. Card (eds.). Handbook of Labor Economics. Amsterdam: Elsevier.

Hainmueller, Jens (2012). Entropy Balancing: A Multivariate Reweighting Method to Produce Balanced Samples in Observational Studies. Political Analysis 20(1):25–46.

Hainmueller, Jens, Chad Hazlett (2014). Kernel Regularized Least Squares: Reducing Misspecification Bias with a Flexible and Interpretable Machine Learning Approach. Political Analysis 22(2):143–168.

References

Jann, Ben (2017). kmatch: Stata module for multivariate-distance and propensity-score matching. Available from http://ideas.repec.org/c/boc/bocode/s458346.html.

Jann, Ben (2020). dstat: Stata module to compute summary statistics and distribution functions including standard errors and optional covariate balancing. Available from http://ideas.repec.org/c/boc/bocode/s458874.html.

Jann, Ben (2021). Entropy balancing as an estimation command. University of Bern Social Sciences Working Paper No. 39. DOI: 10.7892/boris.157883.

Juhn, Chinhui, Kevin M. Murphy, Brooks Pierce (1993). Wage Inequality and the Rise in Returns to Skill. Journal of Political Economy 101(3):410–442.

Machado, José A. F., José Mata (2005). Counterfactual decomposition of changes in wage distributions using quantile regression. Journal of Applied Econometrics 20(4):445–465.

Melly, Blaise (2005). Decomposition of differences in distribution using quantile regression. Labour Economics 12(4):577–590.

Melly, Blaise, 2006. Estimation of counterfactual distributions using quantile regression. University of St. Gallen, Discussion Paper.

Ñopo, Hugo (2008). Matching as a Tool to Decompose Wage Gaps. The Review of Economics and Statistics 90:290–299.