

```
*****
***** KORAK 1 *****
*****
```

- * Da bismo mogli provesti PSM, potrebne su nam sljedeće varijable:
- * •Učesnik programa: grupa (varijabla koja nam govori da li je neko učestvovao u programu ili je kontrolna grupa)
- * •Ishod/uticaj: Varijabla od interesa (npr. status zaposlenja ili primanja)
- * •Karakteristike učesnika i neučesnika koje mogu da utiču na vjerovatnoću učestvovanja, ali i na ishod (spol, starost, itd.)

```
global ishod _____zamijeni_sa_nazivom_varijable_koja_mjeri_ishod_____
global grupa
_____zamijeni_sa_nazivom_varijable_koja_mjeri_pripadnost_grupi_____
global karakteristike
_____zamijeni_sa_nazivom_varijable_koja_mjeri_pripadnost_grupi_____
varijable_karakteristike_____zamijeni_sa_nazivom_varijable_koja_mjeri_pripadnost_grupi_____
```

```
* Primjer
global ishod status2020
global grupa tretiran
global karakteristike starost cekanje2-cekanje3 pol2 oblast2-oblast7
```

```
*****
***** UPOZNAVANJE SA VARIJABLAMA *****
*****
```

- * Upoznati se sa varijablama: provjera vrste varijabli, prosječnih vrijednosti, broja opservacija, itd.

```
describe $ishod $grupa $karakteristike
summarize $ishod $grupa $karakteristike
```

```
* Vidjeti istu statistiku, ali odvojeno za učesnike i neučesnike odvojeno
bysort $grupa: summarize $ishod $karakteristike
```

```
*****
***** REGRESIONI MODEL PRIJE PSM *****
*****
```

- * Uraditi regresiju sa grupom kao zavisnom varijablom (t-test)
- * Zanima nas da li pripadnost grupi utiče na ishod, kada ne kontrolišemo sa ostlae varijable (karakteristike)
- * reg koristimo kada nam je ishod neka numerička varijabla, a logit/logistic kada imamo binarnu (0 ili 1)

```
*ako je ishod broj (npr. visina plaće)
```

```
reg $ishod $grupa
* ako je ishod binarna varijabla (npr. zaposlen ili nije zaposlen)
logit $ishod $grupa
logistic $ishod $grupa
```

* Uraditi regresiju sa grupom kao zavisnom varijablom dok kontrolišemo za ostale varijable (karakteristike)

```
*ako je ishod broj (npr. visina plaće)
reg $ishod $grupa $karakteristike
* ako je ishod binarna varijabla (npr. zaposlen ili nije zaposlen)
logit $ishod $grupa $karakteristike
logistic $ishod $grupa $karakteristike
```

```
*****
***** KORAK 2 *****
*****
```

* Izbrisi varijablu mypscore ako vec postoji (ovo služi samo ako se skripta ponavlja više puta)

```
capture confirm variable skor
if !_rc {
drop skor
drop blok
}
```

* Generiraj skorove

```
pscore $grupa $karakteristike, logit pscore(skor) blockid(blok) comsup
```

```
*****
***** KORAK 3 i 4 *****
*****
*ALGORITAM UPARIVANJA I PROVJERA UPARIVANJA*
*****
```

*** METODA NAJBLIŽEG SUSJEDA (NEAREST NEIGHBOUR MATCHING)

```
***Pristup 1: Komanda psmatch2
psmatch2 $grupa, outcome($ishod) pscore(skor) neighbor(1)
pstest $karakteristike,both graph
```

```
*** Pristup 2: Komanda attnd
attnd $ishod $grupa, pscore(skor) comsup dots
pstest $karakteristike,both graph
```

```
attnd $ishod $grupa, pscore(skor) comsup dots boot reps(50)
pstest $karakteristike,both graph
```

```

*** Pristup 3: komanda teffects
teffects nnmatch ($ishod $karakteristike) ($grupa), nneighbor(1)
tebalance summarize

*** pogledajte rezultate balansiranja za pojedine varijable
*** Npr. starost - ovdje je neophodno upisati naziv varijable !!!
tebalance box starost
tebalance density starost

*** METODA RADIUS MATCHING

***Pristup 1: Komanda psmatch2
psmatch2 $grupa, radius caliper(0.05) outcome($ishod) pscore(skor)
pstest $karakteristike,both graph

*** Pristup 2: Komanda attr

attr $ishod $grupa, radius (0.1) pscore(skor) comsup dots

attr $ishod $grupa, radius (0.1) pscore(skor) comsup dots boot reps(50)

*** METODA KERNEL MATCHING

***Pristup 1: Komanda psmatch2

psmatch2 $grupa, outcome($ishod) pscore(skor) kernel
pstest $karakteristike,both graph

*** Pristup 2: Komanda attnk
atnk $ishod $grupa $karakteristike, pscore(skor) comsup boot reps(50)

//Stratification Matching
atts $ishod $grupa $karakteristike, pscore(skor) comsup blockid (blokovi)
boot reps(50) dots

//prosječne vrijednosti nakon machiranja
by _treated, sort : summarize pol2 [fweight = _weight]

```