HA2_2.1

Ishwara Hegde, Jonathan Neiman and Aleksei Samkov

10/15/2020

Summary

The paper explores the impact of providing microcredit to rural areas in Morocco. The existing literature on microcredit at the time found that microcredit has a positive impact on self-employment activities but no impact on consumption or overall income. This paper contributes to the literature in three ways. First, the program in question is the only microcredit organization in the area. Second, the paper uses a unique sampling strategy that provides sufficient power to estimate impacts on borrowers as well representative households.Third, it provides a strategy to test for externalities on non borrowers.

The authors find that overall the impact of microcredit on the population is fairly limited. Even in the case with no other access to credit, take-up is low. Further, the gains in self-employment investments, sales and profits are offset by declines in employment income.

Dataset

Some useful points about the dataset:

- Create the Output folder in the same directory in which you extracted the file from AEA
- Run the Master do file
- Use output_total, profit_total and ids from the endline_mini… dta file.
- From baseline: * m1–> no. of people in the HH * nadults_resid –> number of mem that are adults * a7_11 –> age of the head of the HH * d2_6==1 –> family doing animal husbandry * d2_6!=1–> family doing non agricultural activities * i1==0 no and i1!=0 is yes–> Outside loan due for the past 12 months * a_31==2–> Spouse responded to the survey * a_31!= 1 or 2–> another HH member (not head / spouse) responded to survey * paire–> 81 village-pair FE
- Run the exporting_csv.do to export the data

Bootstrapping

```
#setting parameters
n<- nrow(data)
N<-nrow(data)
n1 <- sum(data[,(treatment)])
n0 <- n-n1

#Simple means estimator by OLS
ols_out<-tidy(lm(output_total~treatment,data=data))
ols_profit<-tidy(lm(profit_total~treatment,data=data))

#Simple means manually
tau_simple_out<-mean(data[treatment==1, (output_total)])-
```

```
                mean(data[treatment==0,(output_total)])

tau_simple_profit<-mean(data[treatment==1, (profit_total)])-
                mean(data[treatment==0,(profit_total)])


blocks<-data[, .(.N), by = .(paire)]
B<-100
tau_matrix2<-data.table(tau_bs_out=rep(0,B),tau_bs_profit=rep(0,B),
                       b=seq(1:B))

for (i in 1:B){
        block_values2<-data.table(out=rep(-9999,5427),profit=rep(-9999,5427),
                                  treat=rep(-9999,5427),paire=rep(-9999,5427))
        paire_bs<-c(sample(1:81,81,replace = TRUE))
        start_val<-1
        for (j in paire_bs){
                n_block<-as.integer(blocks[paire==j,.(N)])
                block_values2[start_val:(start_val+n_block-1),]<-
                data[paire==j,.(output_total,profit_total,treatment,paire)]
                start_val<-start_val+n_block

        }
        summ_tab2<-
        block_values2[out!=-9999,.(mean_output=mean(out),mean_profit=mean(profit)),
                     by=.(treat)]
        tau_matrix2[i,1:2]<-summ_tab2[treat==1,2:3]-summ_tab2[treat==0,2:3]

}

#Standard errors


bs_se<-tau_matrix2[ ,.(out_sd=sd(tau_bs_out),profit_sd=sd(tau_bs_profit))]
bs_se
```

```
##      out_sd profit_sd
## 1: 4723.288  1689.131
```

```
ols_out
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   33361.     2738.   12.2    1.38e-33
## 2 treatment       256.     3921.    0.0652 9.48e- 1
```

```
ols_profit
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   10128.     1181.     8.57  1.41e-17
## 2 treatment      -645.     1692.    -0.381 7.03e- 1
```

We notice that in both cases the simple estimator has a larger bootstrap variance than just standard OLS.

#Debiased Estimator

```r
# Prepare data


data_db_model  <- na.roughfix(data[,.(m1,nadults_resid,a7_11,d2_6,
                                       i1,a3_1,paire,output_total,profit_total)])



# Cross fitting
index_1 <- sample(1:n,floor(n/2))
index_2 <- setdiff(1:n,index_1)
index_11 <- index_1[data$treatment[index_1]==1]
index_10 <- setdiff(index_1, index_11)
index_21 <- index_2[data$treatment[index_2]==1]
index_20 <- setdiff(index_2, index_21)

data_1  <- data_db_model[index_1,]
data_2  <- data_db_model[index_2,]
data_11 <- data_db_model[index_11,]
data_10 <- data_db_model[index_10,]
data_21 <- data_db_model[index_21,]
data_20 <- data_db_model[index_20,]

# Random forest

#The minus meanns that all columns except that one
fit11_output <- randomForest(output_total ~ . - profit_total,  data = data_11)
fit21_output <- randomForest(output_total ~ . - profit_total,  data = data_21)

fit10_output <- randomForest(output_total ~ . - profit_total,  data = data_10)
fit20_output <- randomForest(output_total ~ . - profit_total,  data = data_20)

fit11_profit <- randomForest(profit_total ~ . - output_total,  data = data_11)
fit21_profit <- randomForest(profit_total ~ . - output_total,  data = data_21)

fit10_profit <- randomForest(profit_total ~ . - output_total,  data = data_10)
fit20_profit <- randomForest(profit_total ~ . - output_total,  data = data_20)

# Output
m21_output <- predict(fit11_output,  data_2)
m11_output <- predict(fit21_output,  data_1)
m1_output <- rep(0,n)
m1_output[index_1] <- m11_output
m1_output[index_2] <- m21_output

m20_output <- predict(fit11_output,  data_2)
m10_output <- predict(fit21_output,  data_1)
m0_output <- rep(0,n)
m0_output[index_1] <- m10_output
m0_output[index_2] <- m20_output

# Profit
m21_profit <- predict(fit11_profit,  data_2)
```

```r
m11_profit <- predict(fit21_profit,  data_1)
m1_profit <- rep(0,n)
m1_profit[index_1] <- m11_profit
m1_profit[index_2] <- m21_profit

m20_profit <- predict(fit11_profit,  data_2)
m10_profit <- predict(fit21_profit,  data_1)
m0_profit <- rep(0,n)
m0_profit[index_1] <- m10_profit
m0_profit[index_2] <- m20_profit


# Construct \tau_db

W_1 <- data$treatment
W_0 <- 1-W_1
gamma_1_simple <- n/n1
gamma_0_simple <- n/n0

tau_db_output <- mean(m1_output-m0_output) +
  mean(W_1*(data$output_total-m1_output))*gamma_1_simple
- mean(W_0*(data$output_total-m0_output))*gamma_0_simple
```

```
## [1] -1390.422
```

```r
tau_db_profit <- mean(m1_profit-m0_profit) +
  mean(W_1*(data$profit_total-m1_profit))*gamma_1_simple
- mean(W_0*(data$profit_total-m0_profit))*gamma_0_simple
```

```
## [1] -661.0388
```

Debiased Standard Errors

```r
B<-100
tau_matrix3<-data.table(tau_bs_out=rep(0,B),tau_bs_profit=rep(0,B),
                       b=seq(1:B))

for (i in 1:B){
        block_values2<-data.table(out=rep(-9999,5427),profit=rep(-9999,5427),
                                  treat=rep(-9999,5427),paire=rep(-9999,5427),
                                  m1=rep(-9999,5427),nadults_resid=rep(-9999,5427),
                                  a7_11=rep(-9999,5427),d2_6=rep(-9999,5427),
                                  i1=rep(-9999,5427),a3_1=rep(-9999,5427))

        paire_bs<-c(sample(1:81,81,replace = TRUE))
        start_val<-1
        for (j in paire_bs){
                n_block<-as.integer(blocks[paire==j,.(N)])
                block_values2[start_val:(start_val+n_block-1),]<-
                data[paire==j,.(output_total,profit_total,treatment,paire,
                                 m1,nadults_resid,a7_11,d2_6,i1,a3_1)]
                start_val<-start_val+n_block
```

```
}
n<-nrow(block_values2[treat!=-9999])
n1 <- sum(block_values2[treat!=-9999,(treat)])
n0 <- n-n1
block_values2<-block_values2[treat!=-9999]



data_db_model  <-
na.roughfix(block_values2[treat!=-9999,.(m1,nadults_resid,a7_11,d2_6,
                               i1,a3_1,paire,out,profit)])

# Cross fitting
index_1 <- sample(1:n,floor(n/2))
index_2 <- setdiff(1:n,index_1)
index_11 <- index_1[block_values2$treat[index_1]==1]
index_10 <- setdiff(index_1, index_11)
index_21 <- index_2[block_values2$treat[index_2]==1]
index_20 <- setdiff(index_2, index_21)

data_1  <- data_db_model[index_1,]
data_2  <- data_db_model[index_2,]
data_11 <- data_db_model[index_11,]
data_10 <- data_db_model[index_10,]
data_21 <- data_db_model[index_21,]
data_20 <- data_db_model[index_20,]

# Random forest

#The minus meanns that all columns except that one
fit11_output <- randomForest(out ~ . - profit,  data = data_11)
fit21_output <- randomForest(out ~ . - profit,  data = data_21)

fit10_output <- randomForest(out ~ . - profit,  data = data_10)
fit20_output <- randomForest(out ~ . - profit,  data = data_20)

fit11_profit <- randomForest(profit ~ . - out,  data = data_11)
fit21_profit <- randomForest(profit ~ . - out,  data = data_21)

fit10_profit <- randomForest(profit ~ . - out,  data = data_10)
fit20_profit <- randomForest(profit ~ . - out,  data = data_20)

# Output
m21_output <- predict(fit11_output,  data_2)
m11_output <- predict(fit21_output,  data_1)
m1_output <- rep(0,n)
m1_output[index_1] <- m11_output
m1_output[index_2] <- m21_output

m20_output <- predict(fit11_output,  data_2)
m10_output <- predict(fit21_output,  data_1)
m0_output <- rep(0,n)
m0_output[index_1] <- m10_output
```

```
        m0_output[index_2] <- m20_output

        # Profit
        m21_profit <- predict(fit11_profit,  data_2)
        m11_profit <- predict(fit21_profit,  data_1)
        m1_profit <- rep(0,n)
        m1_profit[index_1] <- m11_profit
        m1_profit[index_2] <- m21_profit

        m20_profit <- predict(fit11_profit,  data_2)
        m10_profit <- predict(fit21_profit,  data_1)
        m0_profit <- rep(0,n)
        m0_profit[index_1] <- m10_profit
        m0_profit[index_2] <- m20_profit

        W_1 <- block_values2$treat
        W_0 <- 1-W_1
        gamma_1_simple <- n/n1
        gamma_0_simple <- n/n0

        tau_db_output <- mean(m1_output-m0_output) +
          mean(W_1*(block_values2$out-m1_output))*gamma_1_simple
        - mean(W_0*(block_values2$out-m0_output))*gamma_0_simple

        tau_db_profit <- mean(m1_profit-m0_profit) +
          mean(W_1*(block_values2$profit-m1_profit))*gamma_1_simple
        - mean(W_0*(block_values2$profit-m0_profit))*gamma_0_simple


  tau_matrix3[i,1]<-tau_db_output
  tau_matrix3[i,2]<-tau_db_profit

}

#Standard errors


bs_se2<-tau_matrix3[ ,.(out_sd=sd(tau_bs_out),profit_sd=sd(tau_bs_profit))]
bs_se2
```

```
##      out_sd profit_sd
## 1: 828.9038  342.3852
```

Now the standard deviations are much smaller. This makes sense since the de- biased estimators have lower conditional means and therefore lower variance than the simple difference in means. This follows from the Variance decomposition formula. “‘