Setup The Environment

Load The Packages

```r
library(readstata13)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.1     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## Warning: package 'ggplot2' was built under R version 3.5.2
## Warning: package 'tibble' was built under R version 3.5.2
## Warning: package 'tidyr' was built under R version 3.5.2
## Warning: package 'purrr' was built under R version 3.5.2
## Warning: package 'dplyr' was built under R version 3.5.2
## Warning: package 'stringr' was built under R version 3.5.2
## Warning: package 'forcats' was built under R version 3.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(outliers)
library(ggplot2)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.2
## Loading required package: lattice
##
## Attaching package: 'caret'
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(randomForest)
```

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:outliers':
##
##     outlier
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(lfe)
```

```
## Warning: package 'lfe' was built under R version 3.5.2
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:tidyr':
##
##     expand
```

```r
library(grf)
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 3.5.2
##
## Attaching package: 'cowplot'
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

Data Preparation And Exploration

Load The Datasets

```r
endlines <- read.dta13("data/2013-0533_data_endlines1and2.dta",
                       convert.factors = FALSE,
                       generate.factors = TRUE)
str(endlines)
```

```
## 'data.frame':    6863 obs. of  187 variables:
##  $ hhid                      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ areaid                    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ treatment                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ w                         : num  0.82 1 1 1 1 ...
##  $ w1                        : num  0.777 1 1 1 1 ...
##  $ w2                        : num  0.82 1 1 1 1 ...
##  $ sample1                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ sample2                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ old_biz                   : int  0 0 1 1 1 1 0 0 1 1 ...
##  $ any_old_biz               : int  0 0 1 1 1 1 0 0 1 1 ...
##  $ area_pop_base             : int  272 272 272 272 272 272 272 272 272 272 ...
##  $ area_debt_total_base      : num  81050 81050 81050 81050 81050 ...
##  $ area_business_total_base  : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ area_exp_pc_mean_base     : num  1335 1335 1335 1335 1335 ...
##  $ area_literate_head_base   : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ area_literate_base        : num  0.534 0.534 0.534 0.534 0.534 ...
##  $ visitday_1                : int  22 22 23 22 22 23 23 22 22 22 ...
```

```
##  $ visitmonth_1            : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ visityear_1             : int  2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
##  $ visitday_2              : int  16 16 16 16 17 13 16 16 16 17 ...
##  $ visitmonth_2            : int  12 12 12 12 12 5 12 12 12 12 ...
##  $ visityear_2             : int  2009 2009 2009 2009 2009 2010 2009 2009 2009 2009 ...
##  $ hhsize_1                : int  3 4 5 5 6 6 4 4 7 6 ...
##  $ hhsize_adj_1            : num  2.8 3.24 4.18 4.03 5.41 ...
##  $ adults_1                : int  3 2 2 2 4 3 4 2 7 4 ...
##  $ children_1              : int  0 2 3 3 2 3 0 2 0 2 ...
##  $ male_head_1             : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ head_age_1              : int  20 34 40 37 32 40 43 31 62 64 ...
##  $ head_noeduc_1           : int  1 0 0 0 0 1 1 0 0 1 ...
##  $ women1845_1             : int  2 1 1 1 1 1 2 1 2 1 ...
##  $ anychild1318_1          : int  0 0 1 1 1 1 1 0 1 0 ...
##  $ hhsize_2                : int  3 4 6 7 6 7 6 4 7 6 ...
##  $ hhsize_adj_2            : num  2.42 3.51 5.35 6.08 5.41 ...
##  $ adults_2                : int  2 2 3 4 4 6 6 2 6 4 ...
##  $ children_2              : int  1 2 3 3 2 1 0 2 0 2 ...
##  $ male_head_2             : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ head_age_2              : int  32 37 40 40 35 44 45 33 62 68 ...
##  $ head_noeduc_2           : int  0 0 0 0 0 1 0 0 0 1 ...
##  $ women1845_2             : int  1 1 1 1 1 2 3 1 2 1 ...
##  $ anychild1318_2          : int  0 1 1 1 1 1 1 0 1 0 ...
##  $ spouse_literate_1       : int  1 1 1 NA 1 0 0 0 1 0 ...
##  $ spouse_works_wage_1     : int  0 1 0 1 0 0 0 0 0 0 ...
##  $ ownland_hyderabad_1     : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ ownland_village_1       : int  0 0 1 0 1 0 0 0 0 1 ...
##  $ spouse_literate_2       : int  1 1 1 1 1 0 1 0 0 0 ...
##  $ spouse_works_wage_2     : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ ownland_hyderabad_2     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ownland_village_2       : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ spandana_1              : int  1 0 0 0 0 1 0 1 0 0 ...
##  $ othermfi_1              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ anymfi_1                : int  1 0 0 0 0 1 0 1 0 0 ...
##  $ anybank_1               : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ anyinformal_1           : int  1 0 1 1 0 1 0 0 0 0 ...
##  $ anyloan_1               : int  1 0 0 1 1 1 1 1 1 1 ...
##  $ everlate_1              : int  1 0 1 1 0 0 0 1 0 1 ...
##  $ mfi_loan_cycles_1       : num  1 0 0 0 0 3 0 1 0 0 ...
##  $ spandana_amt_1          : int  18000 0 0 0 0 15000 0 15000 0 0 ...
##  $ othermfi_amt_1          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ anymfi_amt_1            : num  18000 0 0 0 0 15000 0 15000 0 0 ...
##  $ bank_amt_1              : int  0 0 0 0 0 0 0 0 0 30000 ...
##  $ informal_amt_1          : num  93540 0 60000 60000 0 ...
##  $ anyloan_amt_1           : int  115780 0 0 51700 23000 15000 15000 15000 9500 30000 ...
##  $ spandana_2              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ othermfi_2              : int  0 0 0 0 0 0 0 1 1 0 ...
##  $ anymfi_2                : int  0 0 0 0 0 0 0 1 1 0 ...
##  $ anybank_2               : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ anyinformal_2           : int  0 0 0 1 1 1 1 1 0 0 ...
##  $ anyloan_2               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ everlate_2              : int  1 0 1 1 1 1 0 1 0 1 ...
##  $ mfi_loan_cycles_2       : num  NA 0 0 0 0 3 0 2 1 0 ...
##  $ spandana_amt_2          : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ othermfi_amt_2           : num  0 0 0 0 0 0 0 26000 22000 0 ...
##  $ anymfi_amt_2             : num  0 0 0 0 0 ...
##  $ bank_amt_2               : num  0 0 0 0 0 ...
##  $ informal_amt_2           : num  0 0 0 462303 45814 ...
##  $ anyloan_amt_2            : int  11000 25000 5000 565000 55000 111000 8000 46000 22000 74000 ...
##  $ bizassets_1              : num  0 0 2000 0 31700 0 0 0 0 12000 ...
##  $ bizinvestment_1          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bizrev_1                 : num  0 0 1800 5000 12400 7560 2170 0 2300 NA ...
##  $ bizexpense_1             : num  0 0 205 205 8750 ...
##  $ bizprofit_1              : num  0 0 1595 4795 3650 ...
##  $ bizemployees_1           : int  0 0 0 0 0 0 0 0 0 9 ...
##  $ any_biz_1                : int  0 0 1 1 1 1 1 0 1 1 ...
##  $ total_biz_1              : int  0 0 1 1 1 1 2 0 2 1 ...
##  $ any_new_biz_1            : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ biz_stop_1               : int  NA NA 0 0 0 0 0 NA 0 0 ...
##  $ newbiz_1                 : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ female_biz_1             : int  0 0 0 0 0 0 1 0 1 1 ...
##  $ female_biz_new_1         : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ bizassets_2              : num  0 0 0 2915 34902 ...
##  $ bizinvestment_2          : num  0 0 0 0 0 ...
##  $ bizrev_2                 : num  0 0 2499 2499 74634 ...
##  $ bizexpense_2             : num  0 0 450 416 NA ...
##  $ bizprofit_2              : num  0 0 2049 2082 NA ...
##  $ bizemployees_2           : int  0 0 0 1 0 0 0 0 2 0 ...
##  $ any_biz_2                : int  0 0 1 1 1 1 1 0 1 1 ...
##  $ total_biz_2              : int  0 0 1 1 1 2 1 0 3 1 ...
##  $ any_new_biz_2            : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ biz_stop_2               : int  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
##  - attr(*, "datalabel")= chr "Endline data for \"The miracle of microfinance?\" (Banerjee et al.), A
##  - attr(*, "time.stamp")= chr " 5 May 2014 11:57"
##  - attr(*, "formats")= chr  "%10.0g" "%8.0g" "%9.0g" "%9.0g" ...
##  - attr(*, "types")= int  65529 65529 65530 65527 65527 65527 65530 65530 65530 65530 ...
##  - attr(*, "val.labels")= Named chr  "" "" "TREAT" "" ...
##   ..- attr(*, "names")= chr  "" "" "TREAT" "" ...
##  - attr(*, "var.labels")= chr  "Household ID" "Area ID" "Treatment area" "Raw weight" ...
##  - attr(*, "version")= int 117
##  - attr(*, "label.table")=List of 2
##   ..$ YESNO: Named int  0 1
##   .. ..- attr(*, "names")= chr  "No" "Yes"
##   ..$ TREAT: Named int  0 1
##   .. ..- attr(*, "names")= chr  "Control" "Treatment"
##  - attr(*, "expansion.fields")=List of 25
##   ..$ : chr  "_dta" "__XijVarLab1" "spandana_duration_yrsEL1_ Yrs since took loan"
##   ..$ : chr  "_dta" "__XijVarLabTotal" "1"
##   ..$ : chr  "hhid" "destring" "Characters removed were:"
##   ..$ : chr  "visitday_2" "destring" "Characters removed were:"
##   ..$ : chr  "visitmonth_2" "destring" "Characters removed were:"
##   ..$ : chr  "visityear_2" "destring" "Characters removed were:"
##   ..$ : chr  "_dta" "__XijVarLabvalue" "Value"
##   ..$ : chr  "_dta" "__JVarLab" "VarName"
##   ..$ : chr  "_dta" "ReS_Xij" "value"
##   ..$ : chr  "_dta" "ReS_str" "1"
##   ..$ : chr  "_dta" "ReS_j" "varname"
```

```
##   ..$ : chr  "_dta" "ReS_ver" "v.2"
##   ..$ : chr  "_dta" "ReS_i" "formid"
##   ..$ : chr  "hhid" "tostring" "converted to string"
##   ..$ : chr  "festival_exp_annual_2" "destring" "Characters removed were:"
##   ..$ : chr  "visitday_1" "destring" "Characters removed were:"
##   ..$ : chr  "visitmonth_1" "destring" "Characters removed were:"
##   ..$ : chr  "visityear_1" "destring" "Characters removed were:"
##   ..$ : chr  "_dta" "_lang_list" "default"
##   ..$ : chr  "_dta" "_lang_c" "default"
##   ..$ : chr  "_dta" "note1" "householdid identifies panel HHs"
##   ..$ : chr  "festival_exp_annual_1" "destring" "Characters removed were:  ,"
##   ..$ : chr  "areaid" "destring" "Characters removed were:"
##   ..$ : chr  "_dta" "__XijVarLabmfi_duration_yrsEL1_" "Yrs since took loan"
##   ..$ : chr  "_dta" "note0" "1"
##  - attr(*, "byteorder")= chr "LSF"
##  - attr(*, "orig.dim")= int  6863 187
```

Split Endline1 And Endline2

```
endline1 <- endlines %>%
  filter(sample1 == 1) %>%
  select(colnames(endlines)[1:16], contains("_1"))
str(endline1)
```

```
## 'data.frame':    6863 obs. of  102 variables:
##  $ hhid                     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ areaid                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ treatment                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ w                        : num  0.82 1 1 1 1 ...
##  $ w1                       : num  0.777 1 1 1 1 ...
##  $ w2                       : num  0.82 1 1 1 1 ...
##  $ sample1                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ sample2                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ old_biz                  : int  0 0 1 1 1 1 0 0 1 1 ...
##  $ any_old_biz              : int  0 0 1 1 1 1 0 0 1 1 ...
##  $ area_pop_base            : int  272 272 272 272 272 272 272 272 272 272 ...
##  $ area_debt_total_base     : num  81050 81050 81050 81050 81050 ...
##  $ area_business_total_base : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ area_exp_pc_mean_base    : num  1335 1335 1335 1335 1335 ...
##  $ area_literate_head_base  : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ area_literate_base       : num  0.534 0.534 0.534 0.534 0.534 ...
##  $ visitday_1               : int  22 22 23 22 22 23 23 22 22 22 ...
##  $ visitmonth_1             : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ visityear_1              : int  2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
##  $ hhsize_1                 : int  3 4 5 5 6 6 4 4 7 6 ...
##  $ hhsize_adj_1             : num  2.8 3.24 4.18 4.03 5.41 ...
##  $ adults_1                 : int  3 2 2 2 4 3 4 2 7 4 ...
##  $ children_1               : int  0 2 3 3 2 3 0 2 0 2 ...
##  $ male_head_1              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ head_age_1               : int  20 34 40 37 32 40 43 31 62 64 ...
##  $ head_noeduc_1            : int  1 0 0 0 0 1 1 0 0 1 ...
##  $ women1845_1              : int  2 1 1 1 1 1 2 1 2 1 ...
##  $ anychild1318_1           : int  0 0 1 1 1 1 1 0 1 0 ...
```

```
##  $ spouse_literate_1         : int  1 1 1 NA 1 0 0 0 1 0 ...
##  $ spouse_works_wage_1       : int  0 1 0 1 0 0 0 0 0 0 ...
##  $ ownland_hyderabad_1       : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ ownland_village_1         : int  0 0 1 0 1 0 0 0 0 1 ...
##  $ spandana_1                : int  1 0 0 0 0 1 0 1 0 0 ...
##  $ othermfi_1                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ anymfi_1                  : int  1 0 0 0 0 1 0 1 0 0 ...
##  $ anybank_1                 : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ anyinformal_1             : int  1 0 1 1 0 1 0 0 0 0 ...
##  $ anyloan_1                 : int  1 0 0 1 1 1 1 1 1 1 ...
##  $ everlate_1                : int  1 0 1 1 0 0 0 1 0 1 ...
##  $ mfi_loan_cycles_1         : num  1 0 0 0 0 3 0 1 0 0 ...
##  $ spandana_amt_1            : int  18000 0 0 0 0 15000 0 15000 0 0 ...
##  $ othermfi_amt_1            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ anymfi_amt_1              : num  18000 0 0 0 0 15000 0 15000 0 0 ...
##  $ bank_amt_1                : int  0 0 0 0 0 0 0 0 0 30000 ...
##  $ informal_amt_1            : num  93540 0 60000 60000 0 ...
##  $ anyloan_amt_1             : int  115780 0 0 51700 23000 15000 15000 15000 9500 30000 ...
##  $ bizassets_1               : num  0 0 2000 0 31700 0 0 0 0 12000 ...
##  $ bizinvestment_1           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bizrev_1                  : num  0 0 1800 5000 12400 7560 2170 0 2300 NA ...
##  $ bizexpense_1              : num  0 0 205 205 8750 ...
##  $ bizprofit_1               : num  0 0 1595 4795 3650 ...
##  $ bizemployees_1            : int  0 0 0 0 0 0 0 0 0 9 ...
##  $ any_biz_1                 : int  0 0 1 1 1 1 1 0 1 1 ...
##  $ total_biz_1               : int  0 0 1 1 1 1 2 0 2 1 ...
##  $ any_new_biz_1             : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ biz_stop_1                : int  NA NA 0 0 0 0 0 NA 0 0 ...
##  $ newbiz_1                  : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ female_biz_1              : int  0 0 0 0 0 0 1 0 1 1 ...
##  $ female_biz_new_1          : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ wages_nonbiz_1            : int  2000 3900 5000 1500 0 500 0 NA 3950 0 ...
##  $ hours_week_1              : num  48 8 21 77 70 126 0 NA 152 64 ...
##  $ hours_week_biz_1          : int  0 0 21 77 70 126 0 0 152 64 ...
##  $ hours_week_outside_1      : num  48 8 0 0 0 0 0 0 0 0 ...
##  $ hours_headspouse_week_1   : num  48 8 49 77 70 70 0 NA 56 49 ...
##  $ hours_headspouse_outside_1: num  48 4 0 63 0 0 0 NA 0 0 ...
##  $ hours_headspouse_biz_1    : num  0 4 49 14 70 70 0 NA 56 49 ...
##  $ hours_child1620_week_1    : int  48 NA NA NA NA 56 0 NA 0 NA ...
##  $ hours_girl1620_week_1     : int  0 NA NA NA NA NA 0 NA 0 NA ...
##  $ hours_boy1620_week_1      : int  48 NA NA NA NA 56 0 NA 0 NA ...
##  $ total_exp_mo_1            : num  2154 4442 5208 4566 5313 ...
##  $ durables_exp_mo_1         : num  NA 29.2 212.5 154.2 NA ...
##  $ nondurable_exp_mo_1       : num  NA 4413 4995 4412 NA ...
##  $ health_exp_mo_1           : num  250 183 200 0 200 ...
##  $ educ_exp_mo_1             : num  NA 825 977 967 900 ...
##  $ festival_exp_annual_1     : int  600 3300 2000 4000 800 4000 1000 NA 6000 10000 ...
##  $ temptation_exp_mo_1       : num  170 0 100 0 0 0 100 NA 0 12000 ...
##  $ food_exp_mo_1             : num  1084 2175 2510 2519 2225 ...
##  $ total_exp_mo_pc_1         : num  769 1371 1246 1133 982 ...
##  $ durables_exp_mo_pc_1      : num  NA 9 50.8 38.3 NA ...
##  $ nondurable_exp_mo_pc_1    : num  NA 1362 1195 1095 NA ...
##  $ food_exp_mo_pc_1          : num  387 671 600 625 411 ...
##  $ health_exp_mo_pc_1        : num  89.3 56.6 47.8 0 37 ...
```

```
##  $ educ_exp_mo_pc_1         : num  NA 255 234 240 166 ...
##  $ temptation_exp_mo_pc_1   : num  60.7 0 23.9 0 0 ...
##  $ festival_exp_mo_pc_1     : num  17.9 84.9 39.9 82.7 12.3 ...
##  $ home_durable_index_1     : num  2.69 2.2 2.46 1.3 2.65 ...
##  $ girl515_school_1         : num  NA 1 NA 1 NA 0.5 NA NA NA NA ...
##  $ boy515_school_1          : num  NA 1 1 1 1 1 NA 1 NA 1 ...
##  $ girl515_workhrs_pc_1     : num  NA 0 NA 0 NA 0 NA NA NA NA ...
##  $ boy515_workhrs_pc_1      : num  NA 0 0 0 0 0 NA 0 NA 0 ...
##  $ girl1620_school_1        : num  0 NA NA NA NA NA 0 NA 1 NA ...
##  $ boy1620_school_1         : num  1 NA NA NA NA 0 0 NA 1 NA ...
##  $ women_emp_index_1        : num  -0.4154 0.5629 -0.0623 -0.368 -0.3653 ...
##  $ female_biz_pct_1         : num  NA NA 0 0 0 0 0.5 NA 0.5 1 ...
##  $ credit_index_1           : num  1.136 -0.492 -0.417 -0.178 -0.269 ...
##  $ biz_index_all_1          : num  -0.224 -0.224 0.0651 0.0827 0.3603 ...
##  $ biz_index_old_1          : num  NA NA -0.197 -0.162 0.183 ...
##  $ biz_index_new_1          : num  NA NA NA NA NA ...
##  $ income_index_1           : num  -0.160999 0.081633 0.296675 -0.000669 -0.245753 ...
##   [list output truncated]
##  - attr(*, "datalabel")= chr "Endline data for \"The miracle of microfinance?\" (Banerjee et al.), AE
##  - attr(*, "time.stamp")= chr " 5 May 2014 11:57"
##  - attr(*, "formats")= chr  "%10.0g" "%8.0g" "%9.0g" "%9.0g" ...
##  - attr(*, "types")= int  65529 65529 65530 65527 65527 65527 65530 65530 65530 65530 ...
##  - attr(*, "val.labels")= Named chr  "" "" "TREAT" "" ...
##   ..- attr(*, "names")= chr  "" "" "TREAT" "" ...
##  - attr(*, "var.labels")= chr  "Household ID" "Area ID" "Treatment area" "Raw weight" ...
##  - attr(*, "version")= int 117
##  - attr(*, "label.table")=List of 2
##   ..$ YESNO: Named int  0 1
##   .. ..- attr(*, "names")= chr  "No" "Yes"
##   ..$ TREAT: Named int  0 1
##   .. ..- attr(*, "names")= chr  "Control" "Treatment"
##  - attr(*, "expansion.fields")=List of 25
##   ..$ : chr  "_dta" "__XijVarLab1" "spandana_duration_yrsEL1_ Yrs since took loan"
##   ..$ : chr  "_dta" "__XijVarLabTotal" "1"
##   ..$ : chr  "hhid" "destring" "Characters removed were:"
##   ..$ : chr  "visitday_2" "destring" "Characters removed were:"
##   ..$ : chr  "visitmonth_2" "destring" "Characters removed were:"
##   ..$ : chr  "visityear_2" "destring" "Characters removed were:"
##   ..$ : chr  "_dta" "__XijVarLabvalue" "Value"
##   ..$ : chr  "_dta" "__JVarLab" "VarName"
##   ..$ : chr  "_dta" "ReS_Xij" "value"
##   ..$ : chr  "_dta" "ReS_str" "1"
##   ..$ : chr  "_dta" "ReS_j" "varname"
##   ..$ : chr  "_dta" "ReS_ver" "v.2"
##   ..$ : chr  "_dta" "ReS_i" "formid"
##   ..$ : chr  "hhid" "tostring" "converted to string"
##   ..$ : chr  "festival_exp_annual_2" "destring" "Characters removed were:"
##   ..$ : chr  "visitday_1" "destring" "Characters removed were:"
##   ..$ : chr  "visitmonth_1" "destring" "Characters removed were:"
##   ..$ : chr  "visityear_1" "destring" "Characters removed were:"
##   ..$ : chr  "_dta" "_lang_list" "default"
##   ..$ : chr  "_dta" "_lang_c" "default"
##   ..$ : chr  "_dta" "note1" "householdid identifies panel HHs"
##   ..$ : chr  "festival_exp_annual_1" "destring" "Characters removed were:  ,"
```

```
##    ..$ : chr  "areaid" "destring" "Characters removed were:"
##    ..$ : chr  "_dta" "__XijVarLabmfi_duration_yrsEL1_" "Yrs since took loan"
##    ..$ : chr  "_dta" "note0" "1"
##   - attr(*, "byteorder")= chr "LSF"
##   - attr(*, "orig.dim")= int  6863 187
```

Exclude Irrelevant & Redundant Covariates

There are some variables in the dataset that are only relevant when the data were collected, such as the information of when the inspectors visit the households and if the households were included in the endline surveys.

```
endline1 <- endline1 %>%
  select(-c(w, w1, w2, sample1, sample2, visitday_1, visitmonth_1, visityear_1))
```

Since we are going to use `areaid` to do cluster analysis, including the area-level variables doesn't make much sense.

```
endline1 <- endline1 %>%
  select(-starts_with("area_"))
```

The dataset include both total expense and per-capita version for each expense category per month (or annual). To prevent issues with overspecified (irrelevant variables), we exclude the total expenses and only leave the per-capita variables.

```
endline1 <- endline1 %>%
  select(-ends_with("_mo_1"),
         -ends_with("_annual_1"))
```

Business-related Variables

`old_biz` and `any_old_biz` contain similar information, the former indicates how many old businesses a household own prior to the first endline and the latter is a binary variable that indicates whether a household has at least an old business. Here we combine the two variables and assume those households that didn't answer the question was either not a busniess owner at all or not understood the question. Either ways, we could safely consider them as having 0 old busineses.

```
summary(endline1$old_biz)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.000   0.000   0.385   1.000   8.000     101
```

```
summary(endline1$any_old_biz)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  0.0000  0.0000  0.0000  0.3098  1.0000  1.0000     124
```

```
endline1 <- endline1 %>%
  mutate(old_biz = ifelse(any_old_biz == 0 | is.na(any_old_biz) == TRUE,
                          0,
                          old_biz))
# Delete any_old_biz as the information is combined with old_biz
endline1$any_old_biz <- NULL
```

The same reasoning could be apply to `total_biz_1` and `any_biz_1`.

```
endline1 <- endline1 %>%
  mutate(total_biz_1 = ifelse(any_biz_1 == 0 | is.na(any_biz_1) == TRUE,
                              0,
                              total_biz_1))
# Delete any_biz_1 as the information is combined with total_biz_1
endline1$any_biz_1 <- NULL
```

```
endline1 <- endline1 %>%
  mutate(newbiz_1 = ifelse(any_new_biz_1 == 0 | is.na(any_new_biz_1) == TRUE,
                           0,
                           newbiz_1))
# Delete any_biz_1 as the information is combined with newbiz_1
endline1$any_new_biz_1 <- NULL
```

Household-related variables

```
endline1 %>%
  select(hhsize_1, adults_1, children_1) %>%
  mutate(hh_total = adults_1 + children_1) %>%
  filter(hhsize_1 != hh_total) %>%
  nrow()
```

```
## [1] 122
```

```
endline1$hhsize_1 <- NULL
```

Loan-related Variables

```
endline1 %>%
  select(anymfi_1, spandana_1, othermfi_1) %>%
  mutate(mfi = max(spandana_1, othermfi_1)) %>%
  filter(anymfi_1 != mfi) %>%
  nrow()
```

```
## [1] 0
```

```
endline1 %>%
  select(anymfi_amt_1, spandana_amt_1, othermfi_amt_1) %>%
  mutate(total_mfi_amt = spandana_amt_1 + othermfi_amt_1) %>%
  filter(anymfi_amt_1 != total_mfi_amt) %>%
  nrow()
```

```
## [1] 382
```

```
endline1$anymfi_1 <- NULL
endline1$anymfi_amt_1 <- NULL
```

Labor-related Variables

```
endline1 %>%
  select(hours_week_1, hours_week_biz_1, hours_week_outside_1) %>%
  mutate(hours_week_sum = hours_week_biz_1 + hours_week_outside_1) %>%
```

```r
  filter(hours_week_1 != hours_week_sum) %>%
  nrow()
```

```
## [1] 0
```

```r
endline1 %>%
  select(hours_headspouse_week_1, hours_headspouse_biz_1, hours_headspouse_outside_1) %>%
  mutate(hours_headspouse_week_sum = hours_headspouse_biz_1 + hours_headspouse_outside_1) %>%
  filter(hours_headspouse_week_1 != hours_headspouse_week_sum) %>%
  nrow()
```

```
## [1] 0
```

```r
endline1 %>%
  select(hours_child1620_week_1, hours_boy1620_week_1, hours_girl1620_week_1) %>%
  mutate(hours_children_total = hours_boy1620_week_1 + hours_girl1620_week_1) %>%
  filter(hours_child1620_week_1 != hours_children_total) %>%
  nrow()
```

```
## [1] 0
```

```r
endline1$hours_week_1 <- NULL
endline1$hours_headspouse_week_1 <- NULL
endline1$hours_child1620_week_1 <- NULL
```

Missing Values

We will first delete the covariates that contains a huge amount of missing values. Then we will look into the remaining covariates and fill them with custom methods.

First we need to find out which variable contains unreasonable amount of missing value.

```r
na_table <- function(x) {
  na_table <- data.frame()
  for (i in 1:ncol(x)) {
    n_na <- nrow(x[is.na(x[,i]),])
    na_ratio <- n_na / nrow(x)
    na_table[i, 1] <- colnames(x)[[i]]
    na_table[i, 2] <- n_na
    na_table[i, 3] <- na_ratio
    colnames(na_table) <- c("covariate", "n", "ratio")
  }
  return(na_table)
}
```

```r
# set the threshold of na ratio
na_delete_threshold <- 0.1
na_table(endline1) %>% filter(ratio > na_delete_threshold)
```

```
##                covariate    n     ratio
## 1      spouse_literate_1  724 0.1054932
## 2             biz_stop_1 4511 0.6572927
## 3  hours_girl1620_week_1 4689 0.6832289
## 4   hours_boy1620_week_1 4997 0.7281072
## 5       educ_exp_mo_pc_1 1448 0.2109864
## 6       girl515_school_1 3828 0.5577736
## 7        boy515_school_1 3790 0.5522366
```

```
## 8   girl515_workhrs_pc_1 3828 0.5577736
## 9    boy515_workhrs_pc_1 3790 0.5522366
## 10     girl1620_school_1 4689 0.6832289
## 11      boy1620_school_1 4997 0.7281072
## 12      female_biz_pct_1 4495 0.6549614
## 13       biz_index_old_1 4775 0.6957599
## 14       biz_index_new_1 6507 0.9481276
```

We will delete those variables as the information might not be helpful.

```r
# select the variables that has a large amount of missing value
na_delete_col <- (na_table(endline1) %>% filter(ratio > na_delete_threshold))[,1]
# delete those variables
for (col in na_delete_col) {
  endline1[,col] <- NULL
}
```

Filling The Missing Values - Business-related Variables

```r
endline1 <- endline1 %>%
  mutate(bizassets_1 = ifelse(total_biz_1 == 0 | is.na(total_biz_1),
                              0,
                              bizassets_1),
         bizinvestment_1 = ifelse(total_biz_1 == 0 | is.na(total_biz_1),
                                  0,
                                  bizinvestment_1),
         bizrev_1 = ifelse(total_biz_1 == 0 | is.na(total_biz_1),
                           0,
                           bizrev_1),
         bizexpense_1 = ifelse(total_biz_1 == 0 | is.na(total_biz_1),
                               0,
                               bizexpense_1),
         bizprofit_1 = ifelse(total_biz_1 == 0 | is.na(total_biz_1),
                              0,
                              bizprofit_1),
         bizemployees_1 = ifelse(total_biz_1 == 0 | is.na(total_biz_1),
                                 0,
                                 bizemployees_1))
```

For All The Other Variables Except Index Variables

```r
covariates_name <- endline1 %>%
  select(-contains("index")) %>%
  colnames()
for (covar in covariates_name) {
  endline1[is.na(endline1[, covar]), covar] <-
    median(endline1[, covar], na.rm = TRUE)
}
```

Check The Result

```
na_table(endline1) %>% filter(n != 0)
```

```
##              covariate  n        ratio
## 1 home_durable_index_1 22 0.0032055952
## 2    women_emp_index_1  1 0.0001457089
## 3       credit_index_1  1 0.0001457089
## 4      biz_index_all_1 53 0.0077225703
## 5       income_index_1 31 0.0045169751
## 6        labor_index_1 14 0.0020399242
## 7  consumption_index_1 18 0.0026227597
## 8       social_index_1  1 0.0001457089
```

```
endline1 <- na.omit(endline1)
nrow(endline1)
```

```
## [1] 6804
```

Outliers

First we want to know which column (variable) contains outliers and how many of them. Here we will use "Z-score" approach to detect outliers.

```
exp_col <- endline1 %>%
  select(contains("exp_mo_pc"), contains("amt_")) %>%
  colnames()
for (covar in exp_col) {
  covar_outlier <- scores(x = endline1[, covar], type = "iqr", lim = 3)
  endline1 <- endline1[!covar_outlier, ]
}
```

Design Of The Study

We want to study the effect of "availibility of microcredit" on different aspects of the households in Hyderabad, India:

- Business
- ...

However, the fact that in the original study, they didn't collect the baseline data in a very rigorous way and they were not confident enough that the baseline data is representative of the slum of whole. Hence the baseline data was only as a basis for stratification, the descriptive analysis, and to collect **area-level characteristics** that are used as control variables.

Because of the flaw of our datasets, we lose the ability to directly link baseline data with endlines data, hence could not perform the analysis on household-level. To mitigate this issue, we use the "index variables", which were calculated by the authors and were included in our dataset, as our target variables. And we assumed that those variables already include the information we need to analyze the causal effect.

How They Calculate The Results in The Original Paper

For each "target" variable, they run an weighted OLS:

$y_{ia} = \alpha + \beta * Treatment_a + X'\gamma + \epsilon_{ia}$

```
endline1 %>%
  filter(treatment == "Control") %>%
  summarize(mean = mean(spandana_amt_1, na.rm = TRUE))
```

```
##   mean
## 1  NaN
```

Business Index

Treatment & Target Variable

- Treatment Variable: `treatment`
- Target Variable: `biz_index_all_1`

We want to find out whether there are heterogeneous effects of "availibility of Spandana microcredit loan" on business in the area.

```
endline1 %>%
  filter(is.na(treatment) == FALSE) %>% # exclude the observations with NA
  group_by(treatment) %>%
  summarize("Num. of Obs." = n(),
            "Ave. Biz. Index" = mean(biz_index_all_1, na.rm = TRUE))
```

```
## # A tibble: 2 x 3
##   treatment `Num. of Obs.` `Ave. Biz. Index`
##       <int>          <int>             <dbl>
## 1         0           1962           -0.0518
## 2         1           1829           -0.0564
```

Dataset

```
target_index <- "biz_index_all_1"

endline1_biz <- endline1 %>%
  select(everything(),
         -hhid,
         -starts_with("biz"),
         -contains("index"), # prevent confounding
         target_index)       # add target index
str(endline1_biz)
```

```
## 'data.frame':    3791 obs. of  43 variables:
##  $ areaid                      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ treatment                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ old_biz                     : num  0 1 1 1 0 1 0 0 1 0 ...
##  $ hhsize_adj_1                : num  3.24 4.18 4.03 5.41 3.74 ...
##  $ adults_1                    : num  2 2 2 4 4 7 4 3 7 5 ...
##  $ children_1                  : num  2 3 3 2 0 0 0 0 0 0 ...
##  $ male_head_1                 : int  1 1 1 1 1 1 0 0 1 1 ...
##  $ head_age_1                  : int  34 40 37 32 43 62 48 46 65 50 ...
##  $ head_noeduc_1               : num  0 0 0 0 1 0 1 0 1 0 ...
```

```
##  $ women1845_1               : num  1 1 1 1 2 2 1 0 2 1 ...
##  $ anychild1318_1            : num  0 1 1 1 1 1 0 0 1 0 ...
##  $ spouse_works_wage_1       : int  1 0 1 0 0 0 0 0 0 0 ...
##  $ ownland_hyderabad_1       : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ ownland_village_1         : int  0 1 0 1 0 0 0 0 0 0 ...
##  $ spandana_1                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ othermfi_1                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ anybank_1                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ anyinformal_1             : int  0 1 1 0 0 0 1 1 0 0 ...
##  $ anyloan_1                 : num  0 0 1 1 1 1 0 1 1 1 ...
##  $ everlate_1                : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ mfi_loan_cycles_1         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ spandana_amt_1            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ othermfi_amt_1            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bank_amt_1                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ informal_amt_1            : num  0 60000 60000 0 0 0 10000 40000 0 0 ...
##  $ anyloan_amt_1             : num  0 0 51700 23000 15000 9500 0 40000 10000 5000 ...
##  $ total_biz_1               : num  0 1 1 1 2 2 0 0 1 0 ...
##  $ newbiz_1                  : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ female_biz_1              : num  0 0 0 0 1 1 0 0 1 0 ...
##  $ female_biz_new_1          : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ wages_nonbiz_1            : int  3900 5000 1500 0 0 3950 3600 5000 2780 1500 ...
##  $ hours_week_biz_1          : num  0 21 77 70 0 152 0 0 114 0 ...
##  $ hours_week_outside_1      : num  8 0 0 0 0 0 4 118 0 70 ...
##  $ hours_headspouse_outside_1: num  4 0 63 0 0 0 0 0 0 70 ...
##  $ hours_headspouse_biz_1    : num  4 49 14 70 0 56 4 0 66 0 ...
##  $ total_exp_mo_pc_1         : num  1371 1246 1133 982 1059 ...
##  $ durables_exp_mo_pc_1      : num  9 50.8 38.3 26.3 0 ...
##  $ nondurable_exp_mo_pc_1    : num  1362 1195 1095 1132 1059 ...
##  $ food_exp_mo_pc_1          : num  671 600 625 411 612 ...
##  $ health_exp_mo_pc_1        : num  56.6 47.8 0 37 29.4 ...
##  $ temptation_exp_mo_pc_1    : num  0 23.9 0 0 26.7 ...
##  $ festival_exp_mo_pc_1      : num  84.9 39.9 82.7 12.3 22.3 ...
##  $ biz_index_all_1           : num  -0.224 0.0651 0.0827 0.3603 1.3788 ...
##  - attr(*, "na.action")= 'omit' Named int  27 109 185 239 241 290 419 511 529 612 ...
##   ..- attr(*, "names")= chr  "27" "109" "185" "239" ...
```

Two-Models Approach With Sorted Group Average Treatment Effects (GATES)

F-Tests

By Random Forest

```
tm_gates <- function(target, treatment, data,
                     split_ratio=0.5, cluster=0, num.iter=100,
                     ml_method="rf") {

  # a list to store the regression results
  results <- list()

  # TODO implement user specified num.groups
  num.groups <- 5
```

```r
if (cluster != 0) {
  strati_target <- cluster
} else {
  strati_target <- treatment
}

for (i in 1:num.iter) {
  # set seed for reproduction
  set.seed(i)

  # seperate auxi and main sample
  auxi_index <- createDataPartition(data[,strati_target],
                                    p = split_ratio,
                                    list = FALSE)
  auxi <- data[auxi_index, ]
  main <- data[-auxi_index,]

  # seperate treatment & control in the auxiliary sample
  auxi_treat_index <- which(auxi[,treatment] == 1)
  auxi_treat <- auxi[auxi_treat_index, ]
  auxi_contr <- auxi[-auxi_treat_index,]

  # use the specified machine learning method to predict the conditional treatment effect
  if (ml_method == "rf") {
    # fit a random forest on auxi_treat and auxi_contr
    auxi_formula <- as.formula(paste(target, " ~ .", "-", treatment))
    auxi_yi0 <- randomForest(auxi_formula,
                             data = auxi_contr,
                             ntree = 3000,
                             mtry = 3,
                             replace = TRUE,
                             type = "regression")
    auxi_yi1 <- randomForest(auxi_formula,
                             data = auxi_treat,
                             ntree = 3000,
                             mtry = 3,
                             replace = TRUE,
                             type = "regression")
    # predict the baseline effect and conditional treatment effect on main sample
    main_yi0 <- predict(auxi_yi0, newdata = main)
    main_yi1 <- predict(auxi_yi1, newdata = main)
    main$baseline <- main_yi0
    main$cte <- (main_yi1 - main_yi0)
  } else if (ml_method == "crf") {
    # fit a causal random forest on auxi sample
    auxi_X <- auxi %>%
      select(everything(), -target, -treatment)
    auxi_Y <- auxi[, target]
    auxi_W <- auxi[, treatment]
    auxi_crf <- causal_forest(X = auxi_X,
                              Y = auxi_Y,
                              W = auxi_W,
                              honesty = TRUE,
```

```r
                              mtry = 3,
                              num.trees = 3000)
  # predict the conditional treatment effect on main sample
  auxi_crf_pred <- predict(auxi_crf, newdata = main)
  main$cte <- auxi_crf_pred$predictions
}

# TWO-MODELS APPROACH
# Fit regression on conditional treatment effect
tm_exclude_col <- c(target, treatment, cluster,
                    "baseline", "cte")
data_col <- names(main)
tm_formula <- as.formula(
  paste(
    "cte", "~",
    paste(data_col[!data_col %in% tm_exclude_col], collapse = " + ")))

tm_model <- felm(tm_formula,
                 data = main,
                 weights = main$weight)

results[[i]] <- tm_model

# SORTED GROUP AVERAGE TREATMENT EFFECT
# calculate propensity score (treated/all)
# TODO implement option to use non-randomized treatment assignment
prop_score <- nrow(data[data$treatment == 1, ])/nrow(data)
main$prop_score <- prop_score

# divide observations based on their predicted conditional treatment effect
breaks <- quantile(main$cte, seq(0,1, 1/num.groups), include.lowest = TRUE)
breaks[1] <- breaks[1] - 0.001
breaks[6] <- breaks[6] + 0.001
main$treat_group <- cut(main$cte, breaks = breaks)

# calculate the propensity score offset for each observation in main sample
main$prop_offset <- main$treatment - main$prop_score

# construct matrix from each observation's group factor
SGX <- model.matrix(~-1+main$treat_group)
# construct D-p(X)*1(G_k) and weight for each observation
DSG <- data.frame(main$prop_offset*SGX)
colnames(DSG) <- c("G1", "G2", "G3", "G4", "G5")
main[,c("G1", "G2", "G3", "G4", "G5", "weight")] <- cbind(
  DSG$G1, DSG$G2, DSG$G3, DSG$G4, DSG$G5,
  1/prop_score*(1-prop_score))

# fit weighted ols
if (ml_method == "rf") {
  gates_formula <- as.formula(paste(target,
                                    "~",
                                    "-1+baseline+cte+G1+G2+G3+G4+G5",
                                    "|0|0|",
```

```r
                                          cluster))
    } else if (ml_method == "crf") {
      gates_formula <- as.formula(paste(target,
                                          "~",
                                          "cte+G1+G2+G3+G4+G5",
                                          "|0|0|",
                                          cluster))
    }

    gates_model <- felm(gates_formula,
                        data = main,
                        weights = main$weight)

    results[[num.iter+i]] <- gates_model
  }
  return(results)
}
```

Results (using Random Forest)

```r
tm_gates_biz <- tm_gates("biz_index_all_1", "treatment", endline1_biz,
                         split_ratio = 0.6,
                         cluster="areaid", num.iter=1, ml_method="rf")
```

```
## Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
## rank-deficient or indefinite
```

```r
summary(tm_gates_biz[[1]])
```

```
## Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
## rank-deficient or indefinite

##
## Call:
##    felm(formula = tm_formula, data = main, weights = main$weight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42084 -0.01633 -0.00117  0.01534  0.53618
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 3.104e-02  8.654e-03   3.587 0.000346 ***
## old_biz                    -1.220e-03  5.490e-03  -0.222 0.824205
## hhsize_adj_1               -2.855e-03  3.398e-03  -0.840 0.400972
## adults_1                    6.904e-03  3.200e-03   2.158 0.031126 *
## children_1                  3.093e-03  2.204e-03   1.404 0.160622
## male_head_1                -3.783e-03  4.189e-03  -0.903 0.366605
## head_age_1                  5.416e-05  1.313e-04   0.412 0.680111
## head_noeduc_1              -3.496e-04  2.681e-03  -0.130 0.896266
## women1845_1                 1.888e-03  1.981e-03   0.953 0.340561
## anychild1318_1              4.535e-03  2.585e-03   1.755 0.079530 .
## spouse_works_wage_1         9.558e-03  3.328e-03   2.872 0.004135 **
## ownland_hyderabad_1         1.757e-02  5.339e-03   3.291 0.001021 **
## ownland_village_1          -5.692e-03  3.009e-03  -1.891 0.058755 .
```

```
## spandana_1                        NA         NA      NA       NA
## othermfi_1                        NA         NA      NA       NA
## anybank_1                         NA         NA      NA       NA
## anyinformal_1             -5.908e-03  3.957e-03  -1.493 0.135624
## anyloan_1                 -2.019e-03  4.274e-03  -0.472 0.636708
## everlate_1                -5.550e-03  2.478e-03  -2.240 0.025262 *
## mfi_loan_cycles_1         -1.376e-03  2.499e-03  -0.551 0.582021
## spandana_amt_1                    NA         NA      NA       NA
## othermfi_amt_1                    NA         NA      NA       NA
## bank_amt_1                        NA         NA      NA       NA
## informal_amt_1             2.974e-08  6.304e-08   0.472 0.637108
## anyloan_amt_1             -1.276e-07  5.480e-08  -2.328 0.020031 *
## total_biz_1               -3.932e-03  4.389e-03  -0.896 0.370500
## newbiz_1                  -5.902e-04  8.768e-03  -0.067 0.946343
## female_biz_1              -1.517e-03  3.850e-03  -0.394 0.693699
## female_biz_new_1           5.980e-02  1.261e-02   4.743 2.31e-06 ***
## wages_nonbiz_1            -5.013e-07  4.144e-07  -1.210 0.226617
## hours_week_biz_1           6.464e-05  3.678e-05   1.758 0.078994 .
## hours_week_outside_1      -1.808e-04  3.274e-05  -5.523 3.93e-08 ***
## hours_headspouse_outside_1 -6.098e-05  5.358e-05  -1.138 0.255283
## hours_headspouse_biz_1     1.285e-04  5.285e-05   2.432 0.015149 *
## total_exp_mo_pc_1          3.282e-05  3.139e-05   1.045 0.295974
## durables_exp_mo_pc_1       5.809e-05  3.699e-05   1.570 0.116545
## nondurable_exp_mo_pc_1    -6.439e-05  3.134e-05  -2.054 0.040106 *
## food_exp_mo_pc_1           1.420e-05  9.303e-06   1.527 0.127073
## health_exp_mo_pc_1        -3.990e-05  1.674e-05  -2.384 0.017258 *
## temptation_exp_mo_pc_1     6.872e-05  1.634e-05   4.207 2.75e-05 ***
## festival_exp_mo_pc_1       8.031e-05  2.287e-05   3.512 0.000458 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04395 on 1481 degrees of freedom
## Multiple R-squared(full model): 0.2303   Adjusted R-squared: 0.2126
## Multiple R-squared(proj model): 0.2303   Adjusted R-squared: 0.2126
## F-statistic(full model):13.03 on 34 and 1481 DF, p-value: < 2.2e-16
## F-statistic(proj model): 3.008 on 40 and 1481 DF, p-value: 1.686e-09
```

```r
summary(tm_gates_biz[[2]])
```

```
##
## Call:
##    felm(formula = gates_formula, data = main, weights = main$weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8038 -0.0254  0.0228  0.0359  3.3986
##
## Coefficients:
##           Estimate Cluster s.e. t value Pr(>|t|)
## baseline  1.200743     0.036475  32.920   <2e-16 ***
## cte       0.437941     0.225216   1.945    0.052 .
## G1        0.018615     0.030054   0.619    0.536
## G2       -0.001261     0.009898  -0.127    0.899
## G3       -0.011524     0.013571  -0.849    0.396
## G4        0.010626     0.026085   0.407    0.684
```

```
## G5       -0.017062     0.034360  -0.497    0.620
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1977 on 1509 degrees of freedom
## Multiple R-squared(full model): 0.7323   Adjusted R-squared: 0.731
## Multiple R-squared(proj model): 0.7323   Adjusted R-squared: 0.731
## F-statistic(full model, *iid*):589.6 on 7 and 1509 DF, p-value: < 2.2e-16
## F-statistic(proj model): 273.2 on 7 and 102 DF, p-value: < 2.2e-16
```

Results (using Causal Random Forest)

```
tm_gates_biz_crf <- tm_gates("biz_index_all_1", "treatment", endline1_biz,
                             split_ratio = 0.6,
                             cluster="areaid", num.iter=1, ml_method="crf")
```

```
## Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
## rank-deficient or indefinite
```

```
summary(tm_gates_biz_crf[[1]])
```

```
## Warning in chol.default(mat, pivot = TRUE, tol = tol): the matrix is either
## rank-deficient or indefinite
```

```
##
## Call:
##    felm(formula = tm_formula, data = main, weights = main$weight)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0178043 -0.0039987  0.0002497  0.0038231  0.0293723
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.406e-03  1.195e-03   3.686 0.000236 ***
## old_biz                 -2.535e-04  7.582e-04  -0.334 0.738156
## hhsize_adj_1             6.507e-04  4.693e-04   1.387 0.165761
## adults_1                 9.204e-04  4.420e-04   2.082 0.037468 *
## children_1               1.742e-04  3.043e-04   0.572 0.567154
## male_head_1             -4.955e-04  5.785e-04  -0.857 0.391843
## head_age_1               2.339e-05  1.814e-05   1.290 0.197400
## head_noeduc_1           -5.845e-04  3.703e-04  -1.579 0.114623
## women1845_1              9.041e-05  2.736e-04   0.330 0.741080
## anychild1318_1           1.765e-03  3.570e-04   4.943 8.55e-07 ***
## spouse_works_wage_1      4.808e-04  4.596e-04   1.046 0.295611
## ownland_hyderabad_1     -3.303e-03  7.374e-04  -4.480 8.05e-06 ***
## ownland_village_1        4.962e-04  4.156e-04   1.194 0.232747
## spandana_1                      NA         NA      NA       NA
## othermfi_1                      NA         NA      NA       NA
## anybank_1                       NA         NA      NA       NA
## anyinformal_1           -2.413e-03  5.465e-04  -4.416 1.08e-05 ***
## anyloan_1               -5.128e-03  5.903e-04  -8.686  < 2e-16 ***
## everlate_1              -2.094e-04  3.423e-04  -0.612 0.540663
## mfi_loan_cycles_1       -3.174e-04  3.451e-04  -0.920 0.357835
## spandana_amt_1                  NA         NA      NA       NA
```

```
## othermfi_amt_1                        NA         NA      NA       NA
## bank_amt_1                            NA         NA      NA       NA
## informal_amt_1              -6.184e-08  8.706e-09  -7.103 1.89e-12 ***
## anyloan_amt_1                9.307e-09  7.569e-09   1.230 0.219008
## total_biz_1                  1.319e-03  6.061e-04   2.176 0.029711 *
## newbiz_1                     3.445e-03  1.211e-03   2.845 0.004503 **
## female_biz_1                 1.265e-04  5.317e-04   0.238 0.812043
## female_biz_new_1            -8.605e-04  1.741e-03  -0.494 0.621219
## wages_nonbiz_1              -7.617e-07  5.724e-08 -13.308  < 2e-16 ***
## hours_week_biz_1            -1.017e-05  5.079e-06  -2.002 0.045414 *
## hours_week_outside_1        -1.482e-05  4.521e-06  -3.277 0.001073 **
## hours_headspouse_outside_1 -3.276e-05  7.400e-06  -4.427 1.03e-05 ***
## hours_headspouse_biz_1      -2.069e-05  7.299e-06  -2.835 0.004645 **
## total_exp_mo_pc_1            1.243e-05  4.335e-06   2.867 0.004199 **
## durables_exp_mo_pc_1        -5.845e-06  5.108e-06  -1.144 0.252691
## nondurable_exp_mo_pc_1      -1.392e-05  4.329e-06  -3.217 0.001324 **
## food_exp_mo_pc_1            -3.891e-06  1.285e-06  -3.028 0.002500 **
## health_exp_mo_pc_1          -2.675e-06  2.312e-06  -1.157 0.247503
## temptation_exp_mo_pc_1       1.348e-05  2.256e-06   5.975 2.87e-09 ***
## festival_exp_mo_pc_1        -2.434e-05  3.158e-06  -7.708 2.33e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00607 on 1481 degrees of freedom
## Multiple R-squared(full model): 0.555   Adjusted R-squared: 0.5448
## Multiple R-squared(proj model): 0.555   Adjusted R-squared: 0.5448
## F-statistic(full model):54.33 on 34 and 1481 DF, p-value: < 2.2e-16
## F-statistic(proj model): 0.7823 on 40 and 1481 DF, p-value: 0.8339
```

```
summary(tm_gates_biz_crf[[2]])
```

```
##
## Call:
##    felm(formula = gates_formula, data = main, weights = main$weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4280 -0.2058 -0.1035  0.0507  3.6261
##
## Coefficients:
##             Estimate Cluster s.e. t value Pr(>|t|)
## (Intercept) -0.030719     0.012768  -2.406   0.0163 *
## cte          9.240015     1.160508   7.962  3.3e-15 ***
## G1          -0.017546     0.015348  -1.143   0.2531
## G2          -0.009313     0.032010  -0.291   0.7711
## G3          -0.062288     0.051613  -1.207   0.2277
## G4           0.010351     0.059133   0.175   0.8611
## G5           0.015779     0.057997   0.272   0.7856
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3719 on 1509 degrees of freedom
## Multiple R-squared(full model): 0.05304   Adjusted R-squared: 0.04928
## Multiple R-squared(proj model): 0.05304   Adjusted R-squared: 0.04928
## F-statistic(full model, *iid*):14.09 on 6 and 1509 DF, p-value: 1.168e-15
```

```
## F-statistic(proj model): 11.43 on 6 and 102 DF, p-value: 9.539e-10
```

CAUSAL FOREST

Var_imp_plot function

```r
var_imp_plot <- function(forest, decay.exponent = 2L, max.depth = 4L) {

  # Calculate variable importance of all features
  # (from print.R)
  split.freq <- split_frequencies(forest, max.depth)
  split.freq <- split.freq / pmax(1L, rowSums(split.freq))
  weight <- seq_len(nrow(split.freq)) ^ -decay.exponent
  var.importance <- t(split.freq) %*% weight / sum(weight)

  # Format data frame
  p <- ncol(forest$X.orig)

  var.names <- colnames(forest$X.orig)[seq_len(p)]
  if (is.null(var.names)) {
    var.names <- paste0('x', seq_len(p))
  }
  df <- tibble(Variable = var.names,
               Importance = as.numeric(var.importance)) %>%
    arrange(Importance) %>%
    mutate(Variable = factor(Variable, levels = unique(Variable)))

  # Plot results
  p <- ggplot(df, aes(Variable, Importance)) +
    geom_bar(stat = 'identity') +
    coord_flip() +
    ggtitle('Variable Importance') +
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5))
  print(p)
}
```

Trend Plots Function

```r
trend_plots <- function(crf, test) {
  # Get the variable importance table
  var_imp <- crf %>%
    variable_importance() %>%
    as.data.frame() %>%
    mutate(variable = colnames(crf$X.orig)) %>%
    arrange(desc(V1))
  # for the first four most important variable
  # create a plot that shows if there are trend of correlation
  p1 <- ggplot(test, aes(x = test[, var_imp$variable[1]], y = preds)) +
    geom_point() +
```

```
    geom_smooth(method = "loess", span = 1) +
    theme_light() +
    labs(x = var_imp$variable[1], y = "pred. CTE")
  p2 <- ggplot(test, aes(x = test[, var_imp$variable[2]], y = preds)) +
    geom_point() +
    geom_smooth(method = "loess", span = 1) +
    theme_light() +
    labs(x = var_imp$variable[2], y = "pred. CTE")
  p3 <- ggplot(test, aes(x = test[, var_imp$variable[3]], y = preds)) +
    geom_point() +
    geom_smooth(method = "loess", span = 1) +
    theme_light() +
    labs(x = var_imp$variable[3], y = "pred. CTE")
  p4 <- ggplot(test, aes(x = test[, var_imp$variable[4]], y = preds)) +
    geom_point() +
    geom_smooth(method = "loess", span = 1) +
    theme_light() +
    labs(x = var_imp$variable[4], y = "pred. CTE")

  # combine those plots
  cowplot::plot_grid(p1, p2, p3, p4, ncol = 2)
}
```

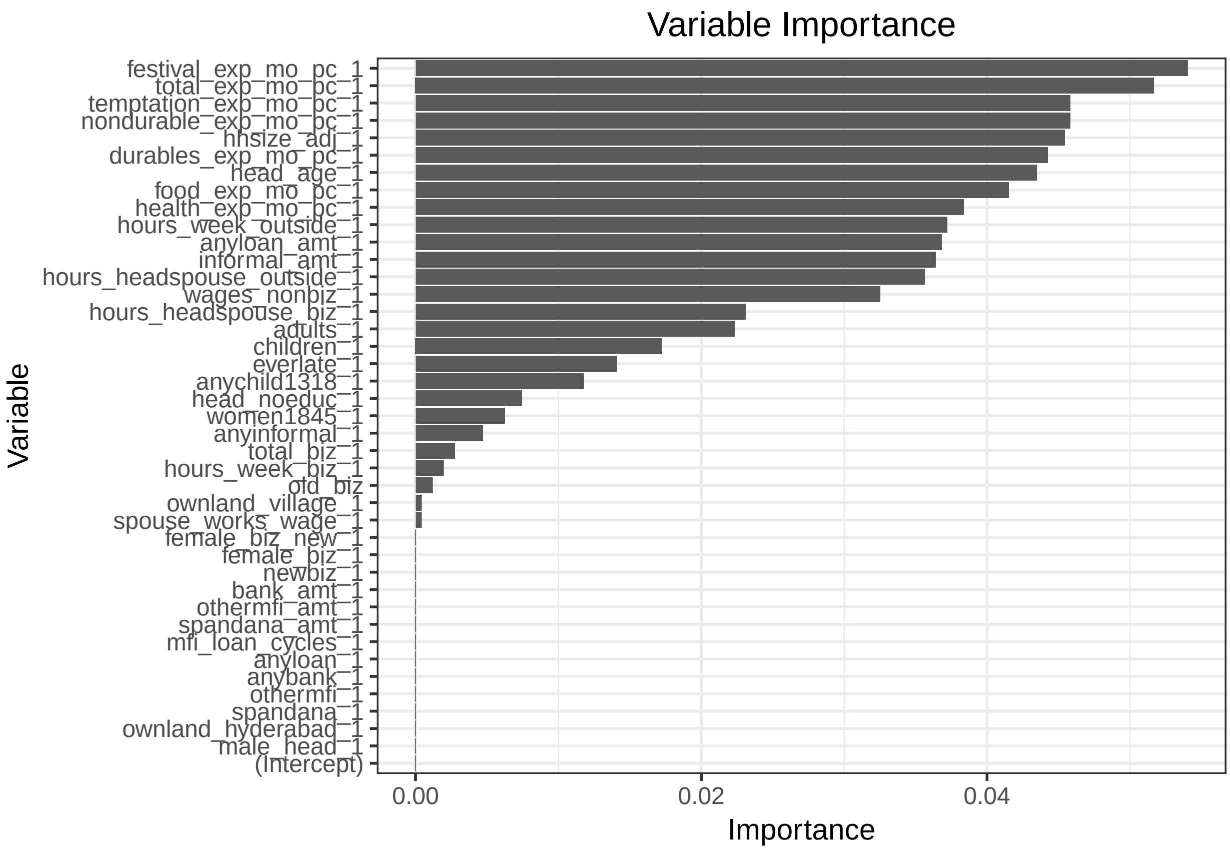
Business Index

```
target_index <- "biz_index_all_1"

endline1_biz <- endline1 %>%
  select(everything(),
         -hhid,
         -starts_with("biz"),
         -contains("index"), # prevent confounding
         target_index)       # add target index
str(endline1_biz)
```

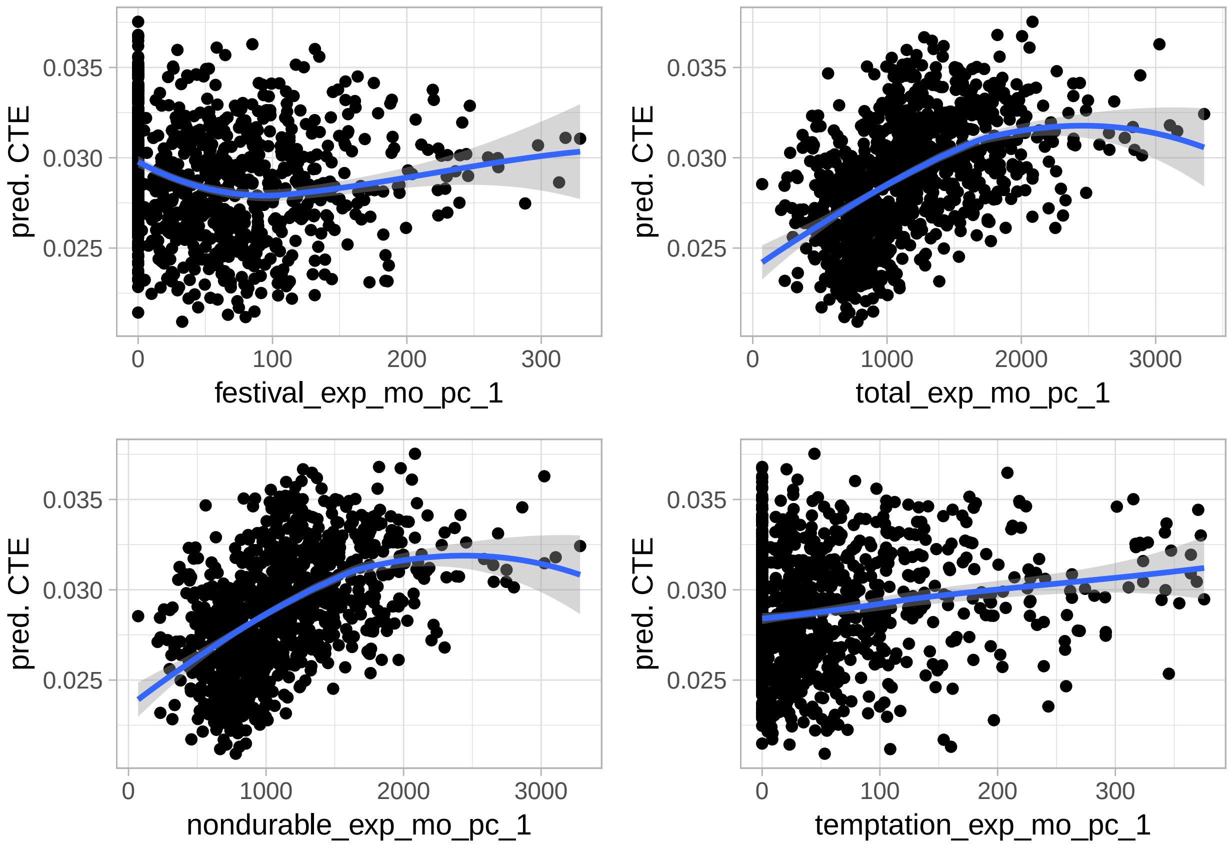
```
## 'data.frame':    3791 obs. of  43 variables:
##  $ areaid                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ treatment                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ old_biz                  : num  0 1 1 1 0 1 0 0 1 0 ...
##  $ hhsize_adj_1             : num  3.24 4.18 4.03 5.41 3.74 ...
##  $ adults_1                 : num  2 2 2 4 4 7 4 3 7 5 ...
##  $ children_1               : num  2 3 3 2 0 0 0 0 0 0 ...
##  $ male_head_1              : int  1 1 1 1 1 1 0 0 1 1 ...
##  $ head_age_1               : int  34 40 37 32 43 62 48 46 65 50 ...
##  $ head_noeduc_1            : num  0 0 0 0 1 0 1 0 1 0 ...
##  $ women1845_1              : num  1 1 1 1 2 2 1 0 2 1 ...
##  $ anychild1318_1           : num  0 1 1 1 1 1 0 0 1 0 ...
##  $ spouse_works_wage_1      : int  1 0 1 0 0 0 0 0 0 0 ...
##  $ ownland_hyderabad_1      : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ ownland_village_1        : int  0 1 0 1 0 0 0 0 0 0 ...
##  $ spandana_1               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ othermfi_1               : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ anybank_1                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ anyinformal_1              : int  0 1 1 0 0 0 1 1 0 0 ...
##  $ anyloan_1                  : num  0 0 1 1 1 1 0 1 1 1 ...
##  $ everlate_1                 : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ mfi_loan_cycles_1          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ spandana_amt_1             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ othermfi_amt_1             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bank_amt_1                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ informal_amt_1             : num  0 60000 60000 0 0 0 10000 40000 0 0 ...
##  $ anyloan_amt_1              : num  0 0 51700 23000 15000 9500 0 40000 10000 5000 ...
##  $ total_biz_1                : num  0 1 1 1 2 2 0 0 1 0 ...
##  $ newbiz_1                   : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ female_biz_1               : num  0 0 0 0 1 1 0 0 1 0 ...
##  $ female_biz_new_1           : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ wages_nonbiz_1             : int  3900 5000 1500 0 0 3950 3600 5000 2780 1500 ...
##  $ hours_week_biz_1           : num  0 21 77 70 0 152 0 0 114 0 ...
##  $ hours_week_outside_1       : num  8 0 0 0 0 0 4 118 0 70 ...
##  $ hours_headspouse_outside_1: num  4 0 63 0 0 0 0 0 0 70 ...
##  $ hours_headspouse_biz_1     : num  4 49 14 70 0 56 4 0 66 0 ...
##  $ total_exp_mo_pc_1          : num  1371 1246 1133 982 1059 ...
##  $ durables_exp_mo_pc_1       : num  9 50.8 38.3 26.3 0 ...
##  $ nondurable_exp_mo_pc_1     : num  1362 1195 1095 1132 1059 ...
##  $ food_exp_mo_pc_1           : num  671 600 625 411 612 ...
##  $ health_exp_mo_pc_1         : num  56.6 47.8 0 37 29.4 ...
##  $ temptation_exp_mo_pc_1     : num  0 23.9 0 0 26.7 ...
##  $ festival_exp_mo_pc_1       : num  84.9 39.9 82.7 12.3 22.3 ...
##  $ biz_index_all_1            : num  -0.224 0.0651 0.0827 0.3603 1.3788 ...
##  - attr(*, "na.action")= 'omit' Named int  27 109 185 239 241 290 419 511 529 612 ...
##   ..- attr(*, "names")= chr  "27" "109" "185" "239" ...
```

```r
# test/train
set.seed(123)
idx.train <- caret::createDataPartition(y = endline1_biz$treatment, p = 0.75, list = FALSE)
train <- endline1_biz[idx.train, ] # training set
test <-  endline1_biz[-idx.train, ]

# train data
Y <- train$biz_index_all_1
X <- train %>%
  select(-treatment, -target_index, -areaid)
X.clusters <- train$areaid
W <- train$treatment

# model
forest <- causal_forest(
  model.matrix(~., data = X),
  Y,
  W,
  clusters = X.clusters,
  mtry = 3,
  num.trees = 3000,
  honesty = TRUE)

var_imp_plot(forest)
```




```
# test data
test_Y <- test$biz_index_all_1
test_X <- test %>%
  select(-treatment, -target_index, -areaid)
test_clusters <- test$areaid
test_W <- test$treatment

# prediction
preds <- predict(
  object = forest,
  newdata = model.matrix(~ ., data = test_X, estimate.variance = TRUE))
test$preds <- preds$predictions
trend_plots(forest, test)
```




```
Y.forest = regression_forest(X, Y, clusters = X.clusters)
Y.hat = predict(Y.forest)$predictions
W.forest = regression_forest(X, W, clusters = X.clusters)
W.hat = predict(W.forest)$predictions

cf.raw = causal_forest(X, Y, W,
                       Y.hat = Y.hat, W.hat = W.hat,
                       clusters = X.clusters)

varimp = variable_importance(cf.raw)
selected.idx = which(varimp > mean(varimp))

cf = causal_forest(X[,selected.idx], Y, W,
                   Y.hat = Y.hat, W.hat = W.hat,
                   clusters = X.clusters,
                   samples_per_cluster = 10,
                   tune.parameters = TRUE)

tau.hat = predict(cf)$predictions
```

Confidence Interval for Average Treatment Effects

```
high_effect = tau.hat > median(tau.hat)
ate.high = average_treatment_effect(cf, subset = high_effect)
ate.low = average_treatment_effect(cf, subset = !high_effect)
```

```r
paste("95% CI for difference in ATE:",
round(ate.high[1] - ate.low[1], 3), "+/-",
round(qnorm(0.975) * sqrt(ate.high[2]^2 + ate.low[2]^2), 3))
```

```
## [1] "95% CI for difference in ATE: 0.064 +/- 0.063"
```

Best Linear Predictor

```r
test_calibration(cf)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with heteroskedasticity-robust (HC3) SEs:
##
##                                Estimate Std. Error t value Pr(>|t|)
## mean.forest.prediction          0.96282    0.45942  2.0957  0.03619 *
## differential.forest.prediction  0.70222    0.48056  1.4613  0.14405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Heterogeneity with 0.1 significance.

The Effects of `hours_week_biz_1` and `total_exp_mo_pc_1`

```r
area.mat = model.matrix(~ -1 + areaid, data = train)
area.size = colSums(area.mat)

dr.score = tau.hat + W / cf$W.hat * (Y - cf$Y.hat - (1 - cf$W.hat) * tau.hat) -
  (1 - W) / (1 - cf$W.hat) * (Y - cf$Y.hat + cf$W.hat * tau.hat)

area.score = area.mat * dr.score / area.size
```

```r
area.hours_biz = area.mat * X$hours_week_biz_1/ area.size
high_hours_biz = area.hours_biz > median(area.hours_biz)
t.test(area.score[high_hours_biz], area.score[!high_hours_biz])
```

```
##
##  Welch Two Sample t-test
##
## data:  area.score[high_hours_biz] and area.score[!high_hours_biz]
## t = 1.8499, df = 782.44, p-value = 0.06471
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.797348e-06  6.058944e-05
## sample estimates:
##    mean of x    mean of y
## 3.046754e-05 1.071498e-06
```

```r
area.total_exp = area.mat * X$total_exp_mo_pc_1 / area.size
high_total_exp = area.total_exp > median(area.total_exp)
t.test(area.score[high_total_exp], area.score[!high_total_exp])
```

```
##
##  Welch Two Sample t-test
##
## data:  area.score[high_total_exp] and area.score[!high_total_exp]
## t = 1.4898, df = 1752.1, p-value = 0.1365
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.191008e-06  3.067541e-05
## sample estimates:
##    mean of x    mean of y
## 1.557909e-05 2.336888e-06
```

```
area.food_exp = area.mat * X$food_exp_mo_pc_1 / area.size
high.food_exp = area.food_exp > median(area.food_exp)
t.test(area.score[high.food_exp], area.score[!high.food_exp])
```

```
##
##  Welch Two Sample t-test
##
## data:  area.score[high.food_exp] and area.score[!high.food_exp]
## t = 0.9228, df = 2626.9, p-value = 0.3562
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -9.229124e-06  2.563766e-05
## sample estimates:
##    mean of x    mean of y
## 1.306012e-05 4.855856e-06
```

There is heterogeneity along `hours_week_biz_1` and `total_exp_mo_pc_1` with 0.05 significance.

Credit Index

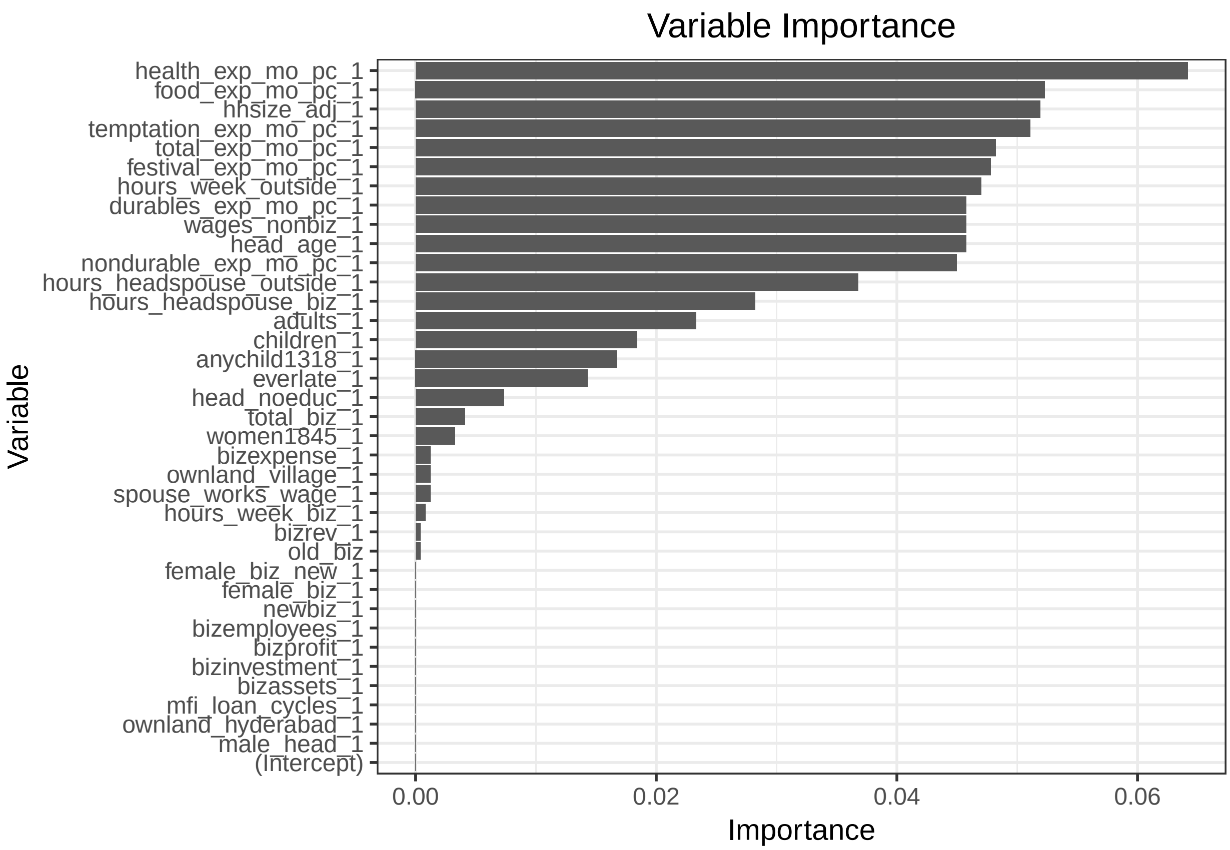
```
target_index <- "credit_index_1"

endline1_credit <- endline1 %>%
  select(everything(),
         -hhid,
         -spandana_1,
         -othermfi_1,
         -anybank_1,
         -anyinformal_1,
         -anyloan_1,
         -contains("amt"),
         -contains("index"), # prevent confounding
         target_index)       # add target index
str(endline1_credit)
```

```
## 'data.frame':    3791 obs. of  39 variables:
##  $ areaid                    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ treatment                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ old_biz                   : num  0 1 1 1 0 1 0 0 1 0 ...
##  $ hhsize_adj_1              : num  3.24 4.18 4.03 5.41 3.74 ...
##  $ adults_1                  : num  2 2 2 4 4 7 4 3 7 5 ...
##  $ children_1                : num  2 3 3 2 0 0 0 0 0 0 ...
```

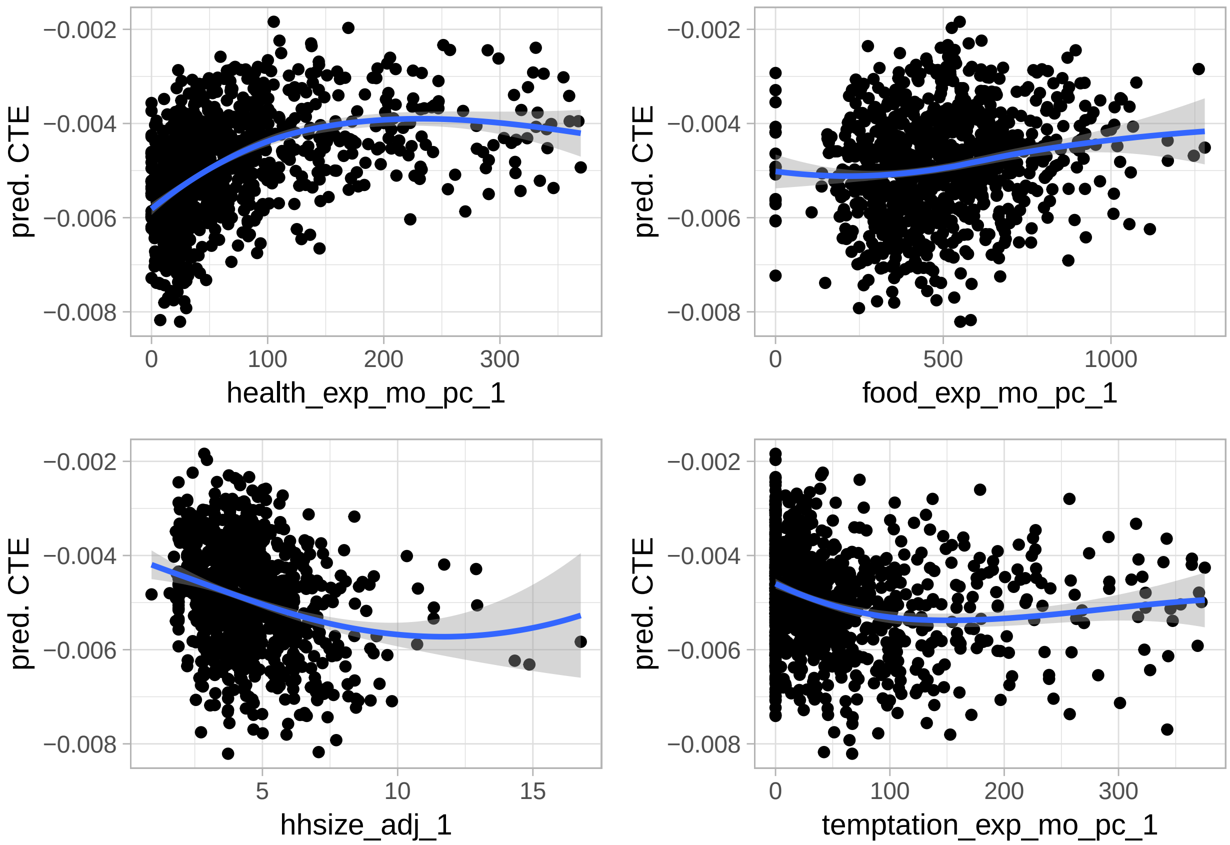
```
##  $ male_head_1                 : int  1 1 1 1 1 1 0 0 1 1 ...
##  $ head_age_1                  : int  34 40 37 32 43 62 48 46 65 50 ...
##  $ head_noeduc_1               : num  0 0 0 0 1 0 1 0 1 0 ...
##  $ women1845_1                 : num  1 1 1 1 2 2 1 0 2 1 ...
##  $ anychild1318_1              : num  0 1 1 1 1 1 0 0 1 0 ...
##  $ spouse_works_wage_1         : int  1 0 1 0 0 0 0 0 0 0 ...
##  $ ownland_hyderabad_1         : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ ownland_village_1           : int  0 1 0 1 0 0 0 0 0 0 ...
##  $ everlate_1                  : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ mfi_loan_cycles_1           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bizassets_1                 : num  0 2000 0 31700 0 0 0 0 0 0 ...
##  $ bizinvestment_1             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bizrev_1                    : num  0 1800 5000 12400 2170 2300 0 0 0 0 ...
##  $ bizexpense_1                : num  0 205 205 8750 10658 ...
##  $ bizprofit_1                 : num  0 1595 4795 3650 0 ...
##  $ bizemployees_1              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ total_biz_1                 : num  0 1 1 1 2 2 0 0 1 0 ...
##  $ newbiz_1                    : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ female_biz_1                : num  0 0 0 0 1 1 0 0 1 0 ...
##  $ female_biz_new_1            : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ wages_nonbiz_1              : int  3900 5000 1500 0 0 3950 3600 5000 2780 1500 ...
##  $ hours_week_biz_1            : num  0 21 77 70 0 152 0 0 114 0 ...
##  $ hours_week_outside_1        : num  8 0 0 0 0 0 4 118 0 70 ...
##  $ hours_headspouse_outside_1: num  4 0 63 0 0 0 0 0 0 70 ...
##  $ hours_headspouse_biz_1      : num  4 49 14 70 0 56 4 0 66 0 ...
##  $ total_exp_mo_pc_1           : num  1371 1246 1133 982 1059 ...
##  $ durables_exp_mo_pc_1        : num  9 50.8 38.3 26.3 0 ...
##  $ nondurable_exp_mo_pc_1      : num  1362 1195 1095 1132 1059 ...
##  $ food_exp_mo_pc_1            : num  671 600 625 411 612 ...
##  $ health_exp_mo_pc_1          : num  56.6 47.8 0 37 29.4 ...
##  $ temptation_exp_mo_pc_1      : num  0 23.9 0 0 26.7 ...
##  $ festival_exp_mo_pc_1        : num  84.9 39.9 82.7 12.3 22.3 ...
##  $ credit_index_1              : num  -0.492 -0.417 -0.178 -0.269 -0.274 ...
##  - attr(*, "na.action")= 'omit' Named int  27 109 185 239 241 290 419 511 529 612 ...
##   ..- attr(*, "names")= chr  "27" "109" "185" "239" ...
```

```r
# test/train
set.seed(123)
idx.train <- caret::createDataPartition(y = endline1_credit$treatment, p = 0.75, list = FALSE)
train <- endline1_credit[idx.train, ] # training set
test <-  endline1_credit[-idx.train, ]

# train data
Y <- train$credit_index_1
X <- train %>%
  select(-treatment, -target_index, -areaid)
X.clusters <- train$areaid
W <- train$treatment

# model
forest <- causal_forest(
  model.matrix(~., data = X),
  Y,
  W,
```

```
  clusters = X.clusters,
  mtry = 3,
  num.trees = 3000,
  honesty = TRUE)

var_imp_plot(forest)
```




```
# test data
test_Y <- test$credit_index_1
test_X <- test %>%
  select(-treatment, -target_index, -areaid)
test_clusters <- test$areaid
test_W <- test$treatment

# prediction
preds <- predict(
  object = forest,
  newdata = model.matrix(~ ., data = test_X, estimate.variance = TRUE))
test$preds <- preds$predictions
trend_plots(forest, test)
```




```
Y.forest = regression_forest(X, Y, clusters = X.clusters)
Y.hat = predict(Y.forest)$predictions
W.forest = regression_forest(X, W, clusters = X.clusters)
W.hat = predict(W.forest)$predictions

cf.raw = causal_forest(X, Y, W,
                       Y.hat = Y.hat, W.hat = W.hat,
                       clusters = X.clusters)

varimp = variable_importance(cf.raw)
selected.idx = which(varimp > mean(varimp))

cf = causal_forest(X[,selected.idx], Y, W,
                   Y.hat = Y.hat, W.hat = W.hat,
                   clusters = X.clusters,
                   samples_per_cluster = 10,
                   tune.parameters = TRUE)

tau.hat = predict(cf)$predictions
```

Confidence Interval for Average Treatment Effects

```
high_effect = tau.hat > median(tau.hat)
ate.high = average_treatment_effect(cf, subset = high_effect)
ate.low = average_treatment_effect(cf, subset = !high_effect)
```

```r
paste("95% CI for difference in ATE:",
round(ate.high[1] - ate.low[1], 3), "+/-",
round(qnorm(0.975) * sqrt(ate.high[2]^2 + ate.low[2]^2), 3))
```

```
## [1] "95% CI for difference in ATE: -0.009 +/- 0.038"
```

Best Linear Predictor

```r
test_calibration(cf)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with heteroskedasticity-robust (HC3) SEs:
##
##                                Estimate Std. Error t value Pr(>|t|)
## mean.forest.prediction          0.81545    1.34879  0.6046  0.54551
## differential.forest.prediction -2.03389    0.91265 -2.2286  0.02592 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is heterogeneity.

The Effects of `total_exp_mo_pc_1` and `food_exp_mo_pc_1` and `nondurable_exp_mo_pc_1`

```r
area.mat = model.matrix(~ -1 + areaid, data = train)
area.size = colSums(area.mat)

dr.score = tau.hat + W / cf$W.hat * (Y - cf$Y.hat - (1 - cf$W.hat) * tau.hat) -
  (1 - W) / (1 - cf$W.hat) * (Y - cf$Y.hat + cf$W.hat * tau.hat)

area.score = area.mat * dr.score / area.size
```

```r
area.total_exp = area.mat * X$total_exp_mo_pc_1 / area.size
high_total_exp = area.total_exp > median(area.total_exp)
t.test(area.score[high_total_exp], area.score[!high_total_exp])
```

```
##
##  Welch Two Sample t-test
##
## data:  area.score[high_total_exp] and area.score[!high_total_exp]
## t = -0.37244, df = 1959, p-value = 0.7096
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.180659e-05  8.037937e-06
## sample estimates:
##     mean of x     mean of y
## -5.070996e-06 -3.186669e-06
```

The t-test shows no significant of different treatment effects between high food expenditure and

```
area.food_exp = area.mat * X$food_exp_mo_pc_1 / area.size
high.food_exp = area.food_exp > median(area.food_exp)
t.test(area.score[high.food_exp], area.score[!high.food_exp])
```

```
##
##  Welch Two Sample t-test
##
## data:  area.score[high.food_exp] and area.score[!high.food_exp]
## t = -0.66349, df = 1948.3, p-value = 0.5071
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.327842e-05  6.565125e-06
## sample estimates:
##     mean of x     mean of y
## -5.807156e-06 -2.450510e-06
```

```
area.nondurable_exp = area.mat * X$nondurable_exp_mo_pc_1 / area.size
high.nondurable_exp = area.nondurable_exp > median(area.nondurable_exp)
t.test(area.score[high.nondurable_exp], area.score[!high.nondurable_exp])
```

```
##
##  Welch Two Sample t-test
##
## data:  area.score[high.nondurable_exp] and area.score[!high.nondurable_exp]
## t = -0.31138, df = 1962.4, p-value = 0.7555
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.149771e-05  8.346943e-06
## sample estimates:
##     mean of x     mean of y
## -4.916524e-06 -3.341141e-06
```

There is heterogeneity along `total_exp_mo_pc_1` and `food_exp_mo_pc_1` and `nondurable_exp_mo_pc_1`.

Consumption Index

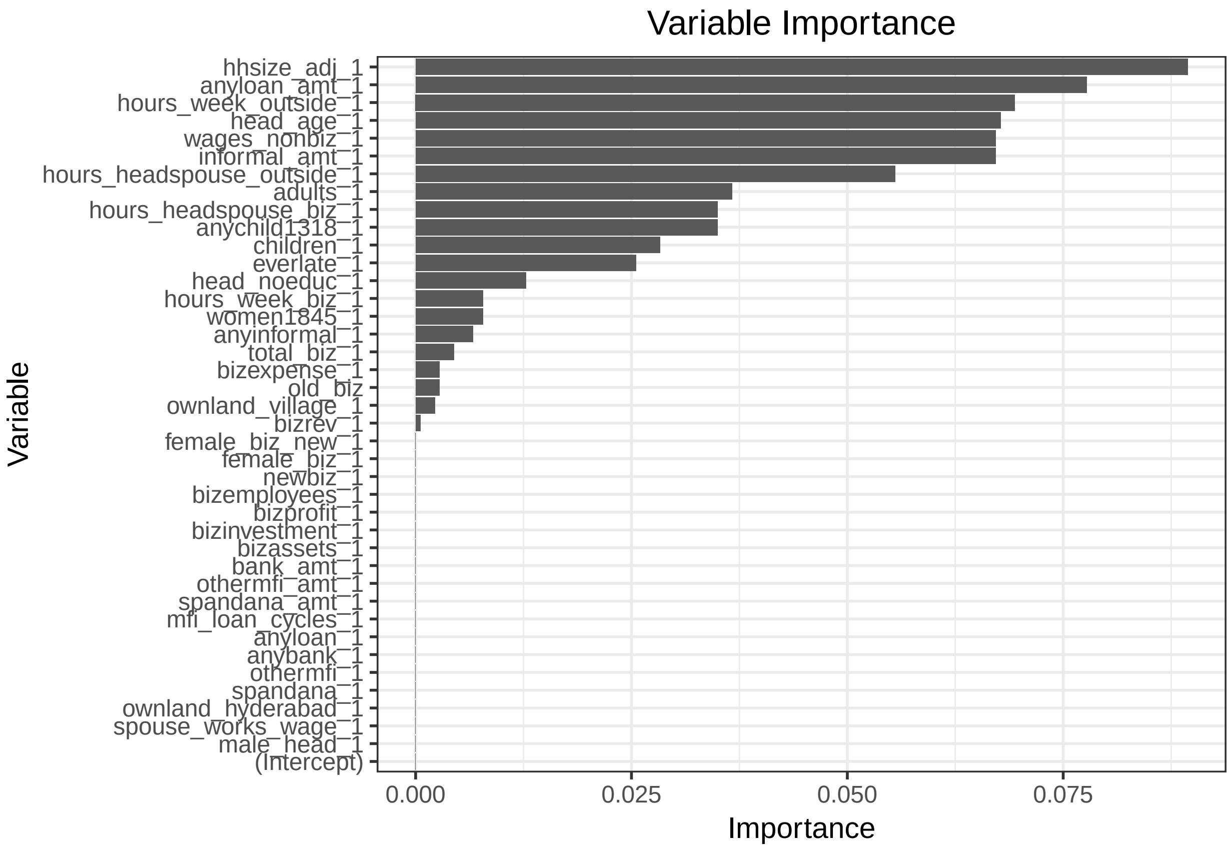
```
target_index <- "consumption_index_1"

endline1_consumption <- endline1 %>%
  select(everything(),
         -hhid,
         -contains("exp"),
         bizexpense_1,
         -contains("index"), # prevent confounding
         target_index)       # add target index
str(endline1_consumption)
```

```
## 'data.frame':    3791 obs. of  42 variables:
##  $ areaid                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ treatment               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ old_biz                 : num  0 1 1 1 0 1 0 0 1 0 ...
##  $ hhsize_adj_1            : num  3.24 4.18 4.03 5.41 3.74 ...
##  $ adults_1                : num  2 2 2 4 4 7 4 3 7 5 ...
##  $ children_1              : num  2 3 3 2 0 0 0 0 0 0 ...
```

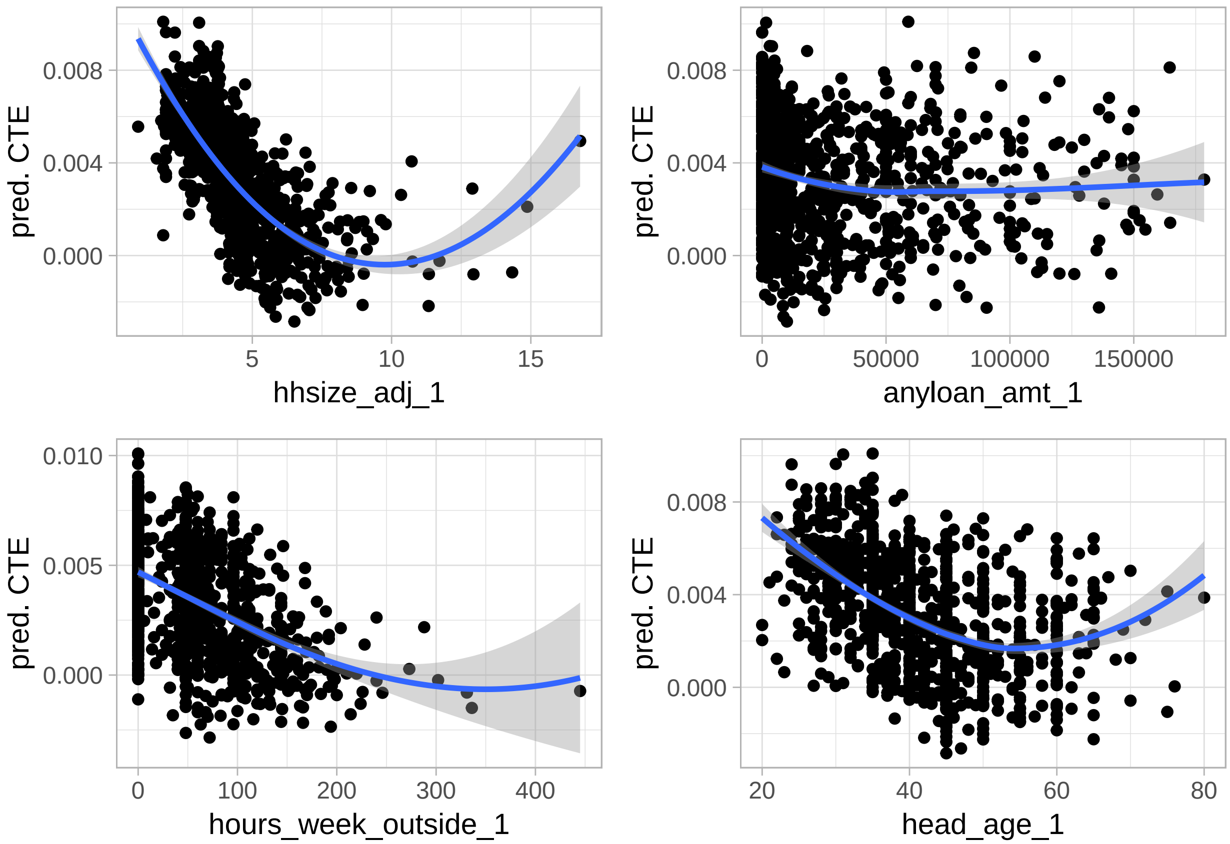
```
##  $ male_head_1                : int  1 1 1 1 1 1 0 0 1 1 ...
##  $ head_age_1                 : int  34 40 37 32 43 62 48 46 65 50 ...
##  $ head_noeduc_1              : num  0 0 0 0 1 0 1 0 1 0 ...
##  $ women1845_1                : num  1 1 1 1 2 2 1 0 2 1 ...
##  $ anychild1318_1             : num  0 1 1 1 1 1 0 0 1 0 ...
##  $ spouse_works_wage_1        : int  1 0 1 0 0 0 0 0 0 0 ...
##  $ ownland_hyderabad_1        : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ ownland_village_1          : int  0 1 0 1 0 0 0 0 0 0 ...
##  $ spandana_1                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ othermfi_1                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ anybank_1                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ anyinformal_1              : int  0 1 1 0 0 0 1 1 0 0 ...
##  $ anyloan_1                  : num  0 0 1 1 1 1 0 1 1 1 ...
##  $ everlate_1                 : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ mfi_loan_cycles_1          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ spandana_amt_1             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ othermfi_amt_1             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bank_amt_1                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ informal_amt_1             : num  0 60000 60000 0 0 0 10000 40000 0 0 ...
##  $ anyloan_amt_1              : num  0 0 51700 23000 15000 9500 0 40000 10000 5000 ...
##  $ bizassets_1                : num  0 2000 0 31700 0 0 0 0 0 0 ...
##  $ bizinvestment_1            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bizrev_1                   : num  0 1800 5000 12400 2170 2300 0 0 0 0 ...
##  $ bizprofit_1                : num  0 1595 4795 3650 0 ...
##  $ bizemployees_1             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ total_biz_1                : num  0 1 1 1 2 2 0 0 1 0 ...
##  $ newbiz_1                   : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ female_biz_1               : num  0 0 0 0 1 1 0 0 1 0 ...
##  $ female_biz_new_1           : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ wages_nonbiz_1             : int  3900 5000 1500 0 0 3950 3600 5000 2780 1500 ...
##  $ hours_week_biz_1           : num  0 21 77 70 0 152 0 0 114 0 ...
##  $ hours_week_outside_1       : num  8 0 0 0 0 0 4 118 0 70 ...
##  $ hours_headspouse_outside_1: num  4 0 63 0 0 0 0 0 0 70 ...
##  $ hours_headspouse_biz_1     : num  4 49 14 70 0 56 4 0 66 0 ...
##  $ bizexpense_1               : num  0 205 205 8750 10658 ...
##  $ consumption_index_1        : num  -0.0296 -0.088 -0.2396 -0.2112 -0.1171 ...
##  - attr(*, "na.action")= 'omit' Named int  27 109 185 239 241 290 419 511 529 612 ...
##   ..- attr(*, "names")= chr  "27" "109" "185" "239" ...
```

```r
# test/train
set.seed(123)
idx.train <- caret::createDataPartition(y = endline1_consumption$treatment, p = 0.75, list = FALSE)
train <- endline1_consumption[idx.train, ] # training set
test <-  endline1_consumption[-idx.train, ]

# train data
Y <- train$consumption_index_1
X <- train %>%
  select(-treatment, -target_index, -areaid)
X.clusters <- train$areaid
W <- train$treatment

# model
forest <- causal_forest(
```

```r
  model.matrix(~., data = X),
  Y,
  W,
  clusters = X.clusters,
  mtry = 3,
  num.trees = 3000,
  honesty = TRUE)

var_imp_plot(forest)
```




```r
# test data
test_Y <- test$consumption_index_1
test_X <- test %>%
  select(-treatment, -target_index, -areaid)
test_clusters <- test$areaid
test_W <- test$treatment

# prediction
preds <- predict(
  object = forest,
  newdata = model.matrix(~ ., data = test_X, estimate.variance = TRUE))
test$preds <- preds$predictions
trend_plots(forest, test)
```




```r
Y.forest = regression_forest(X, Y, clusters = X.clusters)
Y.hat = predict(Y.forest)$predictions
W.forest = regression_forest(X, W, clusters = X.clusters)
W.hat = predict(W.forest)$predictions

cf.raw = causal_forest(X, Y, W,
                       Y.hat = Y.hat, W.hat = W.hat,
                       clusters = X.clusters)

varimp = variable_importance(cf.raw)
selected.idx = which(varimp > mean(varimp))

cf = causal_forest(X[,selected.idx], Y, W,
                   Y.hat = Y.hat, W.hat = W.hat,
                   clusters = X.clusters,
                   samples_per_cluster = 10,
                   tune.parameters = TRUE)

tau.hat = predict(cf)$predictions
```

Confidence Interval for Average Treatment Effects

```r
high_effect = tau.hat > median(tau.hat)
ate.high = average_treatment_effect(cf, subset = high_effect)
ate.low = average_treatment_effect(cf, subset = !high_effect)
```

```r
paste("95% CI for difference in ATE:",
round(ate.high[1] - ate.low[1], 3), "+/-",
round(qnorm(0.975) * sqrt(ate.high[2]^2 + ate.low[2]^2), 3))
```

```
## [1] "95% CI for difference in ATE: -0.063 +/- 0.088"
```

Best Linear Predictor

```r
test_calibration(cf)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with heteroskedasticity-robust (HC3) SEs:
##
##                                Estimate Std. Error t value Pr(>|t|)
## mean.forest.prediction          0.42059    7.98278  0.0527   0.9580
## differential.forest.prediction -0.88460    1.12757 -0.7845   0.4328
```

No heterogeneity.

The Effects of `anyloan_amt_1` and `hhsize_adj_1`

```r
area.mat = model.matrix(~ -1 + areaid, data = train)
area.size = colSums(area.mat)

dr.score = tau.hat + W / cf$W.hat * (Y - cf$Y.hat - (1 - cf$W.hat) * tau.hat) -
  (1 - W) / (1 - cf$W.hat) * (Y - cf$Y.hat + cf$W.hat * tau.hat)

area.score = area.mat * dr.score / area.size
```

```r
area.anyloan_amt = area.mat * X$anyloan_amt_1 / area.size
high_anyloan_amt = area.anyloan_amt > median(area.anyloan_amt)
t.test(area.score[high_anyloan_amt], area.score[!high_anyloan_amt])
```

```
##
##  Welch Two Sample t-test
##
## data:  area.score[high_anyloan_amt] and area.score[!high_anyloan_amt]
## t = -0.4362, df = 2696, p-value = 0.6627
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.544176e-05  1.618228e-05
## sample estimates:
##     mean of x     mean of y
## -3.316638e-06  1.313103e-06
```

```r
area.hhsize_adj = area.mat * X$hhsize_adj_1 / area.size
high.hhsize_adj = area.hhsize_adj > median(area.hhsize_adj)
t.test(area.score[high.hhsize_adj], area.score[!high.hhsize_adj])
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  area.score[high.hhsize_adj] and area.score[!high.hhsize_adj]
## t = 1.6081, df = 2041, p-value = 0.108
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.744812e-06  3.786092e-05
## sample estimates:
##     mean of x     mean of y
##  7.528887e-06 -9.529166e-06
```

No heterogeneity along `anyloan_amt_1` and `hhsize_adj_1`.