

Decompose Right Hand Side Variables from Linear Regression

Fan Wang

2020-04-01

Contents

Decompose RHS	1
-------------------------	---

Decompose RHS

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools](#) Package, [R4Econ](#) Repository ([bookdown site](#)), or [Intro Stats with R](#) Repository.

One runs a number of regressions. With different outcomes, and various right hand side variables.

What is the remaining variation in the left hand side variable if right hand side variable one by one is set to the average of the observed values.

- Dependency: *R4Econ/linreg/ivreg/ivregdfrow.R*

The code below does not work with categorical variables (except for dummies). Dummy variable inputs need to be converted to zero/one first.

```
ff_lr_decompose <- function(df, vars.y, vars.x, vars.c, vars.z, vars.other.keep,
                             list.vars.tomean, list.vars.tomean.name.suffix,
                             df.reg.out = NULL,
                             graph=FALSE, graph.nrow=2) {

  vars.xc <- c(vars.x, vars.c)

  # Regressions
  # regf.iv from C:\Users\fan\R4Econ\linreg\ivreg\ivregdfrow.R
  if(is.null(df.reg.out)) {
    df.reg.out <- as_tibble(
      bind_rows(lapply(vars.y, regf.iv,
                       vars.x=vars.x, vars.c=vars.c, vars.z=vars.z, df=df)))
  }

  # Select Variables
  str.esti.suffix <- '_Estimate'
  arr.esti.name <- paste0(vars.xc, str.esti.suffix)
  str.outcome.name <- 'vars_var.y'
  arr.columns2select <- c(arr.esti.name, str.outcome.name)
  # arr.columns2select

  # Generate dataframe for coefficients
  df.coef <- df.reg.out[,c(arr.columns2select)] %>%
    mutate_at(vars(arr.esti.name), as.numeric) %>% column_to_rownames(str.outcome.name)
```

```

# df.coef
# str(df.coef)

# Decomposition Step 1: gather
df.decompose <- df %>%
  filter(svmthRound %in% c(12, 18, 24)) %>%
  select(one_of(c(vars.other.keep, vars.xc, vars.y))) %>%
  drop_na() %>%
  gather(variable, value, -one_of(c(vars.other.keep, vars.xc)))

# Decomposition Step 2: mutate_at(vars, funs(mean = mean(.)))
# the xc averaging could have taken place earlier, no difference in mean across variables
df.decompose <- df.decompose %>%
  group_by(variable) %>%
  mutate_at(vars(c(vars.xc, 'value')), funs(mean = mean(.))) %>%
  ungroup()

# Decomposition Step 3 With Loop
for (i in 1:length(list.vars.tomean)) {
  var.decomp.cur <- (paste0('value', list.vars.tomean.name.suffix[[i]]))
  vars.tomean <- list.vars.tomean[[i]]
  var.decomp.cur
  df.decompose <- df.decompose %>%
    mutate (!!var.decomp.cur) :=
      ff_lr_decompose_valadj(., df.coef, vars.tomean, str.esti.suffix))
}

# Additional Statistics
df.decompose.var.frac <- df.decompose %>%
  select(variable, contains('value')) %>%
  group_by(variable) %>%
  summarize_all(funs(mean = mean, var = var)) %>%
  select(variable, matches('value')) %>% select(variable, ends_with("_var")) %>%
  mutate_if(is.numeric, funs( frac = (./value_var))) %>%
  mutate_if(is.numeric, round, 3)

# Graph
g.graph.dist <- NULL
if (graph) {
  g.graph.dist <- df.decompose %>%
    select(variable, contains('value'), -value_mean) %>%
    rename(outcome = variable) %>%
    gather(variable, value, -outcome) %>%
    ggplot(aes(x=value, color = variable, fill = variable)) +
      geom_line(stat = "density") +
      facet_wrap(~ outcome, scales='free', nrow=graph.nrow)
}

# Return
return(list(dfmain = df.decompose,
           dfsumm = df.decompose.var.frac,
           graph = g.graph.dist))

```

```

}

# Support Function
ff_lr_decompose_valadj <- function(df, df.coef, vars.tomean, str.esti.suffix) {
  new_value <- (df$value +
    rowSums((df[paste0(vars.tomean, '_mean')] - df[vars.tomean])
      *df.coef[df$variable, paste0(vars.tomean, str.esti.suffix)]))
  return(new_value)
}

```

Decomposition Program

```

# Library
library(tidyverse)
library(AER)

# Load Sample Data
setwd('C:/Users/fan/R4Econ/_data/')
df <- read_csv('height_weight.csv')

```

Prepare Decomposition Data

```

## Parsed with column specification:
## cols(
##   S.country = col_character(),
##   vil.id = col_double(),
##   indi.id = col_double(),
##   sex = col_character(),
##   svymthRound = col_double(),
##   momEdu = col_double(),
##   wealthIdx = col_double(),
##   hgt = col_double(),
##   wgt = col_double(),
##   hgt0 = col_double(),
##   wgt0 = col_double(),
##   prot = col_double(),
##   cal = col_double(),
##   p.A.prot = col_double(),
##   p.A.nProt = col_double()
## )

# Source Dependency
source('C:/Users/fan/R4Econ/linreg/ivreg/ivregdfrow.R')

# Setting
options(repr.matrix.max.rows=50, repr.matrix.max.cols=50)

```

Data Cleaning.

```

# Convert Variable for Sex which is categorical to Numeric
df <- df
df$male <- (as.numeric(factor(df$sex)) - 1)
summary(factor(df$sex))

```

```
## Female    Male
```

```
## 16446 18619
```

```
summary(df$male)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.000  0.000    1.000   0.531   1.000   1.000
```

Parameters.

```
var.y1 <- c('hgt')
var.y2 <- c('wgt')
vars.y <- c(var.y1, var.y2)
vars.x <- c('prot')
vars.c <- c('male', 'wgt0', 'hgt0', 'svymthRound')
vars.other.keep <- c('S.country', 'vil.id', 'indi.id', 'svymthRound')
```

```
# Decompose sequence
```

```
vars.tomean.first <- c('male', 'hgt0')
var.tomean.first.name.suffix <- '_A'
vars.tomean.third <- c(vars.tomean.first, 'prot')
var.tomean.third.name.suffix <- '_B'
vars.tomean.fourth <- c(vars.tomean.third, 'svymthRound')
var.tomean.fourth.name.suffix <- '_C'
list.vars.tomean = list(vars.tomean.first,
                        vars.tomean.third,
                        vars.tomean.fourth)
list.vars.tomean.name.suffix <- list(var.tomean.first.name.suffix,
                                     var.tomean.third.name.suffix,
                                     var.tomean.fourth.name.suffix)
```

```
df.use <- df %>% filter(S.country == 'Guatemala') %>%
  filter(svymthRound %in% c(12, 18, 24))
vars.z <- NULL
list.out <-
  ff_lr_decompose(df=df.use, vars.y, vars.x, vars.c, vars.z, vars.other.keep,
                  list.vars.tomean, list.vars.tomean.name.suffix,
                  graph=TRUE, graph.nrow=1)
options(repr.matrix.max.rows=10, repr.matrix.max.cols=50)
list.out$dfmain
```

Example Guatemala OLS

```
## # A tibble: 1,382 x 19
```

```
##      S.country vil.id indi.id svymthRound  prot  male  wgt0  hgt0 variable value prot_mean male_mean w
##      <chr>      <dbl>  <dbl>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <chr>    <dbl>    <dbl>    <dbl>
##    1 Guatemala      3    1352          18  13.3     1 2545.   47.4 hgt      70.2     20.6     0.550
##    2 Guatemala      3    1352          24  46.3     1 2545.   47.4 hgt      75.8     20.6     0.550
##    3 Guatemala      3    1354          12   1      1 3634.   51.2 hgt      66.3     20.6     0.550
##    4 Guatemala      3    1354          18   9.8     1 3634.   51.2 hgt      69.2     20.6     0.550
##    5 Guatemala      3    1354          24  15.4     1 3634.   51.2 hgt      75.3     20.6     0.550
##    6 Guatemala      3    1356          12   8.6     1 3912.   51.9 hgt      68.1     20.6     0.550
##    7 Guatemala      3    1356          18  17.8     1 3912.   51.9 hgt      74.1     20.6     0.550
##    8 Guatemala      3    1356          24  30.5     1 3912.   51.9 hgt      77.1     20.6     0.550
##    9 Guatemala      3    1357          12   1      1 3791.   52.6 hgt      71.5     20.6     0.550
##   10 Guatemala      3    1357          18  12.7     1 3791.   52.6 hgt      77.8     20.6     0.550
```

```
## # ... with 1,372 more rows, and 2 more variables: value_B <dbl>, value_C <dbl>
```

```
options(repr.plot.width = 10, repr.plot.height = 4)
list.out$dfsumm
```

```
## # A tibble: 2 x 11
##   variable value_var value_mean_var value_A_var value_B_var value_C_var value_var_frac value_mean_var
##   <chr>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1 hgt           21.9           NA           20.3           18.4           8.40           1
## 2 wgt          2965693.           NA          2863501.       2659434.       2346297.       1
```

```
df.use <- df %>% filter(S.country == 'Guatemala') %>%
  filter(svmthRound %in% c(12, 18, 24))
vars.z <- c('vil.id')
list.out <- ff_lr_decompose(
  df=df.use, vars.y, vars.x, vars.c, vars.z, vars.other.keep,
  list.vars.tomean, list.vars.tomean.name.suffix,
  graph=TRUE, graph.nrow=1)
```

Example Guatemala IV = vil.id

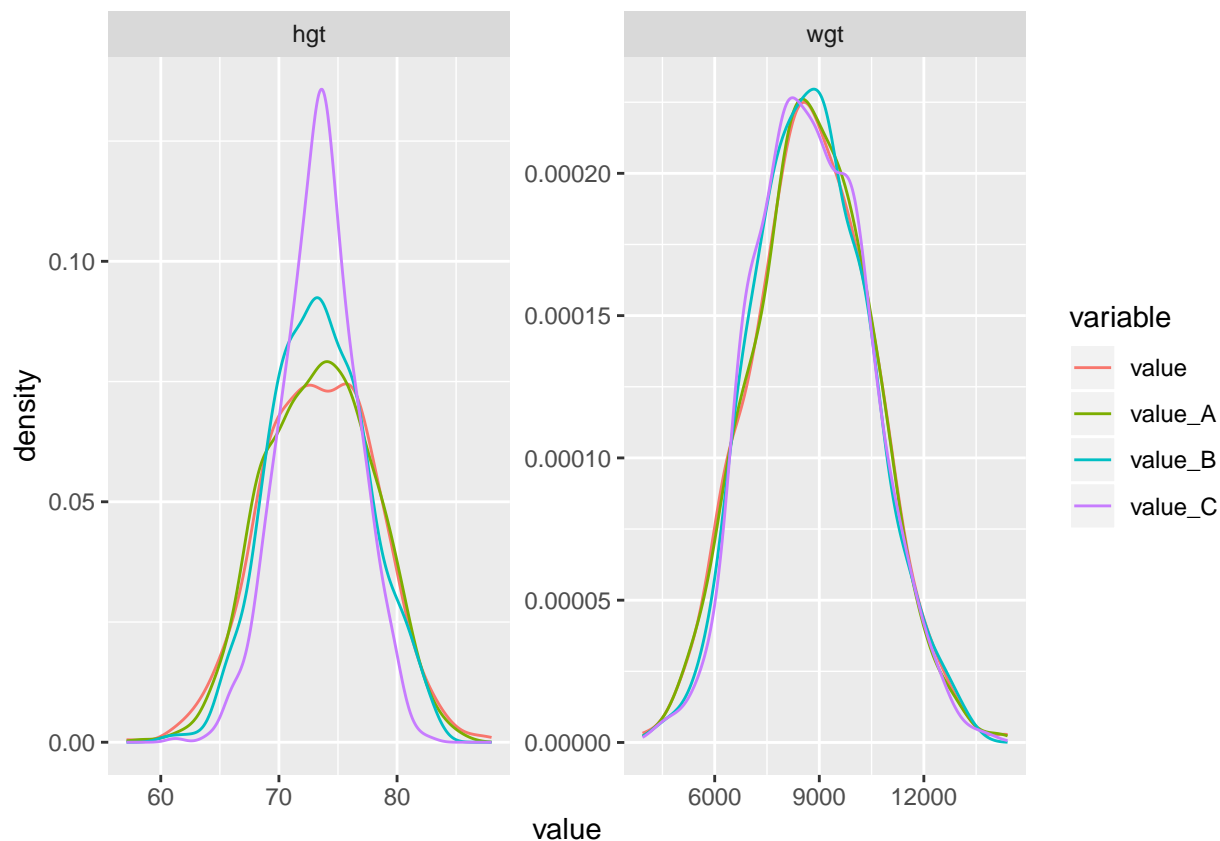
```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
list.out$dfsumm
```

```
## # A tibble: 2 x 11
##   variable value_var value_mean_var value_A_var value_B_var value_C_var value_var_frac value_mean_var
##   <chr>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1 hgt           21.9           NA           20.2           16.3           10.0           1
## 2 wgt          2965693.           NA          2876683.       2676220.       2583301.       1
```

```
options(repr.plot.width = 10, repr.plot.height = 2)
list.out$graph
```



```
df.use <- df %>% filter(S.country == 'Cebu') %>%
  filter(svmthRound %in% c(12, 18, 24))
vars.z <- NULL
list.out <- ff_lr_decompose(
  df=df.use, vars.y, vars.x, vars.c, vars.z, vars.other.keep,
  list.vars.tomean, list.vars.tomean.name.suffix,
  graph=TRUE, graph.nrow=1)
options(repr.matrix.max.rows=10, repr.matrix.max.cols=50)
list.out$dfmain
```

Example Cebu OLS

A tibble: 7,262 x 19

##	S.country	vil.id	indi.id	svmthRound	prot	male	wgt0	hgt0	variable	value	prot_mean	male_mean	w
##	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	
##	1 Cebu	1	1	12	11.3	1	2044.	44.2	hgt	70.8	17.0	0.526	
##	2 Cebu	1	2	12	5.9	0	2840.	49.7	hgt	72.2	17.0	0.526	
##	3 Cebu	1	2	18	0.5	0	2840.	49.7	hgt	76.5	17.0	0.526	
##	4 Cebu	1	2	24	14.1	0	2840.	49.7	hgt	79.2	17.0	0.526	
##	5 Cebu	1	3	12	21.4	0	3446.	51.7	hgt	68	17.0	0.526	
##	6 Cebu	1	3	18	23.6	0	3446.	51.7	hgt	71.6	17.0	0.526	
##	7 Cebu	1	3	24	20.6	0	3446.	51.7	hgt	76.7	17.0	0.526	
##	8 Cebu	1	4	12	0.7	0	3091.	50.2	hgt	69.1	17.0	0.526	
##	9 Cebu	1	4	18	7.2	0	3091.	50.2	hgt	74.3	17.0	0.526	
##	10 Cebu	1	4	24	10.3	0	3091.	50.2	hgt	78.1	17.0	0.526	

```
## # ... with 7,252 more rows, and 2 more variables: value_B <dbl>, value_C <dbl>
```

```
options(repr.plot.width = 10, repr.plot.height = 4)
list.out$dfsumm
```

```
## # A tibble: 2 x 11
```

```
##   variable value_var value_mean_var value_A_var value_B_var value_C_var value_var_frac value_mean_var
##   <chr>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1 hgt           24.4           NA           22.6           21.3           10.0           1
## 2 wgt          3337461.         NA          3218987.       3039514.       2558514.       1
```

```
df.use <- df %>% filter(S.country == 'Cebu') %>%
  filter(svmthRound %in% c(12, 18, 24))
vars.z <- c('wealthIdx')
list.out <- ff_lr_decompose(
  df=df.use, vars.y, vars.x, vars.c, vars.z, vars.other.keep,
  list.vars.tomean, list.vars.tomean.name.suffix,
  graph=TRUE, graph.nrow=1)
```

Example Cebu IV

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

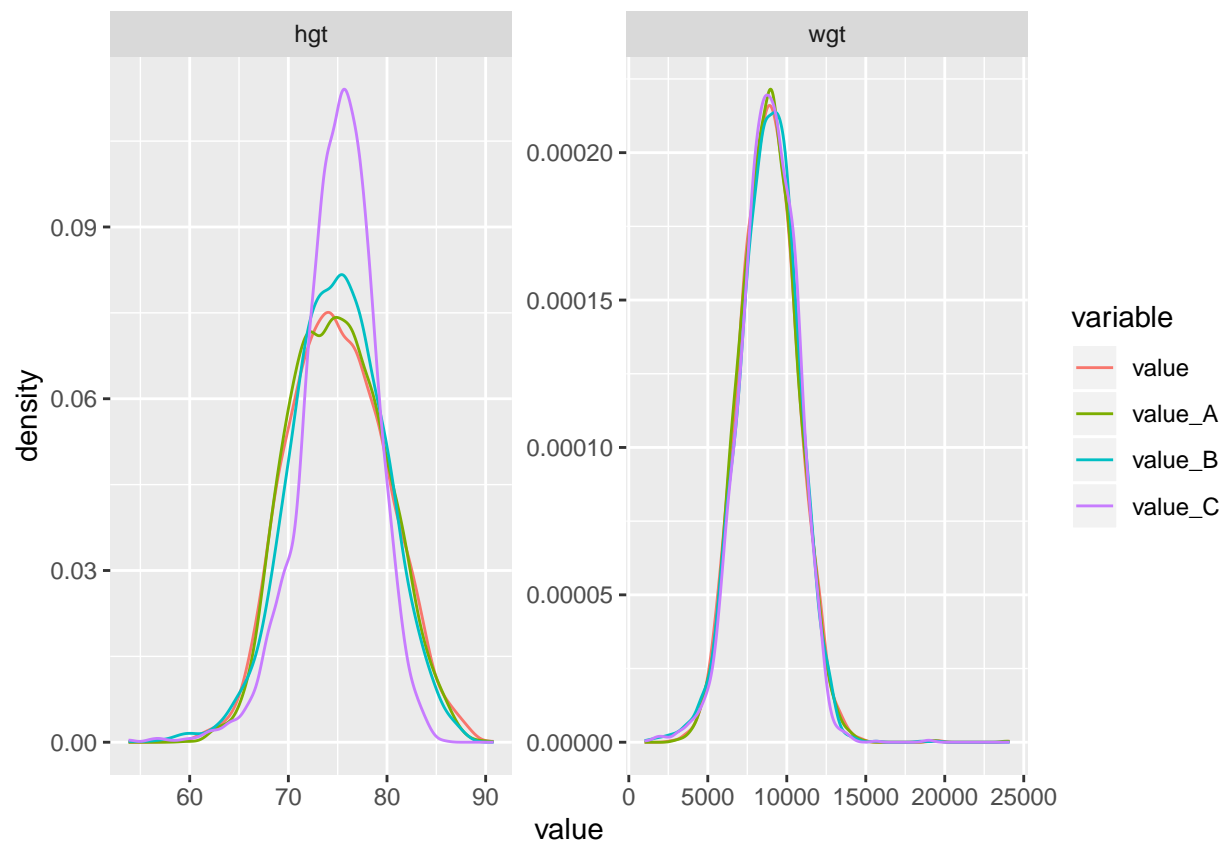
```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
list.out$dfsumm
```

```
## # A tibble: 2 x 11
```

```
##   variable value_var value_mean_var value_A_var value_B_var value_C_var value_var_frac value_mean_var
##   <chr>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1 hgt           24.4           NA           22.6           22.2           14.4           1
## 2 wgt          3337461.         NA          3237415.       3385815.       3158659.       1
```

```
options(repr.plot.width = 10, repr.plot.height = 2)
list.out$graph
```



Examples Line by Line The examples are just to test the code with different types of variables.

```
df.use <- df %>% filter(S.country == 'Guatemala') %>%
  filter(svmthRound %in% c(12, 18, 24))
dim(df.use)
```

```
## [1] 2022 16
```

Setting Up Parameters.

```
# Define Left Hand Side Variables
var.y1 <- c('hgt')
var.y2 <- c('wgt')
vars.y <- c(var.y1, var.y2)
# Define Right Hand Side Variables
vars.x <- c('prot')
vars.c <- c('male', 'wgt0', 'hgt0', 'svmthRound')
# vars.z <- c('p.A.prot')
vars.z <- c('vil.id')
# vars.z <- NULL
vars.xc <- c(vars.x, vars.c)

# Other variables to keep
vars.other.keep <- c('S.country', 'vil.id', 'indi.id', 'svmthRound')

# Decompose sequence
vars.tomean.first <- c('male', 'hgt0')
```



```

var.tomean.first.name.suffix <- '_mh02m'
vars.tomean.second <- c(vars.tomean.first, 'hgt0', 'wgt0')
var.tomean.second.name.suffix <- '_mh0me2m'
vars.tomean.third <- c(vars.tomean.second, 'prot')
var.tomean.third.name.suffix <- '_mh0mep2m'
vars.tomean.fourth <- c(vars.tomean.third, 'svymthRound')
var.tomean.fourth.name.suffix <- '_mh0mepm2m'
list.vars.tomean = list(
#           vars.tomean.first,
vars.tomean.second,
vars.tomean.third,
vars.tomean.fourth
)
list.vars.tomean.name.suffix <- list(
#           var.tomean.first.name.suffix,
var.tomean.second.name.suffix,
var.tomean.third.name.suffix,
var.tomean.fourth.name.suffix
)

```

```

# Regressions
# regf.iv from C:\Users\fan\R4Econ\linreg\ivreg\ivregdfrow.R
df.reg.out <- as_tibble(
  bind_rows(lapply(vars.y, regf.iv,
    vars.x=vars.x, vars.c=vars.c, vars.z=vars.z, df=df)))

```

Obtain Regression Coefficients from somewhere

```

## Warning: attributes are not identical across measure variables;
## they will be dropped

```

```

## Warning: attributes are not identical across measure variables;
## they will be dropped

```

```

# Regressions
# reg1 <- regf.iv(var.y = var.y1, vars.x, vars.c, vars.z, df.use)
# reg2 <- regf.iv(var.y = var.y2, vars.x, vars.c, vars.z, df.use)
# df.reg.out <- as_tibble(bind_rows(reg1, reg2))

```

```

options(repr.matrix.max.rows=50, repr.matrix.max.cols=50)
df.reg.out

```

```

## # A tibble: 2 x 37
##   X.Intercept._Es~ X.Intercept._Pr~ X.Intercept._St~ X.Intercept._zv~ hgt0_Estimate hgt0_Pr...z... hg
##   <chr>           <chr>           <chr>           <chr>           <chr>           <chr>           <chr>
## 1 22.2547168993562 8.9088080511633~ 1.21637209166939 18.2959778934199 0.6834853337~ 4.5575874740~ 0.
## 2 -1101.090058068~ 0.0051062029326~ 393.210441213089 -2.800256408938~ 75.486789661~ 3.0043362381~ 9.
## # ... with 27 more variables: male_Std.Error <chr>, male_zvalue <chr>, prot_Estimate <chr>, prot_Pr.
## #   Sargan_df1 <chr>, svymthRound_Estimate <chr>, svymthRound_Pr...z... <chr>, svymthRound_Std.Error <chr>,
## #   vars_vars.c <chr>, vars_vars.x <chr>, vars_vars.z <chr>, Weakinstruments_df1 <chr>, Weakinstruments
## #   Weakinstruments_statistic <chr>, wgt0_Estimate <chr>, wgt0_Pr...z... <chr>, wgt0_Std.Error <chr>,
## #   Wu.Hausman_df2 <chr>, Wu.Hausman_p.value <chr>, Wu.Hausman_statistic <chr>

```

```

# Select Variables
str.esti.suffix <- '_Estimate'
arr.esti.name <- paste0(vars.xc, str.esti.suffix)
str.outcome.name <- 'vars_var.y'
arr.columns2select <- c(arr.esti.name, str.outcome.name)
arr.columns2select

## [1] "prot_Estimate"          "male_Estimate"          "wgt0_Estimate"          "hgt0_Estimate"          "svymthRound_Estimate"

# Generate dataframe for coefficients
df.coef <- df.reg.out[,c(arr.columns2select)] %>% mutate_at(vars(arr.esti.name), as.numeric) %>% column_to_rownames(var = "variable")
df.coef

##      prot_Estimate male_Estimate wgt0_Estimate hgt0_Estimate svymthRound_Estimate
## hgt      -0.2714772      1.244735  0.0004430418      0.6834853      1.133919
## wgt     -59.0727542     489.852902  0.7696158110     75.4867897     250.778883

str(df.coef)

## 'data.frame':  2 obs. of  5 variables:
## $ prot_Estimate      : num  -0.271 -59.073
## $ male_Estimate      : num   1.24 489.85
## $ wgt0_Estimate      : num  0.000443 0.769616
## $ hgt0_Estimate      : num   0.683 75.487
## $ svymthRound_Estimate: num   1.13 250.78

# Decomposition Step 1: gather
df.decompose_step1 <- df.use %>%
  filter(svymthRound %in% c(12, 18, 24)) %>%
  select(one_of(c(vars.other.keep, vars.xc, vars.y))) %>%
  drop_na() %>%
  gather(variable, value, -one_of(c(vars.other.keep, vars.xc)))
options(repr.matrix.max.rows=20, repr.matrix.max.cols=20)
dim(df.decompose_step1)

```

Decomposition Step 1

```

## [1] 1382  10

df.decompose_step1

## # A tibble: 1,382 x 10
##   S.country vil.id indi.id svymthRound prot male wgt0 hgt0 variable value
##   <chr>      <dbl> <dbl>      <dbl> <dbl> <dbl> <dbl> <dbl> <chr>    <dbl>
## 1 Guatemala    3   1352      18 13.3    1 2545.  47.4 hgt      70.2
## 2 Guatemala    3   1352      24 46.3    1 2545.  47.4 hgt      75.8
## 3 Guatemala    3   1354      12  1      1 3634.  51.2 hgt      66.3
## 4 Guatemala    3   1354      18  9.8    1 3634.  51.2 hgt      69.2
## 5 Guatemala    3   1354      24 15.4    1 3634.  51.2 hgt      75.3
## 6 Guatemala    3   1356      12  8.6    1 3912.  51.9 hgt      68.1
## 7 Guatemala    3   1356      18 17.8    1 3912.  51.9 hgt      74.1
## 8 Guatemala    3   1356      24 30.5    1 3912.  51.9 hgt      77.1
## 9 Guatemala    3   1357      12  1      1 3791.  52.6 hgt      71.5
## 10 Guatemala   3   1357      18 12.7    1 3791.  52.6 hgt      77.8
## # ... with 1,372 more rows

```

```

# Decomposition Step 2: mutate_at(vars, funs(mean = mean(.)))
# the xc averaging could have taken place earlier, no difference in mean across variables
df.decompose_step2 <- df.decompose_step1 %>%
  group_by(variable) %>%
  mutate_at(vars(c(vars.xc, 'value')), funs(mean = mean(.))) %>%
  ungroup()

options(repr.matrix.max.rows=20, repr.matrix.max.cols=20)
dim(df.decompose_step2)

```

Decomposition Step 2

```
## [1] 1382 16
```

```
df.decompose_step2
```

```
## # A tibble: 1,382 x 16
##   S.country vil.id indi.id svymthRound  prot  male  wgt0  hgt0 variable value prot_mean male_mean w
##   <chr>      <dbl> <dbl>      <dbl> <dbl> <dbl> <dbl> <dbl> <chr>      <dbl>      <dbl>      <dbl>
## 1 Guatemala    3   1352        18 13.3    1 2545.  47.4 hgt       70.2       20.6      0.550
## 2 Guatemala    3   1352        24 46.3    1 2545.  47.4 hgt       75.8       20.6      0.550
## 3 Guatemala    3   1354        12  1      1 3634.  51.2 hgt       66.3       20.6      0.550
## 4 Guatemala    3   1354        18  9.8    1 3634.  51.2 hgt       69.2       20.6      0.550
## 5 Guatemala    3   1354        24 15.4    1 3634.  51.2 hgt       75.3       20.6      0.550
## 6 Guatemala    3   1356        12  8.6    1 3912.  51.9 hgt       68.1       20.6      0.550
## 7 Guatemala    3   1356        18 17.8    1 3912.  51.9 hgt       74.1       20.6      0.550
## 8 Guatemala    3   1356        24 30.5    1 3912.  51.9 hgt       77.1       20.6      0.550
## 9 Guatemala    3   1357        12  1      1 3791.  52.6 hgt       71.5       20.6      0.550
## 10 Guatemala   3   1357        18 12.7    1 3791.  52.6 hgt       77.8       20.6      0.550
## # ... with 1,372 more rows
```

```

ff_lr_decompose_valadj <- function(df, df.coef, vars.tomean, str.esti.suffix) {
  new_value <- (df$value +
    rowSums((df[paste0(vars.tomean, '_mean')] - df[vars.tomean])
      *df.coef[df$variable, paste0(vars.tomean, str.esti.suffix)]))
  return(new_value)
}

## # Decomposition Step 3: mutate_at(vars, funs(mean = mean(.)))
# var.decomp.one <- (paste0('value', list.vars.tomean.name.suffix[[1]]))
# var.decomp.two <- (paste0('value', list.vars.tomean.name.suffix[[2]]))
# var.decomp.thr <- (paste0('value', list.vars.tomean.name.suffix[[3]]))
# df.decompose_step3 <- df.decompose_step2 %>%
#   mutate((!var.decomp.one) := f_decompose_here(., df.coef, list.vars.tomean[[1]]
#   (!var.decomp.two) := f_decompose_here(., df.coef, list.vars.tomean[[2]]
#   (!var.decomp.thr) := f_decompose_here(., df.coef, list.vars.tomean[[3]]

# options(repr.matrix.max.rows=10, repr.matrix.max.cols=20)
# dim(df.decompose_step3)
# df.decompose_step3

```

Decomposition Step 3 Non-Loop

```
df.decompose_step3 <- df.decompose_step2
for (i in 1:length(list.vars.tomean)) {
  var.decomp.cur <- (paste0('value', list.vars.tomean.name.suffix[[i]]))
  vars.tomean <- list.vars.tomean[[i]]
  var.decomp.cur
  df.decompose_step3 <- df.decompose_step3 %>%
    mutate(!!var.decomp.cur) :=
      ff_lr_decompose_valadj(., df.coef, vars.tomean, str.esti.suffix))
}
options(repr.matrix.max.rows=10, repr.matrix.max.cols=20)
dim(df.decompose_step3)
```

Decomposition Step 3 With Loop

```
## [1] 1382 19
```

```
df.decompose_step3
```

```
## # A tibble: 1,382 x 19
##   S.country vil.id indi.id svymthRound  prot  male  wgt0  hgt0 variable value prot_mean male_mean w
##   <chr>      <dbl>  <dbl>      <dbl> <dbl> <dbl> <dbl> <dbl> <chr>      <dbl>      <dbl>      <dbl>
## 1 Guatemala    3    1352         18 13.3    1 2545.  47.4 hgt      70.2      20.6      0.550
## 2 Guatemala    3    1352         24 46.3    1 2545.  47.4 hgt      75.8      20.6      0.550
## 3 Guatemala    3    1354         12  1      1 3634.  51.2 hgt      66.3      20.6      0.550
## 4 Guatemala    3    1354         18  9.8    1 3634.  51.2 hgt      69.2      20.6      0.550
## 5 Guatemala    3    1354         24 15.4    1 3634.  51.2 hgt      75.3      20.6      0.550
## 6 Guatemala    3    1356         12  8.6    1 3912.  51.9 hgt      68.1      20.6      0.550
## 7 Guatemala    3    1356         18 17.8    1 3912.  51.9 hgt      74.1      20.6      0.550
## 8 Guatemala    3    1356         24 30.5    1 3912.  51.9 hgt      77.1      20.6      0.550
## 9 Guatemala    3    1357         12  1      1 3791.  52.6 hgt      71.5      20.6      0.550
## 10 Guatemala   3    1357         18 12.7    1 3791.  52.6 hgt      77.8      20.6      0.550
## # ... with 1,372 more rows, and 3 more variables: value_mh0me2m <dbl>, value_mh0mep2m <dbl>, value_mh0mepm2m <dbl>
```

```
df.decompose_step3 %>%
  select(variable, contains('value')) %>%
  group_by(variable) %>%
  summarize_all(funs(mean = mean, var = var)) %>%
  select(matches('value')) %>% select(ends_with("_var")) %>%
  mutate_if(is.numeric, funs( frac = (./value_var))) %>%
  mutate_if(is.numeric, round, 3)
```

Decomposition Step 4 Variance

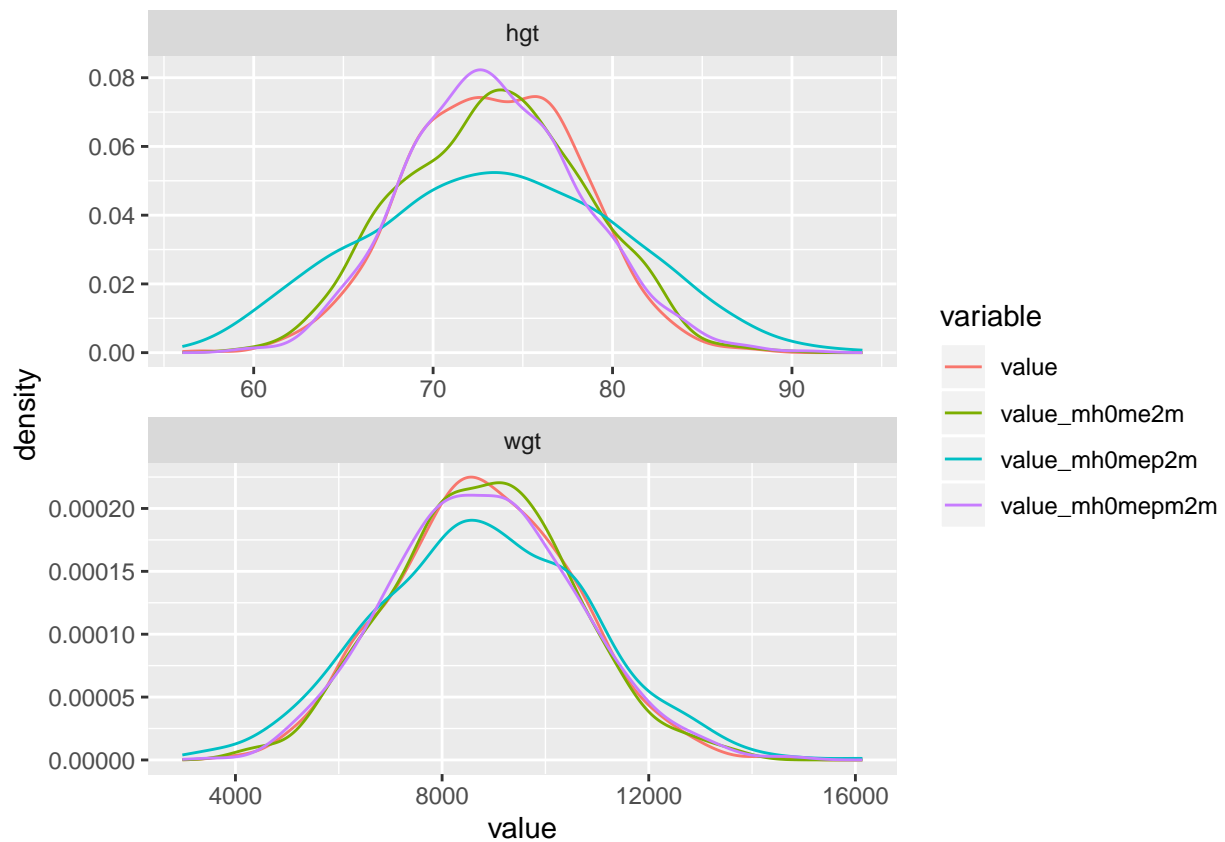
```
## # A tibble: 2 x 10
##   value_var value_mean_var value_mh0me2m_v~ value_mh0mep2m~ value_mh0mepm2m~ value_var_frac value_mh0mepm2m_var_frac
##   <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1    21.9        NA        25.4        49.0        23.1          1
## 2 2965693.      NA    2949188.    4192770.    3147507.          1
## # ... with 1 more variable: value_mh0mepm2m_var_frac <dbl>
```

Graphical Results Graphically, difficult to pick up exact differences in variance, a 50 percent reduction in variance visually does not look like 50 percent. Intuitively, we are kind of seeing standard deviation, not variance on the graph if we think about the x-scale.

```
df.decompose_step3 %>%
  select(variable, contains('value'), -value_mean)

## # A tibble: 1,382 x 5
##   variable value value_mh0me2m value_mh0mep2m value_mh0mepm2m
##   <chr>     <dbl>     <dbl>         <dbl>         <dbl>
## 1 hgt       70.2       73.2          71.2          71.7
## 2 hgt       75.8       78.8          85.8          79.4
## 3 hgt       66.3       63.6          58.3          65.6
## 4 hgt       69.2       66.5          63.6          64.1
## 5 hgt       75.3       72.6          71.2          64.9
## 6 hgt       68.1       64.3          61.1          68.4
## 7 hgt       74.1       70.3          69.6          70.0
## 8 hgt       77.1       73.3          76.0          69.7
## 9 hgt       71.5       66.8          61.5          68.8
## 10 hgt      77.8       73.1          71.0          71.5
## # ... with 1,372 more rows
```

```
options(repr.plot.width = 10, repr.plot.height = 4)
df.decompose_step3 %>%
  select(variable, contains('value'), -value_mean) %>%
  rename(outcome = variable) %>%
  gather(variable, value, -outcome) %>%
  ggplot(aes(x=value, color = variable, fill = variable)) +
    geom_line(stat = "density") +
    facet_wrap(~ outcome, scales='free', nrow=2)
```



```
head(df.decompose_step2[vars.tomean.first],3)
```

Additional Decomposition Testings

```
## # A tibble: 3 x 2
##   male hgt0
##   <dbl> <dbl>
## 1     1  47.4
## 2     1  47.4
## 3     1  51.2
```

```
head(df.decompose_step2[paste0(vars.tomean.first, '_mean')], 3)
```

```
## # A tibble: 3 x 2
##   male_mean hgt0_mean
##   <dbl>    <dbl>
## 1    0.550    49.8
## 2    0.550    49.8
## 3    0.550    49.8
```

```
head(df.coef[df.decompose_step2$variable,
             paste0(vars.tomean.first, str.esti.suffix)], 3)
```

```
##      male_Estimate hgt0_Estimate
## hgt      1.244735    0.6834853
## hgt.1     1.244735    0.6834853
## hgt.2     1.244735    0.6834853
```

```
df.decompose.tomean.first <- df.decompose_step2 %>%
  mutate(pred_new = df.decompose_step2$value +
    rowSums((df.decompose_step2[paste0(vars.tomean.first, '_mean')]
      - df.decompose_step2[vars.tomean.first])
      *df.coef[df.decompose_step2$variable,
        paste0(vars.tomean.first, str.esti.suffix)])) %>%
  select(variable, value, pred_new)
head(df.decompose.tomean.first, 10)
```

```
## # A tibble: 10 x 3
##   variable value pred_new
##   <chr>    <dbl>    <dbl>
## 1 hgt      70.2    71.2
## 2 hgt      75.8    76.8
## 3 hgt      66.3    64.7
## 4 hgt      69.2    67.6
## 5 hgt      75.3    73.7
## 6 hgt      68.1    66.1
## 7 hgt      74.1    72.1
## 8 hgt      77.1    75.1
## 9 hgt      71.5    69.0
## 10 hgt     77.8    75.3
```

```
df.decompose.tomean.first %>%
  group_by(variable) %>%
  summarize_all(funs(mean = mean, sd = sd))
```

```
## # A tibble: 2 x 5
```

```
##   variable value_mean pred_new_mean value_sd pred_new_sd
##   <chr>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 hgt        73.4        73.4        4.68        4.53
## 2 wgt       8808.        8808.       1722.       1695.
```

Note the r-square from regression above matches up with the 1 - ratio below. This is the proper decomposition method that is equivalent to r2.

```
df.decompose_step2 %>%
  mutate(pred_new = df.decompose_step2$value +
    rowSums((df.decompose_step2[paste0(vars.tomean.second, '_mean')]
      - df.decompose_step2[vars.tomean.second])
      *df.coef[df.decompose_step2$variable,
        paste0(vars.tomean.second, str.esti.suffix)])) %>%
  select(variable, value, pred_new) %>%
  group_by(variable) %>%
  summarize_all(funs(mean = mean, var = var)) %>%
  mutate(ratio = (pred_new_var/value_var))
```

```
## # A tibble: 2 x 6
##   variable value_mean pred_new_mean value_var pred_new_var ratio
##   <chr>      <dbl>      <dbl>      <dbl>      <dbl> <dbl>
## 1 hgt        73.4        73.4        21.9        25.4  1.16
## 2 wgt       8808.        8808.     2965693.    2949188. 0.994
```