Thesis

Kyle Brinkman

6/25/2020

S8A__hh__income

Here we have data on household income sources by person. we tie this to hhid to apply into the larger FD set

Household income sources, descriptions taken from “Baseline survey correct names” Kindly explain the major sources of income are for the household. Could you please provide an estimate of the contribution of different household members in total household income in the past 12 months Main income of the household S8Aq1 Husband S8Aq2 Wife S8Aq3 Children S8Aq4 Other S8Aq5 Specify S8Aq6

For Aq1, Likely who does most work 1=adult males in household, 2=adult females in household, 3=male children in household, 4=female children in household, 5= male/female children in household, 6=hired labor, -66=other, specify

##Demographic data

descriptions taken from “Baseline survey correct names”

```r
S2_hh_demography <- read_dta("Kenya_FD_materials/BaselineSurvey/S2_hh_demography.dta")

# length(unique(S2_hh_demography$hhid))
#122 unique households, there are 120 hhids in the entire full set.

#Getting HH Head
hhhead <- S2_hh_demography %>% filter(S2q9 == 1) %>%
  select(hhid = hhid,
         gender = S2q2,
         age = S2q3,
         maritalstatus = S2q10,
         job1 = S2q5,
         job2 = S2q6
         )
#getting hh composition by age, gender, and place in hh
hhcomp <- S2_hh_demography %>%
  select(hhid = hhid,
         gender = S2q2,
         age = S2q3,
         hhrelation = S2q9
         )
#using age to set weigth of individual
hhcomp <- hhcomp %>% group_by(hhid) %>%
  mutate(weight = ifelse(age<5, 0.26, ifelse(age >= 4.99 & age < 15, 0.65,1)))
#now set household weight by the individuals residing in it
```

```
hhcomp <- hhcomp %>% group_by(hhid) %>%
  mutate(wcap = sum(weight),
         cap = n())
```

These codes will be used to isolate the sample to CSA-able income generating activities assigned to hhids

Gender: Male=1 / Female=2 S2Aq2 Age (years) S2Aq3 Highest level of education attained (code A)S2Aq4 Primary occupation (code B)S2Aq5 Secondary occupation (code B)S2Aq6 Labor contribution to occupations (code C)[prim]S2Aq7 How many months has [name] been in location in past year? S2Aq8 Relation to HH (code D) S2Aq9 Marital status (>12yrs)(code E)S2Aq10

Codes A: 1 = No formal schooling, 2 = Primary incomplete, 3 = Primary complete, 4 = Secondary incomplete, 5 = Secondary complete, 6 = Tertiary /university incomplete, 7 = tertiary/University complete, 8 =Adult education incomplete, 9 = Adult education complete, 10 = Don’t know

Codes B: 1=farming crop, 2=farming livestock, 3=salaried employment, 4=self-employed off-farm, 5=casual laborer on-farm, 6=casual labored off-farm, 7=school/college, 8=non-school child, 9=herding, 10-household chores, 11=other, specify

Codes C: 1=full time, 2=part time, 3=occasionally, 4=not a worker

S2q9 Codes D: 1= Household head, 2= Spouse, 3= Son/daughter, 4= Parent living with son/daughter, 5= Son/daughter in-law, 6=Grandchild, 7= other relative, 8= Hired worker 9= Other, specify_

Codes E: 0=single, 1=married, 2=widowed, 3=divorced, 4=other, specify

age groups 0-4 years are weighted as 0.24 of an adult, children aged 5-14 years be weighted as 0.65 and all people aged 15 years and older be assigned a value of unity.

##Crop production

Here we take crop production metrics such as size of plots and crop type by household

```
S3A_crop_prod <- read_dta("Kenya_FD_materials/BaselineSurvey/S3A_crop_prod.dta")

#This contains information on each household’s plots, crops in them, the size, and who works them mainl
crop_prod <- S3A_crop_prod %>%
  select(hhid, S3Aq1, S3Aq4, S3Aq5, S3Aq8) %>%
  rename(plotnumber = S3Aq1,
         crop = S3Aq4,
         cropsize = S3Aq5,
         mainworker = S3Aq8) %>%
  group_by(hhid) %>%
  mutate(totalplot = sum(cropsize),
         plots = length((unique(plotnumber)))
         )

#We take the total size of land and number of plots each household has
hhplots <- crop_prod %>%
  select(hhid, totalplot, plots) %>%
  distinct()

# summary(hhplots)
# The median total plot size is 2 and the mean is slightly higher
# p.plotsize <-
#   ggplot(hhplots, aes(x = totalplot)) +
#   geom_histogram() +
#   geom_vline(aes(xintercept=3),
#             color="blue", linetype="dashed", size=1)
# p.plotsize

# The distribution tapers off quickly after 3;there’s one household, 101, which has a huge amount of la

#Here’s a created indicator for a large amount of land use
hhplots <- hhplots %>% ungroup() %>%
  mutate(bigland = ifelse(totalplot > 3, 1, 0))

# table(hhplots$bigland)
```

122 farming households, ALL HHIDS are farmers mainworker is “Who does the most work”, like S8 income source 1=adult males in household, 2=adult females in household, 3=male children in household, 4=female children in household, 5= male/female children in household, 6=hired labor, -66=other, specify

##Financial Diary Data

We’ll first use FD data to create variables on CSA participation and CSV indicators

```
##       hhid           usecount
##  Min.   :  3.00   Min.   : 1.00
##  1st Qu.: 32.50   1st Qu.: 1.00
##  Median : 62.00   Median : 3.00
##  Mean   : 63.72   Mean   : 3.24
##  3rd Qu.: 94.50   3rd Qu.: 4.00
##  Max.   :122.00   Max.   :15.00
```

33 non CSV, 87 CSV 45 non CSA, 75 CSA

15 non CSV, non CSA 18 non CSV, CSA 30 CSV, non CSA 57 CSV, CSA

let’s create some group variables for each, make it easier later

###Idenitifying households CSV is bianary variable on if a hhid is in a CSV village So CSA participating hhids can be idenified by taking HHIDs with CSAWhat which are not NA. From there I can say the degree of CSA participation of a household. This allows for identifying the three groups of non-CSV, CSV CSAers, CSV non-CSAers. And if any, non-CSV CSAers

##Codes

Variables FARMPROD ‘farm product’ FARMQ ‘quantity of farm product’ number of unit FARMUN ‘farm product unit’ so acres, kilograms, meters etc CSAWHAT ‘what type of CSA’ CSAQ ‘how much CSA’ CSAUN ‘Unit of CSA’ CSACOST ‘Costs of CSA’ PURBUS ‘Purchases made for business?’ SALBUS ‘Sales made for business?’

CSAUnit is 29 unique

CSA practices Most likely under CSAWhat

CS01 “High yielding varieties” CS02 “Improved seeds” CS03 “Early maturing varieties” CS04 “Drough tolerant varieties” CS05 “Pest and disease tolerant varieties” CS06 “Cross breeding Galla goat” CS07 “Cross breeding Red Masai sheep” CS08 “Fodder storage” CS09 “Intercropping” CS10 “Crop cover” CS11 “Micro catchment” CS12 “Ridges and bounds” CS13 “Hedges” CS14 “Fertilizer” CS15 “Manure and compost” CS16 “Mango trees planted” CS17 “Other trees planted” CS18 “Improved poultry, rainbow or croilers” CS24 “Other activities”

CSAWhat is likely what we want to use, 71780 rows including NA and 16 other unique variables on CSA practice

Loan type unique(FD$LoanCreditType)length(unique(FD$LoanCreditType)) LoanCreditType variable TYPS loan types TY01 “loan repayment received” TY02 “credit repayment received” TY03 “money borrowed” TY04 “gift or remittances received” TY05 “advance payment received” TY06 “CBO share received” TY07 “loan repayment paid” TY08 “credit repayment paid” TY09 “money lent” TY10 “gift or remittances given” TY11 “advance paymetn paid” TY12 “CBO contribution paid”

Income Subsetting

Here we need to create a set of incomes per household over the course of the study.

So now we have positive, incoming transactions of all types, with indicators of their affiliation with CSA activity

##Time and months

```r
Income$date <- lubridate::ymd("2019-03-05") + lubridate::weeks(Income$WeekNr)

#This gives the full date of each transaction based on the start date March 5th and given week number
#When it comes to plotting, it should be done with full date format to make sure the breaks are at the
Income$dateFloor <- lubridate::floor_date(Income$date, unit = "month")
Income$month <- lubridate::month(Income$date)
Income$year <- lubridate::year(Income$date)
#gives month set at the first and year
#Here we will also put transactions within their given months,
```

##Variables calculated per month, income, observations, mean income, CV

```
## Warning: Column ‘hhid‘ has different attributes on LHS and RHS of join
```

Now to merge with the main set. Participation to income

In this process we also take out households which we have little data on and might skew analysis due to not incorporating seasonal variation

##Quantiles

Here we break down the households into equal quintiles of mean monthly income.

```
## # A tibble: 117 x 8
##     hhid mmoInc sdmoInc    CV   CSV   CSA partGroup strata
##    <dbl>  <dbl>   <dbl> <dbl> <dbl> <dbl>     <dbl>  <dbl>
##  1     1  2690.   1659. 0.617     1     0         1      2
##  2     2  5667.   3756. 0.663     1     0         1      3
##  3     3  1933.   1565. 0.810     1     1         3      4
##  4     4  6395.   8188. 1.28      1     1         3      2
##  5     5  4759.   3882. 0.816     1     1         3      6
##  6     6  7996.   3758. 0.470     1     1         3      6
##  7     7  2411.   2003. 0.831     1     1         3      4
##  8     8  2161.   1623. 0.751     0     0         0      8
##  9     9  3012.   2178. 0.723     0     0         0      9
## 10    10   645.    468. 0.726     0     0         0      9
## # ... with 107 more rows
```

```
##      40%
## 2958.432
##      20%
## 1817.681
```

From this we not only get households into quintile groups but the lines which divide them from eachother.

Our poverty line will be between the second and third quintile, q[2]

##Poverty and poverty persistence

```
## # A tibble: 117 x 11
##     hhid mmoInc sdmoInc    CV   CSV   CSA partGroup strata povcount   obs
##    <dbl>  <dbl>   <dbl> <dbl> <dbl> <dbl>     <dbl>  <dbl>    <dbl> <int>
##  1     1  2690.   1659. 0.617     1     0         1      2        8    13
##  2     2  5667.   3756. 0.663     1     0         1      3        3    13
##  3     3  1933.   1565. 0.810     1     1         3      4       10    13
##  4     4  6395.   8188. 1.28      1     1         3      2        5    13
##  5     5  4759.   3882. 0.816     1     1         3      6        4    13
##  6     6  7996.   3758. 0.470     1     1         3      6        0    13
##  7     7  2411.   2003. 0.831     1     1         3      4        9    13
##  8     8  2161.   1623. 0.751     0     0         0      8       10    13
##  9     9  3012.   2178. 0.723     0     0         0      9        6    13
## 10    10   645.    468. 0.726     0     0         0      9       12    13
## # ... with 107 more rows, and 1 more variable: povshare <dbl>
```

##Household metrics

We track which households identify into each group and create variables by those groupings This includes in group CV and income. Additionally we count how many households fit into these groupings and sub groups.

```
hhchars <- hhchars %>% group_by(partGroup)

pg.metrics <- summarise(hhchars, mCV = mean(CV),
                         medianCV = median(CV),
          gmmoInc = mean(mmoInc),
          households = n())
pg.metrics <- left_join(pg.metrics, partGroup.index, by = "partGroup")
```

```
## Warning: Column ‘partGroup‘ has different attributes on LHS and RHS of join
```

```
#participation based metrics

hhchars <- hhchars %>% group_by(quintile)

q.metrics <- summarise(hhchars, mCV = mean(CV),
                       medianCV = median(CV),
          gmmoInc = mean(mmoInc),
          households = n())
#income quintile based metrics


hhchars <- hhchars %>% group_by(quintile, partGroup)

c.metrics <- summarise(hhchars, mCV = mean(CV),
                       medianCV = median(CV),
          gmmoInc = mean(mmoInc),
          households = n())

c.metrics <- left_join(c.metrics, partGroup.index, by = "partGroup")
```

```
## Warning: Column ‘partGroup‘ has different attributes on LHS and RHS of join
```

```
#for only csa and csv
hhchars <- hhchars %>% group_by(CSA)

metrics.csa <-  summarise(hhchars, mCV = mean(CV),
                          medianCV = median(CV),
          gmmoInc = mean(mmoInc),
          households = n())


hhchars <- hhchars %>% group_by(CSV)

metrics.csv <-  summarise(hhchars, mCV = mean(CV),
                          medianCV = median(CV),
          gmmoInc = mean(mmoInc),
          households = n())


hhchars <- ungroup(hhchars)
#combined groupings of participation and quintile metrics. Far smaller in counts, but more granular

#Now we incorporate quintile characteristics into the main set.
trunk <- hhchars %>% ungroup() %>%  select(hhid, quintile)

Income <-
  left_join(Income, trunk, by = "hhid")

Income %>% group_by(quintile)
```

```
## # A tibble: 1,394 x 16
## # Groups:   quintile [5]
##     hhid dateFloor  moInc strata   obs mmoInc sdmoInc    CV   CSV   CSA
##    <dbl> <date>     <dbl>  <dbl> <int>  <dbl>   <dbl> <dbl> <dbl> <dbl>
##  1     1 2019-03-01 1775       2    13  2690.   1659. 0.617     1     0
##  2     1 2019-04-01 3515       2    13  2690.   1659. 0.617     1     0
##  3     1 2019-05-01 1422.      2    13  2690.   1659. 0.617     1     0
##  4     1 2019-06-01 1545       2    13  2690.   1659. 0.617     1     0
##  5     1 2019-07-01 6872.      2    13  2690.   1659. 0.617     1     0
##  6     1 2019-08-01 2130       2    13  2690.   1659. 0.617     1     0
##  7     1 2019-09-01 4260       2    13  2690.   1659. 0.617     1     0
##  8     1 2019-10-01 2128.      2    13  2690.   1659. 0.617     1     0
##  9     1 2019-11-01 2938.      2    13  2690.   1659. 0.617     1     0
## 10     1 2019-12-01 4156.      2    13  2690.   1659. 0.617     1     0
## # ... with 1,384 more rows, and 6 more variables: partGroup <dbl>, pov <dbl>,
## #   povcount <dbl>, povshare <dbl>, month <chr>, quintile <int>
```

```r
summarise(Income, minc = mean(moInc))
```

```
## # A tibble: 1 x 1
##    minc
##   <dbl>
## 1 8113.
```

```r
hhchars <- left_join(hhchars, hhplots, by = "hhid")
```

```
## Warning: Column ‘hhid‘ has different attributes on LHS and RHS of join
```

```r
hhchars <- left_join(hhchars, hhhead, by = "hhid")
```

```
## Warning: Column ‘hhid‘ has different attributes on LHS and RHS of join
```

```r
unique(hhchars$gender)
```

```
## [1] "2" "1" NA
```

```r
hhchars <- hhchars %>% mutate(
  femalehh = as.numeric(gender) - 1
)
# table(hhchars$femalehh)
#1 is female
```

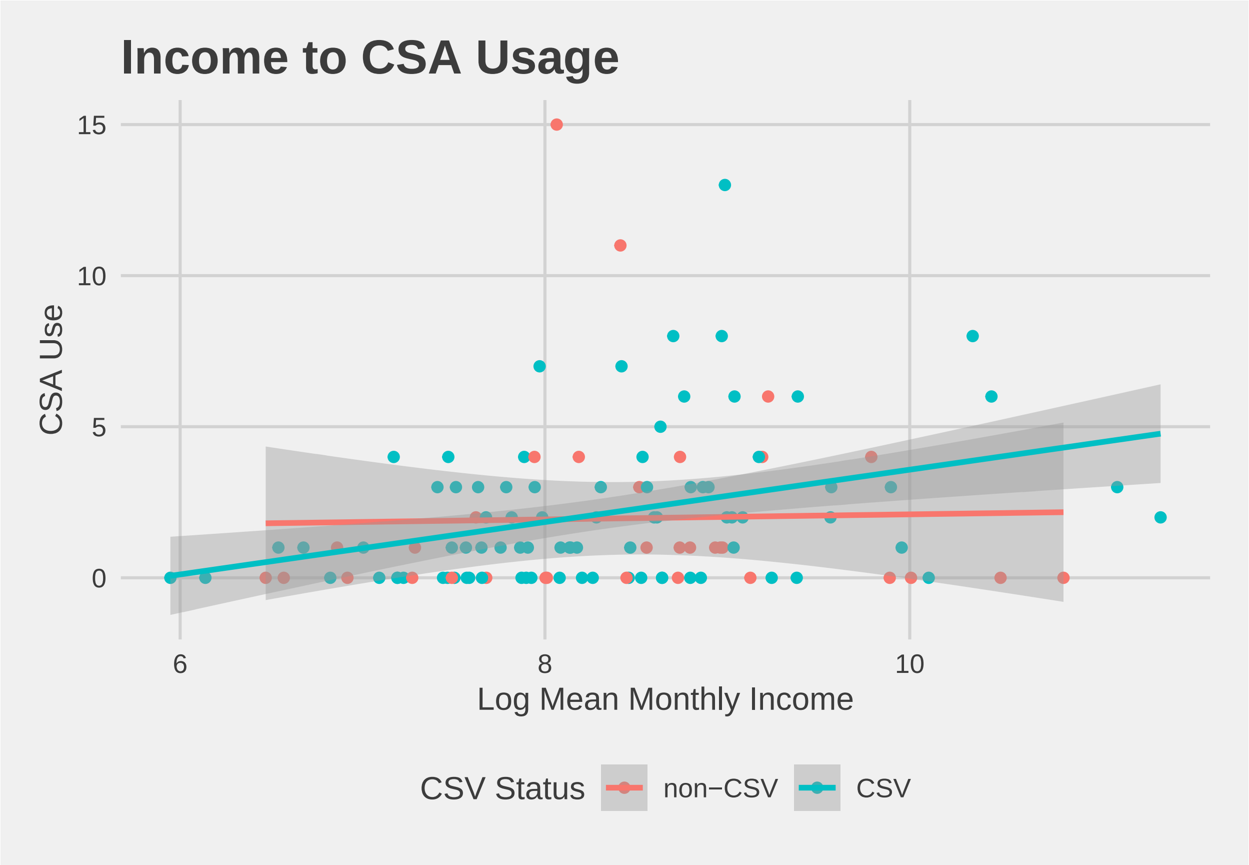
```r
hhchars <- hhchars %>%
  mutate(
    persistent = ifelse(povshare > .5, 1, 0)
  )
#about half and half
```

```r
hhchars <- hhchars %>%
  mutate(
    upperquint = ifelse(quintile > 2, 1, 0)
  )
#about half and half
```

```r
hhchars <- left_join(hhchars, CSAcount, by = "hhid")
hhchars$usecount[is.na(hhchars$usecount)] <- 0
```

```r
library(ggthemes)
p.iuc <- ggplot(hhchars, aes(x = log(mmoInc), y = usecount, color = factor(CSV))) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_fivethirtyeight() +
  labs(title = "Income to CSA Usage",y = "CSA Use", x = "Log Mean Monthly Income", color = "CSV Status")
  scale_color_discrete(name = "CSV Status", labels = c("non-CSV", "CSV")) +
  theme(axis.title = element_text())
p.iuc
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
hhcount <- left_join(hhchars, hhwcap, by = "hhid")
```

```
## Warning: Column `hhid` has different attributes on LHS and RHS of join
```

```
poorcount <- hhcount %>% ungroup() %>% filter(quintile<=2)

sum(hhcount$cap)
```

```
## [1] 701
```

```
sum(poorcount$cap)
```

```
## [1] 308
```

```
sum(poorcount$cap)/sum(hhcount$cap)*100
```

```
## [1] 43.93723
```

```
sum(poorcount$wcap)/sum(hhcount$wcap)*100
```

```
## [1] 44.76056
```

```
(nrow(poorcount))/(nrow(hhcount))*100
```

```
## [1] 40.17094
```

```
S3C_connectivity <- read_dta("Kenya_FD_materials/BaselineSurvey/S3C_connectivity.dta")

hhdist <-  S3C_connectivity %>% select(hhid, dist_fmarket = S3Cq1b)

hhchars <- left_join(hhchars, hhdist, "hhid")
```

```
## Warning: Column ‘hhid‘ has different attributes on LHS and RHS of join
```

```
offinc <- aginc %>% filter(SalesForBusiness == 1) %>% group_by(hhid) %>% mutate(agsales = total)

offinc <- offinc %>% select(hhid, agsales) %>% group_by(hhid) %>% mutate(totalag = sum(agsales))

offinc <- offinc %>% select(hhid, totalag) %>% distinct()

offinc <- left_join(hhwcap,offinc,  by = "hhid")
```

```
## Warning: Column ‘hhid‘ has different attributes on LHS and RHS of join
```

```
offinc <- left_join(hhchars,offinc,  by = "hhid")
```

```
## Warning: Column ‘hhid‘ has different attributes on LHS and RHS of join
```

```
offinc <- offinc %>% mutate(agshare = ((totalag/wcap)/(mmoInc*obs)))
offinc <- offinc %>% ungroup() %>%  select(hhid, offinc = agshare)


###2 THIS IS LIKELY "FARM"
oninc <- aginc %>% filter(SalesForBusiness == 2) %>% group_by(hhid) %>% mutate(agsales = total)

oninc <- oninc %>% select(hhid, agsales) %>% group_by(hhid) %>% mutate(totalag = sum(agsales))

oninc <- oninc %>% select(hhid, totalag) %>% distinct

oninc <- left_join(hhwcap,oninc,  by = "hhid")
```

```
## Warning: Column ‘hhid‘ has different attributes on LHS and RHS of join
```

```
oninc <- left_join(hhchars,oninc,  by = "hhid")
```

```
## Warning: Column ‘hhid‘ has different attributes on LHS and RHS of join
```

Table 1: Group Variables

	Mean CV	Max CV	Min CV	SD CV	Mean Income	Max Income	Min Income	SD Income	Mean Poverty Freq	Obs
Overall	0.77	1.69	0.35	0.27	3952.36	87134.49	382.44	12186.19	42.65	117
Participation										
non-CSA	0.82	1.54	0.35	0.29	2766.62	51168.61	382.44	10098.04	51.72	43
CSA	0.74	1.69	0.37	0.25	5045.63	87134.49	691.44	13273.23	37.38	74
non-CSV	0.77	1.54	0.35	0.27	5000.77	51168.61	644.92	10716.48	39.82	33
CSV	0.77	1.69	0.43	0.27	3602.91	87134.49	382.44	12770.11	43.77	84
Participation Overlap										
non-CSV, non-CSA	0.83	1.54	0.35	0.29	3011.88	51168.61	644.92	15150.35	45.64	15
CSV, non-CSA	0.81	1.52	0.43	0.30	2656.12	24439.98	382.44	4861.54	54.98	28
non-CSV, CSA	0.72	1.32	0.37	0.26	5722.96	17840.58	953.88	3993.02	34.97	18
CSV, CSA	0.75	1.69	0.43	0.25	4647.65	87134.49	691.44	15057.79	38.16	56
Household Head Gender										
Male	0.78	1.54	0.37	0.27	3873.89	87134.49	463.35	11384.69	42.63	93
Female	0.75	1.69	0.35	0.28	4662.12	68704.90	382.44	15337.70	41.25	23
Income										
Low-income	0.72	1.41	0.37	0.25	1814.61	2936.62	382.44	716.72	76.07	47
Higher-income	0.80	1.69	0.35	0.28	6622.01	87134.49	2991.15	14501.48	20.22	70
Plot size										
Smaller land	0.77	1.69	0.35	0.27	4279.06	87134.49	382.44	13550.60	41.68	92
Larger land	0.77	1.38	0.37	0.26	3653.12	11852.64	712.03	3300.64	46.26	25

Table 2: ANOVA of CV

	Sum Sq	Df	F value	Pr(>F)
femalehh	0.022	1	0.298	0.587
CSA	0.263	1	3.618	0.060
CSV	0.006	1	0.077	0.781
bigland	0.000	1	0.001	0.979
upperquint	0.223	1	3.072	0.082
CSA:CSV	0.013	1	0.184	0.669
Residuals	7.928	109	NA	NA

```
oninc <- oninc %>% mutate(agshare = ((totalag/wcap)/(mmoInc*obs)))
oninc <- oninc %>% ungroup() %>%  select(hhid, oninc = agshare)


hhchars <- left_join(hhchars, offinc, "hhid")
hhchars <- left_join(hhchars, oninc, "hhid")
```

##Tables

F tests, but assume normal distribution, unlikely

```
cv.rem1 <-
  lm_robust(CV ~ CSV + femalehh  + totalplot + log(mmoInc), hhchars,
            se_type = "stata")

summary(cv.rem1, diagnostics = T)
```

Call: lm_robust(formula = CV ~ CSV + femalehh + totalplot + log(mmoInc), data = hhchars, se_type = "stata")

Standard error type: HC1

Table 3: ANOVA of Mean income

	Sum Sq	Df	F value	Pr(>F)
femalehh	47568004	1	0.322	0.572
CSA	84011527	1	0.569	0.452
CSV	30882162	1	0.209	0.648
bigland	174309398	1	1.180	0.280
CSA:CSV	522358420	1	3.537	0.063
Residuals	16245262483	110	NA	NA

Table 4: ANOVA of Poverty Persistence

	Sum Sq	Df	F value	Pr(>F)
femalehh	0.011	1	0.263	0.609
CSA	0.004	1	0.091	0.764
CSV	0.000	1	0.011	0.917
bigland	0.005	1	0.106	0.746
upperquint	22.056	1	512.247	0.000
CSA:CSV	0.031	1	0.714	0.400
Residuals	4.693	109	NA	NA

Table 5: ANOVA of Poverty Persistence, Income Groups Omitted

	Sum Sq	Df	F value	Pr(>F)
femalehh	0.016	1	0.068	0.795
CSA	0.748	1	3.075	0.082
CSV	0.702	1	2.887	0.092
bigland	0.106	1	0.437	0.510
CSA:CSV	0.088	1	0.364	0.548
Residuals	26.749	110	NA	NA

Table 6: ANOVA of CSA Use

	Sum Sq	Df	F value	Pr(>F)
femalehh	15.721	1	2.208	0.140
CSV	0.772	1	0.108	0.743
bigland	0.075	1	0.010	0.919
upperquint	52.554	1	7.380	0.008
Residuals	790.440	111	NA	NA

Table 7: ANOVA of CV

	Sum Sq	Df	F value	Pr(>F)
femalehh	0.013	1	0.182	0.671
CSV	0.001	1	0.014	0.905
usecount	0.122	1	1.673	0.199
bigland	0.001	1	0.017	0.896
upperquint	0.245	1	3.348	0.070
CSV:usecount	0.103	1	1.410	0.238
Residuals	7.979	109	NA	NA

Table 8: ANOVA of Mean income

	Sum Sq	Df	F value	Pr(>F)
femalehh	45204931	1	0.305	0.582
CSV	22551383	1	0.152	0.697
usecount	133610978	1	0.901	0.344
bigland	252857356	1	1.706	0.194
CSV:usecount	412473692	1	2.783	0.098
Residuals	16305547760	110	NA	NA

Table 9: ANOVA of Poverty Persistence

	Sum Sq	Df	F value	Pr(>F)
femalehh	0.005	1	0.137	0.712
CSV	0.000	1	0.008	0.927
usecount	0.010	1	0.255	0.615
bigland	0.013	1	0.322	0.572
upperquint	20.580	1	519.615	0.000
CSV:usecount	0.401	1	10.119	0.002
Residuals	4.317	109	NA	NA

Table 10: P-Values of ANOVA Tests

	CV (2)	CV (7)	Income (3)	Income (8)	Poverty (4)	Poverty (5)	Poverty (9)	CSA Use (6)
bigland	0.979	0.896	0.280	0.194	0.746	0.510	0.572	0.919
CSA	0.060	-	0.452	-	0.764	0.082	-	-
CSA:CSV	0.669	-	0.063	-	0.400	0.548	-	-
CSV	0.781	0.905	0.648	0.697	0.917	0.092	0.927	0.743
CSV:usecount	-	0.238	-	0.098	-	-	0.002	-
femalehh	0.587	0.671	0.572	0.582	0.609	0.795	0.712	0.140
upperquint	0.082	0.070	-	-	0.000	-	0.000	0.008
usecount	-	0.199	-	0.344	-	-	0.615	-

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.5437450 0.216566 2.51076 0.01349 0.11461 0.97288 111 CSV -0.0045258 0.059690 -0.07582 0.93970 -0.12280 0.11375 111 femalehh -0.0294316 0.066994 -0.43932 0.66129 -0.16218 0.10332 111 totalplot -0.0007004 0.005425 -0.12910 0.89751 -0.01145 0.01005 111 log(mmoInc) 0.0284906 0.025309 1.12573 0.26271 -0.02166 0.07864 111

Multiple R-squared: 0.01351 , Adjusted R-squared: -0.02204 F-statistic: 0.4093 on 4 and 111 DF, p-value: 0.8016

```r
cv.rem2 <-
  lm_robust(CV ~ CSA + femalehh  + totalplot + log(mmoInc), hhchars,
            se_type = "stata")

summary(cv.rem2, diagnostics = T)
```

Call: lm_robust(formula = CV ~ CSA + femalehh + totalplot + log(mmoInc), data = hhchars, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.5317890 0.20641 2.57642 0.01130 0.12278 0.94080 111 CSA -0.1007697 0.05811 -1.73423 0.08565 -0.21591 0.01437 111 femalehh -0.0433691 0.06602 -0.65695 0.51258 -0.17418 0.08745 111 totalplot -0.0002793 0.00494 -0.05654 0.95501 -0.01007 0.00951 111 log(mmoInc) 0.0374239 0.02610 1.43369 0.15447 -0.01430 0.08915 111

Multiple R-squared: 0.04431 , Adjusted R-squared: 0.009876 F-statistic: 0.9793 on 4 and 111 DF, p-value: 0.4219

```r
cv.rem3 <- lm_robust(CV ~ usecount + femalehh  + totalplot, hhchars,
                     se_type = "stata")

summary(cv.rem3, diagnostics = T)
```

Call: lm_robust(formula = CV ~ usecount + femalehh + totalplot, data = hhchars, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.7961261 0.040632 19.59368 5.551e-38 0.71562 0.87663 112 usecount -0.0082676 0.009357 -0.88360 3.788e-01 -0.02681 0.01027 112 femalehh -0.0355814 0.064387 -0.55262 5.816e-01 -0.16316 0.09199 112 totalplot -0.0004308 0.005269 -0.08176 9.350e-01 -0.01087 0.01001 112

Multiple R-squared: 0.008446 , Adjusted R-squared: -0.01811 F-statistic: 0.3302 on 3 and 112 DF, p-value: 0.8035

```r
cv.rem4 <-
  lm_robust(CV ~ usecount + femalehh  + totalplot + log(mmoInc),
            hhchars,
            se_type = "stata")

summary(cv.rem4, diagnostics = T)
```

Call: lm_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc), data = hhchars, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.5065881 0.214625 2.3603 0.0200 0.08129 0.931882 111 usecount -0.0114806 0.010078 -1.1391 0.2571 -0.03145 0.008490 111 femalehh -0.0412344 0.063454 -0.6498 0.5171 -0.16697 0.084504 111 totalplot -0.0009515 0.005462 -0.1742 0.8620 -0.01177 0.009871 111 log(mmoInc) 0.0357767 0.026715 1.3392 0.1832 -0.01716 0.088714 111

Multiple R-squared: 0.02598 , Adjusted R-squared: -0.009125 F-statistic: 0.6789 on 4 and 111 DF, p-value: 0.608

```r
cv.rem5 <-
  lm_robust(
    CV ~ usecount + femalehh  + totalplot + log(mmoInc) + dist_fmarket + oninc:
      usecount,
    hhchars,
    se_type = "stata"
  )

summary(cv.rem5, diagnostics = T)
```

Call: lm_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:usecount, data = hhchars, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.392121 0.230154 1.7037 0.091806 -0.06498 0.849226 92 usecount -0.023411 0.008932 -2.6211 0.010254 -0.04115 -0.005672 92 femalehh -0.044339 0.074174 -0.5978 0.551460 -0.19166 0.102977 92 totalplot -0.001516 0.005712 -0.2654 0.791305 -0.01286 0.009828 92 log(mmoInc) 0.051374 0.029044 1.7688 0.080234 -0.00631 0.109057 92 dist_fmarket -0.003948 0.008158 -0.4839 0.629594 -0.02015 0.012255 92 usecount:oninc 0.123744 0.032866 3.7652 0.000293 0.05847 0.189018 92

Multiple R-squared: 0.09667 , Adjusted R-squared: 0.03776 F-statistic: 3.207 on 6 and 92 DF, p-value: 0.006632

```r
cv.rem6 <-
  lm_robust(
    CV ~ CSA + femalehh  + totalplot + log(mmoInc) + dist_fmarket + oninc:
      CSA,
    hhchars,
    se_type = "stata"
  )

summary(cv.rem6, diagnostics = T)
```

Call: lm_robust(formula = CV ~ CSA + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:CSA, data = hhchars, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.4789921 0.226357 2.11609 0.037038 0.029427 0.928557 92 CSA -0.1702408 0.063952 -2.66199 0.009168 -0.297256 -0.043226 92 femalehh -0.0597470 0.075482 -0.79154 0.430664 -0.209660 0.090166 92 totalplot -0.0003765 0.005320 -0.07077 0.943737 -0.010942 0.010189 92 log(mmoInc) 0.0478072 0.028994 1.64885 0.102589 -0.009778 0.105392 92 dist_fmarket -0.0063975 0.008242 -0.77618 0.439634 -0.022767 0.009972 92 CSA:oninc 0.5188196 0.308218 1.68329 0.095710 -0.093329 1.130968 92

Multiple R-squared: 0.1064 , Adjusted R-squared: 0.04809 F-statistic: 1.805 on 6 and 92 DF, p-value: 0.1067

```r
cv.rem7 <-
  lm_robust(
    CV ~ usecount + femalehh  + totalplot + log(mmoInc) + dist_fmarket + oninc*
      usecount,
    hhchars,
    se_type = "stata"
  )

summary(cv.rem7, diagnostics = T)
```

Call: lm_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * usecount, data = hhchars, se_type = "stata")

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.380789 0.244212 1.5593 0.12241 -0.104308 0.865886 91 usecount -0.022398 0.009483 -2.3620 0.02031 -0.041234 -0.003561 91 femalehh -0.046296 0.072956 -0.6346 0.52730 -0.191215 0.098623 91 totalplot -0.001718 0.005606 -0.3065 0.75989 -0.012854 0.009417 91 log(mmoInc) 0.051997 0.029913 1.7383 0.08554 -0.007421 0.111415 91 dist_fmarket -0.004022 0.008237 -0.4883 0.62650 -0.020385 0.012340 91 oninc 0.067040 0.263108 0.2548 0.79945 -0.455592 0.589672 91 usecount:oninc 0.114303 0.046084 2.4803 0.01497 0.022762 0.205844 91

Multiple R-squared: 0.09726 , Adjusted R-squared: 0.02781 F-statistic: 2.542 on 7 and 91 DF, p-value: 0.0195

```r
cv.rem8 <-
  lm_robust(
    CV ~ CSA + femalehh  + totalplot + log(mmoInc) + dist_fmarket + oninc*CSA,
    hhchars,
    se_type = "stata"
  )

summary(cv.rem8, diagnostics = T)
```

Call: lm_robust(formula = CV ~ CSA + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * CSA, data = hhchars, se_type = "stata")

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.4619663 0.267092 1.72961 0.08709 -0.06858 0.992512 91 CSA -0.1642217 0.075113 -2.18633 0.03136 -0.31342 -0.015019 91 femalehh -0.0605521 0.076048 -0.79624 0.42797 -0.21161 0.090507 91 totalplot -0.0004624 0.005242 -0.08821 0.92990 -0.01087 0.009950 91 log(mmoInc) 0.0492021 0.031619 1.55609 0.12316 -0.01361 0.112009 91 dist_fmarket -0.0064929 0.008303 -0.78202 0.43623 -0.02299 0.009999 91 oninc 0.0617173 0.376913 0.16374 0.87030 -0.68697 0.810408 91 CSA:oninc 0.4552605 0.508199 0.89583 0.37271 -0.55421 1.464735 91

Multiple R-squared: 0.1066 , Adjusted R-squared: 0.0379 F-statistic: 1.531 on 7 and 91 DF, p-value: 0.1668

```r
cv.rem9 <-
  lm_robust(
    CV ~ CSA + CSV + log(mmoInc) + dist_fmarket + oninc,
    hhchars,
    se_type = "stata"
  )

summary(cv.rem9, diagnostics = T)
```

Call: lm_robust(formula = CV ~ CSA + CSV + log(mmoInc) + dist_fmarket + oninc, data = hhchars, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.288897 0.227695 1.2688 0.20765 -0.163197 0.74099 94 CSA -0.114324 0.062544 -1.8279 0.07074 -0.238507 0.00986 94 CSV 0.047270 0.062120 0.7610 0.44859 -0.076070 0.17061 94 log(mmoInc) 0.059713 0.029052 2.0554 0.04262 0.002029 0.11740 94 dist_fmarket -0.004251 0.007883 -0.5392 0.59103 -0.019903 0.01140 94 oninc 0.349388 0.257688 1.3559 0.17839 -0.162258 0.86103 94

Multiple R-squared: 0.09521 , Adjusted R-squared: 0.04708 F-statistic: 1.695 on 5 and 94 DF, p-value: 0.1435

```r
cv.remiv1 <- iv_robust(CV ~ CSA | CSV,
                       hhchars,
                       se_type = "stata",
                       diagnostics = T)

# print(summary(cv.remiv1, diagnostics = T))



cv.remiv2 <-
  iv_robust(
    CV ~ CSA + femalehh + totalplot |
      CSV + femalehh + totalplot,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

# print(summary(cv.remiv2, diagnostics = T))




cv.remiv3 <- iv_robust(CV ~ usecount | CSV,
                       hhchars,
                       se_type = "stata",
                       diagnostics = T)

# print(summary(cv.remiv3, diagnostics = T))



cv.remiv4 <-
  iv_robust(
    CV ~ usecount + femalehh + totalplot |
      CSV + femalehh + totalplot,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

# print(summary(cv.remiv4, diagnostics = T))
```

```r
cv.remiv5 <-
  iv_robust(
    CV ~ usecount + femalehh + totalplot + log(mmoInc) |
      CSV + femalehh + totalplot + log(mmoInc),
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

# print(summary(cv.remiv5, diagnostics = T))



cv.remiv6 <-
  iv_robust(
    CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket |
      CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

# print(summary(cv.remiv6, diagnostics = T))


cv.remiv7 <-
  iv_robust(
    CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc|
      CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

print(summary(cv.remiv7, diagnostics = T))
```

Call: iv_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc | CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc, data = hhchars, se_type = "stata", diagnostics = T)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 1.91366 40.3833 0.04739 0.9623 -78.2910 82.118 92 usecount 0.71544 18.5004 0.03867 0.9692 -36.0280 37.459 92 femalehh 0.83270 22.6338 0.03679 0.9707 -44.1199 45.785 92 totalplot 0.01704 0.4972 0.03427 0.9727 -0.9704 1.004 92 log(mmoInc) -0.37357 10.9280 -0.03418 0.9728 -22.0775 21.330 92 dist_fmarket 0.02206 0.6921 0.03188 0.9746 -1.3525 1.397 92 oninc 1.00994 16.8643 0.05989 0.9524 -32.4840 34.504 92

Multiple R-squared: -54.62 , Adjusted R-squared: -58.25 F-statistic: 0.0203 on 6 and 92 DF, p-value: 1

Diagnostics: numdf dendf value p.value Weak instruments 1 92 0.002 0.969 Wu-Hausman 1 91 0.126 0.723 Overidentifying 0 NA NA NA

```r
cv.remiv8 <-
  iv_robust(
    CV ~ CSA + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:CSA |
      CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:CSV,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

print(summary(cv.remiv8, diagnostics = T))
```

Call: iv_robust(formula = CV ~ CSA + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:CSA | CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:CSV, data = hhchars, se_type = “stata”, diagnostics = T)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.399097 0.25722 1.55158 0.1242 -0.11176 0.90996 92 CSA 0.191437 0.80290 0.23843 0.8121 -1.40319 1.78607 92 femalehh 0.005868 0.17553 0.03343 0.9734 -0.34274 0.35448 92 totalplot -0.003184 0.01169 -0.27235 0.7860 -0.02640 0.02004 92 log(mmoInc) 0.027748 0.07221 0.38428 0.7017 -0.11566 0.17116 92 dist_fmarket 0.001395 0.02237 0.06238 0.9504 -0.04304 0.04583 92 CSA:oninc 0.246395 0.55148 0.44678 0.6561 -0.84890 1.34169 92

Multiple R-squared: -0.2212 , Adjusted R-squared: -0.3009 F-statistic: 0.4762 on 6 and 92 DF, p-value: 0.8244

Diagnostics: numdf dendf value p.value
Weak instruments (CSA) 2 92 0.394 0.67521
Weak instruments (CSA:oninc) 2 92 5.538 0.00536 ** Wu-Hausman 2 90 0.587 0.55800
Overidentifying 0 NA NA NA
— Signif. codes: 0 ‘’ ***0.001*** ” *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

```r
cv.remiv9 <-
  iv_robust(
    CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:usecount|
      CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:CSV,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

print(summary(cv.remiv9, diagnostics = T))
```

Call: iv_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:usecount | CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc:CSV, data = hhchars, se_type = “stata”, diagnostics = T)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.256606 0.516764 0.4966 0.6207 -0.76973 1.28294 92 usecount -0.048702 0.134392 -0.3624 0.7179 -0.31562 0.21821 92 femalehh -0.095953 0.190349 -0.5041 0.6154 -0.47400 0.28210 92 totalplot -0.001451 0.007168 -0.2024 0.8401 -0.01569 0.01279 92 log(mmoInc) 0.081257 0.110809 0.7333 0.4652 -0.13882 0.30133 92 dist_fmarket -0.006264 0.011752 -0.5330 0.5953 -0.02961 0.01708 92 usecount:oninc -0.073578 0.425468 -0.1729 0.8631 -0.91859 0.77144 92

Multiple R-squared: -0.2263 , Adjusted R-squared: -0.3063 F-statistic: 0.6594 on 6 and 92 DF, p-value: 0.6825

Diagnostics: numdf dendf value p.value
Weak instruments (usecount) 2 92 2.795 0.0663 . Weak instruments (usecount:oninc) 2 92 4.765 0.0107 *
Wu-Hausman 2 90 0.316 0.7301
Overidentifying 0 NA NA NA
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

```
cv.remiv10 <-
  iv_robust(
    CV ~ CSA + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc*CSA |
      CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc*CSA ,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

print(summary(cv.remiv10, diagnostics = T))
```

Call: iv_robust(formula = CV ~ CSA + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * CSA | CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * CSA, data = hhchars, se_type = “stata”, diagnostics = T)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.4619663 0.267092 1.72961 0.08709 -0.06858 0.992512 91 CSA -0.1642217 0.075113 -2.18633 0.03136 -0.31342 -0.015019 91 femalehh -0.0605521 0.076048 -0.79624 0.42797 -0.21161 0.090507 91 totalplot -0.0004624 0.005242 -0.08821 0.92990 -0.01087 0.009950 91 log(mmoInc) 0.0492021 0.031619 1.55609 0.12316 -0.01361 0.112009 91 dist_fmarket -0.0064929 0.008303 -0.78202 0.43623 -0.02299 0.009999 91 oninc 0.0617173 0.376913 0.16374 0.87030 -0.68697 0.810408 91 CSA:oninc 0.4552605 0.508199 0.89583 0.37271 -0.55421 1.464735 91

Multiple R-squared: 0.1066 , Adjusted R-squared: 0.0379 F-statistic: 1.531 on 7 and 91 DF, p-value: 0.1668

Diagnostics: numdf dendf value p.value
Weak instruments 2 90 8.462e+30 <2e-16 *** Wu-Hausman 0 91 NA NA
Overidentifying 1 NA 4.290e-01 0.513
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

```
cv.remiv11 <-
  iv_robust(
    CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc*usecount|
      CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc*usecount,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

print(summary(cv.remiv11, diagnostics = T))
```

Call: iv_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * usecount | CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * usecount, data = hhchars, se_type = “stata”, diagnostics = T)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.380789 0.244212 1.5593 0.12241 -0.104308 0.865886 91 usecount -0.022398 0.009483 -2.3620 0.02031 -0.041234 -0.003561 91 femalehh -0.046296 0.072956 -0.6346 0.52730 -0.191215 0.098623 91 totalplot -0.001718 0.005606 -0.3065 0.75989 -0.012854 0.009417 91 log(mmoInc) 0.051997 0.029913 1.7383 0.08554 -0.007421 0.111415 91 dist_fmarket -0.004022 0.008237 -0.4883 0.62650 -0.020385 0.012340 91 oninc 0.067040 0.263108 0.2548 0.79945 -0.455592 0.589672 91 usecount:oninc 0.114303 0.046084 2.4803 0.01497 0.022762 0.205844 91

Multiple R-squared: 0.09726 , Adjusted R-squared: 0.02781 F-statistic: 2.542 on 7 and 91 DF, p-value: 0.0195

Diagnostics: numdf dendf value p.value
Weak instruments 2 90 1.333e+31 <2e-16 *** Wu-Hausman 0 91 NA NA
Overidentifying 1 NA 4.720e-01 0.492
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

```r
cv.remiv10a <-
  iv_robust(
    CV ~ CSA + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc*CSA |
      CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc*CSV ,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

print(summary(cv.remiv10a, diagnostics = T))
```

Call: iv_robust(formula = CV ~ CSA + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * CSA | CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * CSV, data = hhchars, se_type = “stata”, diagnostics = T)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.620323 0.68148 0.9103 0.3651 -0.73336 1.97401 91 CSA 0.279641 0.85490 0.3271 0.7443 -1.41851 1.97779 91 femalehh 0.039338 0.16806 0.2341 0.8155 -0.29449 0.37317 91 totalplot -0.005993 0.01106 -0.5421 0.5891 -0.02795 0.01597 91 log(mmoInc) -0.010056 0.07494 -0.1342 0.8935 -0.15891 0.13880 91 dist_fmarket 0.007819 0.02199 0.3555 0.7230 -0.03587 0.05151 91 oninc -0.642020 1.72239 -0.3727 0.7102 -4.06333 2.77929 91 CSA:oninc 1.500675 2.86287 0.5242 0.6014 -4.18606 7.18741 91

Multiple R-squared: -0.7837 , Adjusted R-squared: -0.921 F-statistic: 0.5316 on 7 and 91 DF, p-value: 0.8085

Diagnostics: numdf dendf value p.value Weak instruments (CSA) 2 91 0.566 0.570 Weak instruments (CSA:oninc) 2 91 1.377 0.258 Wu-Hausman 2 89 1.224 0.299 Overidentifying 0 NA NA NA

```r
cv.remiv11a <-
  iv_robust(
    CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc*usecount|
      CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc*CSV,
    hhchars,
    se_type = "stata",
    diagnostics = T
  )

print(summary(cv.remiv11a, diagnostics = T))
```

Call: iv_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * usecount | CSV + femalehh + totalplot + log(mmoInc) + dist_fmarket + oninc * CSV, data = hhchars, se_type = “stata”, diagnostics = T)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 0.5477943 0.403802 1.35659 0.1783 -0.25431 1.34990 91 usecount 0.1288747 0.313650 0.41089 0.6821 -0.49415 0.75190 91 femalehh 0.0978465 0.295528 0.33109 0.7413 -0.48918 0.68488 91 totalplot 0.0011452 0.009175 0.12482 0.9009 -0.01708 0.01937 91 log(mmoInc) -0.0163883 0.132374 -0.12380 0.9017 -0.27933 0.24656 91 dist_fmarket -0.0004158 0.018063 -0.02302 0.9817 -0.03630 0.03546 91 oninc 0.7125043 1.790688 0.39789 0.6916 -2.84448 4.26949 91 usecount:oninc -0.1142721 0.672626 -0.16989 0.8655 -1.45036 1.22182 91

Multiple R-squared: -1.728 , Adjusted R-squared: -1.938 F-statistic: 0.5348 on 7 and 91 DF, p-value: 0.8061

Diagnostics: numdf dendf value p.value Weak instruments (usecount) 2 91 0.763 0.469 Weak instruments (usecount:oninc) 2 91 0.902 0.409 Wu-Hausman 2 89 0.887 0.415 Overidentifying 0 NA NA NA

```
bptest(cv.rem1)
```

```
studentized Breusch-Pagan test
```

data: cv.rem1 BP = 0.6896, df = 4, p-value = 0.9526

```
bptest(cv.rem2)
```

```
studentized Breusch-Pagan test
```

data: cv.rem2 BP = 1.7711, df = 4, p-value = 0.7778

```
bptest(cv.rem3)
```

```
studentized Breusch-Pagan test
```

data: cv.rem3 BP = 0.47246, df = 3, p-value = 0.9249

```
bptest(cv.rem4)
```

```
studentized Breusch-Pagan test
```

data: cv.rem4 BP = 1.3545, df = 4, p-value = 0.8521

```
bptest(cv.rem5)
```

```
studentized Breusch-Pagan test
```

data: cv.rem5 BP = 5.6574, df = 6, p-value = 0.4626

```
bptest(cv.rem6)
```

```
studentized Breusch-Pagan test
```

data: cv.rem6 BP = 8.0488, df = 6, p-value = 0.2346

```r
bptest(cv.rem7)
```

```
studentized Breusch-Pagan test
```

data: cv.rem7 BP = 6.3123, df = 7, p-value = 0.5038

```r
bptest(cv.rem8)
```

```
studentized Breusch-Pagan test
```

data: cv.rem8 BP = 7.9756, df = 7, p-value = 0.3348

```r
texreg::texreg(
  l = list(
    cv.rem1,
    cv.rem2,
    cv.rem3,
    cv.rem4,
    cv.rem5,
    cv.rem6,
    cv.rem7,
    cv.rem8
  ),
  stars = c(0.01, 0.05, 0.1),
  include.ci = F,
  caption.above = TRUE,
  caption = "OLS Models",
  # booktabs = TRUE,
  # custom.header = list(
  #   "OLS" = 1:4,
  #   "IV, use count and CSA instrumented by CSV" = 5:9
  # ),
  bold = 0.05,
  custom.coef.names = c(
    "Intercept",
    "CSV",
    "Female Head",
    "Total Plot Size",
    "Log Mean Income",
    "CSA",
    "CSA Use Count",
    "Distance to Market",
    "Use CountxOn-Farm %",
    "CSAxOn-Farm %",
    "On-Farm Income Share"
  ),
  #  custom.note = "%stars. Models 5 and 6 instrument CSA with CSV, Models 7, 8, and 9 instrument use c
)
```

Table 11: OLS Models

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8
Intercept	$\mathbf{0.54}^{**}$	$\mathbf{0.53}^{**}$	$\mathbf{0.80}^{***}$	$\mathbf{0.51}^{**}$	0.39^{*}	$\mathbf{0.48}^{**}$	0.38	0.46^{*}
	(0.22)	(0.21)	(0.04)	(0.21)	(0.23)	(0.23)	(0.24)	(0.27)
CSV	−0.00							
	(0.06)							
Female Head	−0.03	−0.04	−0.04	−0.04	−0.04	−0.06	−0.05	−0.06
	(0.07)	(0.07)	(0.06)	(0.06)	(0.07)	(0.08)	(0.07)	(0.08)
Total Plot Size	−0.00	−0.00	−0.00	−0.00	−0.00	−0.00	−0.00	−0.00
	(0.01)	(0.00)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)
Log Mean Income	0.03	0.04		0.04	0.05^{*}	0.05	0.05^{*}	0.05
	(0.03)	(0.03)		(0.03)	(0.03)	(0.03)	(0.03)	(0.03)
CSA		-0.10^{*}				$\mathbf{-0.17}^{***}$		$\mathbf{-0.16}^{**}$
		(0.06)				(0.06)		(0.08)
CSA Use Count			−0.01	−0.01	$\mathbf{-0.02}^{**}$		$\mathbf{-0.02}^{**}$	
			(0.01)	(0.01)	(0.01)		(0.01)	
Distance to Market					−0.00	−0.01	−0.00	−0.01
					(0.01)	(0.01)	(0.01)	(0.01)
Use CountxOn-Farm %					$\mathbf{0.12}^{***}$		$\mathbf{0.11}^{**}$	
					(0.03)		(0.05)	
CSAxOn-Farm %						0.52^{*}		0.46
						(0.31)		(0.51)
On-Farm Income Share							0.07	0.06
							(0.26)	(0.38)
R^2	0.01	0.04	0.01	0.03	0.10	0.11	0.10	0.11
Adj. R^2	−0.02	0.01	−0.02	−0.01	0.04	0.05	0.03	0.04
Num. obs.	116	116	116	116	99	99	99	99
RMSE	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27

$^{***}p < 0.01$; $^{**}p < 0.05$; $^{*}p < 0.1$

```r
texreg::texreg(
  l = list(
    cv.remiv1,
    cv.remiv2,
    cv.remiv3,
    # cv.remiv4,
    cv.remiv5,
    # cv.remiv6,
    cv.remiv7,
    cv.remiv8,
    cv.remiv9,
    cv.remiv10a,
    cv.remiv11a
  ),
  stars = c(0.01, 0.05, 0.1),
  include.ci = F,
  caption.above = TRUE,
  caption = "Instrumented Variable Models",
  # booktabs = TRUE,
  # custom.header = list(
  #   "OLS" = 1:4,
  #   "IV, use count and CSA instrumented by CSV" = 5:9
  # ),
  bold = 0.05,
  custom.coef.names = c(
    "Intercept",
    "CSA",
    "Female Head",
    "Total Plot Size",
    "CSA Use Count",
    "Log Mean Income",
    "Distance to Market",
    "On-Farm Income Share",
    "CSAxOn-Farm %",
    "Use CountxOn-Farm %"
  ),
   custom.note = "%stars. Models 1, 2, 6, 8 instrument CSA with CSV, Models 3, 4, 5, 7, and 9 instrumen
)
```

```r
#Rerunning iv

# texreg::texreg(
#   l = list(
#     cv.remiv10a,
#     cv.remiv11a
#   ),
#   # no.margin = T,
#   include.ci = F,
#   stars = c(0.01, 0.05, 0.1),
#   caption.above = TRUE,
#   caption = "Respecified Instrumented Variable Models",
#   # booktabs = TRUE,
#   # custom.header = list(
```

Table 12: Instrumented Variable Models

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9
Intercept	$\mathbf{0.78}^{***}$	$\mathbf{0.84}^{**}$	0.80	0.09	1.91	0.40	0.26	0.62	0.55
	(0.29)	(0.41)	(0.85)	(12.52)	(40.38)	(0.26)	(0.52)	(0.68)	(0.40)
CSA	−0.02	−0.09				0.19		0.28	
	(0.46)	(0.62)				(0.80)		(0.85)	
Female Head		−0.04		−0.20	0.83	0.01	−0.10	0.04	0.10
		(0.12)		(4.89)	(22.63)	(0.18)	(0.19)	(0.17)	(0.30)
Total Plot Size		0.00		−0.01	0.02	−0.00	−0.00	−0.01	0.00
		(0.01)		(0.14)	(0.50)	(0.01)	(0.01)	(0.01)	(0.01)
CSA Use Count			−0.01	−0.16	0.72		−0.05		0.13
			(0.41)	(4.45)	(18.50)		(0.13)		(0.31)
Log Mean Income				0.13	−0.37	0.03	0.08	−0.01	−0.02
				(2.77)	(10.93)	(0.07)	(0.11)	(0.07)	(0.13)
Distance to Market					0.02	0.00	−0.01	0.01	−0.00
					(0.69)	(0.02)	(0.01)	(0.02)	(0.02)
On-Farm Income Share					1.01			−0.64	0.71
					(16.86)			(1.72)	(1.79)
CSAxOn-Farm %						0.25		1.50	
						(0.55)		(2.86)	
Use CountxOn-Farm %							−0.07		−0.11
							(0.43)		(0.67)
R^2	0.01	0.02	0.00	−2.05	−54.62	−0.22	−0.23	−0.78	−1.73
Adj. R^2	−0.00	−0.00	−0.01	−2.16	−58.25	−0.30	−0.31	−0.92	−1.94
Num. obs.	117	116	117	116	99	99	99	99	99
RMSE	0.27	0.27	0.27	0.48	2.10	0.31	0.31	0.38	0.47

$^{***}p < 0.01$; $^{**}p < 0.05$; $^{*}p < 0.1$. Models 1, 2, 6, 8 instrument CSA with CSV, Models 3, 4, 5, 7, and 9 instrument use count with CSV.

```
#   #   "OLS" = 1:4,
#   #   "IV, use count and CSA instrumented by CSV" = 5:9
#   # ),
#   bold = 0.05,
#   custom.coef.names = c(
#     "Intercept",
#     "CSA",
#     "Female Head",
#     "Total Plot Size",
#     "Log Mean Income",
#     "Distance to Market",
#     "On-Farm Income Share",
#     "CSAxOn-Farm %",
#     "CSA Use Count",
#     "Use CountxOn-Farm %"
#   ),
#    # custom.note = "%stars. Model 1 instruments CSA with CSV in interaction, Model 2 instruments use
# )


#non-robust
# texreg::texreg(
#   l = list(
#     cv.lm1,
#     cv.lm2,
#     cv.lm3,
#     cv.lm4,
#     cv.lm5,
#     cv.lm6,
#     cv.lm7,
#     cv.lm8
#   ),
#   stars = c(0.01, 0.05, 0.1),
#   include.ci = F,
#   caption.above = TRUE,
#   caption = "OLS Models",
#   # booktabs = TRUE,
#   # custom.header = list(
#   #   "OLS" = 1:4,
#   #   "IV, use count and CSA instrumented by CSV" = 5:9
#   # ),
#   bold = 0.05,
#   custom.coef.names = c(
#     "Intercept",
#     "CSV",
#     "Female Head",
#     "Total Plot Size",
#     "Log Mean Income",
#     "CSA",
#     "CSA Use Count",
#     "Distance to Market",
#     "Use CountxOn-Farm %",
#     "CSAxOn-Farm %",
#      "On-Farm Income Share"
#   ),
#   #  custom.note = "%stars. Models 5 and 6 instrument CSA with CSV, Models 7, 8, and 9 instrument use
# )
```

```
initalincome <- Income %>% filter(dateFloor == "2019-03-01") %>% select(hhid, moInc)

hhcharsnew <- left_join(hhchars, initalincome, "hhid")

useshare1 <- lm_robust(
    usecount ~ CV + CSV + log(moInc) + femalehh + totalplot + dist_fmarket + oninc,
    hhcharsnew,
    se_type = "stata"
  )

summary(useshare1, diagnostics = T)
```

Call: lm_robust(formula = usecount ~ CV + CSV + log(moInc) + femalehh + totalplot + dist_fmarket + oninc, data = hhcharsnew, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) -0.72646 2.52821 -0.2873 0.77452 -5.74995 4.2970 89 CV -0.48285 1.03636 -0.4659 0.64242 -2.54209 1.5764 89 CSV 0.12783 0.93848 0.1362 0.89196 -1.73692 1.9926 89 log(moInc) 0.46972 0.22445 2.0928 0.03922 0.02375 0.9157 89 femalehh -1.35299 0.59223 -2.2846 0.02472 -2.52975 -0.1762 89 totalplot -0.00228 0.05743 -0.0397 0.96842 -0.11639 0.1118 89 dist_fmarket -0.03369 0.11374 -0.2962 0.76774 -0.25969 0.1923 89 oninc -0.93723 2.65837 -0.3526 0.72525 -6.21935 4.3449 89

Multiple R-squared: 0.07324 , Adjusted R-squared: 0.0003438 F-statistic: 1.941 on 7 and 89 DF, p-value: 0.07225

```
useshare2 <- lm_robust(
    usecount ~ CV + CSV + log(mmoInc) + femalehh + totalplot + dist_fmarket + oninc,
    hhchars,
    se_type = "stata"
  )

summary(useshare2, diagnostics = T)
```

Call: lm_robust(formula = usecount ~ CV + CSV + log(mmoInc) + femalehh + totalplot + dist_fmarket + oninc, data = hhchars, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) -1.84224 2.21241 -0.83268 0.407203 -6.2369 2.55244 91 CV -1.22174 1.04603 -1.16797 0.245869 -3.2996 0.85608 91 CSV 0.06188 0.85689 0.07222 0.942587 -1.6402 1.76398 91 log(mmoInc) 0.65361 0.23731 2.75420 0.007105 0.1822 1.12500 91 femalehh -1.25640 0.58289 -2.15547 0.033764 -2.4142 -0.09856 91 totalplot -0.02803 0.05365 -0.52247 0.602615 -0.1346 0.07854 91 dist_fmarket -0.03841 0.10937 -0.35122 0.726232 -0.2557 0.17884 91 oninc -0.49926 2.65585 -0.18798 0.851307 -5.7748 4.77627 91

Multiple R-squared: 0.08891 , Adjusted R-squared: 0.01882 F-statistic: 2.219 on 7 and 91 DF, p-value: 0.03964

```r
useshare3 <- lm_robust(
    usecount ~ CV + CSV + log(moInc) + log(mmoInc) + femalehh + totalplot + dist_fmarket + oninc,
    hhcharsnew,
    se_type = "stata"
  )

summary(useshare3, diagnostics = T)
```

Call: lm_robust(formula = usecount ~ CV + CSV + log(moInc) + log(mmoInc) + femalehh + totalplot + dist_fmarket + oninc, data = hhcharsnew, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) -1.75781 2.53042 -0.69467 0.48909 -6.7865 3.27087 88 CV -1.02753 1.14551 -0.89700 0.37217 -3.3040 1.24894 88 CSV 0.07252 0.94355 0.07686 0.93891 -1.8026 1.94762 88 log(moInc) 0.13059 0.33809 0.38624 0.70025 -0.5413 0.80248 88 log(mmoInc) 0.50219 0.36326 1.38246 0.17033 -0.2197 1.22408 88 femalehh -1.30282 0.58879 -2.21271 0.02951 -2.4729 -0.13273 88 totalplot -0.02303 0.05981 -0.38511 0.70108 -0.1419 0.09583 88 dist_fmarket -0.03564 0.11129 -0.32027 0.74952 -0.2568 0.18552 88 oninc -0.60192 2.65648 -0.22658 0.82127 -5.8811 4.67728 88

Multiple R-squared: 0.08406 , Adjusted R-squared: 0.0007881 F-statistic: 1.836 on 8 and 88 DF, p-value: 0.0809

```r
useshare4 <- lm_robust(
    usecount ~ CV  + CSV*log(mmoInc) + femalehh + totalplot + dist_fmarket + oninc,
    hhcharsnew,
    se_type = "stata"
  )

summary(useshare4, diagnostics = T)
```

Call: lm_robust(formula = usecount ~ CV + CSV * log(mmoInc) + femalehh + totalplot + dist_fmarket + oninc, data = hhcharsnew, se_type = “stata”)

Standard error type: HC1

Coefficients: Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF (Intercept) 3.664460 3.93005 0.93242 0.35361 -4.14328 11.47220 90 CV -1.161369 1.03024 -1.12728 0.26262 -3.20812 0.88538 90 CSV -7.731147 4.37620 -1.76663 0.08068 -16.42523 0.96294 90 log(mmoInc) 0.006453 0.41759 0.01545 0.98770 -0.82317 0.83608 90 femalehh -1.183015 0.56174 -2.10600 0.03799 -2.29900 -0.06703 90 totalplot -0.011166 0.05334 -0.20932 0.83467 -0.11714 0.09481 90 dist_fmarket -0.063978 0.11618 -0.55070 0.58320 -0.29478 0.16682 90 oninc -0.449811 2.70995 -0.16598 0.86854 -5.83361 4.93399 90 CSV:log(mmoInc) 0.917906 0.48749 1.88292 0.06294 -0.05058 1.88639 90

Multiple R-squared: 0.1114 , Adjusted R-squared: 0.03243 F-statistic: 2.016 on 8 and 90 DF, p-value: 0.05328

```r
texreg::texreg(
 l = list(useshare1, useshare2, useshare3, useshare4),
  stars = c(0.01, 0.05, 0.1),
  include.ci = F,
  caption.above = TRUE,
  caption = "Predicting CSA Use Count",
 # custom.model.names = c("Initial income", "Mean Income", "Both"),
  # booktabs = TRUE,
  # custom.header = list(
```

Table 13: Predicting CSA Use Count

	Model 1	Model 2	Model 3	Model 4
Intercept	−0.73	−1.84	−1.76	3.66
	(2.53)	(2.21)	(2.53)	(3.93)
CV	−0.48	−1.22	−1.03	−1.16
	(1.04)	(1.05)	(1.15)	(1.03)
CSV	0.13	0.06	0.07	−7.73*
	(0.94)	(0.86)	(0.94)	(4.38)
Log Inital Income	0.47**		0.13	
	(0.22)		(0.34)	
Female Head	−1.35**	−1.26**	−1.30**	−1.18**
	(0.59)	(0.58)	(0.59)	(0.56)
Total Plot Size	−0.00	−0.03	−0.02	−0.01
	(0.06)	(0.05)	(0.06)	(0.05)
Distance to Market	−0.03	−0.04	−0.04	−0.06
	(0.11)	(0.11)	(0.11)	(0.12)
On-Farm Income Share	−0.94	−0.50	−0.60	−0.45
	(2.66)	(2.66)	(2.66)	(2.71)
Log Mean Income		0.65***	0.50	0.01
		(0.24)	(0.36)	(0.42)
CSV x Log Mean Income				0.92*
				(0.49)
R^2	0.07	0.09	0.08	0.11
Adj. R^2	0.00	0.02	0.00	0.03
Num. obs.	97	99	97	99
RMSE	2.91	2.87	2.91	2.85

$^{***}p < 0.01$; $^{**}p < 0.05$; $^{*}p < 0.1$

```
  #   "OLS" = 1:4,
  #   "IV, use count and CSA instrumented by CSV" = 5:9
  # ),
  # bold = 0.05,
  custom.coef.names = c(
    "Intercept",
    "CV",
    "CSV",
    "Log Inital Income",
    "Female Head",
    "Total Plot Size",
    "Distance to Market",
    "On-Farm Income Share",
    "Log Mean Income",
    "CSV x Log Mean Income"
    ),
  #  custom.note = "%stars.",
 # siunitx = T
)
```

```
print(summary(cv.remiv1, diagnostics = T))
```

```
##
## Call:
## iv_robust(formula = CV ~ CSA | CSV, data = hhchars, se_type = "stata",
##     diagnostics = T)
##
## Standard error type:  HC1
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)  0.78005     0.2900  2.68935 0.008224   0.2055   1.3546 115
## CSA         -0.01629     0.4568 -0.03566 0.971615  -0.9211   0.8885 115
##
## Multiple R-squared:  0.007386 ,  Adjusted R-squared:  -0.001245
## F-statistic: 0.001272 on 1 and 115 DF,  p-value: 0.9716
##
## Diagnostics:
##                     numdf dendf value p.value
## Weak instruments        1   115 1.422   0.236
## Wu-Hausman              1   114 0.019   0.892
## Overidentifying         0    NA    NA      NA
```

```
print(summary(cv.remiv2, diagnostics = T))
```

```
##
## Call:
## iv_robust(formula = CV ~ CSA + femalehh + totalplot | CSV + femalehh +
##     totalplot, data = hhchars, se_type = "stata", diagnostics = T)
##
## Standard error type:  HC1
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)  0.8381841   0.407340  2.05770  0.04194  0.03109  1.64528 112
## CSA         -0.0940585   0.615650 -0.15278  0.87885 -1.31389  1.12577 112
## femalehh    -0.0400572   0.120513 -0.33239  0.74022 -0.27884  0.19872 112
## totalplot    0.0001608   0.005405  0.02974  0.97632 -0.01055  0.01087 112
##
## Multiple R-squared:  0.02449 ,   Adjusted R-squared:  -0.001637
## F-statistic: 0.05837 on 3 and 112 DF,  p-value: 0.9814
##
## Diagnostics:
##                     numdf dendf value p.value
## Weak instruments        1   112 0.842   0.361
## Wu-Hausman              1   111 0.000   0.989
## Overidentifying         0    NA    NA      NA
```

```
print(summary(cv.remiv3, diagnostics = T))
```

```
##
```

```
## Call:
## iv_robust(formula = CV ~ usecount | CSV, data = hhchars, se_type = "stata",
##     diagnostics = T)
##
## Standard error type:  HC1
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)  0.79945     0.8517  0.93861   0.3499  -0.8877   2.4866 115
## usecount    -0.01436     0.4112 -0.03494   0.9722  -0.8288   0.8001 115
##
## Multiple R-squared:  9.239e-06 , Adjusted R-squared:  -0.008686
## F-statistic: 0.001221 on 1 and 115 DF,  p-value: 0.9722
##
## Diagnostics:
##                  numdf dendf value p.value
## Weak instruments     1   115 0.047   0.829
## Wu-Hausman           1   114 0.000   0.986
## Overidentifying      0    NA    NA      NA
```

```
# print(summary(cv.remiv4, diagnostics = T))
```

```r
print(summary(cv.remiv5, diagnostics = T))
```

```
##
## Call:
## iv_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) |
##     CSV + femalehh + totalplot + log(mmoInc), data = hhchars,
##     se_type = "stata", diagnostics = T)
##
## Standard error type:  HC1
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)  0.089961    12.5182  0.007186   0.9943 -24.7158  24.8957 111
## usecount    -0.159283     4.4527 -0.035772   0.9715  -8.9826   8.6640 111
## femalehh    -0.202855     4.8913 -0.041473   0.9670  -9.8952   9.4895 111
## totalplot   -0.005437     0.1387 -0.039214   0.9688  -0.2802   0.2693 111
## log(mmoInc)  0.127870     2.7734  0.046106   0.9633  -5.3678   5.6236 111
##
## Multiple R-squared:  -2.05 , Adjusted R-squared:  -2.16
## F-statistic: 0.1709 on 4 and 111 DF,  p-value: 0.9529
##
## Diagnostics:
##                  numdf dendf value p.value
## Weak instruments     1   111 0.002   0.968
## Wu-Hausman           1   110 0.005   0.945
## Overidentifying      0    NA    NA      NA
```

```r
print(summary(cv.remiv7, diagnostics = T))
```

```
##
## Call:
## iv_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) +
##     dist_fmarket + oninc | CSV + femalehh + totalplot + log(mmoInc) +
##     dist_fmarket + oninc, data = hhchars, se_type = "stata",
##     diagnostics = T)
##
## Standard error type:  HC1
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   1.91366    40.3833  0.04739   0.9623 -78.2910   82.118 92
## usecount      0.71544    18.5004  0.03867   0.9692 -36.0280   37.459 92
## femalehh      0.83270    22.6338  0.03679   0.9707 -44.1199   45.785 92
## totalplot     0.01704     0.4972  0.03427   0.9727  -0.9704    1.004 92
## log(mmoInc)  -0.37357    10.9280 -0.03418   0.9728 -22.0775   21.330 92
## dist_fmarket  0.02206     0.6921  0.03188   0.9746  -1.3525    1.397 92
## oninc         1.00994    16.8643  0.05989   0.9524 -32.4840   34.504 92
##
## Multiple R-squared:  -54.62 ,    Adjusted R-squared:  -58.25
## F-statistic: 0.0203 on 6 and 92 DF,  p-value: 1
##
## Diagnostics:
##                  numdf dendf value p.value
## Weak instruments     1    92 0.002   0.969
## Wu-Hausman           1    91 0.126   0.723
## Overidentifying      0    NA    NA      NA
```

```r
print(summary(cv.remiv8, diagnostics = T))
```

```
##
## Call:
## iv_robust(formula = CV ~ CSA + femalehh + totalplot + log(mmoInc) +
##     dist_fmarket + oninc:CSA | CSV + femalehh + totalplot + log(mmoInc) +
##     dist_fmarket + oninc:CSV, data = hhchars, se_type = "stata",
##     diagnostics = T)
##
## Standard error type:  HC1
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   0.399097    0.25722  1.55158   0.1242 -0.11176  0.90996 92
## CSA           0.191437    0.80290  0.23843   0.8121 -1.40319  1.78607 92
## femalehh      0.005868    0.17553  0.03343   0.9734 -0.34274  0.35448 92
## totalplot    -0.003184    0.01169 -0.27235   0.7860 -0.02640  0.02004 92
## log(mmoInc)   0.027748    0.07221  0.38428   0.7017 -0.11566  0.17116 92
## dist_fmarket  0.001395    0.02237  0.06238   0.9504 -0.04304  0.04583 92
## CSA:oninc     0.246395    0.55148  0.44678   0.6561 -0.84890  1.34169 92
##
## Multiple R-squared:  -0.2212 ,   Adjusted R-squared:  -0.3009
## F-statistic: 0.4762 on 6 and 92 DF,  p-value: 0.8244
##
## Diagnostics:
##                              numdf dendf value p.value
## Weak instruments (CSA)           2    92 0.394 0.67521
## Weak instruments (CSA:oninc)     2    92 5.538 0.00536 **
## Wu-Hausman                       2    90 0.587 0.55800
## Overidentifying                  0    NA    NA      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(summary(cv.remiv9, diagnostics = T))
```

```
##
## Call:
## iv_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) +
##     dist_fmarket + oninc:usecount | CSV + femalehh + totalplot +
##     log(mmoInc) + dist_fmarket + oninc:CSV, data = hhchars, se_type = "stata",
##     diagnostics = T)
##
## Standard error type:  HC1
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)     0.256606   0.516764  0.4966   0.6207 -0.76973  1.28294 92
## usecount       -0.048702   0.134392 -0.3624   0.7179 -0.31562  0.21821 92
## femalehh       -0.095953   0.190349 -0.5041   0.6154 -0.47400  0.28210 92
## totalplot      -0.001451   0.007168 -0.2024   0.8401 -0.01569  0.01279 92
## log(mmoInc)     0.081257   0.110809  0.7333   0.4652 -0.13882  0.30133 92
## dist_fmarket   -0.006264   0.011752 -0.5330   0.5953 -0.02961  0.01708 92
## usecount:oninc -0.073578   0.425468 -0.1729   0.8631 -0.91859  0.77144 92
##
## Multiple R-squared:  -0.2263 ,   Adjusted R-squared:  -0.3063
## F-statistic: 0.6594 on 6 and 92 DF,  p-value: 0.6825
##
## Diagnostics:
##                                   numdf dendf value p.value
## Weak instruments (usecount)           2    92 2.795  0.0663 .
## Weak instruments (usecount:oninc)     2    92 4.765  0.0107 *
## Wu-Hausman                            2    90 0.316  0.7301
## Overidentifying                       0    NA    NA      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

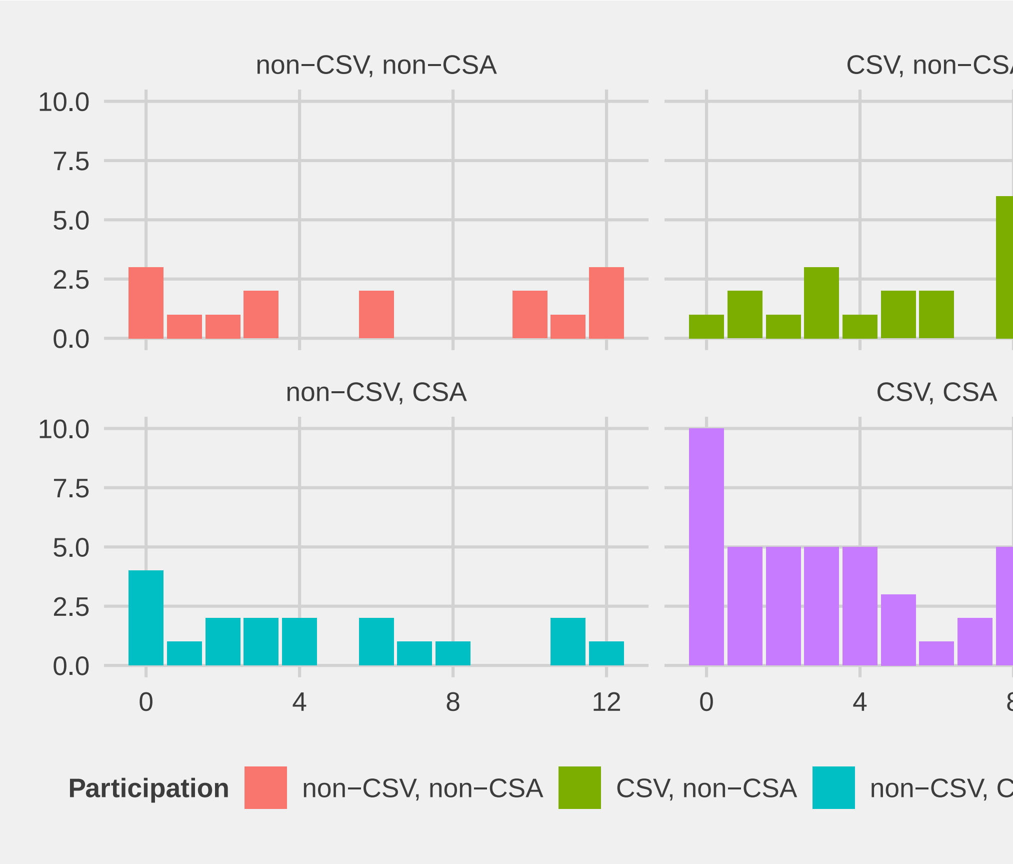
```r
print(summary(cv.remiv10a, diagnostics = T))
```

```
##
## Call:
## iv_robust(formula = CV ~ CSA + femalehh + totalplot + log(mmoInc) +
##     dist_fmarket + oninc * CSA | CSV + femalehh + totalplot +
##     log(mmoInc) + dist_fmarket + oninc * CSV, data = hhchars,
##     se_type = "stata", diagnostics = T)
##
## Standard error type:  HC1
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   0.620323    0.68148  0.9103   0.3651 -0.73336  1.97401 91
## CSA           0.279641    0.85490  0.3271   0.7443 -1.41851  1.97779 91
## femalehh      0.039338    0.16806  0.2341   0.8155 -0.29449  0.37317 91
## totalplot    -0.005993    0.01106 -0.5421   0.5891 -0.02795  0.01597 91
## log(mmoInc)  -0.010056    0.07494 -0.1342   0.8935 -0.15891  0.13880 91
## dist_fmarket  0.007819    0.02199  0.3555   0.7230 -0.03587  0.05151 91
## oninc        -0.642020    1.72239 -0.3727   0.7102 -4.06333  2.77929 91
## CSA:oninc     1.500675    2.86287  0.5242   0.6014 -4.18606  7.18741 91
##
## Multiple R-squared:  -0.7837 ,   Adjusted R-squared:  -0.921
## F-statistic: 0.5316 on 7 and 91 DF,  p-value: 0.8085
##
## Diagnostics:
##                               numdf dendf value p.value
## Weak instruments (CSA)            2    91 0.566   0.570
## Weak instruments (CSA:oninc)      2    91 1.377   0.258
## Wu-Hausman                        2    89 1.224   0.299
## Overidentifying                   0    NA    NA      NA
```

```r
print(summary(cv.remiv11a, diagnostics = T))
```

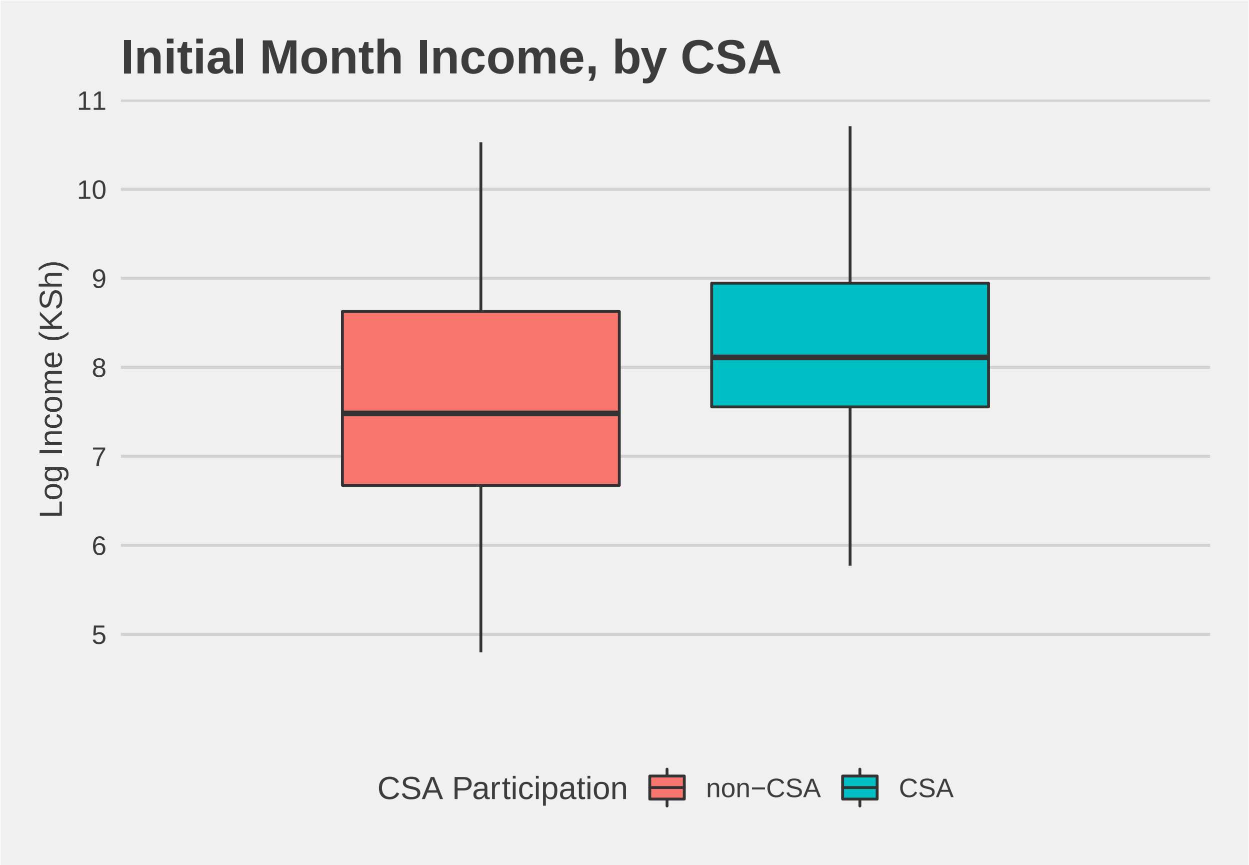
```
##
## Call:
## iv_robust(formula = CV ~ usecount + femalehh + totalplot + log(mmoInc) +
##     dist_fmarket + oninc * usecount | CSV + femalehh + totalplot +
##     log(mmoInc) + dist_fmarket + oninc * CSV, data = hhchars,
##     se_type = "stata", diagnostics = T)
##
## Standard error type:  HC1
##
## Coefficients:
##                  Estimate Std. Error  t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)     0.5477943   0.403802  1.35659   0.1783 -0.25431  1.34990 91
## usecount        0.1288747   0.313650  0.41089   0.6821 -0.49415  0.75190 91
## femalehh        0.0978465   0.295528  0.33109   0.7413 -0.48918  0.68488 91
## totalplot       0.0011452   0.009175  0.12482   0.9009 -0.01708  0.01937 91
## log(mmoInc)    -0.0163883   0.132374 -0.12380   0.9017 -0.27933  0.24656 91
## dist_fmarket   -0.0004158   0.018063 -0.02302   0.9817 -0.03630  0.03546 91
## oninc           0.7125043   1.790688  0.39789   0.6916 -2.84448  4.26949 91
## usecount:oninc -0.1142721   0.672626 -0.16989   0.8655 -1.45036  1.22182 91
##
## Multiple R-squared:  -1.728 ,    Adjusted R-squared:  -1.938
## F-statistic: 0.5348 on 7 and 91 DF,  p-value: 0.8061
##
## Diagnostics:
##                                  numdf dendf value p.value
## Weak instruments (usecount)          2    91 0.763   0.469
## Weak instruments (usecount:oninc)    2    91 0.902   0.409
## Wu-Hausman                           2    89 0.887   0.415
## Overidentifying                      0    NA    NA      NA
```

##Plots

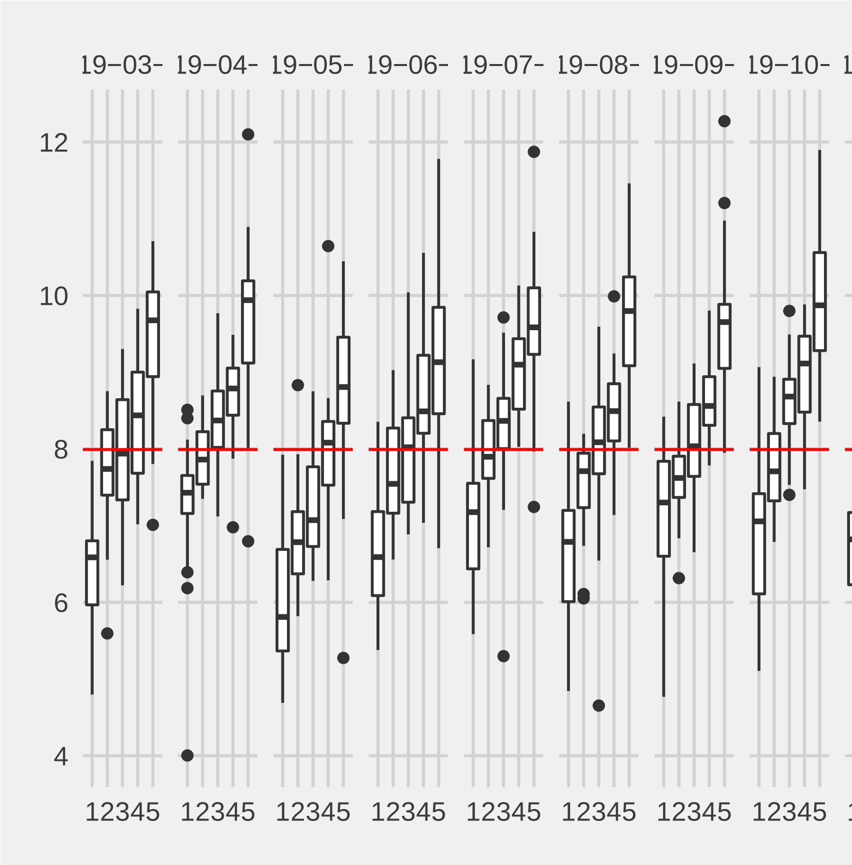


Poverty count by participation group

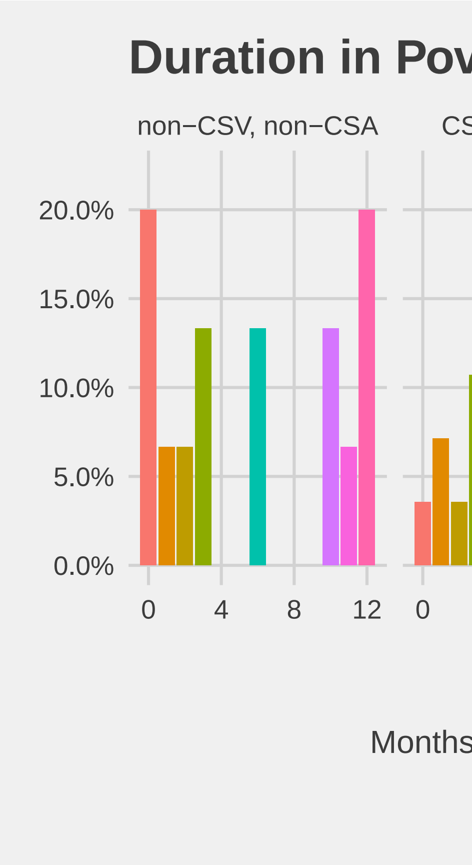
Participation group by initial income



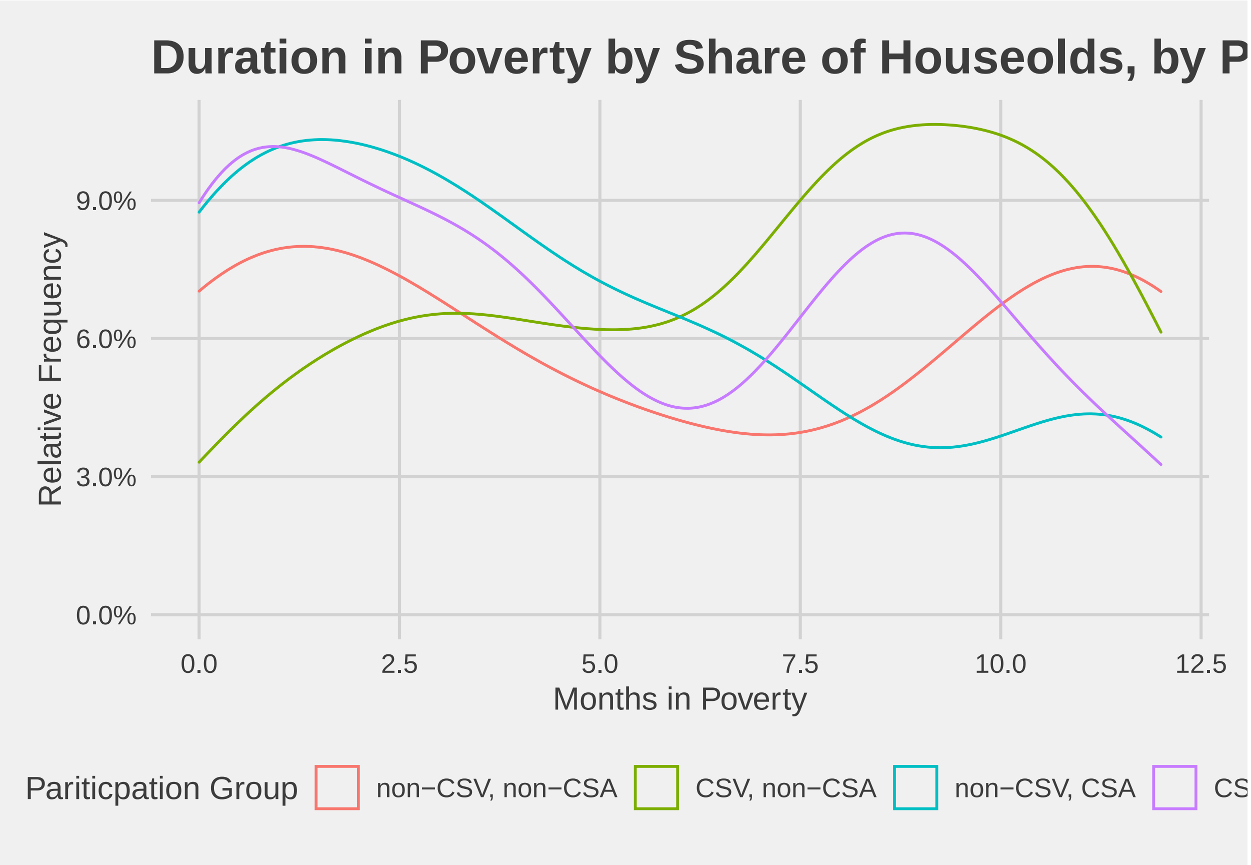
```
## # A tibble: 2 x 4
##     CSA  mInc medInc count
##   <dbl> <dbl>  <dbl> <int>
## 1     0 5749.  1775     41
## 2     1 6421.  3330.    73
```

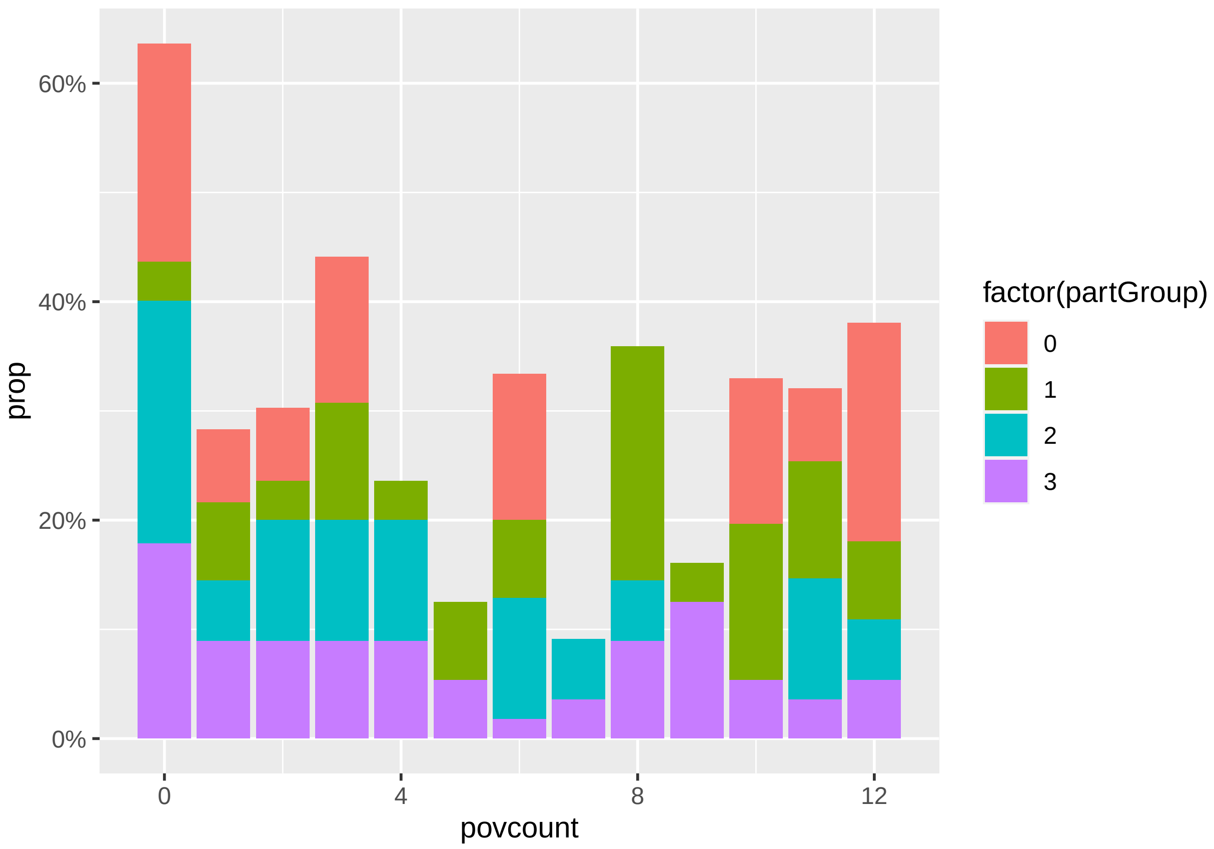


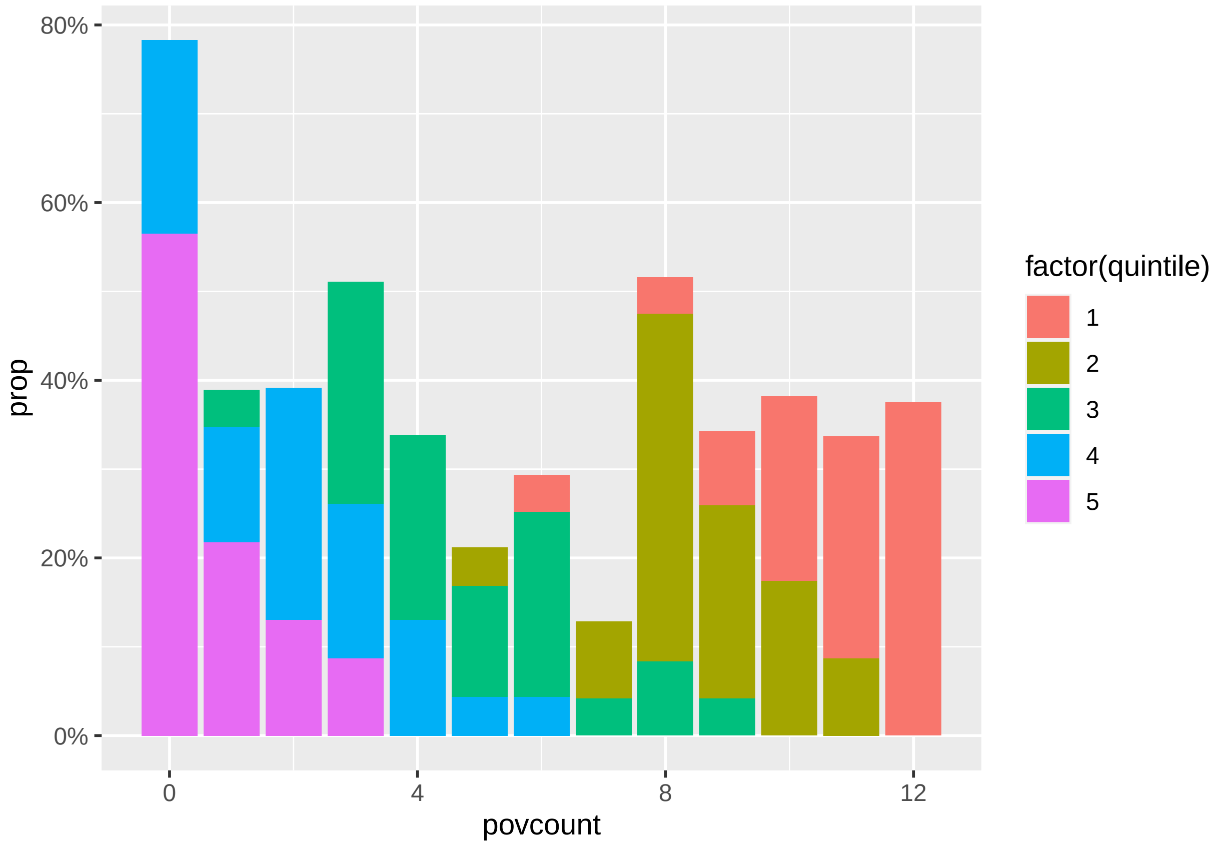
Income box plots over each month with poverty line



Share of households by number of months spent under pov line, by CSA participation

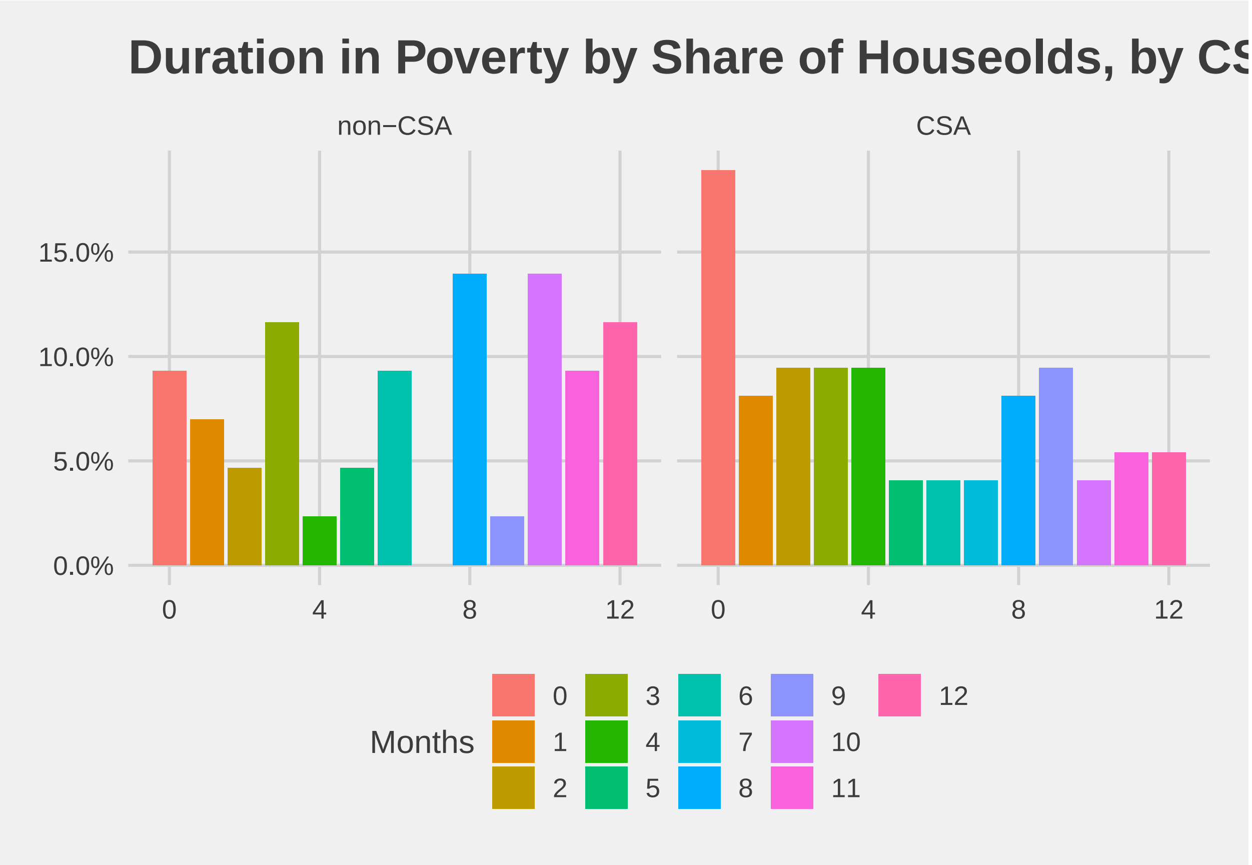

Duration in Poverty by Share of Houseolds, by P
Relative Frequency
9.0%
6.0%
3.0%
0.0%
0.0
2.5
5.0
7.5
10.0
12.5
Months in Poverty
Pariticpation Group
non−CSV, non−CSA
CSV, non−CSA
non−CSV, CSA
CS

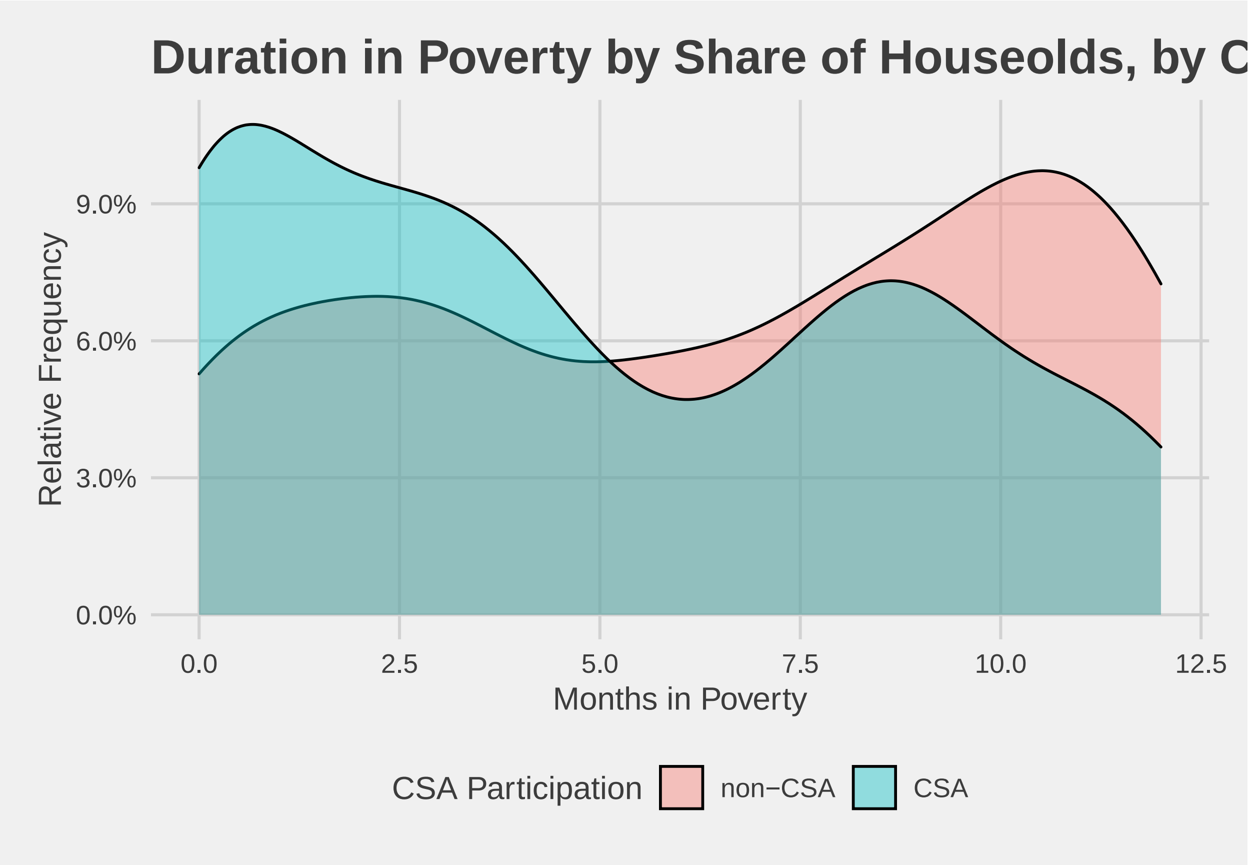

60%
40%
20%
0%
prop
0
4
8
12
povcount
factor(partGroup)
0
1
2
3



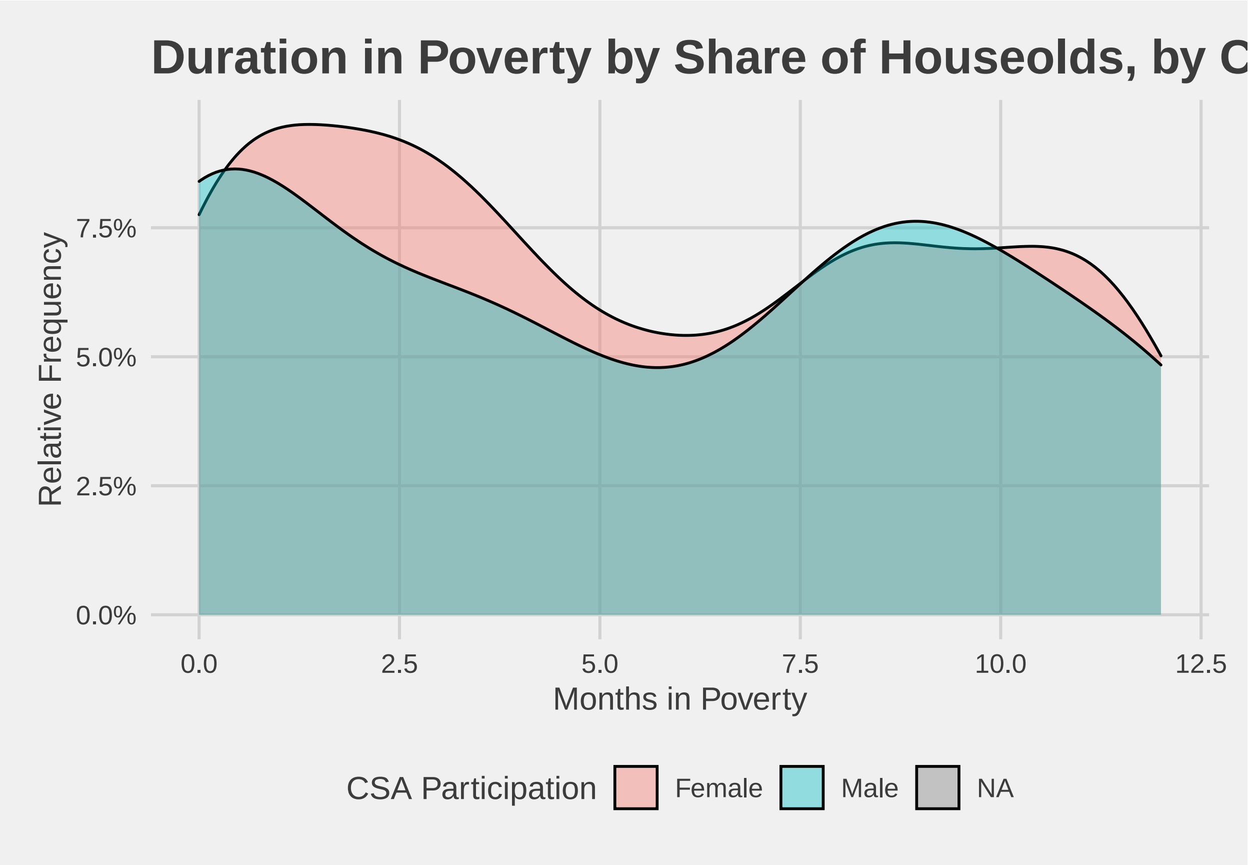
kernel can be found here https://rdrr.io/r/stats/density.html

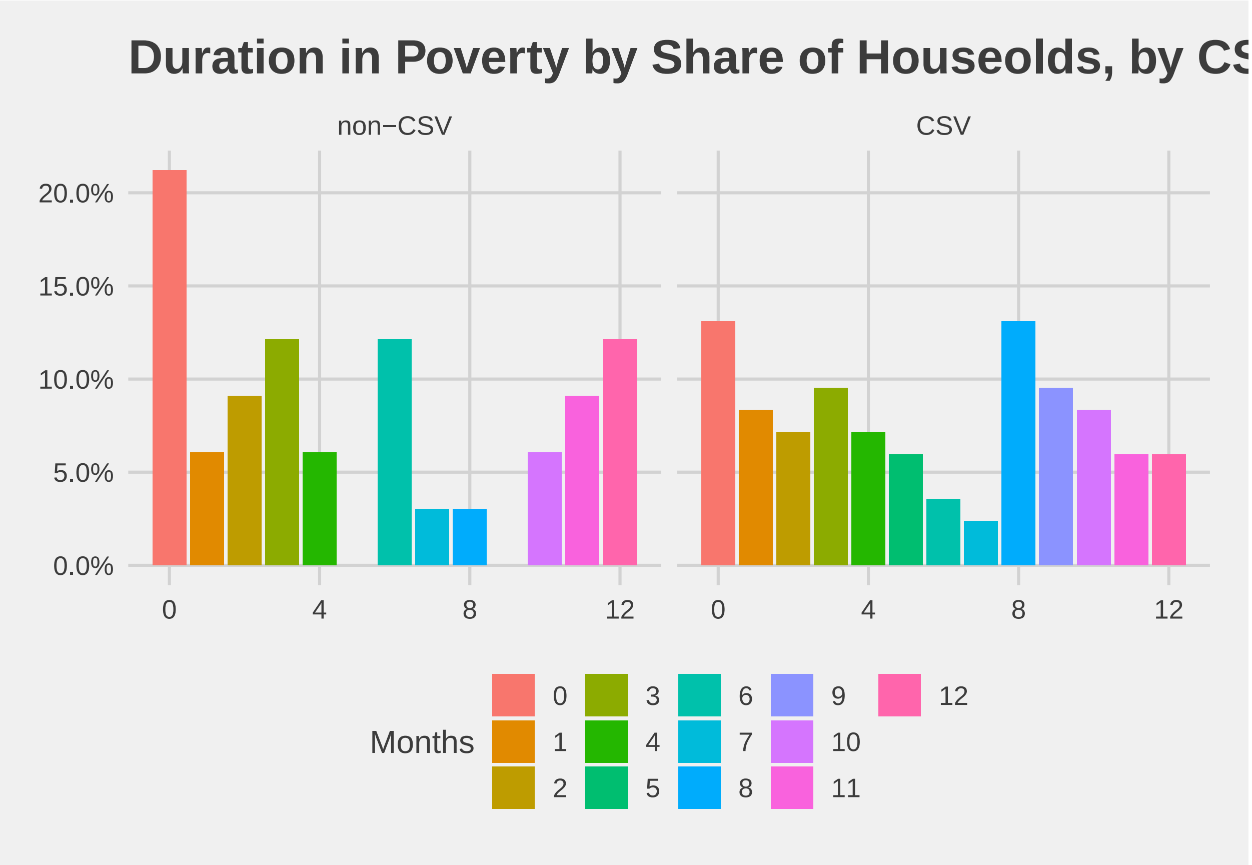
By only CSA


Duration in Poverty by Share of Houseolds, by CS
non−CSA
CSA
15.0%
10.0%
5.0%
0.0%
0
4
8
12
0
4
8
12
Months
0
1
2
3
4
5
6
7
8
9
10
11
12

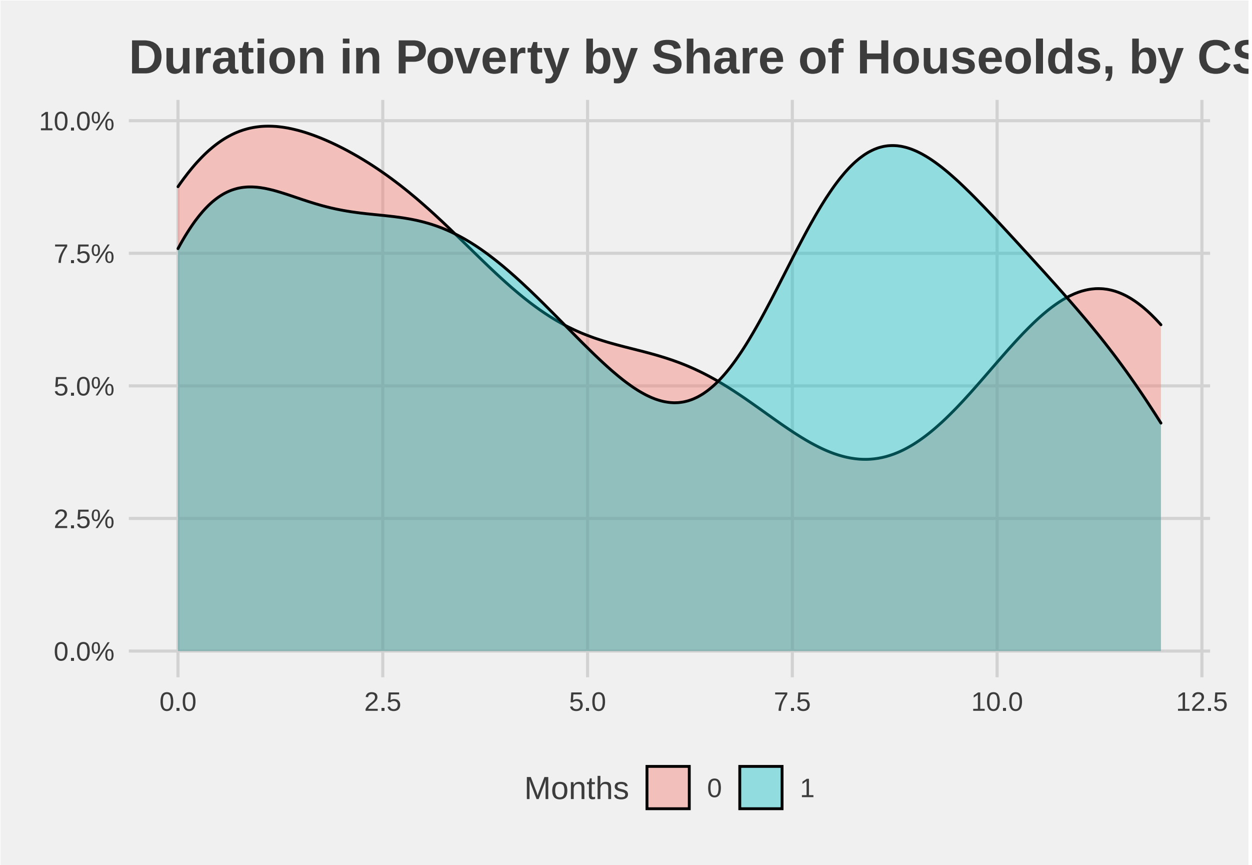


```
## Warning: Groups with fewer than two data points have been dropped.
```


Duration in Poverty by Share of Houseolds, by C
Relative Frequency
7.5%
5.0%
2.5%
0.0%
0.0
2.5
5.0
7.5
10.0
12.5
Months in Poverty
CSA Participation
Female
Male
NA

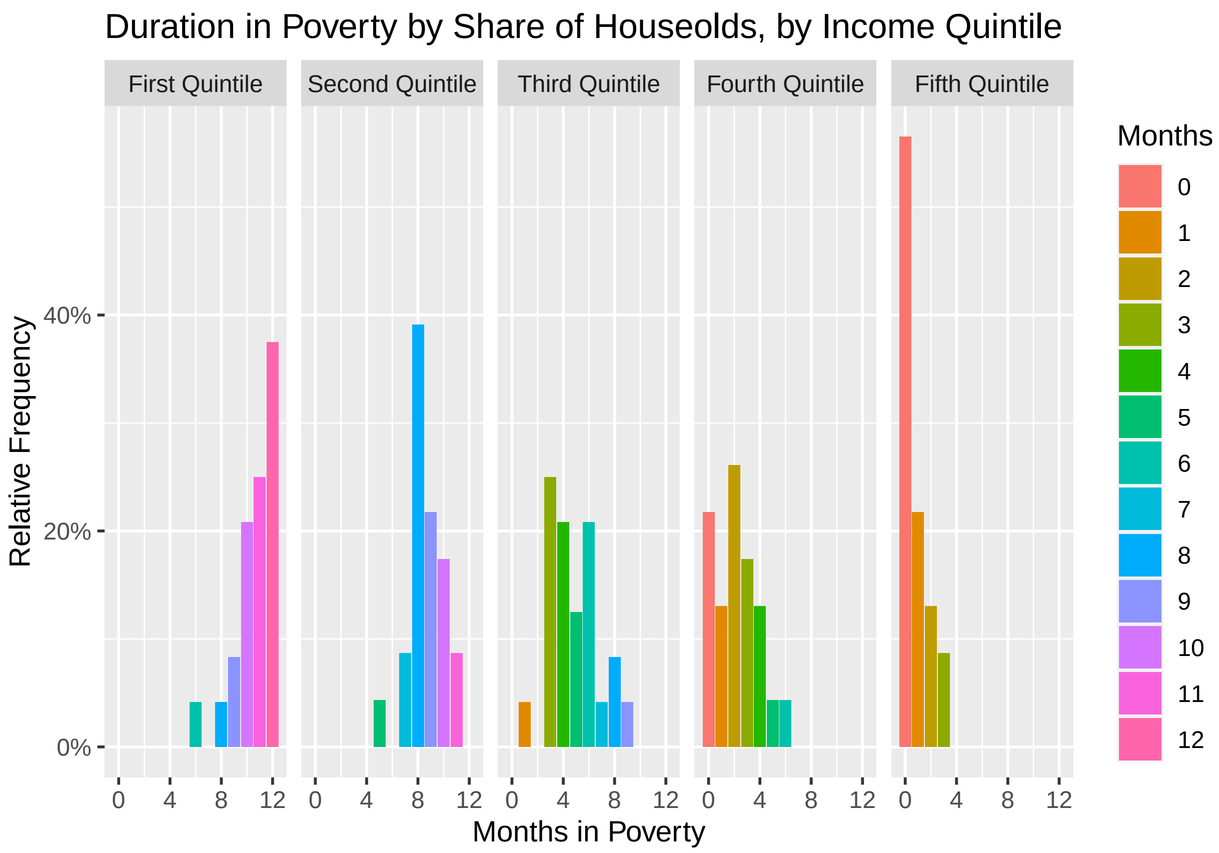

Duration in Poverty by Share of Houseolds, by CS
non−CSV
CSV
20.0%
15.0%
10.0%
5.0%
0.0%
0
4
8
12
Months
0
1
2
3
4
5
6
7
8
9
10
11
12

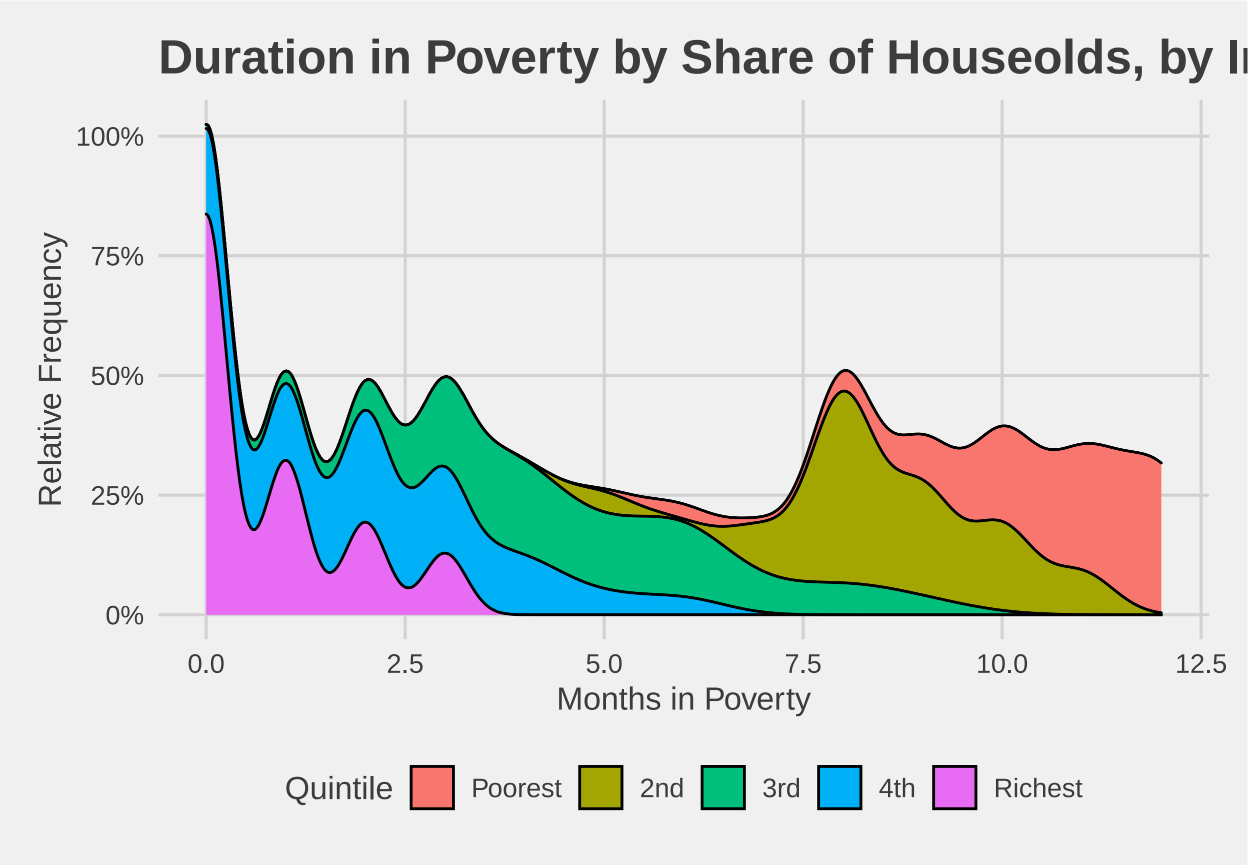
By only CSV



Share of households by number of months spent under pov line, by income quintile

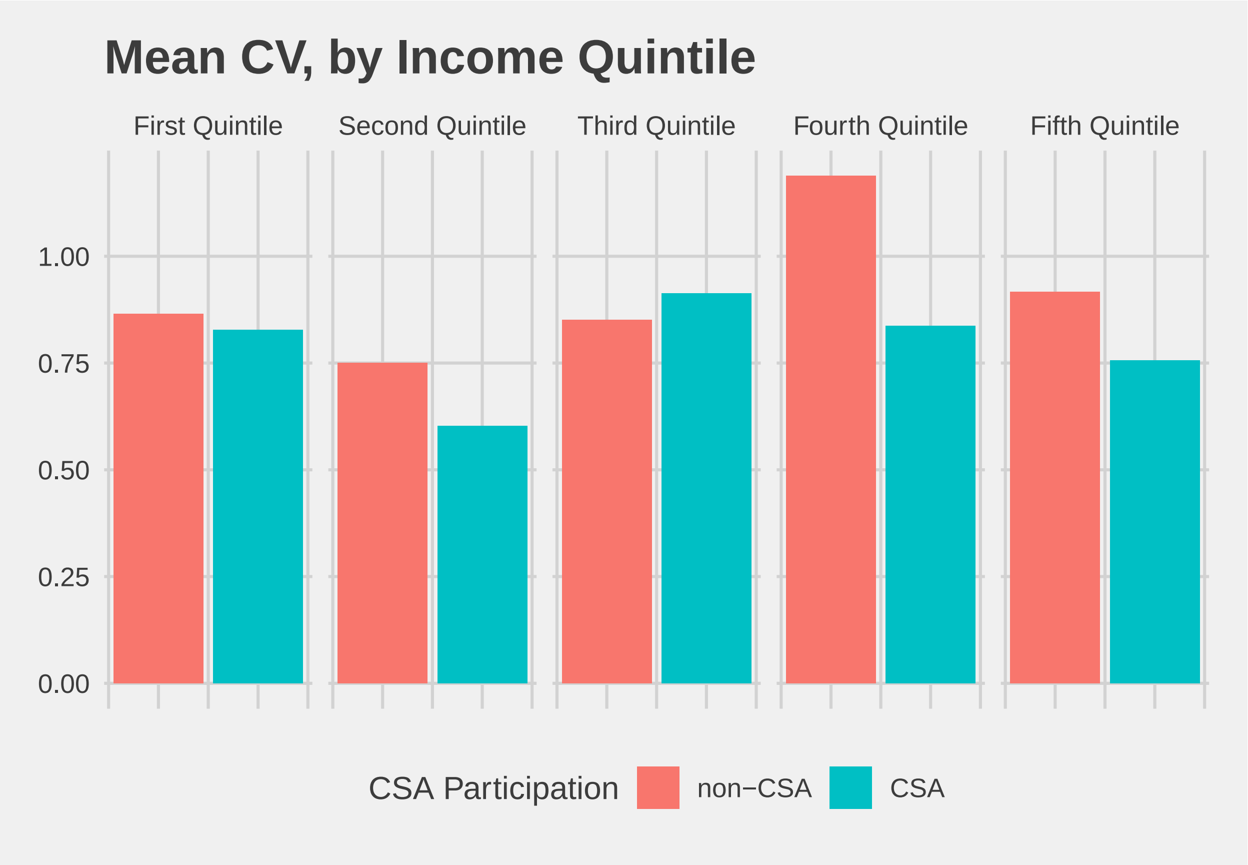
```
## Warning: Ignoring unknown aesthetics: stat
```


Duration in Poverty by Share of Houseolds, by Income Quintile
First Quintile
Second Quintile
Third Quintile
Fourth Quintile
Fifth Quintile
Relative Frequency
0%
20%
40%
0
4
8
12
Months in Poverty
Months
0
1
2
3
4
5
6
7
8
9
10
11
12

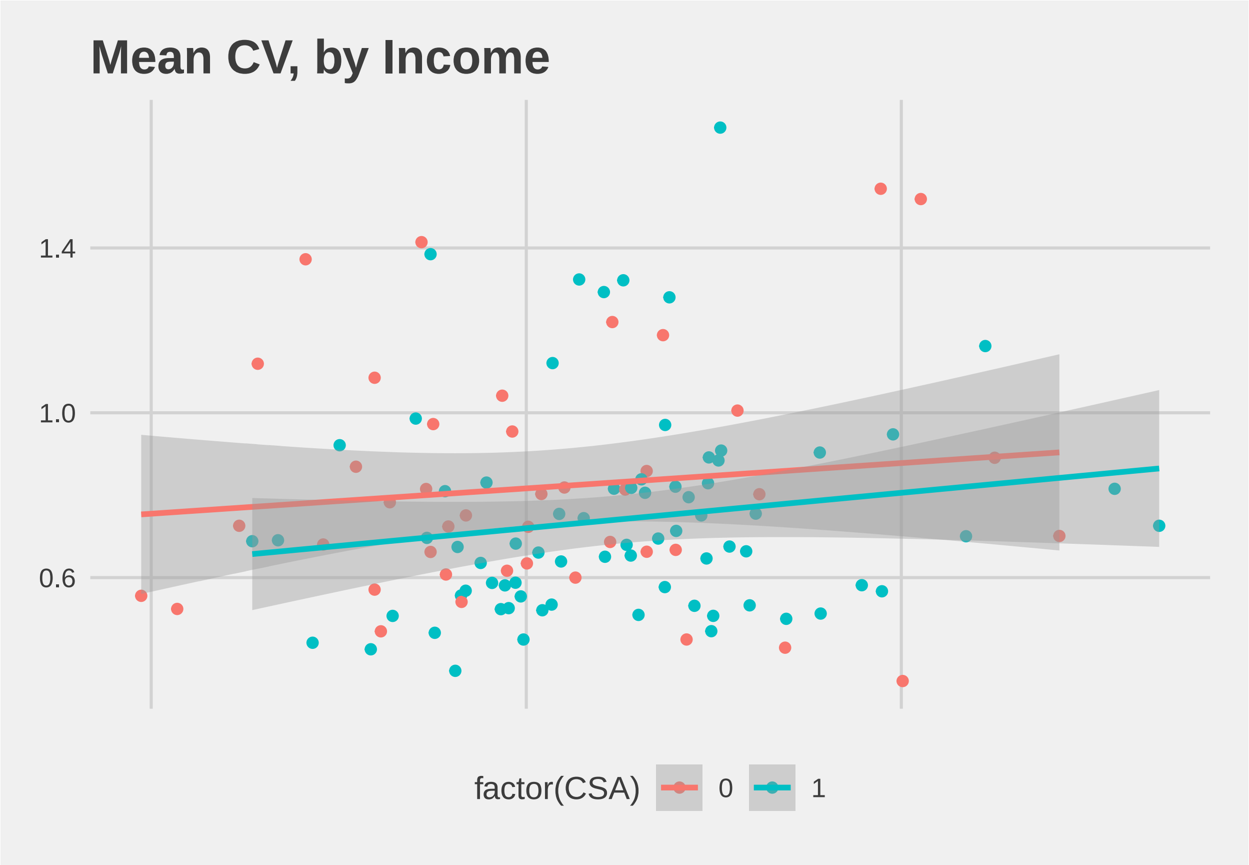


Gridding both CSA and Quintile

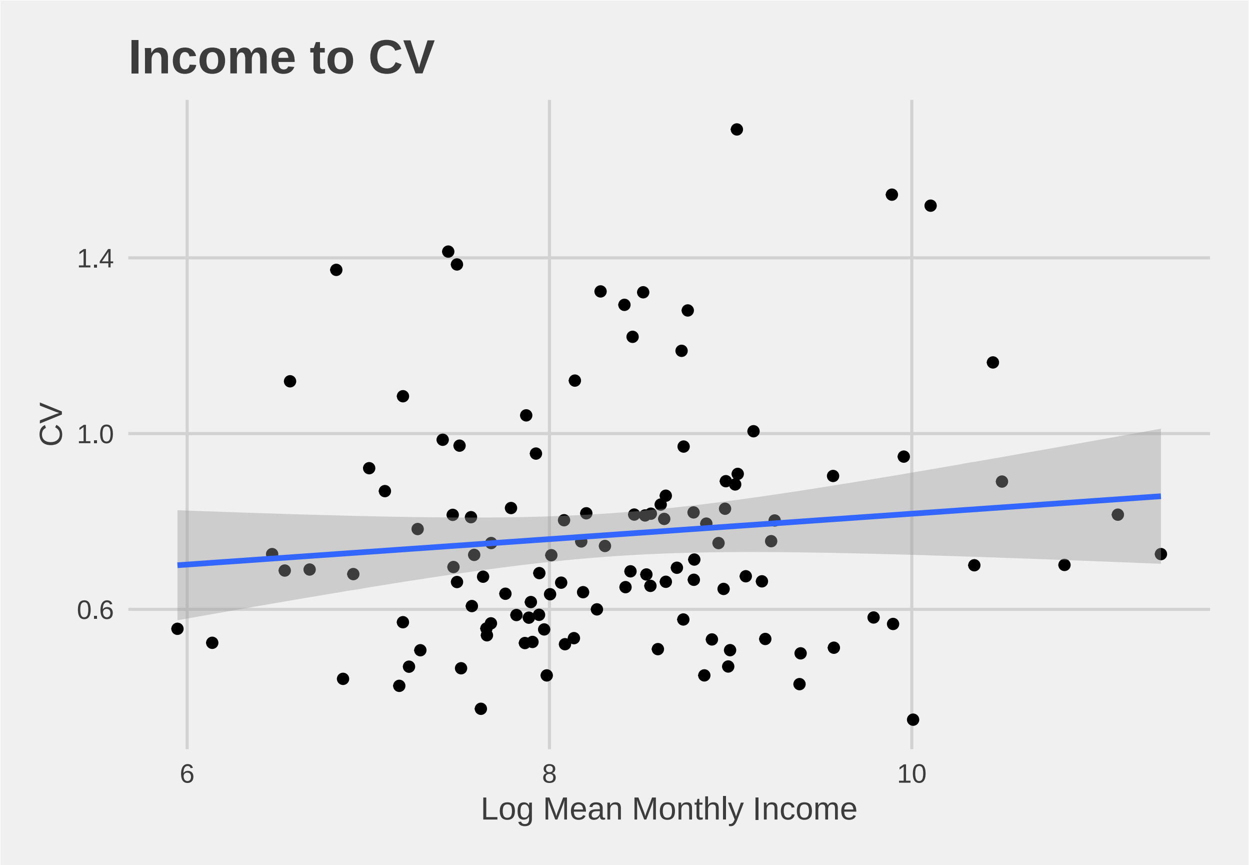
##CV Plots CV by quintile group



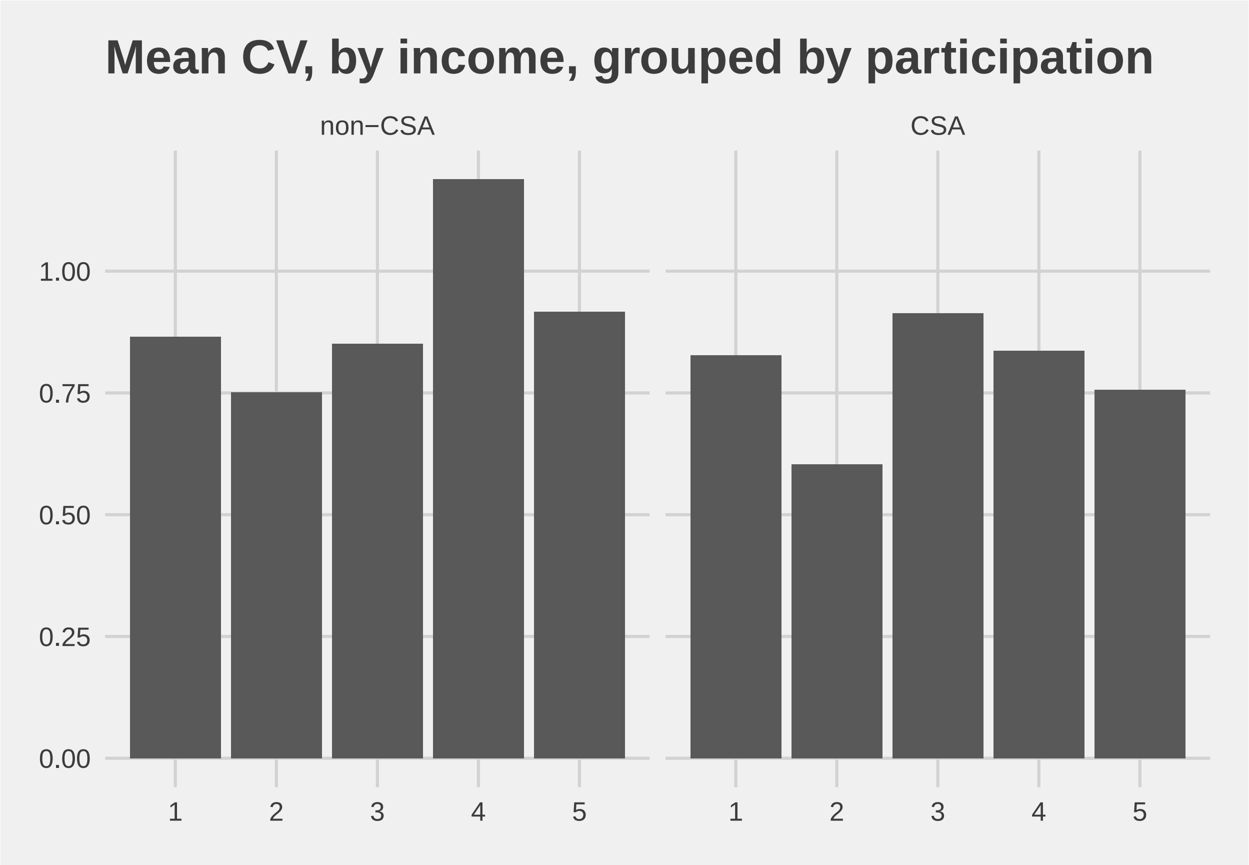
```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

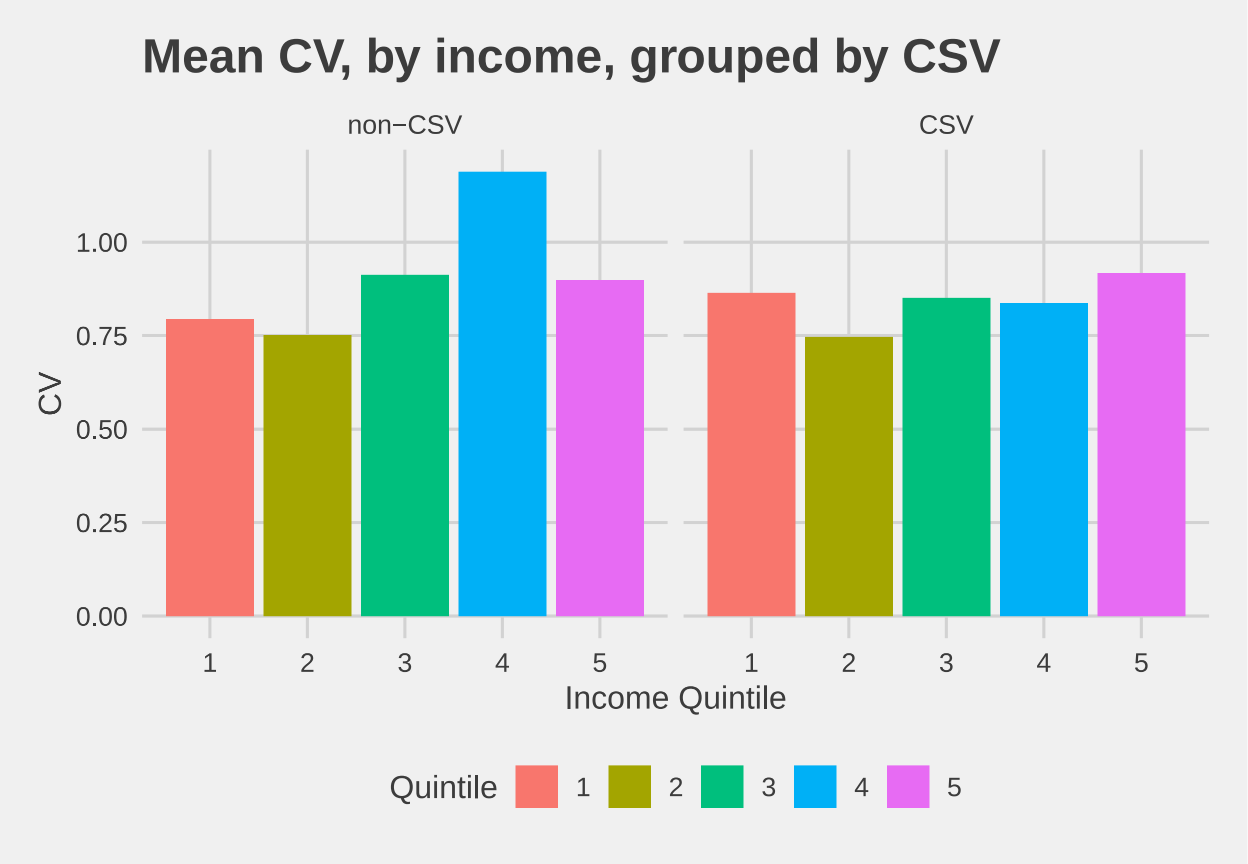


```
## 'geom_smooth()' using formula 'y ~ x'
```

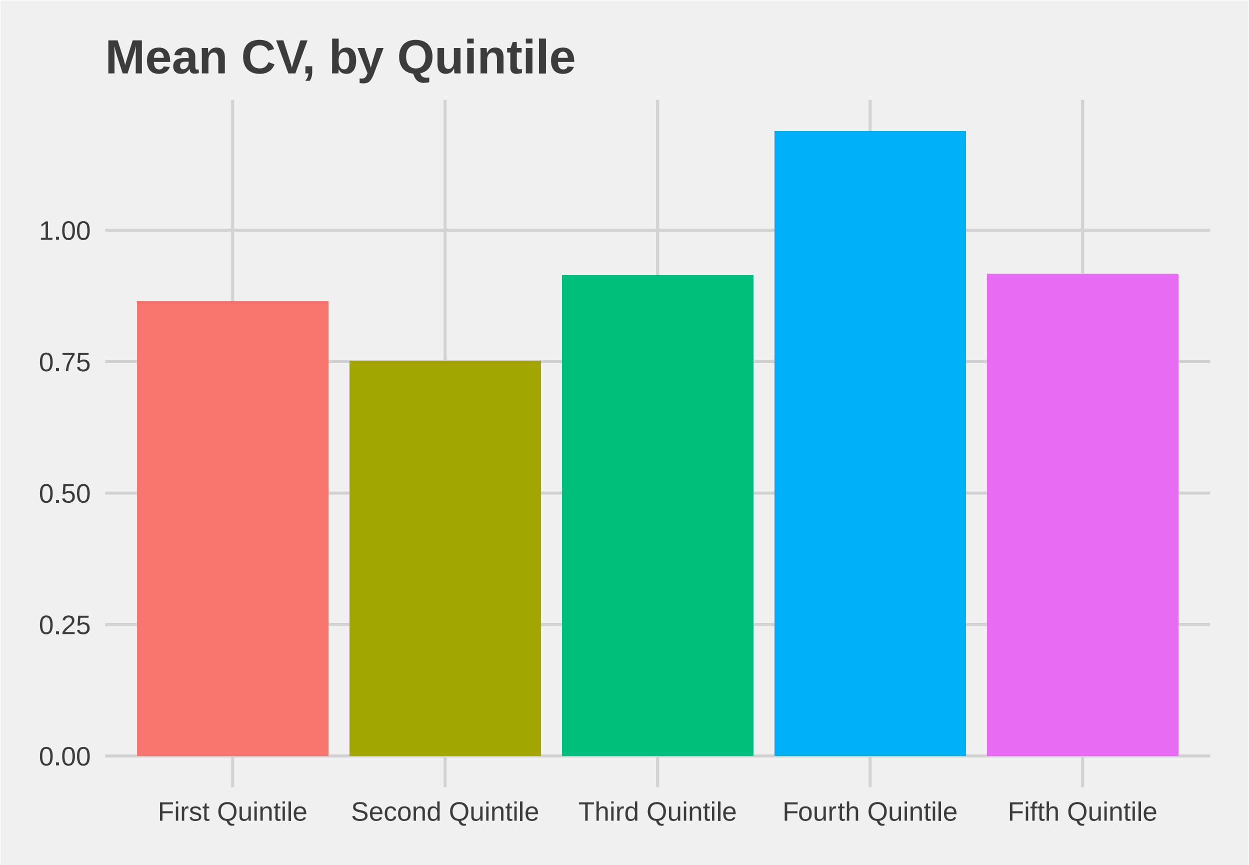


CV by participation group

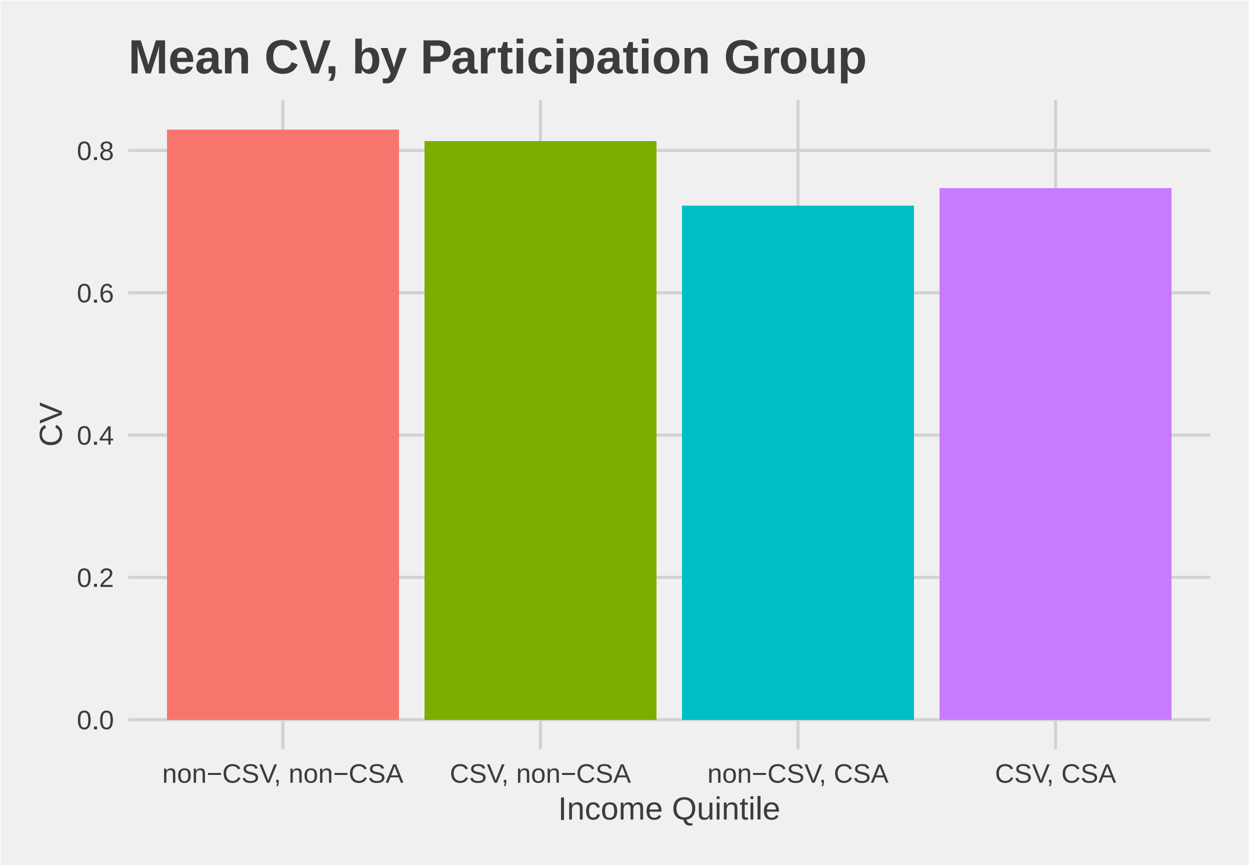

Mean CV, by income, grouped by participation
non−CSA
CSA
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
1
2
3
4
5



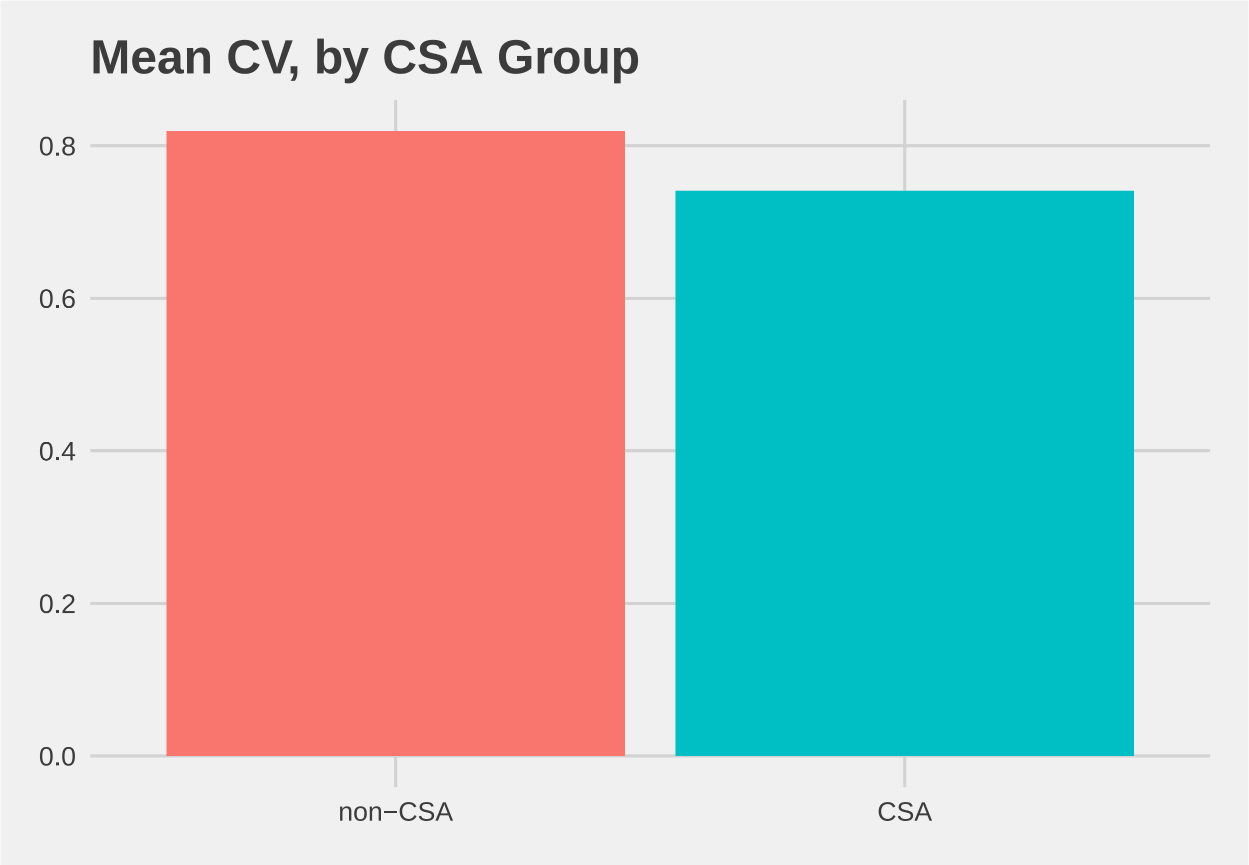
Non-grouped Cv, by quintile



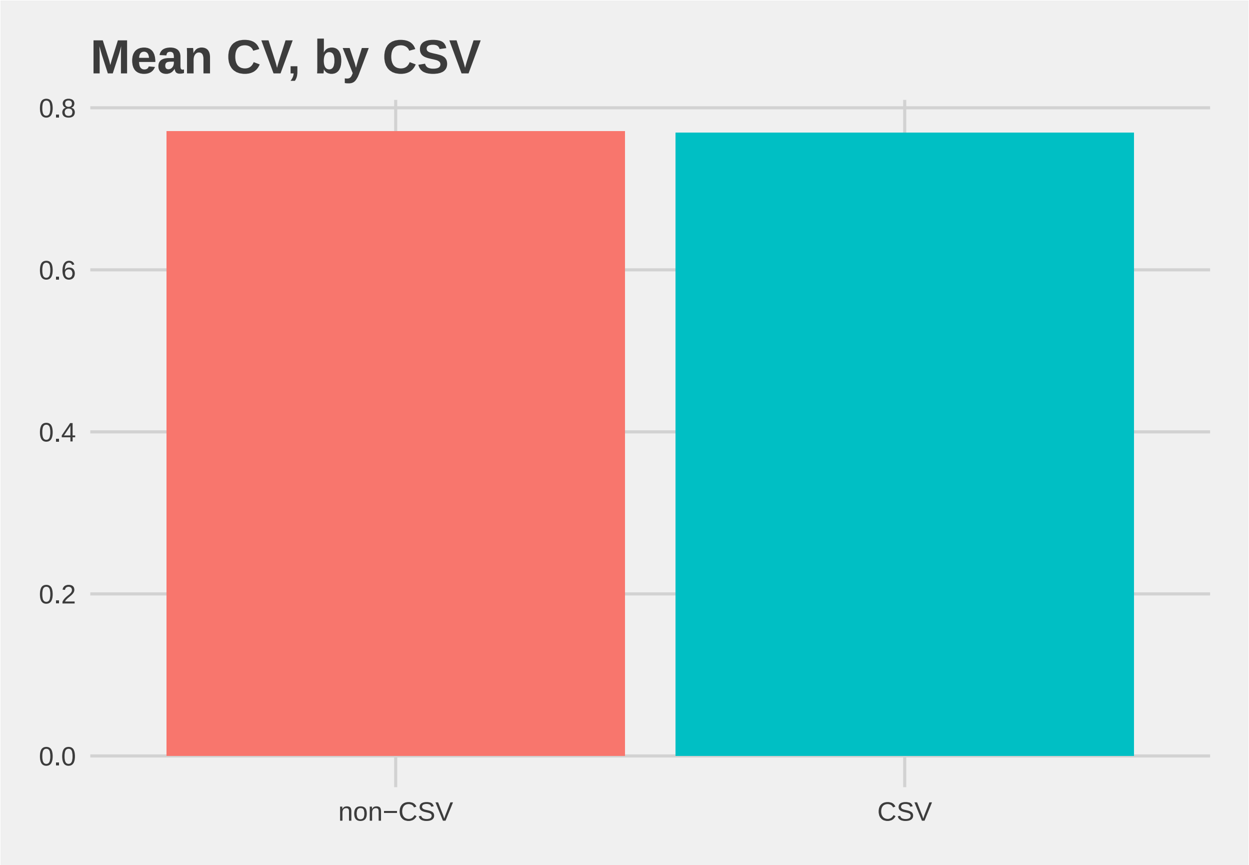
Non-grouped Cv, by participation



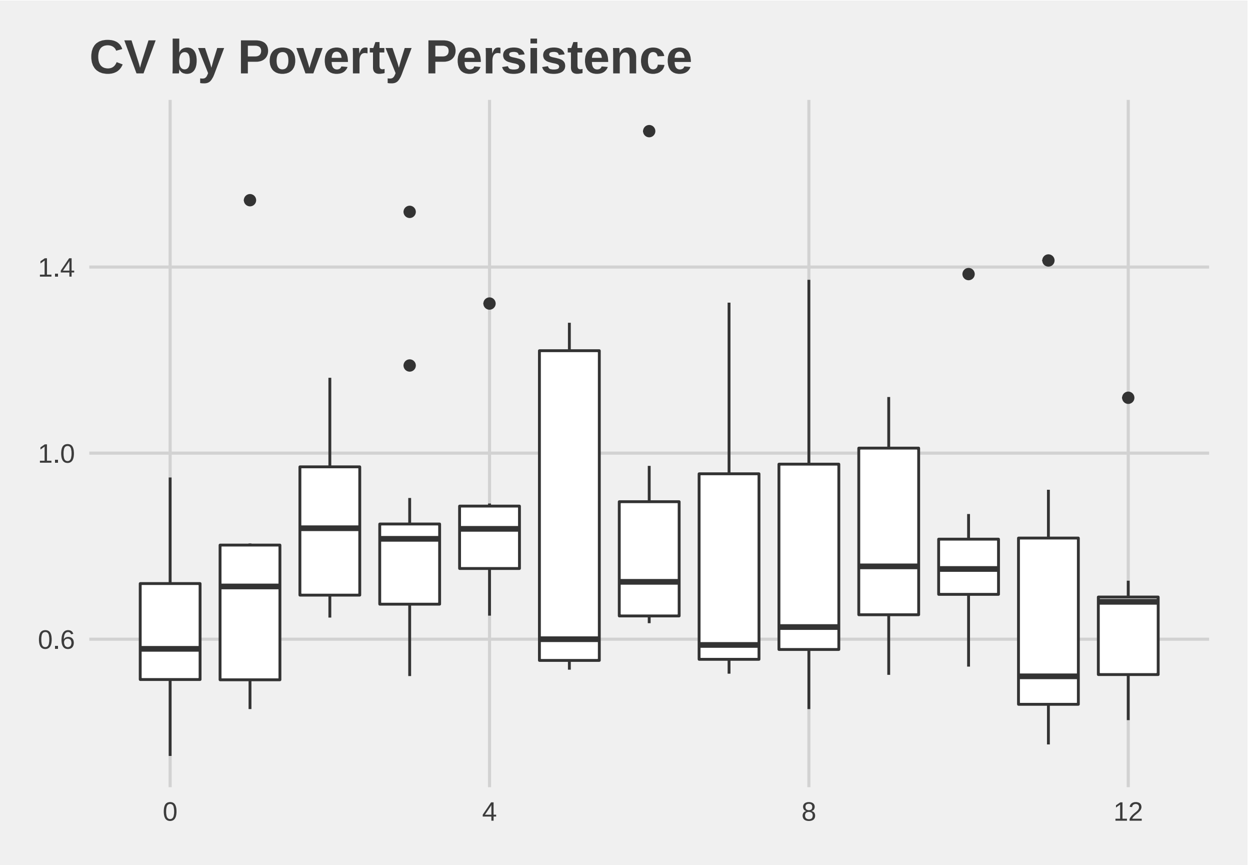
Mean CV by CSA



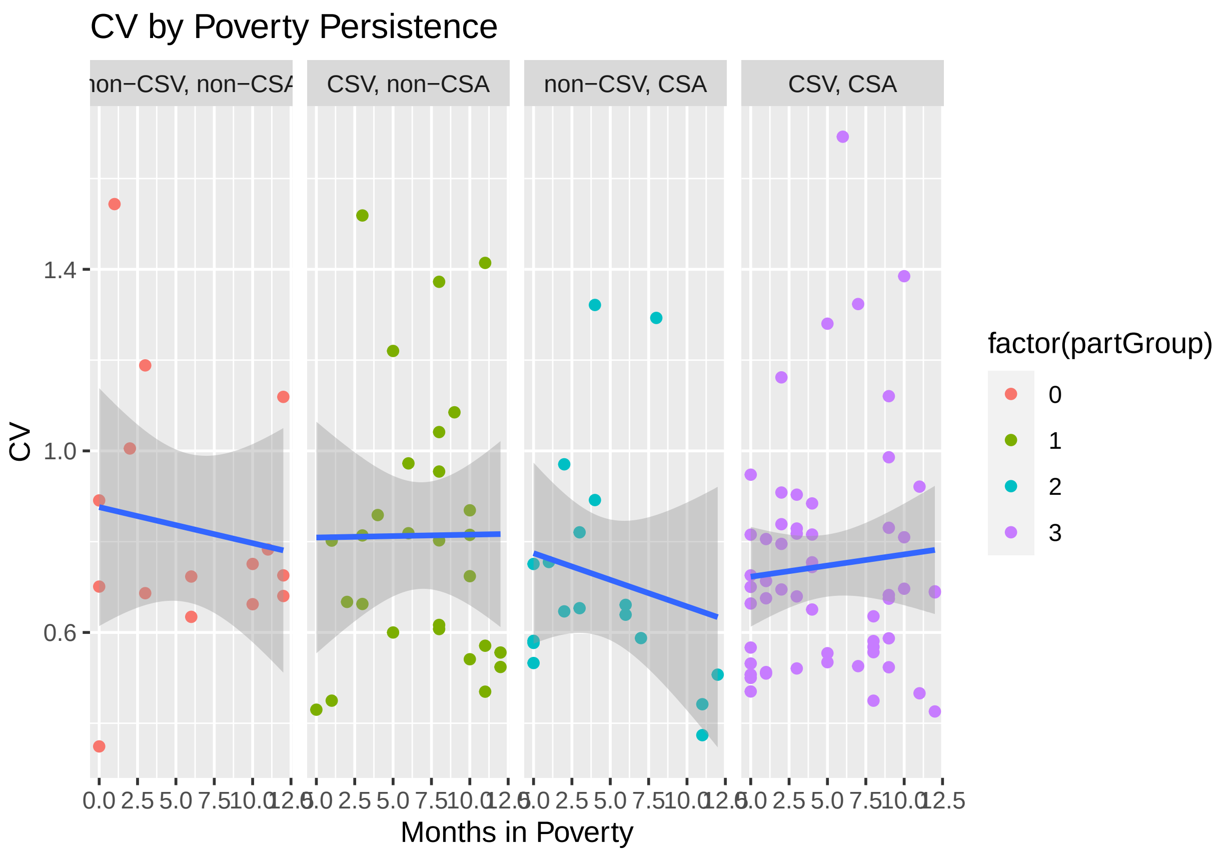
Mean CV by CSV

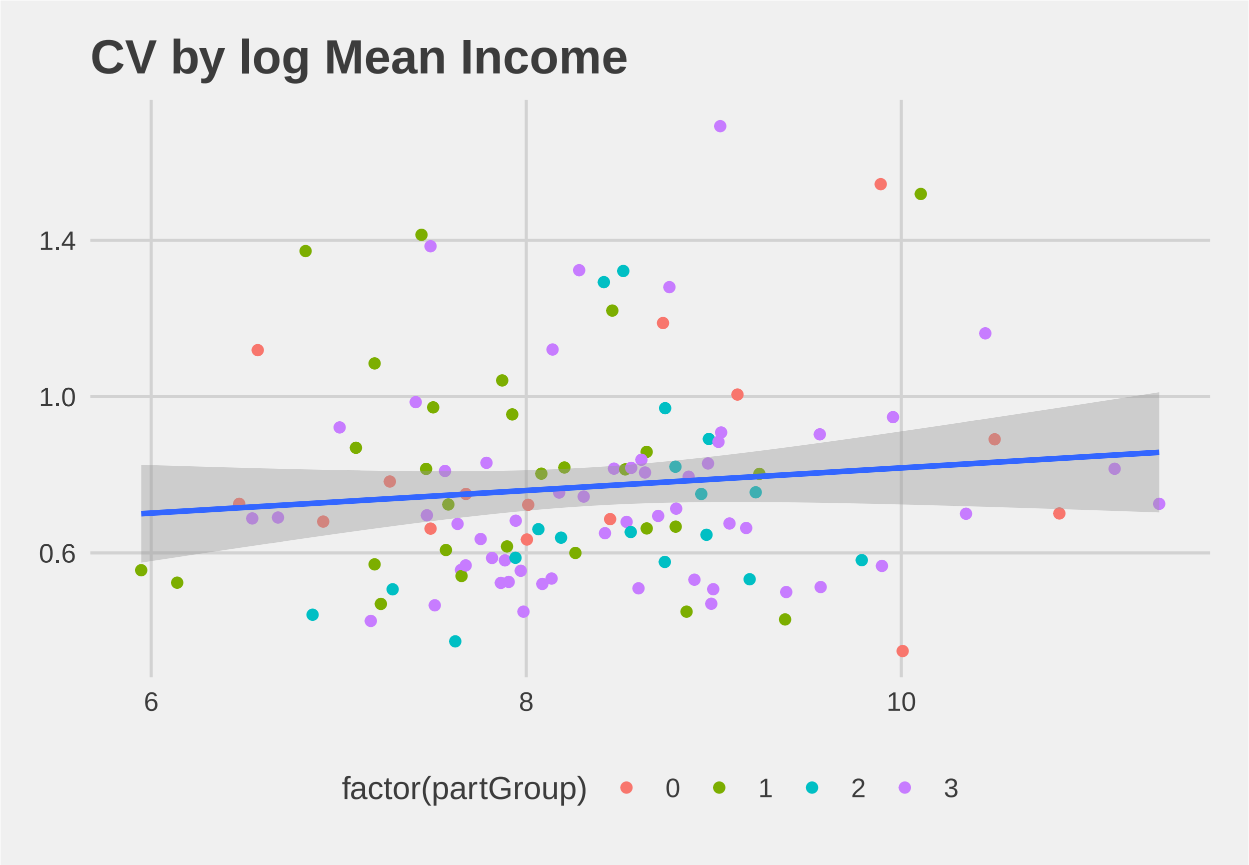


```
## ‘geom_smooth()‘ using formula ’y ~ x’
```

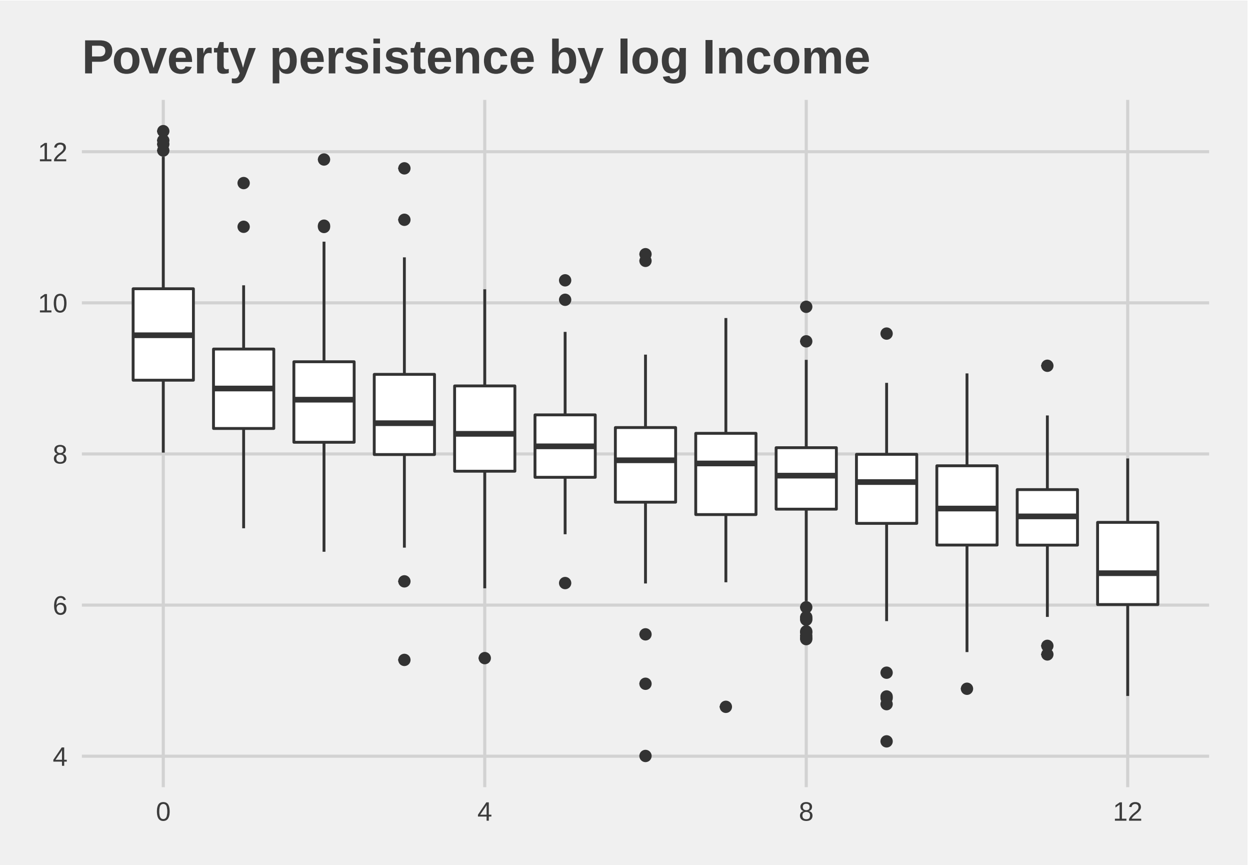


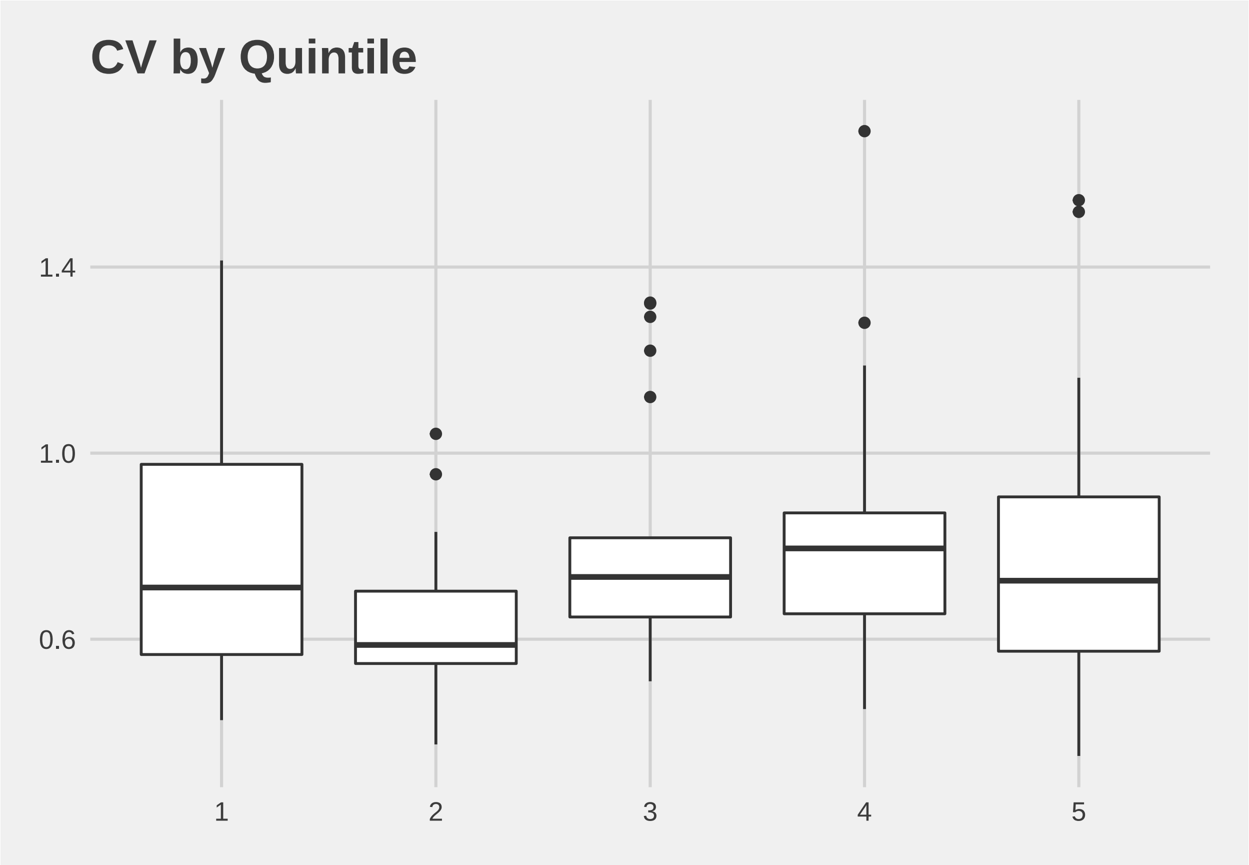
```
## 'geom_smooth()' using formula 'y ~ x'
```


CV by Poverty Persistence
non−CSV, non−CSA
CSV, non−CSA
non−CSV, CSA
CSV, CSA
CV
1.4
1.0
0.6
0.0
2.5
5.0
7.5
10.0
12.5
Months in Poverty
factor(partGroup)
0
1
2
3



Here's the same thing, but using household monthly income instead of the mean of them. Note: this means more observations of different income levels, but the same poverty persistence for each household, so we're just increasing the density of observations for incomes, based on endline poverty measures.


Poverty persistence by log Income
12
10
8
6
4
0
4
8
12



```
citation()
```

```
##
## To cite R in publications use:
##
##   R Core Team (2020). R: A language and environment for statistical
##   computing. R Foundation for Statistical Computing, Vienna, Austria.
##   URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {R: A Language and Environment for Statistical Computing},
##     author = {{R Core Team}},
##     organization = {R Foundation for Statistical Computing},
##     address = {Vienna, Austria},
##     year = {2020},
##     url = {https://www.R-project.org/},
##   }
##
## We have invested a lot of time and effort in creating R, please cite it
## when using it for data analysis. See also 'citation("pkgname")' for
## citing R packages.
```

```r
citation("dplyr")
```

```
##
## To cite package 'dplyr' in publications use:
##
##   Hadley Wickham, Romain François, Lionel Henry and Kirill Müller
##   (2020). dplyr: A Grammar of Data Manipulation. R package version
##   0.8.5. https://CRAN.R-project.org/package=dplyr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {dplyr: A Grammar of Data Manipulation},
##     author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller},
##     year = {2020},
##     note = {R package version 0.8.5},
##     url = {https://CRAN.R-project.org/package=dplyr},
##   }
```

```r
citation("tidyr")
```

```
##
## To cite package 'tidyr' in publications use:
##
##   Hadley Wickham and Lionel Henry (2020). tidyr: Tidy Messy Data. R
##   package version 1.0.2. https://CRAN.R-project.org/package=tidyr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {tidyr: Tidy Messy Data},
##     author = {Hadley Wickham and Lionel Henry},
##     year = {2020},
##     note = {R package version 1.0.2},
##     url = {https://CRAN.R-project.org/package=tidyr},
##   }
```

```r
citation("knitr")
```

```
##
## To cite the 'knitr' package in publications use:
##
##   Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report
##   Generation in R. R package version 1.28.
##
##   Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition.
##   Chapman and Hall/CRC. ISBN 978-1498716963
##
##   Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible
##   Research in R. In Victoria Stodden, Friedrich Leisch and Roger D.
##   Peng, editors, Implementing Reproducible Computational Research.
##   Chapman and Hall/CRC. ISBN 978-1466561595
```

```
##
## To see these entries in BibTeX format, use ’print(<citation>,
## bibtex=TRUE)’, ’toBibtex(.)’, or set
## ’options(citation.bibtex.max=999)’.
```

```r
citation("ggplot2")
```

```
##
## To cite ggplot2 in publications, please use:
##
##   H. Wickham. ggplot2: Elegant Graphics for Data Analysis.
##   Springer-Verlag New York, 2016.
##
## A BibTeX entry for LaTeX users is
##
##   @Book{,
##     author = {Hadley Wickham},
##     title = {ggplot2: Elegant Graphics for Data Analysis},
##     publisher = {Springer-Verlag New York},
##     year = {2016},
##     isbn = {978-3-319-24277-4},
##     url = {https://ggplot2.tidyverse.org},
##   }
```

```r
citation("haven")
```

```
##
## To cite package ’haven’ in publications use:
##
##   Hadley Wickham and Evan Miller (2019). haven: Import and Export
##   ’SPSS’, ’Stata’ and ’SAS’ Files. R package version 2.2.0.
##   https://CRAN.R-project.org/package=haven
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {haven: Import and Export ’SPSS’, ’Stata’ and ’SAS’ Files},
##     author = {Hadley Wickham and Evan Miller},
##     year = {2019},
##     note = {R package version 2.2.0},
##     url = {https://CRAN.R-project.org/package=haven},
##   }
```

```r
citation("ggthemes")
```

```
##
## To cite package ’ggthemes’ in publications use:
##
##   Jeffrey B. Arnold (2019). ggthemes: Extra Themes, Scales and Geoms
##   for ’ggplot2’. R package version 4.2.0.
##   https://CRAN.R-project.org/package=ggthemes
##
```

```
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {ggthemes: Extra Themes, Scales and Geoms for 'ggplot2'},
##     author = {Jeffrey B. Arnold},
##     year = {2019},
##     note = {R package version 4.2.0},
##     url = {https://CRAN.R-project.org/package=ggthemes},
##   }
```

```r
citation("car")
```

```
##
## To cite the car package in publications use:
##
##   John Fox and Sanford Weisberg (2019). An {R} Companion to Applied
##   Regression, Third Edition. Thousand Oaks CA: Sage. URL:
##   https://socialsciences.mcmaster.ca/jfox/Books/Companion/
##
## A BibTeX entry for LaTeX users is
##
##   @Book{,
##     title = {An {R} Companion to Applied Regression},
##     edition = {Third},
##     author = {John Fox and Sanford Weisberg},
##     year = {2019},
##     publisher = {Sage},
##     address = {Thousand Oaks {CA}},
##     url = {https://socialsciences.mcmaster.ca/jfox/Books/Companion/},
##   }
```

```r
citation("estimatr")
```

```
##
## To cite package 'estimatr' in publications use:
##
##   Graeme Blair, Jasper Cooper, Alexander Coppock, Macartan Humphreys
##   and Luke Sonnet (2020). estimatr: Fast Estimators for Design-Based
##   Inference. R package version 0.22.0.
##   https://CRAN.R-project.org/package=estimatr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {estimatr: Fast Estimators for Design-Based Inference},
##     author = {Graeme Blair and Jasper Cooper and Alexander Coppock and Macartan Humphreys and Luke So
##     year = {2020},
##     note = {R package version 0.22.0},
##     url = {https://CRAN.R-project.org/package=estimatr},
##   }
```

```r
citation()
```

```
##
## To cite R in publications use:
##
##   R Core Team (2020). R: A language and environment for statistical
##   computing. R Foundation for Statistical Computing, Vienna, Austria.
##   URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {R: A Language and Environment for Statistical Computing},
##     author = {{R Core Team}},
##     organization = {R Foundation for Statistical Computing},
##     address = {Vienna, Austria},
##     year = {2020},
##     url = {https://www.R-project.org/},
##   }
##
## We have invested a lot of time and effort in creating R, please cite it
## when using it for data analysis. See also 'citation("pkgname")' for
## citing R packages.
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.