Hw_2

Assignment 2

- Overview of Hw 2 using R, Python and STATA

A.1 Import and create folders

A.1.1 Import libraries

```r
# R
library(tidyverse)
library(foreign)
library(dplyr)
```

```python
# Python
import re
import os
import sys
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
* Stata
* Not necessary for stata
```

A.1.2 Set or change working directory

```r
# R
path = getwd()
```

```python
# Python
path = os.getcwd() #Main folder
files_folder = path+'/hh02dta_bc' #DTA files

```

```
* Stata

```

A.2 Calculatign consumption

A.2.1 Weekly variables

A.2.1.1 Import weekly variables data "i_cs"

```r
# R
df1 = read.dta(paste(path,'/hh02dta_b1/i_cs.dta', sep=''))
df1 = as_tibble(df1)
df1 = df1 %>% rename(Household_id = folio)
```

```python
# Python
path = os.getcwd() #Main folder
df1 = pd.read_stata(path+'/hh02dta_b1/i_cs.dta')
df1 = df1.rename(columns={'folio':'Household_id'})
```

```
* Stata
use "hh02dta_b1\i_cs.dta", clear

```

A.2.1.2 Select weekly consumption variables

A.2.1.2.1 Convert weekly varibles to monthly vars

```r
# R
#A.2.1.2 Select weekly consumption variables
cons1 = df1 %>%
  select(Household_id,
         intersect(contains('cs02a_'), ends_with('2')))

# A.2.1.2.1 Convert weekly varibles to monthly vars
cons1 = cons1 %>%
  mutate_at(vars(cs02a_12:cs02a_82),.funs = funs(. *4.3))
```

```python
# Python
#A.2.1.2 Select weekly consumption variables
cons1 = [i for i in df1.columns if i.startswith('cs02a_') and i.endswith('2')]
cons1.append('Household_id')
cons1 = df1[cons1]
cons1.set_index('Household_id', inplace=True)

# A.2.1.2.1 Convert weekly varibles to monthly vars
cons1 = cons1*4.3
```

```
* Stata
local weeklyvar "cs02a_12 cs02a_22 cs02a_32 cs02a_42 cs02a_52 cs02a_62 cs02a_72 cs02a_82"
foreach x in `weeklyvar' {
generate ics_monthly_`x' = `x' * 4.3
}
```

A.2.2 Monthly variables

```r
# R
cons2 = df1 %>%
  select(Household_id,
      intersect(contains('cs16'), ends_with('2')))
```

```python
# Python
cons2 = [i for i in df1.columns if i.startswith('cs16') and i.endswith('2')]
cons2.append('Household_id')
cons2 = df1[cons2]
cons2.set_index('Household_id', inplace=True)
```

```
* Stata
local 1month_var "cs16a_2 cs16b_2 cs16c_2 cs16d_2 cs16e_2 cs16f_2 cs16g_2 cs16h_2 cs16i_2"
foreach x in `1month_var' {
generate ics_monthly_`x' = `x'
}
```

A.2.3 3-Month variables

A.2.3.1 Import weekly variables data "i_cs1"

A.2.3.2 Select 3-month variables

A.2.3.2.1 Convert 3-month varibles to monthly vars

```r
# R
#### A.2.3.1 Import weekly variables data "i_cs1"
df2 = read.dta(paste(path,'/hh02dta_b1/i_cs1.dta', sep=''))
df2 = as_tibble(df2)
df2 = df2 %>% rename(Household_id = folio) #Rename

# A.2.3.2 Select 3-month variables
cons3 = df2 %>%
  select(Household_id,
      intersect(contains('cs22'), ends_with('2')))

# A.2.3.2.1 Convert 3-month varibles to monthly vars
cons3 = cons3 %>%
  mutate_at(vars(cs22a_2:cs22h_2),.funs = funs(. /3))
```

```python
# Python
# A.2.3.1 Import weekly variables data "i_cs1"
df2 = pd.read_stata(path+'/hh02dta_b1/i_cs1.dta')
df2 = df2.rename(columns={'folio':'Household_id' })

# A.2.3.2 Select 3-month variables
cons3 = [i for i in df2.columns if i.startswith('cs22') and i.endswith('2')]
cons3.append('Household_id')
```

```python
cons3 = df2[cons3]
cons3.set_index('Household_id', inplace=True)

# A.2.3.2.1 Select 3-month variables
cons3 = cons3/3
```

```
* Stata
* A.2.3.1 Import weekly variables data "i_cs1"
merge m:m folio using "hh02dta_b1\i_cs1.dta"

* A.2.3.2.1 Select 3-month variables
local 3month_var "cs22a_2 cs22b_2 cs22c_2 cs22d_2 cs22e_2 cs22f_2 cs22g_2 cs22h_2"

foreach x in `3month_var' {
generate ics1_monthly_`x' = `x' /3
}

```

A.2.4 Merge consumption variables into one dataframe

```r
# R
merge1 = merge(cons1, cons2, by = 'Household_id', all= TRUE)
cons_merge = merge(merge1, cons3, by = 'Household_id', all= TRUE)
head(cons_merge,3)
```

```python
# Python
cons_merge = pd.concat([cons1, cons2, cons3], axis=1)
```

```
* Stata
* Already merged
```

Homework Questions

Q.1.

- Calculate a measure of total consumption and per capita consumption for each household in the 2002 round.
    - To calculate per capita note you will have to calculate the number of individuals in each household.

Q.1.1 Total consumption

1.1.1 Calculate total consumer spending

- Sum the rows from cons_merge (A.2.4)

```r
# R
total_cons = cons_merge %>%
  mutate(
```

```
    consumption = rowSums(., cs02a_12:cs22h_2)
  )%>% select(Household_id, consumption)
```

```
# Python
cons_merge['total_cons'] = cons_merge.sum(axis=1)
```

```
* Stata
egen total_cons = rowtotal(ics1_monthly_cs22a_2 ics1_monthly_cs22b_2 ics1_monthly_cs22c_2
ics1_monthly_cs22d_2 ics1_monthly_cs22e_2 ics1_monthly_cs22f_2 ics1_monthly_cs22g_2
ics1_monthly_cs22h_2 ics_monthly_cs02a_12 ics_monthly_cs02a_22 ics_monthly_cs02a_32
ics_monthly_cs02a_42 ics_monthly_cs02a_52 ics_monthly_cs02a_62 ics_monthly_cs02a_72
ics_monthly_cs02a_82 ics_monthly_cs16a_2 ics_monthly_cs16b_2 ics_monthly_cs16c_2
ics_monthly_cs16d_2 ics_monthly_cs16e_2 ics_monthly_cs16f_2 ics_monthly_cs16g_2
ics_monthly_cs16h_2 ics_monthly_cs16i_2)
```

1.1.1.1 Basic stats: total consumer spending

```
# R
summary(total_cons$consumption)
```

```
# Python
cons_merge['total_cons'].describe()
```

```
* Stata
sum total_cons
```

1.1.1.2 Graph: total consumer spending

```
# R
hist(total_cons$consumption)
```

```
# Python
from scipy import stats
#Calculates Z-scores
z = np.abs(stats.zscore(cons_merge['total_cons']))
threshold = 3

outlier_removed = cons_merge['total_cons'][(z < 5)].values
import seaborn as sns
ax = sns.distplot(outlier_removed, hist=True, kde=True,
                  bins=int(180/5), color = 'darkblue',
                  hist_kws={'edgecolor':'black'},
                  kde_kws={'linewidth': 3})
ax.set_xlabel('Income')
ax.set_ylabel('Pct of households')
ax.set_title('Kernal density plot of household total income')
plt.show()
```

```
* Stata
egen total_cons_sd =std(total_cons)
```

```
histogram total_cons if total_cons_sd<3, bin(10)
```

Q.1.2 Per capita consumption (Total/house size)

1.2.1 Avg. house size

1.2.1.1 Import house size data set "c_ls"

1.2.1.2 Count family members "ls" in each household "folio"

```r
# R
# 1.2.1.1 Import house size data set "c_ls"
df_housesize = read.dta(paste(path,'/hh02dta_bc/c_ls.dta', sep=''))
df_housesize = as_tibble(df_housesize) #Set as tibble
head(df_housesize,2)

# 1.2.1.2 Count family members "ls" in each household "folio"
family_members = df_housesize %>% group_by(folio) %>% count(folio)
family_members = family_members%>% rename(family_members = n, Household_id=folio)
head(family_members)
```

```python
# Python
# 1.2.1.1 Import house size data set "c_ls"
df_housesize = pd.read_stata(path+'/hh02dta_bc/c_ls.dta')
df_housesize = df_housesize.rename(columns={'folio':'Household_id'})
df_housesize.set_index('Household_id', inplace=True)
merge = pd.merge(df_housesize, cons_merge, left_index=True, right_index=True)

# 1.2.1.2 Count family members "ls" in each household "folio"
merge['family_members'] = merge.groupby(merge.index.get_level_values(0))['ls'].count()
merge.family_members.head(5)
```

```
* Stata
* 1.2.1.1 Import house size data set "c_ls"
drop _merge
merge m:m folio using "hh02dta_bc\c_ls.dta"

*1.2.1.2 Count family members "ls" in each household "folio"
egen family_members = count(ls), by(folio)
histogram family_members, bin(10) title(Family members) xtitle(# of family members)
```

1.3 Per capita consumption (Total/house size)

1.3.1 Merge family_members and consumption data set

```r
# R
percap_consum = merge(family_members, total_cons, by='Household_id')
```

```python
# Python
```

```
merge['percap_consum'] = (merge.total_cons)/(merge.family_members)

```

```
* Stata

```

1.3.1.3 Calculate percapita consumption

```
# R
percap_consum = percap_consum %>%
mutate(percap_consum = percap_consum$consumption/percap_consum$family_members)
percap_consum%>% head(3)
```

```
# Python
merge['percap_consum'] = (merge.total_cons)/(merge.family_members)
```

```
* Stata
gen percap_consum = total_cons / family_members
egen percap_consum_sd = std(percap_consum)
histogram percap_consum  if percap_consum_sd <3, bin(100)
```

Q.2

- Calculate the set of poverty rates nationwide using the FGT indicators of poverty:
- Head count
- Average poverty gap
- Average poverty gap squared.
- Assume the poverty line=500 pesos per person. Provide poverty rates based on household consumption per capita.

2.1 Headcount using 1000 as an example

```
# R
povertyline = 500

below_poverty = percap_consum %>%filter(percap_consum < povertyline )%>%
select(percap_consum)
observations = length(percap_consum$percap_consum)
head_count = length(below_poverty$percap_consum)/observations

print(paste('Headcount: ', round(head_count*100, 2),'%', sep=''), quote=FALSE)
```

```
# Python
povertyline = 500
merge['below_poverty_dummy'] = merge.percap_consum< int(povertyline)
```

```

head_count = merge['below_poverty_dummy'].mean()
print('Headcount: {}'.format(head_count))
```

```
* Stata
gen poverty_line = 500
gen p0 =percap_consum <poverty_line  //Poor households dummy
la var p0 "poverty incidence"
la de p0 0 "non-poor" 1 "poor"
sum p0
local p0_tot=r(mean)
display `p0_tot'
```

2.2 Avg. poverty gap

```
# R
(povertyline-below_poverty$percap_consum)*head_count
```

```
# Python
((povertyline-merge['percap_consum'])*merge['below_poverty_dummy']).mean()
```

```
* Stata
gen p1=(poverty_line-percap_consum)*p0  //calculate the poverty gap for each household
la var p1 "poverty gap"
gen p1z= p1/poverty_line

sum p1z
local p1z_tot=r(mean)
display `p1z_tot'
```

2.3 Avg. poverty gap squared

```
# R
mean((povertyline-below_poverty$percap_consum)**2)
```

```
# Python
(((povertyline-merge['percap_consum'])*merge['below_poverty_dummy'])**2).mean()
```

```
* Stata
gen p2=(poverty_line-percap_consum)^2*p0
la var p2 "poverty gap squared"
gen p2z=p2/(poverty_line^2)

sum p2z
local p2z_tot=r(mean)
display `p2z_tot'
```

Q.3 Repeat 2) by area of residence.

- How does the poverty rate change by rural/urban residence?

3.1. Import residence data from "c_portad"

3.1.1. Merge residence df with percap_consum df from Q.1 & Q.2

```r
# R
# 3.1. Import residence data from "c_portad"
residence_df = read.dta(paste(path,'/hh02dta_bc/c_portad.dta', sep=''))
residence_df = residence_df%>% rename(Household_id = folio)
head(residence_df)

# 3.1.1. Merge residence df with percap_consum df from Q.1 & Q.2
consum_residence_df  = merge(residence_df, percap_consum, by='Household_id')
consum_residence_df  = as_tibble(consum_residence_df)
consum_residence_df %>% head(3)
```

```python
# Python
# 3.1
residence_df = pd.read_stata(path+'/hh02dta_bc/c_portad.dta')
residence_df  = residence_df.rename(columns={'folio':'Household_id'})
residence_df.set_index('Household_id', inplace=True)

# 3.1.1. Merge residence df with percap_consum df from Q.1 & Q.2¶
consum_residence_df  = pd.merge(residence_df, merge, left_index=True, right_index=True)

```

```stata
* Stata
* 3.1. Import residence data from "c_portad"
drop _merge
merge m:m folio using "hh02dta_bc\c_portad.dta"
```

3.1.1.2 Create poverty dummy

```r
# R
consum_residence_df$poverty_dummy <-
as.numeric(consum_residence_df$percap_consum<povertyline)
consum_residence_df %>% filter(poverty_dummy ==1) #Show observations living in poverty
```

```python
# Python
#Created previously
```

```stata
* Stata
*Created in 2.1 (Note don't run again)
gen poverty_line = 500
gen p0 =percap_consum <poverty_line  //Poor households dummy
la var p0 "poverty incidence"
la de p0 0 "non-poor" 1 "poor"
```

3.2 Show poverty by area of residence

```r
# R
consum_residence_df %>%
group_by(estrato) %>% #Groupby estrato
count(poverty_dummy) %>% #Count poverty dummy by each estrato
mutate(obs = sum(n))%>%
mutate(head_count = (n/obs)*100)%>%
filter(poverty_dummy==1) %>%
select(estrato, head_count)
```

```python
# Python
consum_residence_df.groupby('estrato')['below_poverty_dummy'].mean()
```

```
* Stata
tab estrato p0, row
```

Q.4.

Calculate the Gini coefficient overall and by urban and rural areas both using consumption measures. In both R and Stata there are programs which can be downloaded to calculate the Gini. Provide a graph of the Lorenz curve.

4.1 Calculate cumulative sum for population and consumption

Note: Python code runs but does not produce the correct results

```r
# R
gini = percap_consum %>%
arrange(percap_consum) %>% #Sort consumption from least to greatest
mutate(consum_cumulative = cumsum(percap_consum), consum_total = sum(percap_consum),
pop_total = sum(family_members), pop_cumulative = cumsum(family_members))%>% #Calculate
total and cumulative sum for variables
mutate(consum_pct = (consum_cumulative/consum_total), pop_pct= ((pop_cumulative/
pop_total))) #Calculate quintiles
gini %>% head(3) #Show new data
```

```python
# Python
var_list = ['percap_consum', 'family_members']
for i in var_list:
    consum_residence_df[str(i)+'_cumulative'] = consum_residence_df[str(i)].cumsum()
    consum_residence_df[str(i)+'_cumulative_pct'] =
consum_residence_df[str(i)+'_cumulative']/sum(consum_residence_df[str(i)])
```

```
* Stata
sort percap_consum
gen rank_percap_consum  =_n
cumul percap_consum, generate(cum_percap_consum)
```

```
egen total_percap_consum =total(percap_consum)
sort percap_consum


xtile q5=percap_consum, n(5)  //create the variable that indicates the quintiles
sum percap_consum
scalar percap_consum_sum=r(sum)
 forvalues i = 1/5 {
    quietly sum percap_consum if q5==`i'
    scalar percap_consum_`i'_sum=r(sum)
    scalar percap_consum_`i'_share=percap_consum_`i'_sum/percap_consum_sum
    }
scalar list  percap_consum_1_share  percap_consum_2_share  percap_consum_3_share
percap_consum_4_share  percap_consum_5_share


gen cum_total_conspc=0
replace cum_total_conspc=  percap_consum if _n==1
local N=_N
forvalues k=2/`N' {
    quietly replace cum_total_conspc = percap_consum+cum_total_conspc[_n-1] if _n==`k'
    }
gen cuml_conspc=cum_total_conspc/total_percap_consum  //create the variable for the
cumulative consumption share

gen pop_share = rank_percap_consum/35677
```

4.1.1 Plot:

```
# R
library(ggplot2)

ggplot(data= gini, aes(x=pop_pct, y= consum_pct))+
ggtitle("Lorenz curve")+theme(plot.title = element_text(hjust = 0.5))+
geom_line()+
geom_abline(intercept = 0, slope = 1, color='red')+xlab('cum. % of households')+ylab('cum. %
consum/percap')
```

```
# Python
plt.plot(consum_residence_df['percap_consum_cumulative_pct'],consum_residence_df['family_mem
bers_cumulative_pct'])
```

```
* Stata
line cuml_conspc pop_share, title("lorenz curve") xtitle("cum. % of households")
ytitle("cum. % of consumption per capita")

```

4.2 Gini coefficient calculation

```r
# R
cov_consum_V_consum_pct = cov(gini$percap_consum, gini$consum_pct)
mean_cons = mean(gini$percap_consum)
print((2*cov_consum_V_consum_pct)/(mean_cons))
```

```python
# Python
cov_consum_V_consum_pct =
consum_residence_df[['percap_consum','percap_consum_cumulative_pct']].cov().iloc[0,1:]
mean_cons = np.mean(consum_residence_df['percap_consum'])
print((2*cov_consum_V_consum_pct)/mean_cons)
```

```
* Stata
correlate percap_consum cuml_conspc, covariance
scalar conspc_cov=r(cov_12) //Calculate the covariance
su percap_consum
scalar conspc_mean=r(mean)  //Calculate the mean
scalar gini=2*conspc_cov/conspc_mean   //Gini coefficient
display gini
```