Stata Code Sample *

Xinghuan Luo xinghuanluo@uchicago.edu

Dec. 17 2020

Background

This sample is my code for a data task. It has 4 parts:
- Part 0 Initialization
- Part 1 Data Cleaning
- Part 2 Data Exploration
- Part 3 Estimation and Causal Inference

0. Initialization

```
. clear
. set more off
. set varabbrev off
.
. global bfi_test "D:/OneDrive - The University of Chicago/2021 Fall/job searching/BFI/Dube"
.
. local top_file `""scripts" "results" "raw_data""´
. foreach file_path in `top_file´ {
  2. cap mkdir "$bfi_test/`file_path´"
  3. global `file_path´ "$bfi_test/`file_path´"
  4. }
.
. local data_path `""car_data" "market_data" "merged_data""´
. foreach data in `data_path´ {
  2. cap mkdir "$raw_data/`data´"
  3. global `data´ "$raw_data/`data´"
  4. }
.
. local results_path `""tables" "graphs" "latex" "logs""´
. foreach result in `results_path´ {
  2. cap mkdir "$results/`result´"
  3. global `result´ "$results/`result´"
  4. }
.
```

1. Data Cleaning

I cleaned the two new data set below and modified variables for merging them together later. I converted the abbreviation of country name to the full name

```
. use "$merged_data/all_market_data.dta", clear
(Stata file created from 5 csv files using csvconvert)
.
. local country_list `""Belgium" "France" "Germany" "Italy" "United Kingdom""´
. foreach country in `country_list´ {
  2. qui replace ma = "`country´" if ma == substr("`country´", 1, 1)
  3. }
. qui save "$merged_data/all_market_data.dta", replace
.
. use "$merged_data/all_car_data.dta", clear
(Stata file created from 21 csv files using csvconvert)
```

*I finished this markdown file by `markstat`. The source code is in my github repository, here

```
. qui replace ma = "United Kingdom" if ma == "UK"
. qui replace ye = 1900 + ye
.
. qui merge m:1 ye ma using "$merged_data/all_market_data", nogen
. order model, before(loc)
```

I transformed all li(measure of fuel consumption) variables into number so that I could fill in the missing values. I first filled in the missing values in li and used the value of li to fill in other missing values.

```
. qui destring li*, replace force
. qui replace li = li1 + li2 + li3 if mi(li) & !mi(li1, li2, li3)
.
. foreach var in li1 li2 li3 {
  2. qui replace `var´ = 0 if mi(`var´)
  3. }
. foreach var in li1 li2 li3 {
  2. qui replace `var´ = li*3 - (li1 + li2 + li3) if `var´ == 0
  3. }
.
. qui save "$merged_data/cleaning_done_data.dta", replace
```

2. Data Exploration

I saved the original data set so that I could use it later. Then, I created two tempfiles, only_1970 and only_1990.

```
. tempfile original_data
. save `original_data´
file D:\stata\temp_file\ST_4670_000001.tmp saved
.
. preserve
. tempfile only_1970
. keep if ye == 1970
(7,407 observations deleted)
. save `only_1970´
file D:\stata\temp_file\ST_4670_000003.tmp saved
. restore
.
. preserve
. tempfile only_1990
. keep if ye == 1990
(7,281 observations deleted)
. save `only_1990´
file D:\stata\temp_file\ST_4670_000005.tmp saved
. restore
.
. cap program drop data_manipulation
. program define data_manipulation
  1. foreach temp_file of local 0 {
  2.     use `temp_file´, clear
  3.     pctile hp_pct = hp, nq(10)
  4.     xtile decile_grp = hp, cut(hp_pct)
  5.     bysort decile_grp: asgen avg_fuel = li, weight(qu)
  6.     bysort decile_grp: egen mid_hp = median(hp)
  7.     bysort decile_grp: egen num_obs = count(avg_fuel)
  8.     gen log_hp = log(hp)
  9.     qui reg avg_fuel hp log_hp [pweight=qu]
 10.     qui predict y_hat
 11.     save `temp_file´, replace
 12. }
 13. end
.
. cap ssc install asgen
. data_manipulation `only_1970´ `only_1990´
(Stata file created from 21 csv files using csvconvert)
file D:\stata\temp_file\ST_4670_000003.tmp saved
(Stata file created from 21 csv files using csvconvert)
file D:\stata\temp_file\ST_4670_000005.tmp saved
.
. use `only_1970´, clear
(Stata file created from 21 csv files using csvconvert)
```

```
. append using `only_1990´

.
. tempfile all_70_90

. qui save `all_70_90´

.
. tempfile prepare_scatter

. collapse (mean) avg_fuel mid_hp num_obs, by(ye decile_grp)

. foreach var of varlist _all {
  2. rename `var´ unique_`var´
  3. }

. qui save `prepare_scatter´

.
. use `all_70_90´, clear
(Stata file created from 21 csv files using csvconvert)

. qui merge 1:1 _n using `prepare_scatter´, nogen

. local scatter_settings msize(small) jitter(4)

. qui twoway (scatter unique_avg_fuel unique_mid_hp if unique_ye == 1970 [fweight=unique_num_obs], `scatter_settings´ color
> (blue)) ///
>    (scatter unique_avg_fuel unique_mid_hp if unique_ye == 1990 [fweight=unique_num_obs], `scatter_settings´ color(dkorang
> e)), ///
>    graphregion(color(white)) legend(label(1 1970) label(2 1990) nobox region(lcolor(white)))  xlabel(15(15)120 ,labsize(s
> mall)) ///
>    xtitle("Midpoint of Each Horsepower Decile") ytitle("Sales-Weighted Average of Fuel Consumption") ///
>    note("The relative size of the each scatter point represents the number of observations it has.")

. qui graph export "$graphs/only_scatter.png", as(png) replace

.
. local scatter_settings msize(small) jitter(4)

. local line_settings lcolor(gs0) sort

. qui twoway (scatter unique_avg_fuel unique_mid_hp if unique_ye == 1970 [fweight=unique_num_obs], `scatter_settings´ color
> (blue)) ///
>    (line y_hat hp  if ye == 1970,  `line_settings´ lpattern(shortdash dot)) ///
>    (scatter unique_avg_fuel unique_mid_hp if unique_ye == 1990 [fweight=unique_num_obs], `scatter_settings´ color(dkorang
> e)) ///
>    (line y_hat hp  if ye == 1990, `line_settings´ lpattern(longdash)), ///
>     xlabel(15(15)150, labsize(small)) ylabel(5(2.5)15) graphregion(color(white)) ///
>     legend(label(1 1970 ) label(2 1970 ) label(3 1990) label(4 1990) nobox region(lcolor(white))) ///
>     xtitle("Horsepower") ytitle("Sales-Weighted Average of Fuel Consumption" ) ///
>     note("Both scatter points and fitted lines describe the relationship between horsepower and fuel consumption. " ///
>          "The fitted lines are generated by the regression with sales as sample weights. " ///
>          "The relative size of the each scatter point represents the number of observations it has.")

. qui graph export "$graphs/scatter_fitted.png", as(png) replace
```

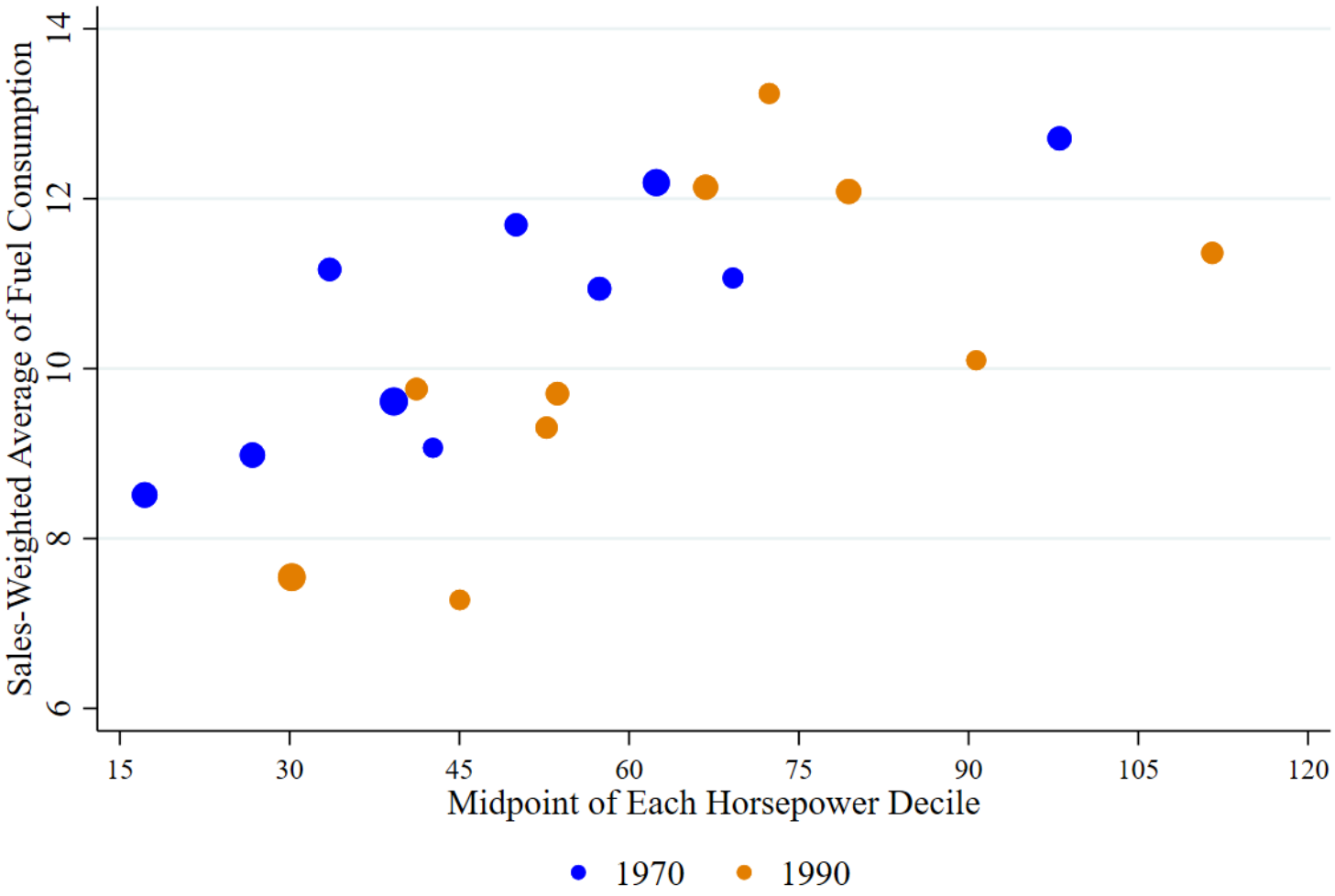

Figure 1: Relationship between Sales-Weighted Average of Fuel Consumption and Horsepower in 1970 and 1990

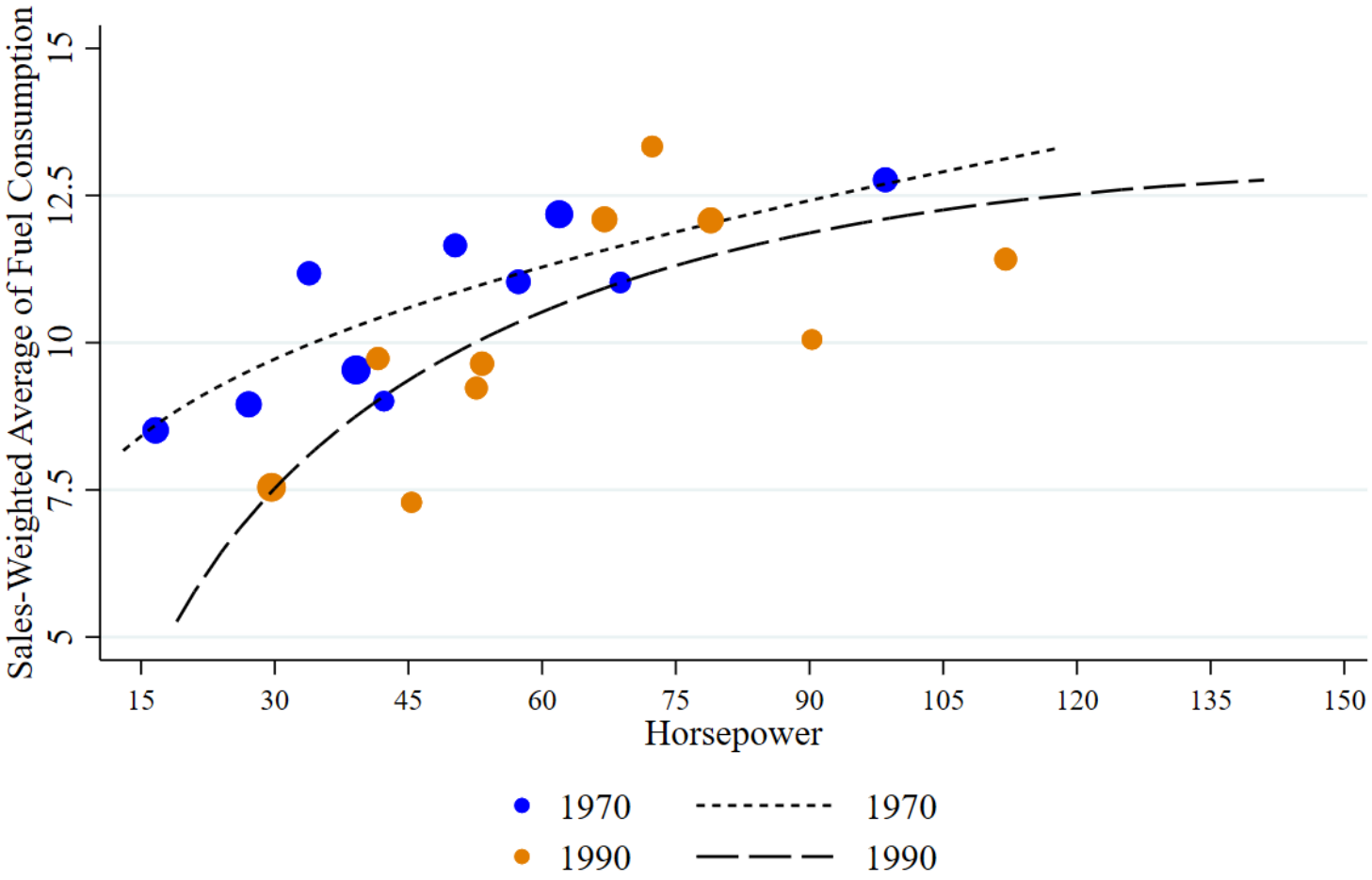

Figure 2: Relationship between Sales-Weighted Average of Fuel Consumption and Horsepower in 1970 and 1990

I first collapsed the data set and used texsave to create the required graph in question 6

```
. collapse (min) min_hp = hp (max) max_hp = hp (mean) mean_fuel = avg_fuel (count) num_obs = avg_fuel if ye == 1990, by(dec
> ile_grp)
. egen hp_interval = concat(min_hp max_hp), punct("--")
. drop decile_grp min_hp max_hp
.
. order hp_interval, first
. label var hp_interval "Horsepower(kW)"
. label var mean_fuel "Fuel Consumption"
. label var num_obs "\(N\)"
.
. replace mean_fuel = round(mean_fuel, .01)
(10 real changes made)
.
. local title title("Sales-Weighted Average of Fuel Consumption by Decile of Horsepower in 1990")
. local footnote footnote("Notes: Horsepower column represents the range of horsepower in each decile group. Fuel Consumpti
> on column represents the sales-weighted average of fuel consumption (liter per km) of each decile gruop.")
. qui texsave using "$tables/summarized_table.tex", varlabels nofix replace `footnote´ frag `title´ marker(tab: tb1)
```

3.Estimation and Causal Inference

```
. use `original_data´, clear
(Stata file created from 21 csv files using csvconvert)
```

Table 1: Sales-Weighted Average of Fuel Consumption by Decile of Horsepower in 1990

Horsepower(kW)	Fuel Consumption	N
19–33	7.55	61
34–40	9.82	37
41–46	7.25	30
48–54	9.46	36
55–57	9.83	41
59–66	12.22	48
67–75	13.04	32
76–85	12.1	49
87–96	10.19	28
96.5–141	11.24	36

Notes: Horsepower column represents the range of horsepower in each decile group. Fuel Consumption column represents the sales-weighted average of fuel consumption (liter per km) of each decile gruop.

```
.
. label var ye "Year"
. label var li "Fuel Consumption"
. label var eurpr "Price in Euro"
.
. bysort ye ma: gen N_jt = pop / 4
. bysort ye ma model: egen total_model_sale = total(qu)
. bysort ye ma : egen total_car_sale = total(qu)
. gen S_ijt = total_model_sale / N_jt
. gen S_0jt = 1 - (total_car_sale / N_jt)
.
. bysort ye ma model: gen Y_ijt = log(S_ijt) - log(S_0jt)
.
. eststo clear
. eststo model1: qui reg Y_ijt li eurpr, r
.
. encode ma, generate(market)
. encode model, generate(model_code)
.
. label var market "Market"
. label var model_code "Model Code"
.
. eststo model2: qui reg Y_ijt li eurpr i.ye i.market i.model_code, r
. qui esttab using "$tables/two_regressions.tex", ///
>    replace p keep(li eurpr) booktabs width(\hsize) nofloat label ///
>    mtitles("Model 1" "Model 2") nonumbers ///
>    addnotes("Model 1 is conventional OLS regression for question 2. " ///
>    "Model 2 is OLS regression with fixed effects of car model, market and year for question 3. ")
. * EOF
```

	Model 1	Model 2
Fuel Consumption	-0.0163***	-0.00203
	(0.000)	(0.243)
Price in Euro	-0.0000669***	-0.0000922***
	(0.000)	(0.000)
Observations	7679	7653

p-values in parentheses

Model 1 is conventional OLS regression for question 2.

Model 2 is OLS regression with fixed effects of car model, market and year for question 3.

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 2: OLS Regressions of Problem 2 and 3