Code Sample *

Haoliang Hu huhaoliang@whu.edu.cn

Dec. 15 2023

Background

This sample is my code for a data task. It has 5 parts:
- Part 0 Initialization
- Part 1 Data Cleaning
- Part 2 Data Exploration
- Part 3 Estimation and Causal Inference
- Part 4 Further Analysis

0. Initialization

```
. clear all

. set more off

. set maxvar 20000

. set scheme cleanplots, perm
(set scheme preference recorded)

.
. // If the reader wants to replicate the results, he/she just needs to change
. // this global path and put the data in raw_data file.
. global path "C:\Users\huhu\Desktop\Code Task\David Chan Data Task\"

. global D    "$path\data"      //data file

. global Out  "$path\out"       //result: graph and table

. cd "$D"                       //set current working directory
C:\Users\huhu\Desktop\Code Task\David Chan Data Task\data

.
.
. import delimited "test_data.csv",clear
(encoding automatically selected: ISO-8859-2)
(8 vars, 8,831 obs)

. save raw_data.dta, replace
file raw_data.dta saved

.
. *********************Question 0***************************************
. // summarize data
. qui summarize, detail

. qui duplicates list arrive leave

.
```

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

Summarize the data. I find this data set contains entries of patient flow. The shift duration typically lasts around 9 to 10 hours, and there are rare situations where the shift lasts only 2 hours. And I checked the duplication of the arrive and leave times of patients and found some observations that may appear to be data entry errors as they have exactly the same arrive and leave times, which may result from coding errors of the ED system. In the following analysis, I take them as true data for convenience.

Group	Obs	arrive	leave
1	2558	03jul1982 07:27:03	03jul1982 08:34:46
1	2659	03jul1982 07:27:03	03jul1982 08:34:46
2	7905	09jun1982 06:11:17	09jun1982 08:43:06
2	7973	09jun1982 06:11:17	09jun1982 08:43:06
3	2564	11jul1982 04:57:33	11jul1982 05:25:57
3	2710	11jul1982 04:57:33	11jul1982 05:25:57
4	6565	13jun1982 10:00:11	13jun1982 12:02:31
4	6833	13jun1982 10:00:11	13jun1982 12:02:31
5	3563	16jun1982 18:25:49	16jun1982 21:47:53
5	3777	16jun1982 18:25:49	16jun1982 21:47:53
6	7893	18jun1982 12:38:47	18jun1982 14:54:13
6	8632	18jun1982 12:38:47	18jun1982 14:54:13
7	6710	24may1982 14:20:20	24may1982 15:46:38
7	6795	24may1982 14:20:20	24may1982 15:46:38
8	8804	28may1982 10:30:29	28may1982 11:17:27
8	8812	28may1982 10:30:29	28may1982 11:17:27
9	2522	29may1982 10:55:13	29may1982 15:35:56
9	5467	29may1982 10:55:13	29may1982 15:35:56
10	1079	30may1982 08:53:35	30may1982 10:03:29
10	1122	30may1982 08:53:35	30may1982 10:03:29

Table 1: Possible Data Entry Errors

1. Data Cleaning

First I transfer the a.m./p.m. to 24-hours in order to transfer the original datatime format into stata time format. Notably, we should use double here to generate the new stata datatime variable. Finally we get there are **7.43%** patients arriving before their physician's shift starts and **19.01%** patients discharged after their physician's shift ends.

```
. *********************Question 1*****************************************
. // transfer datetime
. gen shift_date_time = date(shift_date, "DMY")

. format shift_date_time %td

.
. // extract hour and AM/PM indicator from shift_start and shift_end
. // replace noon to 12 pm for conviency
. qui replace shift_start="12 p.m." if shift_start=="noon"

. qui replace shift_end="12 p.m." if shift_end=="noon"

.
. gen hour_start = real(substr(shift_start, 1, strpos(shift_start, " ") - 1))

. gen am_pm_start = substr(shift_start, -4, 4)

.
```

```
. // do the same for shift_end
. gen hour_end = real(substr(shift_end, 1, strpos(shift_end, " ") - 1))
. gen am_pm_end = substr(shift_end, -4, 4)
.
. // convert to 24 hour format
. qui replace hour_start = hour_start + 12 if am_pm_start == "p.m." & hour_start != 12
. qui replace hour_end = hour_end + 12 if am_pm_end == "p.m." & hour_end != 12
.
. // combine data with time
. gen shift_start_data_time = shift_date + " " + string(hour_start, "%02.0f") + ":00:00"
. gen shift_end_data_time = shift_date + " " + string(hour_end, "%02.0f") + ":00:00"
. // adjust for across day pattern
. gen shift_date_time_1 = string(shift_date_time + 1, "%td")
.
. qui replace shift_end_data_time = shift_date_time_1 + " " + string(hour_end, "%02.0f") ///
>  + ":00:00" if shift_start == "7 p.m."
.
.
. // note, use double to ensure precient
. gen double shift_start_time = clock(shift_start_data_time, "DMY hms")
. gen double shift_end_time = clock(shift_end_data_time, "DMY hms")
. format shift_start_time %tc
. format shift_end_time %tc
.
. gen double arrive_time = clock(arrive, "DMY hms")
. gen double leave_time = clock(leave, "DMY hms")
. format arrive_time %tc
. format leave_time %tc
.
. // calculate percentages
. // patients arriving before their physician´s shift starts
. gen arrive_before_shift = arrive_time < shift_start_time
.
. // patients discharged after their physician´s shift ends
. gen leave_after_shift = leave_time > shift_end_time
.
. // calculate percentages
. qui sum arrive_before_shift
. qui sum leave_after_shift
.
```

2. Data Exploration

I calculated the average predicted severity by half-hour of patient arrival. The connected plot and trend line does not show obvious connection between hours of arrival and the predicted severity of the patient. To test formally test whether patient severity is or is not predicted by hour of the day, I regressed the pred_lnlos on the dummies of hour arrival variables. The coefficient plot of dummies shows patients stay shorter at dawn and after lunch and stay longer in the morning. However, the result may result from the limit of usage of some inspect equipment such as CT.

```
. *********************Question 2*****************************************
. // get hours and minutes
. gen arrive_hour = hh(arrive_time)
. gen arrive_minute = mm(arrive_time)
.
```

```
. // compute half-hour
. gen half_hour_interval = arrive_hour + (arrive_minute >= 30)/2

.
. // calculate average servenity by half hours
. bysort half_hour_interval: egen avg_severity = mean(pred_lnlos)

.
. qui twoway (connected avg_severity half_hour_interval,m(o)) ///
> (lfit avg_severity half_hour_interval, lpattern(dash)), ///
> ytitle("Average Severity") xtitle("Half-Hour Interval of Day") ///
> title("Average Severity by Half-Hour of Patient Arrival") ///
> legend(ring(0) position(4))
. qui graph export "$Out\Average_Severity.png", replace

.
. // formally test whether patient severity is or is not predicted by hour of the day
. qui reg avg_severity i.arrive_hour,r

.
. qui coefplot,keep(*.arrive_hour) title("Coefficient of Arrival Hours") baselevels ci vertical ///
> label xlabel(, angle(45) labsize(vsmall)) yline(0)
. qui graph export "$Out\Coef_Hours.png", replace
```

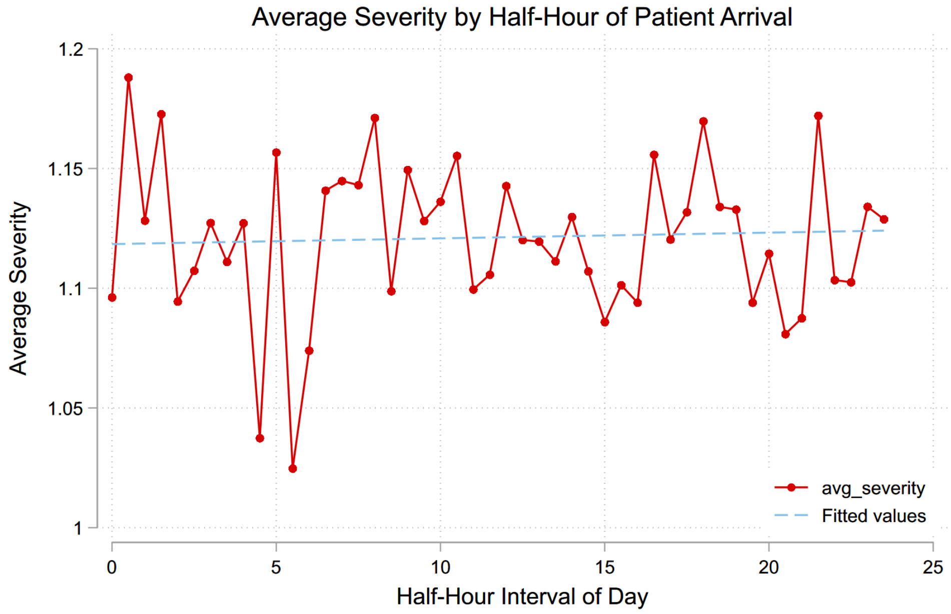

Figure 1: Average Severity by Half-Hour of Patient Arrival

3.Estimation and Causal Inference

(a) I graphed the census variation relative to end of shift as Fig. 3 shows. The census count – in any census scope, would increase at the beginning of the physician's shift and decreases as the time get closer to his end of shift time. And the patient under care still decreases even the time passed the shift time for 4 hours.

(b) As we have the accurate time of the shift time and the patient arrival time. I construct the lower bound of the census under the criterion that we only take the patients who is under

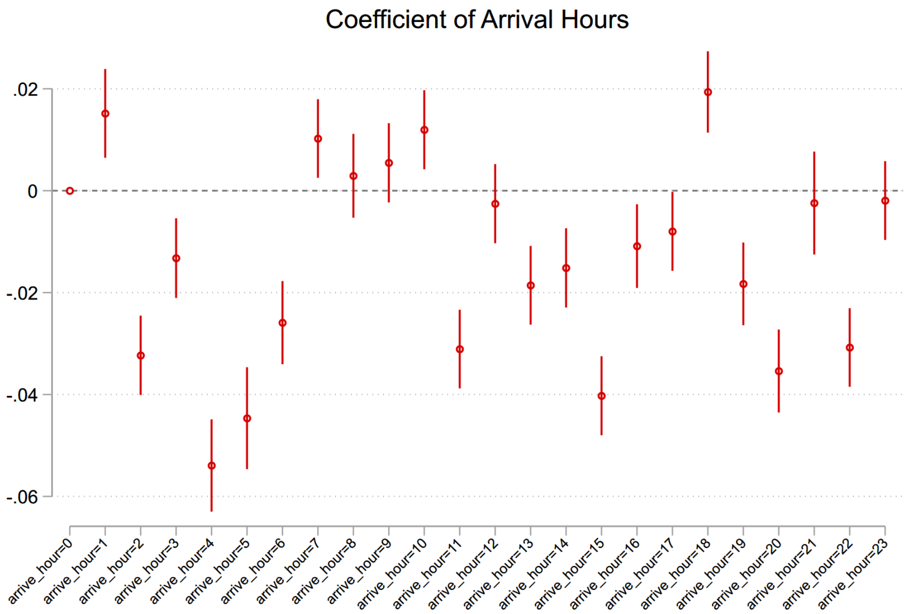

Figure 2: Coefficient of Arrival Hours

care for the whole hours into account. And I construct the upper bound of the census under the criterion that we counts the patients whoever is under care when he arrives the ED at that hour. At last, the finer census – I excluded the patients whose arrival time is among the last 15 minutes of the hour or leave time is among the first 15 minutes of the hour. One issue one may addressed is that the physician may arrive at the EP before his scheduled time, which would affects the power of our above analysis. In the mean time, the time between the arrival time and leave time of patients may not be accounted in the care time, they may wait at the waiting room or doing some paper works before the physician's care.

(c) If we have the ED data, we may construct the census of the co-work of more than one physicians, and under this circumstance, they may behavior differently.

```
. ********************Question 3****************************************
. // extend shift end by 4 hours
. gen double shift_end_4h_more = shift_end_time + 60*60*4000

. format shift_end_4h_more %tc

.
. save temp.dta, replace
file temp.dta saved

.
. use temp.dta, clear

. // creat hourly interval for time shift
. gen hours_of_shift = (shift_end_4h_more - shift_start_time) / 3600000

.
. // create an index for each shift
. gen shift_index = _n

.
. // expand the dataset for each hour of each shift
. qui expand hours_of_shift
.
. // generate hour_id for each hour within the shift
. bysort shift_index: gen hour_id = _n
.
. gen double hour_lb = shift_start_time + (hour_id-1)*3600000

. gen double hour_ub = hour_lb + 3600000
.
. format hour_lb %tc

. format hour_ub %tc
.
. // lower bound census: counts only throught whole hours
. gen patient_under_care_lb = (arrive_time <= hour_lb) & (leave_time > hour_ub)
.
. bysort phys_id shift_date hour_id: egen census_lb = sum(patient_under_care_lb)
.
. // upper bound census: counts if intersects within hours
. gen patient_under_care_ub = (arrive_time <= hour_ub) & (leave_time > hour_lb)
.
. bysort phys_id shift_date hour_id: egen census_ub = sum(patient_under_care_ub)
.
. // finer bound census
. gen patient_under_care_fb = (arrive_time <= hour_ub) & (leave_time > hour_lb)

. qui replace patient_under_care_fb = 0 ///
> if (arrive_time <= hour_ub) & (leave_time > hour_lb) & (leave_time < hour_lb + 900000)

. qui replace patient_under_care_fb = 0 ///
> if (arrive_time <= hour_ub) & (arrive_time > hour_ub - 900000) & (leave_time > hour_lb)
.
. bysort phys_id shift_date hour_id: egen census_fb = sum(patient_under_care_fb)
.
. // end of shift
. gen end_of_shift = hour_id - hours_of_shift + 4

. save patients_all.dta, replace
file patients_all.dta saved
.
. use patients_all.dta, clear

. qui duplicates drop phys_id shift_date shift_start shift_end end_of_shift, force

. keep phys_id shift_date shift_start shift_end hour_id ///
> patient_under_care_lb patient_under_care_ub patient_under_care_fb end_of_shift

. rename hour_id hour

. save census.dta, replace
file census.dta saved
.
. use census.dta, clear

. // How does the census vary with time relative to end of shift
. bysort end_of_shift: egen sum_patient_under_care_fb = sum(patient_under_care_fb)

. bysort end_of_shift: egen sum_patient_under_care_lb = sum(patient_under_care_lb)

. bysort end_of_shift: egen sum_patient_under_care_ub = sum(patient_under_care_ub)
.
. qui duplicates drop end_of_shift, force

. qui twoway (connected sum_patient_under_care_fb end_of_shift) ///
> (connected sum_patient_under_care_lb end_of_shift) ///
> (connected sum_patient_under_care_ub end_of_shift), xline(0, lpattern(dash)) ///
> ytitle("Census Count") xtitle("Time Relative to End of Shift (Hours)") ///
> title("Census Variation Relative to End of Shift") xtick(-9(1)4) ///
> legend(ring(0) pos(11))
```

```
. qui graph export "$Out\Census_Variation.png", replace
.
```

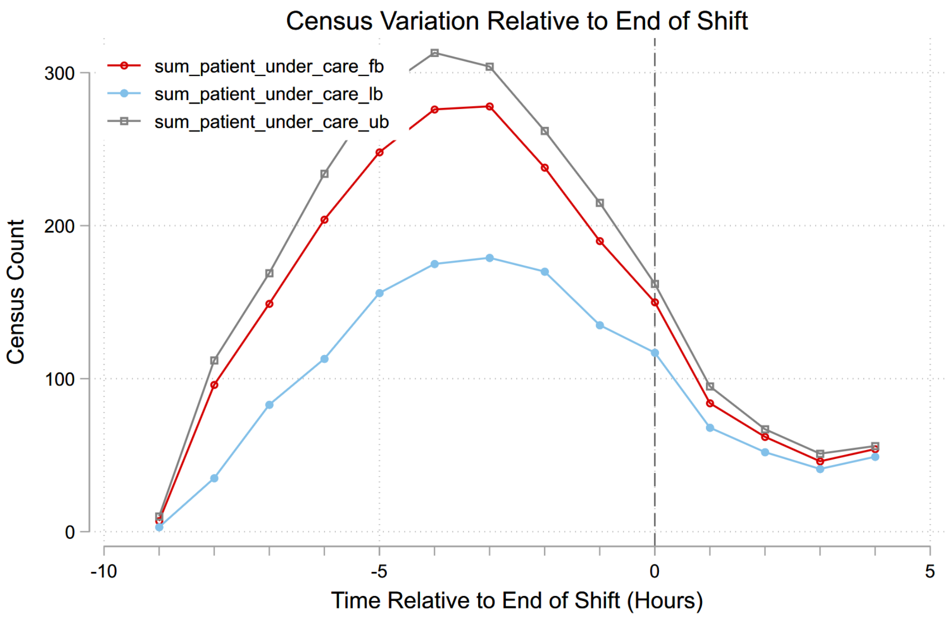

Figure 3: Census Variation Relative to End of Shift

4. Further Analysis

I regressed the length of the stay on the dummies of the physicians, the result shows at the Figure 4, phys_id=16 is the one who is fastest at discharging patients. The potential threats may be that different physicians in different ED may encounter different types of patients, thus leading to different discharge time. So we can control the expected log length of stay, where length of stay is the difference between leave and arrive, based on patient demographics and medical conditions. The result is robust, physician 16 and 30 are two who are fastest at discharging patients.

Moreover, a potential issue may arising from the fact that the patient may be discharged after the physician's shift time. So we just regress the log length of stay on the dummies of time to shift. The result is still robust to this specification: physician 16 and 30 are two who are fastest at discharging patients.

```
. *********************Question 4*****************************************
. use temp.dta, clear

.
. // gen length of stay(seconds)
. gen double log_length_stay = log((leave_time - arrive_time)/1000)
(4 missing values generated)

.
. qui reg log_length_stay i.phys_id, r

. qui coefplot,keep(*.phys_id) title("Coefficient of Physician") baselevels ci vertical ///
> label xlabel(, angle(45) labsize(vsmall)) yline(0)

. qui graph export "$Out\Coef_phys.png", replace
```

```
.
. // control pred_lnlos
. qui reg log_length_stay i.phys_id pred_lnlos, r
. qui coefplot,keep(*.phys_id) title("Coefficient of Physician, Controlled for pred_los") baselevels ci vertical /
> label xlabel(, angle(45) labsize(vsmall)) yline(0)
. qui graph export "$Out\Coef_phys_control.png", replace
.
.
. // los versus time to shift
. use patients_all.dta, clear
. gen double log_length_stay = log((leave_time - arrive_time)/1000)
(52 missing values generated)
.
. qui reg log_length_stay i.phys_id##i.hour_id, r
. qui coefplot,keep(*.phys_id) title("Coefficient of Physician, Controlled for time to shift") baselevels ci verti
> label xlabel(, angle(45) labsize(vsmall)) yline(0)
.
. qui graph export "$Out\Interaction_coef.png", replace
.
```

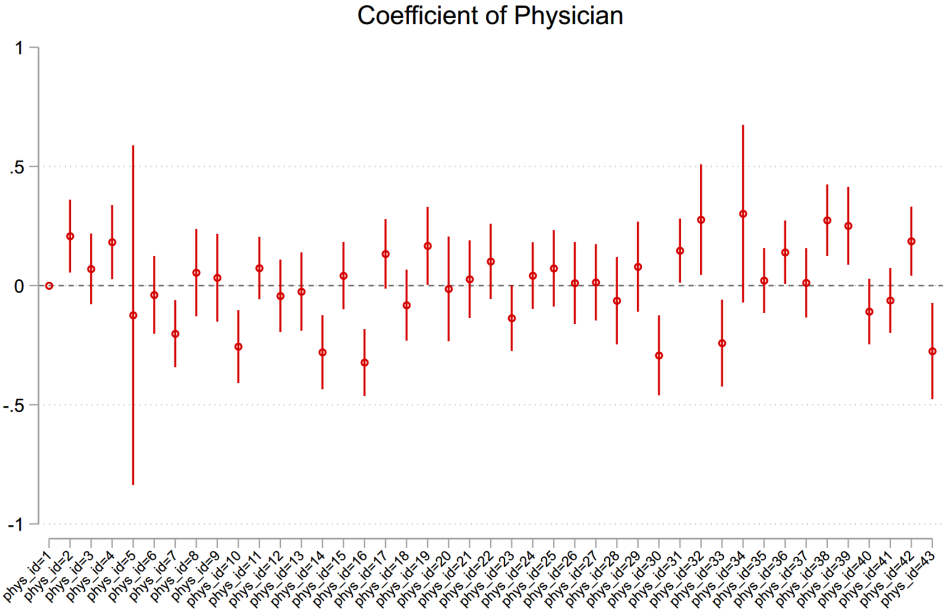

Figure 4: Coefficient of Physician

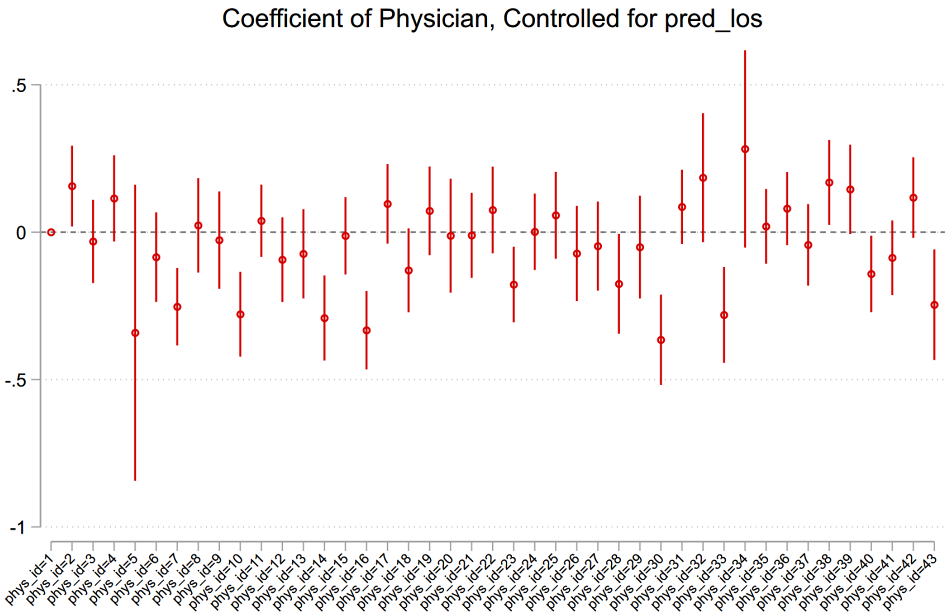

Figure 5: Coefficient of Physician, Controlled for pred_los