Coding Sample in STATA

Variable Construction and Empirical Analysis

Lingyun (Leone) QU

November 9, 2022

I use this coding sample to illustrate my STATA coding techniques and characteristic variable construction skills. Based on Japan ' s prefecture-level panel data on digitalization development from 2011 to 2019, I empirically analyze the impacts of digitalization development and digitalization spillover effects on Japan ' s prefectural economic activities. I utilizes the Principal Component Analysis (PCA) method for constructing the digitalization development index and Directed Acyclic Graphs (DAGs, **not shown in this STATA coding sample**) for measuring the direction of digitalization spillover effects , empirically investigate the impacts of digitalization development and digitalization spillover on economic performance. After benchmark OLS regression with fixed effects, I used total production instead of GDP per capita to conduct the robustness test, tested the endogeneity with availability rate of high-speed Internet in primary schools and 2SLS method, and developed two further mechanism tests with population aging and industrial structure.
The result of the analysis could be found in my writing sample, *Does Digitalization Spillover Negatively Influence the Economy? Empirical Evidence from Japan*.

1 Variable Construction

```
1 * a sample analysis job
2 use digitalization.dta
```

```
* set fixed effect
xtset id year

* digitalization indicator construction.
** variable list: hardward & software & availability rate
** PCA
factortest software_enterprise_1000 info_enterprise_1000 service_enterprise_1000 software_enterprise_per_1000 info_employee_per_1000 ser_employee_per_1000 soft_sell_pctg info_sell_pctg ser_sell_pctg soft_add_pctg info_add_pctg ser_add_pctg internet smartphone computer
factor software_enterprise_1000 info_enterprise_1000 service_enterprise_1000 software_enterprise_per_1000 info_employee_per_1000 ser_employee_per_1000 soft_sell_pctg info_sell_pctg ser_sell_pctg soft_add_pctg info_add_pctg ser_add_pctg internet smartphone computer, pcf
rotate
predict f1 f2
* f1=0.6601/0.8141=0.81083405, f2=0.154/0.8141=0.18916595
gen digital = 0.81083405*f1+0.18916595*f2

*clear empty rows
drop if district == .

* spillover_11: Using three continuous years for calculating the direction of the directed arrow in the given year and the PC (Peter and Clark) algorithm, create the digitalization spillover index, one per country per year ranging from 2013 to 2019. For each base year, the binary variable digitalization spillover index ^^e3^^80^^96Spillover^^e3^^80^^97_(i,j) is given value 1 if the prefecture j experiences digitalization spillover in year i, and other prefectures, both spill-in and without effects, are given value -1.
*  only keep the directed arrows statistically significant at 1%.
```

2 Empirical Analysis

```
*--------
*baseline analysis
*--------
```

```
**GDP per capita ~ spillover: impact of digitalization ↩
    spillover to GDP per capita
xtreg log_productionthousand_per_c spillover_11 lifeinv ↩
    agrinv secuinv consumption consumptionthousand_per_c ↩
    populationthousand,fe
** (-), statistically significant
est store baseline1

**Overall GDP ~ spillover: impact of digitalization ↩
    spillover to Overall GDP
xtreg log_productionmillionen spillover_11 lifeinv agrinv ↩
    secuinv consumption consumptionthousand_per_c ↩
    populationthousand,fe
** (-), statistically significant
est store baseline2
esttab baseline1 baseline2 using baseline.rtf, b(%12.3f) se↩
    (%12.3f) nogap compress s(year N r2) star(* 0.1 ** 0.05↩
     *** 0.01)

*--------
*Robust test: change explained variable
*--------
*GDP per capita ~ digital
xtreg log_productionthousand_per_c digital lifeinv agrinv ↩
    secuinv consumption consumptionthousand_per_c ↩
    populationthousand,fe
** (+), statistically significant
est store robust1

*Overall GDP ~ digital
xtreg log_productionmillionen digital lifeinv agrinv ↩
    secuinv consumption consumptionthousand_per_c ↩
    populationthousand,fe
** (+), statistically significant
est store robust2
esttab robust1 robust2 using robust.rtf, b(%12.3f) se(%12.3↩
    f) nogap compress s(year N r2) star(* 0.1 ** 0.05 *** ↩
    0.01)

*--------
* alternative mechanism
*--------
*GDP per capita ~ population aging:
*diff_region_80_over: the difference between the over-80-↩
    year-old population proportions of the prefecture and ↩
    the regional average.
gen inter_80= spillover_11*Diff80
xtreg log_productionthousand_per_c spillover_11 Diff80 ↩
    inter_80 lifeinv agrinv secuinv consumption ↩
    consumptionthousand_per_c populationthousand,fe
```

```
*spillover (-), aging (-), interaction(-). statistically significant.
est store mechanism1

*GDP per capita ~ industrial structure:
*service: the ratio of entertainment and life-related service industry production to total GDP.
gen inter_life= spillover_11*service
xtreg log_productionthousand_per_c spillover_11 service inter_life lifeinv agrinv secuinv consumption consumptionthousand_per_c populationthousand,fe
*spillover (-), aging (-), interaction(+). statistically significant.
est store mechanism2
esttab mechanism1 mechanism2 using alternativem.rtf, b(%12.3f) se(%12.3f) nogap compress s(year N r2) star(* 0.1 ** 0.05 *** 0.01)

*--------
* Heterogeneity Analysis 1: geographic location
*--------
** gen variable divided by geographic location
gen sea=0
replace sea = 1 if name_kanji北海道==""|name_kanji青森県==""|name_kanji秋田県==""|name_kanji山形県==""|name_kanji新潟県==""|name_kanji富山県==""|name_kanji石川県==""|name_kanji福井県==""|name_kanji京都府==""|name_kanji兵庫県==""|name_kanji鳥取県==""|name_kanji島根県==""|name_kanji山口県==""|name_kanji福岡県==""|name_kanji佐賀県==""|name_kanji長崎県==""

**GDP per capita ~ not facing sea
xtreg log_productionthousand_per_c spillover_11 lifeinv agrinv secuinv consumption consumptionthousand_per_c populationthousand if sea==0,fe
** not significant
est store sea0

**GDP per capita ~ facing sea
xtreg log_productionthousand_per_c spillover_11 lifeinv agrinv secuinv consumption consumptionthousand_per_c populationthousand if sea==1,fe
** (-) significant
est store sea1
esttab sea0 sea1 using hetero_sea.rtf, b(%12.3f) se(%12.3f) nogap compress s(year N r2) star(* 0.1 ** 0.05 *** 0.01)

* Heterogeneity Analysis 2: gdp
bys year: egen gdpm=mean(log_productionmillionen)
gen gdp_01 = (log_productionmillionen> gdpm) if gdpm!=.
```

```
68
69  **GDP per capita ~ spillover | high gdp
70  xtreg log_productionthousand_per_c spillover_11 lifeinv ↩
        agrinv secuinv consumption consumptionthousand_per_c ↩
        populationthousand if gdp_01 ==1,fe
71  * (-) significant
72  est store gdp0
73
74  **GDP per capita ~ spillover | low gdp
75  xtreg log_productionthousand_per_c spillover_11 lifeinv ↩
        agrinv secuinv consumption consumptionthousand_per_c ↩
        populationthousand if gdp_01 ==0,fe
76  * not significant
77  est store gdp1
78  esttab gdp0 gdp1 using gdp.rtf, b(%12.3f) se(%12.3f) nogap ↩
        compress s(year N r2) star(* 0.1 ** 0.05 *** 0.01)
79
80  *Endogeneity
81  *internet_30mbps:  the availability rate of high-speed ↩
        Internet in primary schools to test the endogeneity. ↩
        Since the time range of this paper is 2011-2019, the ↩
        contribution of primary students to the GDP is ↩
        ignorable.
82  ivregress 2sls log_productionthousand_per_c lifeinv agrinv ↩
        secuinv consumption consumptionthousand_per_c ↩
        populationthousand (spillover_11 =  internet_30mbps),r ↩
        first
83  est store endo
84  esttab endo using endo.rtf, b(%12.3f) se(%12.3f) nogap ↩
        compress s(year N r2) star(* 0.1 ** 0.05 *** 0.01)
```

Coding Sample in Matlab

AR Model Selection and Impulse Response Analysis

Lingyun (Leone) QU

November 9, 2022

I use this coding sample to illustrate my Matlab coding techniques, knowledge of time series analysis and interpretation skills.

1 Basic Plotting

```
[CPI,Inflation_date,~]=xlsread('CPI_2015.xlsx',
'data','A6:B229'); %1965 I - 2020 IV
[GDP,GDP_date,~]=xlsread('GDP_growth_rate_yoy_sa.xlsx',
'data','A6:B248'); %1960 II - 2020 IV
[ITR,Interest_date,~]=xlsread('CD91days_GovBond3yr.xlsx',
'data','A21:C140'); %1901 II - 2020 IV
IFR = (log(CPI(5:end,1))-log(CPI(1:end-4,1)))*100
% ignoring first 4 observations of 1st year

%plot
figure(1)
plot((1:243)',GDP,'-k')

% interest rate along GDP data, start from 1991.1
plot((1:243)',GDP,'-k',(124:243)',ITR(:,1),'-r')
%(:,1): all rows from 1 to last, 1st column
% start from 1991 and 1960

% add inflation rate
plot((1:243)',GDP,'-k',(124:243)',ITR(:,1),'-r',(24:243)',IFR,'-')
xticks(3:20:243)
```

```
% start from 3 to 243 and plot every 20
xticklabels({'60.IV','65.IV','75.IV','80.IV','85.IV','90.IV','95.IV',
'00.IV','05.IV','10.IV','15.IV','20.IV',})
axis([1,243,-10,30])
% specify range for x and y: x from 1 to 243, y from -10 to 30
legend('GDP growth rate','CD 91 days','inflation')
% legend on the top
xlabel('Yr.Quarter')
ylabel('%')
title('KR macro variables')
```

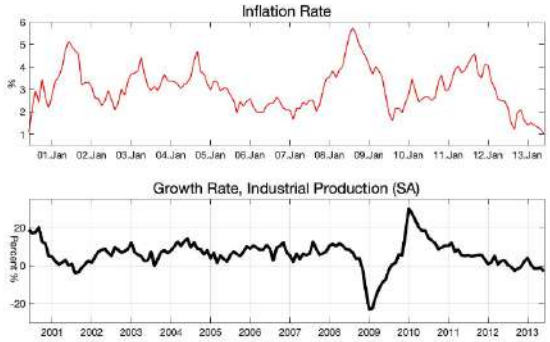

Figure 1: Directed Arrow Graphs (DAGs) of year 2013-2019.

2 Parameter Estimation via OLS

To estimate the parameters using OLS, assume that $yt = a + b_t + ut$.

```
%% olsvar.m
%% factorial regression
function[B,SIG,U,ZZ]=olsvar(y.p)

[T,K]=size(y);
Y=y(p+1:T,:);
Z=ones(T-p,1)

for j=1:p
Z=[Z,y(p+1-j:T-j,:)];
end

ZZ=Z'*Z;
B=(ZZ)\(Z'*Y);
U=Y-Z*B;
SIG=(U'*U)/(T-p-1-k*p)
```

```
end
```

OLS estimates of GDP growth rate: a=0.7313, b=0.1876

OLS estimates of CD 91 days: a=0.0713, b=0.9558

OLS estimates of inflation rate: a=0.2185, b=0.8980

2.1 Residual Plots

To visualize the estimation mistake via OLS, first let $\hat{u}_t = y_t - \hat{a}_t - \hat{b}_t$, where $\hat{a}_t$ and $\hat{b}_t$ are the OLS estimates.

```
function [beta,Sig2,Cov]=AROLS(y,p)
T=size(y,1);
Y=y(p+1:T,1); %y(p+1).....y(T)
X=ones(T-p,1);
for j=1:p
    X=[X,y(p+1-j:T-j,1)];
        % #1: mu parameter, #2: AR parameter a0 - at
end
beta=inv(X'*X)*(X'*Y)
U=Y-X*beta;
Sig2=(U'*U)/(T-p-1);
Cov=Sig2*inv(X'*X)
end
```

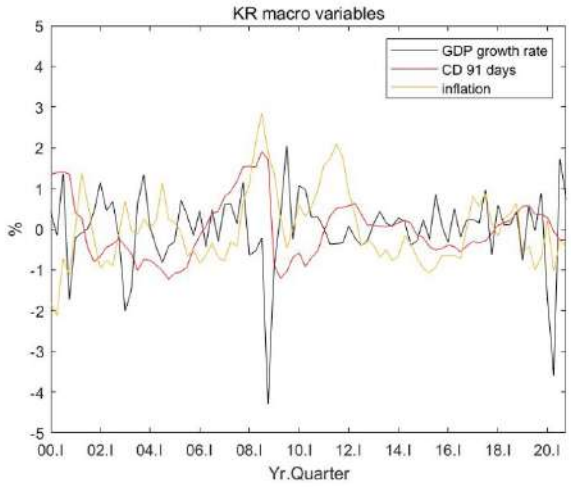

Figure 2: growth rate and inflation rate of Korea, 1965-2020.

According to the plots, the interest rate is the most persistent (with least fluctuations).

3 AR Process Parameter Estimation

To estimate the parameter and determine the existence of causal process, first assume that u_t is generated by $\hat{u}_t = a_1\hat{u}_{t-1} + a_2\hat{u}_{t-2} + e_t$ with e_t from IID(0, σ^2).

```
y=GDP;
T=size(y,1);
p=2;
% specify AR order, AR 2 here

w=[ones(T,1),(1:T)'];
% linear trend: 1 to T. transpose for a column vector

% run regression of the dependent variable on w
% remove repeated linear trend
y=y-w*((w'*w)\(w'*y));

[beta,Sig2,Cov]=AROLS(y,p)

a1=beta(2,1);
a2=beta(3,1);
```

```
17
18 root1=(a1+sqrt(a1^2+4*a2)/(-2*a2));
19 root2=(a1-sqrt(a1^2+4*a2)/(-2*a2));
20 disp([abs(root1),abs(root2)])
21
22 disp(beta)
```

GDP growth rate: a_1=0.1039, a_2=−0.1255, absolute values of both roots are 2.7938 and 2.7938. Since the absolute value of both roots are larger than 1, the estimated values of a_1 and a_2 imply a causal process.

CD 91 days: $a_1 = 1.1469, a_2 = -0.3329$, absolute values of roots are 1.1629 and 1.1629. Since the absolute value of both roots are larger than 1, the estimated values imply a causal process.

Inflation rate: $a_1 = 0.8697, a_2 = -0.1542$, absolute values of both roots are 2.0808 and 2.0808. Since the absolute value of both roots are larger than 1, the estimated values imply a causal process.

4 MA Process Parameter Estimation

To estimate the parameter and determine the existence of invertible process, first assume that u_t is generated by $\hat{u}_t = \hat{e}_t + b_1\hat{e}_{t-1} + b_2\hat{e}_{t-2} + e_t$ with e_t from IID(0, σ^2).

```
1 % AR 1 model
2 y=GDP;
3 T=size(y,1);
4 p=1;
5 % specify AR order, AR 1 here
6 [beta,Sig2,Cov]=AROLS(y,p)
7 disp(beta)
```

GDP growth rate: $b_1 = 0.0556, b_2 = -0.1712$, absolute values of both roots are 2.4119 and 2.4119. Since absolute values of both roots are larger than 1, the estimated values imply an invertible process.

CD 91 days: $a_1 = 0.9650, a_2 = 0.4848$, absolute values of roots are 0.7824 and 2.7125. Since the absolute value of ONLY one root is larger than 1, the estimated values do NOT imply an invertible process.

Inflation rate: $a_1 = 1.0901, a_2 = 0.1105$, absolute values of roots are 4.6872 and 6.8674. Since absolute values of both roots are larger than 1, the estimated values imply an invertible process.

$$\mathrm{E}(\mathrm{y}_t) = E(a) + E(bt) + E(u_t) = a + bE(t) + E(u_t) = \\ a + bE(t) + E(e_t) + b1E(e_{t-1}) + b2E(e_{t-2}) = a + bE(t)$$

Therefore, the estimation of three:

GDP growth rate: $E(y_t) = 0.7313 + 0.1876E(t)$

CD 91 days: $E(y_t) = 0.0713 + 0.9558E(t)$

Inflation rate: $E(y_t) = 0.2185 + 0.8980E(t)$

5 model discussion

To choose the most appropriate model, I compare p-values and sums of squared residuals.

```
1 [pvalue,fval,exitflag]=fminsearch (@(para) ...
      -1*lglkma1(para,Y),[-0.5,1]')
2 disp(pvalue)
```

GDP growth rate: AR(2) model with a sum of squared residual of 70.4359, MA(2S) model with 99.9212. AR(2) model is more desirable.

CD 91 days: AR(2) model with a sum of squared residual of 9.2034, MA(2) model with a sum of squared residual of 481.4962. Therefore, AR(2) model is more desirable.

Inflation rate: AR(2) model with a sum of squared residual of 23.7844, MA(2) model with 246.0836. AR(2) model is more desirable.

Therefore, the AR(2) model with parameters estimated by the OLS is the most appropriate.

6 point estimates of autoregressive coefficients based on VAR(2) model

```
% irfvar1.m
% impulse response function
function[IRF]=irfvar1(A,p,H,BOinv)
K=size(BOinv.1);
Ab=[A:[eye(K*(p-1)),zeros(K*(p-1),K)]];
J=[eye(K,K)zeros(K,K*(p-1))];
IRF=reshape(BOinv',K^2,1);
    for i=1:H
        IRF=([IRF.reshape((J*Ab^i*J'*B0inv)',K^2,1)]);
    end
end
```

```
[T,K]=size(y);

p=2;
H=16;
nrep=20000:

%%
[B,Sig,U,ZZ]=olsvar(y,p);
beta_bar=reshape(B',K*(1+K*p),1);
SIGb = kron(inv(ZZ),Sig);
P=chol(SIGb)';
BOinv=chol(Sig)';

%% IRF point estimates
IRFp=irfvar1(B(2:end,:)',p,H,BOinv);

%% simulation
IRFrmat=zeros(nrep,K^2*(H+1));

% IRF for each bootstrap replication
forj=1:nrep
    beta_sim=beta_bar+P*randn(size(beta_bar));
    Atemp=reshape(beta_sim,K,1+K*p);
    IRFr=irfvar1(Atemp(:,2:end),p,H,BOinv);
    IRFrmat(j,:) =vec(IRFr')';
end

%% Calculate 95 perccent confidence intervals
CI=prctile(IRFrmat,[2.5 97.5]);
CILv=reshape(CI(1,:)'.H+1,K^2); %lower bound
CIHv=reshape(CI(2,:)',H+1,K^2)'; %upper bound

%% plotting the results
name=[];
    for idx=1:K
        for idx2=1:K
            namel=['e_' num2str(idx2)'\rightarrow e_'num2str(idx)];
            name=[name:namel];
```

```
        end
    end
figure(3)

for i=1:K^2
    subplot(K,K.i)
    plot((0:H),IRFp(i.:),'-k',(0:H),CIHv(i,:),'-.b',(0:H),
    CILv(i,:),'-.b'.(0:H).zeros(H+1,1),'-r','linewidth',2)
    xlim([0 H])
    title(name(i,:))
end
legend('IRF','simulated CI');
```

7 analysis of impulse response functions plot

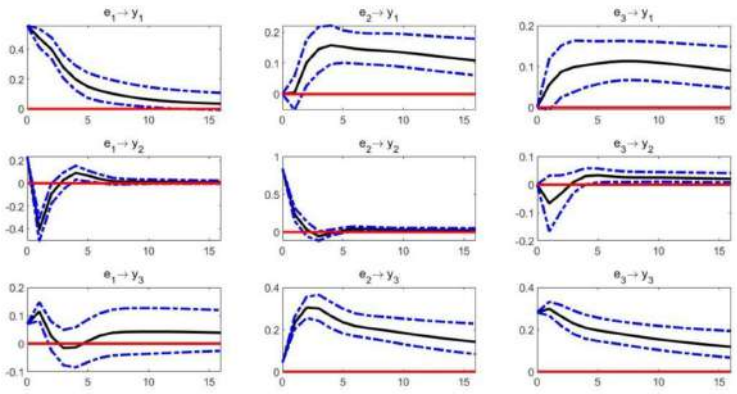

Figure 3: impulse response functions plot with 68% conOdence interval (in Dashed lines)

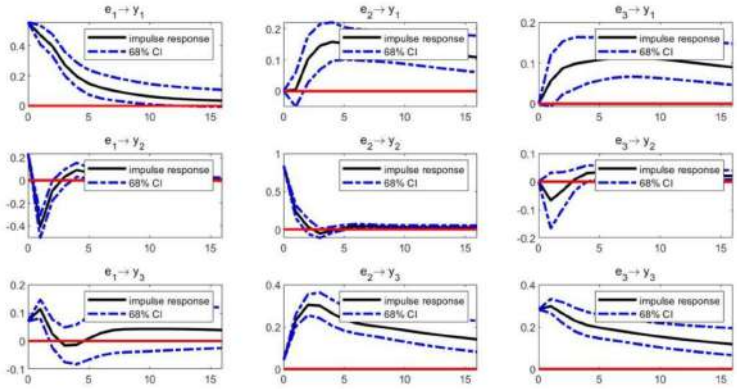

Figure 4: impulse response functions plot with 68% conOdence interval (in Dashed lines) with tags

To identify structural parameters recursively, I assume that the causal chain holds in the order of inflation rates, growth rates and interest rates.

Plots of the 1st column: structural shock of inflation rate (inflation shock)

(1,1) plot, e1 −> y1, shows the inflation response to inflation shock. When the economy experiences an inflation shock of size around 0.55, then in the

first period, the inflation rate goes up by the same amount, but the impact of the inflation shock drops quickly. The dashed lines denote the 68% confidence interval, and when time lag goes beyond 10, then the confidence interval includes the 0 line, indicating that it is impossible to reject the null hypothesis (no impact at all). In other words, beyond 10 quarters, whether the shock is indeed affecting the inflation rate is unsure.

(2,1) plot, e1 −> y2, the response of the growth rate to the inflation shock. The growth rate grows up together with the inflation but drops back, and it fluctuates around 0. The confidence interval includes 0 very quickly and it means that the inflation shock does not really make the economy better or worse.

(3,1) plot, e1 −> y3, the response of the interest rate to the inflation shock. when there is inflation −> interest rate goes up a little bit. When CPI increases by 0.05, the interest rate increase less than 0.2 of the magnitude but the direction is the same. In other words, inflation rate goes up, CB rsaies the interest rate to fight the inflation rate.

2. Plots of the 2nd column: structural shock of growth rate (growth rate shock)

(1,2) plot, e2 −> y1, the inflation response to the growth rate shock. The growth rate shock does not affect the inflation rate right away but the difference takes time. The large 68% confidence interval indicates that no response at all up to 2 quarters; but after that, the inflation rate goes quickly and the magnitude of the change is considerable. The interpretation is that, when the economy is in a boom, the economic growth rate influences the inflation rate much later with a time lag.

(2,2) plot, e2 −> y2, the growth rate response to the growth rate shock. When the economy experiences a growth rate shock of size around 0.8, then in the first period, the inflation rate goes up by the same amount, but the impact of the shock drops dramatically in about 3 periods, and when time lag goes beyond 4, then the confidence interval includes the 0 line indicating that it is impossible to reject the null hypothesis (no impact at all). In other words, beyond 4 quarters, whether the shock is indeed affecting the growth rate is unsure.

(3,2) plot, e2 −> y3, the interest rate response to the growth rate shock. The magnitude of the impulse response is small compared with inflation where the interest rate response is about 0.2, and the growth rate shock is about 0.8, but the interest rate shock response is less than 0.1. Therefore, we may conclude that the CB responds more actively to interest rate than the growth rate. The same direction of change indicates similarities of CB's policies: when

the inflation rate goes up, CB goes against the change by increasing the interest rate; when there is a boom, CB raises the interest rate to stabilize the economy because the export increase and the labor market is on a situation that demand is larger than supply.

3. Plots of the 3rd column: the structural shock of interest rate (interest rate shock)

(1,3) plot, e3 −> y1, the inflation rate response to the interest rate shock. The magnitude of the impulse response is small. Besides, the interest rate shock does not affect the inflation rate right away but the difference takes time. As the 68% confidence interval includes 0 at the 1st period but the impulse response line is significantly larger than 0 afterward, we may conclude that the influence of interest rate shock on the inflation rate is significantly on a 68% confidence level in 15 periods.

(2,3) plot, e3 −> y2, the growth rate response to the interest rate shock. The influence is unclear because the confidence interval includes 0 continuously.

(3,3) plot, e3 −> y2, the interest rate response to the interest rate shock. The magnitude of the impulse response is about 0.3 and the impact is significant on a 68% confidence level throughout the 16 quarters despite the magnitude gradually dying down. The interest rate shock in the 1st period continuously influences the interest rate in the 20 years but the influence is less and less important.

Coding Sample in Python

Deferred Acceptance Algorithm Replication with calculation of Average Matching Rank (AMR) and Fairness (FE) calculation

Lingyun (Leone) QU

November 9, 2022

I use this coding sample to illustrate my Python coding techniques and knowledge of matching theory. I replicated the DA algorithm in Python, simulated the preference by assigning random numbers, and presented calculation of two properties in matching algorithm evaluation.

1 Basic Setting

```
1  # import packages
2  import collections
3  import pandas as pd
4  import numpy as np
5  import random
6  import copy
7
8  # define DA algorithm
9  def find_free_partner(boys, girls, sort_boy_to_girl, ↩
       sort_girl_to_boy):
10
11     # current choice at hand
12     current_boys = dict(zip(boys, [None]*len(boys)))
13     current_girls = dict(zip(girls, [None]*len(girls)))
```

```
14      sumboys= sum(current_girls)
15      # current_boys = {boys[0]:None, boys[1]:None, boys[2]:↩
            None, boys[3]:None}
16      # current_girls = {girls[0]:None, girls[1]:None, girls↩
            [2]:None, boys[3]:None}
17      count = len(boys)
18
19
20      # array: boy's next choice (if rejected)
21      next_select = dict(zip(boys, [None]*len(boys)))
22      for i in range(count):
23          temp = [girls[m-1] for m in sort_boy_to_girl[i]]
24          next_select[boys[i]] = deque(temp)
25
26
27      # dict: girl chooses boy
28      sort_girl = dict(zip(girls, [None]*len(boys)))
29      for i in range(count):
30          # rank boys
31          temp = [[boys[m-1], 4-ind] for ind, m in enumerate(↩
                sort_girl_to_boy[i])]
32
33          name = []
34          match = []
35          for t in temp:
36              name.append(t[0])
37              match.append(t[1])
38
39          sort_girl[girls[i]] = dict(zip(name, match))
40
41
42      while None in current_boys.values():
43          for i in range(count):
44              bid = boys[i]
45              if current_boys[bid]:
46                  # skip if boy is on the string
47                  continue
48              else:
49                  # preferred girl
50                  select = next_select[bid][0]
51                  if current_girls[select] == None:
52                      # match if both are single
53                      current_boys[bid] = select
54                      current_girls[select] = bid
55                      next_select[bid].popleft()
56                  else:
57                      # comparision
58                      # if current > new then stay
59                      if sort_girl[select][current_girls[↩
                            select]] > sort_girl[select][bid]:
```

```
                        next_select[bid].popleft()
                    # else: opposite
                    # if new > current then change
                    # girl chooses new boy, current boy loses
                    # can't propose twice to a same girl
                    else:
                        current_boys[current_girls[select]] = None
                        current_boys[bid] = select
                        current_girls[select] = bid
                        next_select[bid].popleft()
    return current_boys
```

2 Numerical Simulation

```
## initialization
## number of boys = number of girls = size
# take sample 20
size=20
boys = range(1,size)
girls = range(1,size)

# random preference matrix
sort_boy_to_girl= [random.sample(range(0,size),size) for _ in range(0, size)]
sort_girl_to_boy= [random.sample(range(0,size),size) for _ in range(0, size)]

# convert to dataframe for calculation
df = pd.DataFrame(pd.Series(find_free_partner(boys, girls, sort_boy_to_girl, sort_girl_to_boy)), columns=['girls'])
df = df.reset_index().rename(columns = {'index':'boys'})
```

3 Calculation

Average Matched Rank(AMR)

A measure of the average "satisfaction" of participants with the resulting matching, calculated as the overall rank of matched partners averaged over all participants.

$$AMR = \frac{\sum_{i,j \in <X,Y>} P_{i,j} + P_{j,i}}{n_X + n_Y} \tag{1}$$

Fairness

A measure of difference of average matched rank between the two sides of the matching market, calculated as the difference between two average rank of matched partners.

$$Fairness = \frac{\sum_{i,j \in <X,Y>} P_{i,j}}{n_X} - \frac{\sum_{i,j \in <X,Y>} P_{i,j} + P_{j,i}}{n_X + n_Y} \tag{2}$$

```
# define rank_boys: rank for boys
# define rank_girls: rank for girls
rank_boys=0
rank_girls=0

# calculate rank_boys
for i in range(0, size):
    print(sort_girl_to_boy[i])
    print(df['girls'].iloc[i])
    rank_boys+= sort_girl_to_boy[i].index(df['girls'].iloc[↩
        i])
    i+=1
    print(rank_boys)
    print(rank_boys/size)
    count=0

# calculate rank_girls
for i in range(0, size):
    print(sort_boy_to_girl[i])
    print(df['boys'].iloc[i])
    rank_girls+= sort_girl_to_boy[i].index(df['girls'].iloc↩
        [i])
    i+=1
    print(rank_girls)
    print(rank_girls/size)

# calculation AMR and FE
AMR=(rank_boys+rank_girls)/(2*size)
FE=rank_boys/size-rank_girls/size
```

Coding Sample in R

Dataset Cleansing and Visualization

Lingyun (Leone) QU

November 12, 2022

I use this coding sample to illustrate my R coding techniques and aesthetic tastes. I cleaned the data, switched arrangement for those pairs with reversed result under A1 criteria, and consciously excluded SNPs with information inconsistent with the background. Since the case aiming at discussing the impact of certain gene on agents' behaviors, I also considered the distribution of p-values produced from a two-sided test of the null hypothesis for each SNP (single nucleotide polymorphisms, regarded as the independent variable in this analysis) in the GWAS (genome-wide association study, consists of running this regression for all SNPs) results. Therefore, I conducted visualizations of the data by Quantile-Quantile (QQ) plot of the p-values generated from the GWAS, and produced analysis. The data has been masked.

1 Import

```
# Description:    GADM data cleansing and plotting the disaster frequency on the prefecture level Japan map
# Data:    floodcount.csv
# Date:    2022-03-05

# Question 1

# load packages
library("tidyverse")
```

```
9
10
11
12 # load data
13 A<-read.table('sumstats_trait_A.txt',header = T, sep = "\t"↩
       , as.is = T)
14 B<-read.table('sumstats_trait_B.txt',header = T, sep = "\t"↩
       , as.is = T)
```

2 Cleaning

```
1  # Part 1 cleansing
2  # prob 1 wrong minor -> MAF >0.5
3  # gen MAF_new = (1- current MAF) if MAF>0.5, drop if MAF>1
4  A2<-A %>% filter(MAF<1) %>% mutate(MAF_new = ifelse(MAF↩
       >0.5, 1-MAF, MAF))
5
6
7
8
9  # prob 2 wrong match
10 # correct: AT TA CG GC
11 # incorrect: AC AG ATT ...
12 # more convenient to only filter out the correct
13 # exchange position: generate A1_new=A2 then merge
14 A3<-A2 %>%
15     mutate(A1_new = ifelse(MAF>0.5, A2, NA)) %>%
16     mutate(A2_new = ifelse(MAF>0.5, A1, NA))
17 # create merge A1_new and A2_new
18 A4<-unite(A3, "A", A1, A2)
19 A4<-unite(A4, "A_new", A1_new, A2_new)
20 A5<-A4 %>%
21     mutate(A_final=ifelse(A_new=="NA_NA",A,A_new))
22
23
24
25 # filter matches beyond correct merge
26 # there are many mismatch so better stick to the correct ↩
       ones
27 A6<-A5%>%
28     filter(A_final=="A_G"|A_final=="G_A"|A_final=="C_T"|A_↩
           final=="T_C"|A_final=="A_C"|A_final=="C_A"|A_final↩
           =="T_G"|A_final=="G_T")
29
30
```

```
31
32 # do the same to B
33 B<-read.table("sumstats_trait_B.txt",
34                header = T, sep = "\t", as.is = T);
35 B2<-B %>%
36   filter(MAF<1) %>%
37   mutate(MAF_new = ifelse(MAF>0.5, 1-MAF, MAF))
38 B3<-B2 %>%
39   mutate(A1_new = ifelse(MAF>0.5, A2, NA)) %>%
40   mutate(A2_new = ifelse(MAF>0.5, A1, NA))
41 B4<-unite(B3, "A", A1, A2)
42 B4<-unite(B4, "A_new", A1_new, A2_new)
43 B5<-B4 %>%
44   mutate(A_final=ifelse(A_new=="NA_NA",A,A_new))
45 B6<-B5%>%
46     filter(A_final=="A_G"|A_final=="G_A"|A_final=="C_T"|A_↩
       final=="T_C"|A_final=="A_C"|A_final=="C_A"|A_final↩
       =="T_G"|A_final=="G_T")
47
48
49
50 # merge two datasets and only keep intersection of the two ↩
   files
51 merge1 <- merge(A6,B6,by="SNP") %>%
52   filter(A_final.x==A_final.y)
53 count(merge1)
54 summary(merge1$z.x)
55
56 the final count of remaining SNPs: 7377
57 the rsid of the SNP from this set with the highest z-score ↩
   for trait A: 5.75801
```

3 Plotting

```
1 # Part2: visialization with R
2
3 # select needed data
4 qqplot <- merge1 %>%
5   select("z.x","z.y") %>%
6   na.omit()
7
8
9
10 # set the canvas
11 p <- ggplot(qqplot, aes(sample = z.x))
```

```
p + geom_qq(aes(sample=z.x,colour="trait A")) +
  geom_qq(aes(sample=z.y,colour="trait B")) +
  geom_abline(intercept = 0, slope = 1, col= "gray") +
  ylab("The empirical CDF of the p-values from the GWAS summary statistics (log scale)") +
  xlab("The CDF of the p-values under the null hypothesis (log scale)") +
  ggtitle("Quantile-Quantile plot of the empirical CDF of p-values against the CDF \n
          expected if the null hypothesis were true for all SNPs") +
  scale_y_continuous(trans='log10') + scale_x_continuous(trans='log10')
```

4 Graphic Result

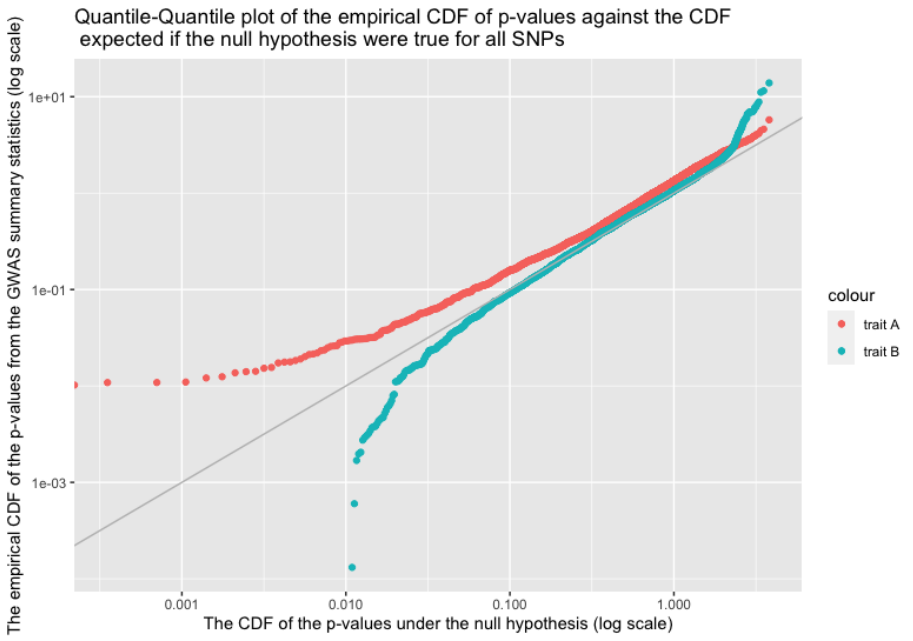

Figure 1: Output

5 Discussion

5.1 Interpretation of the 45 degree line

It is a reference line for normal distribution. It examines the distribution of p-value if they has come from a normal distribution function. Since we are plotting p-values to evaluate many tests simultaneously, if the SNPs were null, then the distribution of their p-values to be distributed uniformly on the unit interval (U (0, 1)), and the qqplot would closely approach the 45 degree line.

Qualitatively, the deviation of a data trend from the 45 degree line in the graph signifies how far the distribution of the p-value against the null-hypothesis is from being normally distributed.

5.2 Interpretation of the Q–Q plot

Because Q–Q plot is a probability plot, and the probability at each p-value must be at least 0, therefore, the qqplot must be monotonically increasing, if not strictly increasing.

a) The trend of trait A QQ plot: The pattern of points seems to indicate that the distribution is skewed or perhaps light-tailed. It first increases slowly from less than 0.001 then constantly increases until 1. It suggests that all SNPs are not likely to be 0, and we may reject null hypothesis for each SNP in the GWAS results at 0.001 confidence level, means that we estimate the SNP to be non-null.

b) The trend of trait B QQ plot: It first increases fastly from 0.01 then slowly increases until 1. It suggests that all SNPs are not likely to be 0, and we may reject null hypothesis for each SNP in the GWAS results at 0.01 confidence level, means that we estimate the SNP to be non-null.