

```
1 . do "/Users/csh/Desktop/判决书/初步结果/论文/定稿/修改1017/2024-00316_程序代码_更新1017.do"

2 . *****关键变量介绍*****
3 . /*交通事故数据集中的关键变量：
> 外地原告：虚拟变量，该变量为0时表示原告、被告均来自本省（即和法院所在省相同），为1表示原告来自外省，被告来自本省，为-1时
> 表示其他情况
> 外地原告_city：虚拟变量，该变量为0时表示原告、被告均来自本市（即和法院所在市相同），为1表示原告来自外市，被告来自本市，
> 为-1时表示其他情况
> 外地被告：虚拟变量，该变量为0时表示原告、被告均来自本省（即和法院所在省相同），为1表示被告来自外省，原告来自本省，为-1时
> 表示其他情况
> 外地被告_city：虚拟变量，该变量为0时表示原告、被告均来自本市（即和法院所在市相同），为1表示被告来自外市，原告来自本市，
> 为-1时表示其他情况
> 外原外被：虚拟变量，该变量为0时表示原告、被告均来自本省（即和法院所在省相同），为1表示原告来自外省，原告来自外省，为-1时
> 表示其他情况
> 外原外被_city：虚拟变量，该变量为0时表示原告、被告均来自本市（即和法院所在市相同），为1表示原告来自外市，被告来自外市，
> 为-1时表示其他情况
> win_rate_yuangao：原告胜率，为0到1的数值变量
> win_rate_beigao：被告胜率，为0到1的数值变量
> win_rate_二审_yuangao
> 电动车原告：原告驾驶非机动车，01虚拟变量，为1表示原告驾驶了非机动车
> 行人原告：原告是行人，01虚拟变量，为1表示原告是行人
> death_and_injury：原告有伤亡，01虚拟变量，为1表示原告有伤亡
> 上诉：是否上诉，01虚拟变量，为1表示上诉
> jianyichengxu：是否采用简易程序，01虚拟变量，为1表示采用
> free_code：是否采用自由裁量，01虚拟变量，为1表示采用
> zhengren：是否传唤证人，01虚拟变量，为1表示传唤
> words_body：文书正文字数
> city_court：法院所在城市
> qu_court：法院所在县区
> reform：i城市t年是否进行司法体制改革，若进行，reform=1，反之为0
> */

4 .
5 . use "/Volumes/Elements2/mark_data/裁判文书全量数据（已完成）/交通事故数据集.dta",clear

6 .
7 . *****表1 交通事故案件的描述性统计*****
8 .
9 . *生成type_city变量，1、2、3、4对应（本市，本市）、（本市、外市）、（外市、本市）、（外市、外市）
10 . gen type_city=0

11 . replace type_city=3 if 外地原告_city==1
    (103,109 real changes made)

12 . replace type_city=2 if 外地被告_city==1
    (213,922 real changes made)

13 . replace type_city=4 if 外原外被_city==1
    (103,531 real changes made)

14 . replace type_city=1 if 外地原告_city==0
    (748,185 real changes made)

15 .
16 . *生成type_pro变量，1、2、3、4对应（本省，本省）、（本省、外省）、（外省、本省）、（外省、外省）
17 . gen type_pro=0

18 . replace type_pro=3 if 外地原告==1
    (83,053 real changes made)

19 . replace type_pro=2 if 外地被告==1
    (144,780 real changes made)

20 . replace type_pro=4 if 外原外被==1
    (60,182 real changes made)

21 . replace type_pro=1 if 外地原告==0
    (973,296 real changes made)

22 .
23 .
24 . *案件变量全样本的描述统计
25 . tabstat win_rate_yuangao 电动车原告 行人原告 death_and_injury 上诉 jianyichengxu free_code zhengren words_body if win_ra
> te_yuangao!=.,stat(N mean sd min max) format(%9.0g)
```

stats	win_ra..	电-原告	行人原告	death_~y	上诉	jianyi~u	free_c~e	zhengren	words_~y
N	1265181	1265181	1265181	1265181	1265181	1265181	1265181	1265181	1265181
mean	.6619277	.0245965	.016615	.3452273	.0841627	.6115457	.0001581	.0049092	2.484797

sd	.3922125	.1548919	.127824	.4754425	.2776317	.487399	.012572	.0698934	1.623514
min	0	0	0	0	0	0	0	0	0
max	1	1	1	1	1	1	1	1	89.128

26 . \*本地-本地、本地-外地、外地-本地、外地-外地的样本数,有近4000的样本不属于4类案件,因此总数要略小于总样本数  
 27 . tabstat type\_pro if type\_pro!=0,by(type\_pro) stat(N)

Summary for variables: type\_pro  
 by categories of: type\_pro

type_pro	N
1	973296
2	144780
3	83053
4	60182
Total	1261311

28 .  
 29 . tabstat type\_city if type\_city!=0,by(type\_city) stat(N)

Summary for variables: type\_city  
 by categories of: type\_city

type_city	N
1	748185
2	213922
3	103109
4	103531
Total	1168747

30 .  
 31 . \*省份层面: 案件关键变量的描述统计, 分案件类型  
 32 . tabstat win\_rate\_yuangao 电动车原告 行人原告 death\_and\_injury 上诉 jianyichengxu free\_code zhengren words\_body if type\_p  
 > ro!=0 ,by(type\_pro) stat(mean)

Summary statistics: mean  
 by categories of: type\_pro

type_pro	win_ra..	电~原告	行人原告	death_~y	上诉	jianyi~u	free_c~e	zhengren	words_~y
1	.6606118	.0237194	.0174582	.3503128	.0849813	.6027467	.0001593	.0049389	2.432942
2	.6809536	.0274693	.0117627	.3361652	.0815168	.6302804	.0001727	.0043376	2.586202
3	.6429223	.0258269	.0171096	.3320049	.0864749	.6391461	.0001565	.005659	2.702141
4	.6653172	.0305241	.0141238	.3000897	.0794922	.6721943	.0001163	.0049184	2.762681
Total	.6620065	.0246133	.0166224	.345087	.0844201	.6116176	.0001586	.0049163	2.483993

33 .  
 34 . \*城市层面: 案件关键变量的描述统计, 分案件类型  
 35 . tabstat win\_rate\_yuangao 电动车原告 行人原告 death\_and\_injury 上诉 jianyichengxu free\_code zhengren words\_body if type\_c  
 > ity!=0,by(type\_city) stat(mean)

Summary statistics: mean  
 by categories of: type\_city

type_city	win_ra..	电~原告	行人原告	death_~y	上诉	jianyi~u	free_c~e	zhengren	words_~y
1	.6611612	.0257984	.0185636	.3567005	.0852757	.6008634	.0001657	.0051458	2.415001
2	.6743008	.0256823	.0127383	.3554006	.0813007	.621563	.0001402	.0039968	2.543592
3	.6417049	.0195618	.017467	.3318915	.0890029	.6103347	.0001358	.0058579	2.540714
4	.6588488	.0201582	.0111078	.3212854	.0902338	.6208672	.0002125	.0044528	2.663369
Total	.6616449	.0247273	.0167401	.3511367	.0853162	.6072597	.0001626	.0049369	2.47163

36 .  
 37 .  
 38 . \*\*\*\*\*表2 本地原告vs外地原告胜率回归结果\*\*\*\*\*  
 39 .  
 40 . \*案件层面控制变量  
 41 . global judge\_yuan "电动车原告 行人原告 death\_and\_injury 上诉 jianyichengxu free\_code zhengren words\_body"

```

42 .
43 . *超过99%的交通事故一审案件都是在基层法院审理，因此基准回归采用的样本为一审在基层法院的案子，only_city_court为1时表示一审
    > 发生在市级或更高级法院的交通事故案件。
44 . gen only_city_court=0

45 . replace only_city_court=1 if city_court!=" " & qu_court==" "
    (44,758 real changes made)

46 .
47 . /*
    > 基准回归样本里仅包含（本地，本地）、（外地，本地）的案件，即外地原告变量为0或为1的样本，不包含为-1时的样本
    > 时间固定效应：year
    > 市级固定效应：n_city_court
    > 县区级固定效应：n_qu_court
    > 下同
    > */
48 . qui reg win_rate_yuangao 外地原告 $judge_yuan i.year i.n_city_court if 外地原告 !=-1 & only_city_court!=1, vce(cluster pr
    > o_pair_id)

49 . est store m1_pro

50 .
51 . qui reghdfe win_rate_yuangao 外地原告 $judge_yuan if 外地原告 !=-1 & only_city_court!=1,absorb(year n_qu_court) vce(clust
    > er pro_pair_id)

52 . est store m2_pro

53 .
54 . qui reg win_rate_yuangao 外地原告_city $judge_yuan i.year i.n_city_court if 外地原告_city !=-1 & only_city_court!=1, vce(
    > cluster city_pair_id)

55 . est store m1_city

56 .
57 . qui reghdfe win_rate_yuangao 外地原告_city $judge_yuan if 外地原告_city !=-1 & only_city_court!=1,absorb(year n_qu_court)
    > vce(cluster city_pair_id)

58 . est store m2_city

59 .
60 .
61 . esttab m1_pro m2_pro m1_city m2_city , b(3) se(3) r2(3) keep(外地原告 外地原告_city $judge_yuan ) starlevel(* 0.1 ** 0.
    > 05 *** 0.01)

```

	(1) win_rate_y~o	(2) win_rate_y~o	(3) win_rate_y~o	(4) win_rate_y~o
外地原告	-0.022*** (0.003)	-0.022*** (0.002)		
电动车原告	0.066*** (0.005)	0.065*** (0.004)	0.071*** (0.004)	0.069*** (0.003)
行人原告	0.076*** (0.005)	0.076*** (0.005)	0.077*** (0.004)	0.077*** (0.003)
death_and_~y	0.034*** (0.006)	0.027*** (0.007)	0.036*** (0.003)	0.029*** (0.003)
上诉	0.005 (0.005)	-0.001 (0.005)	0.005 (0.003)	-0.001 (0.003)
jianyichen~u	0.234*** (0.008)	0.237*** (0.009)	0.238*** (0.005)	0.242*** (0.004)
free_code	-0.088** (0.043)	-0.099** (0.046)	-0.087* (0.049)	-0.093* (0.050)
zhengren	-0.123*** (0.014)	-0.121*** (0.012)	-0.128*** (0.008)	-0.124*** (0.007)
words_body	0.079*** (0.005)	0.084*** (0.006)	0.079*** (0.002)	0.084*** (0.002)
外地原告_c~y			-0.012*** (0.002)	-0.011*** (0.002)

N	1014060	1014032	824089	824054
R-sq	0.278	0.311	0.285	0.319

Standard errors in parentheses  
 \* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```

62 .
63 . *****表3 司法体制改革与非正式制度下的本地偏好*****
64 .
65 . *生成reform和外地原告、外地原告_city的交乘项
66 . gen 外地原告_reform=外地原告*reform if 外地原告!=-1
    (213,117 missing values generated)
67 . gen 外地原告_city_reform=外地原告_city*reform if 外地原告_city!=-1
    (418,172 missing values generated)
68 .
69 . qui reg win_rate_yuangao 外地原告 reform 外地原告_reform $judge_yuan i.year i.n_city_court if 外地原告!=-1 & only_city_c
    > ourt!=1, vce(cluster pro_pair_id)
70 . est store mr1
71 .
72 . qui reghdfe win_rate_yuangao 外地原告 reform 外地原告_reform $judge_yuan if 外地原告!=-1 & only_city_court!=1,absorb(yea
    > r n_qu_court) vce(cluster pro_pair_id)
73 . est store mr2
74 .
75 . qui reg win_rate_yuangao 外地原告_city reform 外地原告_city_reform $judge_yuan i.year i.n_city_court if 外地原告_city!=
    > -1 & only_city_court!=1, vce(cluster city_pair_id)
76 . est store mr3
77 .
78 . qui reghdfe win_rate_yuangao 外地原告_city reform 外地原告_city_reform $judge_yuan if 外地原告_city!=-1 & only_city_cour
    > t!=1,absorb(year n_qu_court) vce(cluster city_pair_id)
79 . est store mr4
80 .
81 . esttab mr1 mr2 mr3 mr4, b(3) se(3) r2(3) keep(外地原告 外地原告_city 外地原告_reform 外地原告_city_reform $judge_yuan re
    > form) starlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) win_rate_y~o	(2) win_rate_y~o	(3) win_rate_y~o	(4) win_rate_y~o
外地原告	-0.023*** (0.005)	-0.023*** (0.005)		
reform	-0.013 (0.011)	-0.014 (0.011)	-0.012** (0.006)	-0.013** (0.006)
外地原告_r~m	0.002 (0.007)	0.001 (0.008)		
电动车原告	0.066*** (0.004)	0.064*** (0.004)	0.071*** (0.004)	0.069*** (0.003)
行人原告	0.076*** (0.005)	0.076*** (0.005)	0.076*** (0.004)	0.077*** (0.003)
death_and_~y	0.034*** (0.006)	0.027*** (0.007)	0.036*** (0.003)	0.029*** (0.003)
上诉	0.006 (0.006)	-0.001 (0.005)	0.005 (0.003)	-0.001 (0.003)
jianyichen~u	0.234*** (0.008)	0.237*** (0.009)	0.238*** (0.005)	0.242*** (0.004)
free_code	-0.089** (0.042)	-0.100** (0.046)	-0.088* (0.049)	-0.094* (0.050)
zhengren	-0.123*** (0.014)	-0.121*** (0.012)	-0.128*** (0.008)	-0.124*** (0.007)
words_body	0.079***	0.084***	0.079***	0.084***

	(0.005)	(0.006)	(0.002)	(0.002)
外地原告_c~y			-0.013*** (0.003)	-0.012*** (0.003)
外地原告_c~m			0.003 (0.005)	0.003 (0.005)
N	1014060	1014032	824089	824054
R-sq	0.278	0.311	0.285	0.319

Standard errors in parentheses

\* p&lt;0.1, \*\* p&lt;0.05, \*\*\* p&lt;0.01

```

82 .
83 . *****表4 本地偏好的稳健性检验结果*****
84 .
85 . *****Panel A: 分数响应模型、Tobit模型*****
86 .
87 . qui fracreg logit win_rate_yuangao 外地原告 $judge_yuan i.year i.n_city_court if 外地原告 !=-1 & only_city_court!=1, vce(
> cluster pro_pair_id) iter(50) tolerance(1e-6)

88 . est store m_pro_fra

89 .
90 . qui fracreg logit win_rate_yuangao 外地原告_city $judge_yuan i.year i.n_city_court if 外地原告_city!=-1 & only_city_cour
> t!=1, vce(cluster city_pair_id) iter(50) tolerance(1e-6)

91 . est store m_city_fra

92 .
93 . qui tobit win_rate_yuangao 外地原告 $judge_yuan i.year i.n_city_court if 外地原告 !=-1 & only_city_court!=1, vce(cluster
> pro_pair_id)

94 . est store m_pro_tobit

95 .
96 . qui tobit win_rate_yuangao 外地原告_city $judge_yuan i.year i.n_city_court if 外地原告_city!=-1 & only_city_court!=1, vc
> e(cluster city_pair_id)

97 . est store m_city_tobit

98 .
99 . esttab m_pro_fra m_city_fra m_pro_tobit m_city_tobit , b(3) se(3) r2(3) pr2(3) keep(外地原告 外地原告_city $judge_yuan
> ) starlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) win_rate_y~o	(2) win_rate_y~o	(3) win_rate_y~o	(4) win_rate_y~o
win_rate_y~o				
外地原告	-0.115*** (0.016)		-0.022*** (0.003)	
电动车原告	0.368*** (0.030)	0.417*** (0.026)	0.066*** (0.005)	0.071*** (0.004)
行人原告	0.427*** (0.037)	0.457*** (0.026)	0.076*** (0.005)	0.077*** (0.004)
death_and_~y	0.147*** (0.034)	0.152*** (0.017)	0.034*** (0.006)	0.036*** (0.003)
上诉	0.009 (0.030)	0.001 (0.019)	0.005 (0.005)	0.005 (0.003)
jianyichen~u	1.093*** (0.036)	1.115*** (0.021)	0.234*** (0.008)	0.238*** (0.005)
free_code	-0.446** (0.220)	-0.437* (0.250)	-0.088** (0.043)	-0.087* (0.049)
zhengren	-0.648*** (0.072)	-0.691*** (0.044)	-0.123*** (0.014)	-0.128*** (0.008)
words_body	0.424*** (0.037)	0.445*** (0.015)	0.079*** (0.005)	0.079*** (0.002)
外地原告_c~y		-0.063***		-0.012***

	(0.011)		(0.002)	
N	1014060	824089	1014060	824089
R-sq				
pseudo R-sq	0.158	0.162	0.334	0.344

Standard errors in parentheses  
 \* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```

100 .
101 .
102 . *****Panel B: 二审胜率替换一审胜率*****
103 .
104 . *生成新的原告胜率win_rate_yuangao_替换二审, 即把有二审的案件的胜率, 替换为二审胜率
105 . gen win_rate_yuangao_替换二审=win_rate_yuangao
    (4,285 missing values generated)

106 . replace win_rate_yuangao_替换二审=win_rate_二审_yuangao if win_rate_二审_yuangao!=.
    (56,596 real changes made)

107 .
108 . qui reg win_rate_yuangao_替换二审 外地原告 $judge_yuan i.year i.n_city_court if 外地原告!=-1 & only_city_court!=1, vce(c
    > luster pro_pair_id)

109 . est store m1_pro_二审

110 .
111 . qui reghdfe win_rate_yuangao_替换二审 外地原告 $judge_yuan if 外地原告!=-1 & only_city_court!=1,absorb(year n_qu_court)
    > vce(cluster pro_pair_id)

112 . est store m2_pro_二审

113 .
114 . qui reg win_rate_yuangao_替换二审 外地原告_city $judge_yuan i.year i.n_city_court if 外地原告_city!=-1 & only_city_court
    > !=1, vce(cluster city_pair_id)

115 . est store m1_city_二审

116 .
117 . qui reghdfe win_rate_yuangao_替换二审 外地原告_city $judge_yuan if 外地原告_city!=-1 & only_city_court!=1,absorb(year n_
    > qu_court) vce(cluster city_pair_id)

118 . est store m2_city_二审

119 .
120 . esttab m1_pro_二审 m2_pro_二审 m1_city_二审 m2_city_二审, b(3) se(3) r2(3) keep(外地原告 外地原告_city $judge_yuan ) st
    > arlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) win_rate_~审	(2) win_rate_~审	(3) win_rate_~审	(4) win_rate_~审
外地原告	-0.020*** (0.002)	-0.021*** (0.002)		
电动车原告	0.064*** (0.004)	0.063*** (0.004)	0.069*** (0.004)	0.067*** (0.003)
行人原告	0.073*** (0.005)	0.072*** (0.005)	0.073*** (0.004)	0.073*** (0.003)
death_and_~y	0.032*** (0.006)	0.025*** (0.006)	0.034*** (0.003)	0.027*** (0.003)
上诉	0.121*** (0.004)	0.116*** (0.004)	0.120*** (0.003)	0.114*** (0.003)
jianyichen~u	0.232*** (0.008)	0.236*** (0.009)	0.237*** (0.005)	0.241*** (0.004)
free_code	-0.092** (0.046)	-0.102** (0.049)	-0.073 (0.045)	-0.080* (0.047)
zhengren	-0.107*** (0.012)	-0.105*** (0.010)	-0.112*** (0.008)	-0.109*** (0.007)
words_body	0.079*** (0.005)	0.084*** (0.006)	0.080*** (0.002)	0.085*** (0.002)

外地原告_c~y			-0.011*** (0.002)	-0.009*** (0.002)
N	1014060	1014032	824089	824054
R-sq	0.290	0.321	0.297	0.329

Standard errors in parentheses  
\* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```

121 .
122 . *****Panel C: 外地被告*****
123 .
124 . qui reg win_rate_beigao 外地被告 $judge_yuan i.year i.n_city_court if 外地被告!=-1 & only_city_court!=1, vce(cluster pro
> _pair_id)

125 . est store m1_pro_beigao

126 .
127 . qui reg win_rate_beigao 外地被告_city $judge_yuan i.year i.n_city_court if 外地被告_city!=-1 & only_city_court!=1, vce(c
> luster city_pair_id)

128 . est store m1_city_beigao

129 .
130 . qui reghdfe win_rate_beigao 外地被告 $judge_yuan if 外地被告!=-1 & only_city_court!=1,absorb(year n_qu_court) vce(clust
> er pro_pair_id)

131 . est store m2_pro_beigao

132 .
133 . qui reghdfe win_rate_beigao 外地被告_city $judge_yuan if 外地被告_city!=-1 & only_city_court!=1,absorb(year n_qu_court)
> vce(cluster city_pair_id)

134 . est store m2_city_beigao

135 .
136 . esttab m1_pro_beigao m2_pro_beigao m1_city_beigao m2_city_beigao , b(3) se(3) r2(3) keep(外地被告 外地被告_city $judge
> _yuan ) starlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) win_rate_b~o	(2) win_rate_b~o	(3) win_rate_b~o	(4) win_rate_b~o
外地被告	-0.012** (0.005)	-0.011** (0.005)		
电动车原告	-0.063*** (0.006)	-0.062*** (0.006)	-0.066*** (0.005)	-0.064*** (0.004)
行人原告	-0.075*** (0.006)	-0.075*** (0.006)	-0.074*** (0.004)	-0.074*** (0.003)
death_and_~y	-0.030*** (0.007)	-0.023*** (0.007)	-0.031*** (0.004)	-0.024*** (0.004)
上诉	-0.007 (0.006)	-0.001 (0.005)	-0.010*** (0.003)	-0.003 (0.003)
jianyichen-u	-0.230*** (0.008)	-0.233*** (0.009)	-0.233*** (0.005)	-0.236*** (0.005)
free_code	0.078* (0.044)	0.087* (0.046)	0.079 (0.049)	0.092* (0.050)
zhengren	0.119*** (0.017)	0.117*** (0.014)	0.111*** (0.010)	0.110*** (0.008)
words_body	-0.080*** (0.006)	-0.085*** (0.006)	-0.081*** (0.003)	-0.086*** (0.003)
外地被告_c~y			-0.006** (0.002)	-0.006*** (0.002)
N	1074407	1074383	932974	932945
R-sq	0.276	0.310	0.283	0.319

Standard errors in parentheses  
\* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```

137 .
138 . *****Panel D: 市级法院案件*****
139 .
140 . qui reg win_rate_yuangao 外地原告 $judge_yuan i.year i.n_city_court if 外地原告 !=-1 & only_city_court==1, vce(cluster pr
> o_pair_id)

141 . est store m1_only_city

142 .
143 . qui reg win_rate_yuangao 外地原告_city $judge_yuan i.year i.n_city_court if 外地原告_city !=-1 & only_city_court==1, vce(
> cluster city_pair_id)

144 . est store m2_only_city

145 .
146 . esttab m1_only_city m2_only_city , b(3) se(3) r2(3) keep(外地原告 外地原告_city $judge_yuan ) starlevel(* 0.1 ** 0.05 **
> * 0.01)

```

	(1) win_rate_y~o	(2) win_rate_y~o
外地原告	<b>-0.010</b> (0.007)	
电动车原告	<b>0.027</b> (0.017)	<b>0.030*</b> (0.016)
行人原告	<b>0.055**</b> (0.022)	<b>0.064***</b> (0.019)
death_and_~y	<b>0.038***</b> (0.012)	<b>0.039***</b> (0.010)
上诉	<b>-0.005</b> (0.009)	<b>-0.004</b> (0.009)
jianyichen~u	<b>0.217***</b> (0.028)	<b>0.232***</b> (0.017)
free_code	<b>-0.188</b> (0.239)	<b>-0.188</b> (0.239)
zhengren	<b>-0.144**</b> (0.065)	<b>-0.124*</b> (0.071)
words_body	<b>0.073***</b> (0.008)	<b>0.073***</b> (0.006)
外地原告_c~y		<b>-0.010</b> (0.007)
N	<b>35945</b>	<b>27205</b>
R-sq	<b>0.288</b>	<b>0.303</b>

Standard errors in parentheses  
\* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```

147 .
148 . *****分城市本地偏好指数估计*****
149 . encode city_court ,gen(n_city_court_new)

150 . gen city_id_new=n_city_court_new
(18,761 missing values generated)

151 .
152 . *其中_b[外地原告_city]表示外地原告_city的系数，即各城市的本地偏好指数
153 . statsby _b[外地原告_city] e(N),by(city_id_new) saving("/Users/csh/Desktop/数据文件/分城市本地偏好指数.dta",replace):reg
> win_rate_yuangao 外地原告_city $judge_yuan i.year i.n_qu_court if 外地原告_city !=-1 & only_city_court!=1 & year<=2017
(running regress on estimation sample)

command: regress win_rate_yuangao 外地原告_city 电动车原告 行人原告 death_and_injury 上诉 jianyichengxu free_code
zhengren words_body i.year i.n_qu_court if 外地原告_city !=-1 & only_city_court!=1 & year<=2017
_stat_1: _b[外地原告_city]
_stat_2: e(N)
by: city_id_new

```

Statsby groups

1	2	3	4	5
---	---	---	---	---

```

.....x.....x..... 50
.....x.....x..... 100
.....x.....x..... 150
.....x.....x..... 200
..x.....x.....x... 250
.....x.....x..... 300
.....x.....x..... 350
.....x.....x.....x..... 400
.....

154 .
155 . *****图1 分城市本地偏好指数*****
156 . use "/Users/csh/Desktop/数据文件/分城市本地偏好指数.dta",clear
      (statsby: regress)

157 . rename _stat_1 本地偏好指数

158 . *要求每个城市的样本量大于30
159 . graph bar 本地偏好指数 if 本地偏好指数 !=. & _stat_2>=30 , over(city_id_new, sort(1) descending reverse) ytitle("城市本地
    > 偏好指数") graphregion(fcolor(white))

160 .
161 . *****关键变量介绍*****
162 . /*贸易和本地偏好数据集中的关键变量（变量后面加2表示卖出地的变量）：
    > 本地偏好指数：根据2013-2017的案件样本，分城市估计外地原告的回归系数，即为各城市的本地偏好指数
    > 本地偏好指数_inv：买入地本地偏好指数（将估计出的本地偏好指数取相反数，即本地偏好指数_inv越大表示本地偏好程度越高）
    > 本地偏好指数2_inv：卖出地本地偏好指数（将估计出的本地偏好指数取相反数，即本地偏好指数2_inv越大表示本地偏好程度越高）
    > 政府支出比例：各城市一般预算支出与地区生产总值的比值
    > 公路密度：各城市公路里程和土地面积的比值
    > 人均_FDI：各城市人均FDI
    > 人均_FDI_diff：城市对的人均FDI差值的绝对值，再取对数
    > gdp_diff：城市对的人均GDP差值的绝对值，再取对数
    > distance：城市对之间的距离
    > altitude_diff：城市对之间的海拔差距
    > mean_remote：城市对的偏远度均值
    > tri_diff：城市对的崎岖度差距
    > 贸易总额_双边：城市对的贸易量
    > 贸易金额_工业品_双边：城市对工业品的贸易量
    > 人均地区生产总值元：各城市的人均GDP
    > 年底总人口户籍人口万人：各城市年底总人口户籍人口
    > 方言距离2015：城市对的方言距离
    > barrier_贸易总额：城市对的贸易成本，计算公式见原文
    > w_3_本地偏好指数：各城市的非当地本地偏好综合指数（delta=3）
    > w_5_本地偏好指数：各城市的非当地本地偏好综合指数（delta=5）
    > w_7_本地偏好指数：各城市的非当地本地偏好综合指数（delta=7）
    > 接受程度：基于2017年中国流动人口卫生计生动态监测调查得到的“接受”程度
    > 排斥程度：基于2017年中国流动人口卫生计生动态监测调查得到的“排斥”程度
    > Scc：城市对的产业结构相似度
    > ln_贸易总量：省份对的贸易量（对数），以2017年区域间投入产出表计算
    > ln_贸易总量_发票：省份对的贸易量（对数），以2017年增值税发票数据计算
    > ln_贸易总量_工业品：省份对的工业品贸易量（对数），以2017年区域间投入产出表计算
    > ln_贸易总量_工业品_发票：省份对的工业品贸易量（对数），以2017年增值税发票数据计算
    > ln_贸易总量_水电燃气：省份对的水电燃气贸易量（对数），以2017年区域间投入产出表计算
    > ln_贸易总量_水电燃气_发票：省份对的水电燃气贸易量（对数），以2017年增值税发票数据计算
    > */

163 .
164 . *导入数据集
165 . use "/Users/csh/Desktop/数据文件/贸易和本地偏好数据集.dta",clear

166 .
167 . *****表II1 主要指标描述性统计*****
168 . gen 政府支出比例=一般预算支出亿元/地区生产总值亿元
      (8,737 missing values generated)

169 . gen 政府支出比例2=一般预算支出亿元2/地区生产总值亿元2
      (8,773 missing values generated)

170 .
171 . gen 公路密度=公路里程公里/土地面积平方公里
      (39,794 missing values generated)

172 . gen 公路密度2=公路里程公里2/土地面积平方公里2
      (39,509 missing values generated)

173 .
174 . gen 人均_FDI=外商直接投资额实际使用额万美元/年底总人口户籍人口万人
      (60,595 missing values generated)

175 . gen 人均_FDI2=外商直接投资额实际使用额万美元2/年底总人口户籍人口万人2

```

(60,477 missing values generated)

```
176 .
177 .
178 . *城市对数据描述统计
179 . tabstat distance mean_remote altitude_diff river tri_diff 贸易总额_双边,stat(N mean sd min max )
```

stats	distance	mean_r-e	altitu~f	river	tri_diff	~额_双边
N	106755	106755	106755	106755	106197	106755
mean	1380.644	1415.437	883.6499	.3472999	272.1524	7.69e+08
sd	808.5291	286.6335	974.2794	.4761143	322.5692	8.28e+09
min	1.789435	1015.436	.0025749	0	.0064902	.01
max	4485.797	3304.075	5045.906	1	1898.01	1.02e+12

```
180 .
181 . *城市层面数据描述统计
182 . preserve

183 . duplicates drop cityname 人均地区生产总值元 年底总人口户籍人口万人 政府支出比例 公路密度,force
```

Duplicates in terms of **cityname** 人均地区生产总值元 年底总人口户籍人口万人 政府支出比例 公路密度

(106,418 observations deleted)

```
184 . tabstat 人均地区生产总值元 年底总人口户籍人口万人 政府支出比例 公路密度,stat(N mean sd min max )
```

stats	人均~元	年底~人	政府~例	公路密度
N	291	322	309	208
mean	53891.13	424.928	.278085	1.115582
sd	32295.1	321.0333	.2461975	.5537304
min	10707	20.7255	.0727967	.0389156
max	183544	3389.82	1.926349	2.307394

```
185 . restore

186 .
187 . *****表5 本地偏好与地级市贸易的回归结果*****
188 . *为变量取对数
189 . replace distance=log(distance)
    (106,755 real changes made)

190 . replace altitude_diff=log(altitude_diff)
    (106,755 real changes made)

191 . replace mean_remote=log(mean_remote)
    (106,755 real changes made)

192 . replace tri_diff=log(tri_diff)
    (106,197 real changes made)

193 . replace 贸易总额_双边=log(贸易总额_双边)
    (106,755 real changes made)

194 . replace 贸易金额_工业品_双边=log(贸易金额_工业品_双边)
    (99,749 real changes made)

195 . replace 人均地区生产总值元=log(人均地区生产总值元)
    (92,546 real changes made)

196 . replace 人均地区生产总值元2=log(人均地区生产总值元2)
    (92,422 real changes made)

197 . replace 年底总人口户籍人口万人=log(年底总人口户籍人口万人)
    (102,790 real changes made)

198 . replace 年底总人口户籍人口万人2=log(年底总人口户籍人口万人2)
    (102,768 real changes made)

199 . replace 方言距离2015 =log(方言距离2015)
    (73,329 real changes made, 354 to missing)

200 . gen barrier_贸易总额_ln=log(barrier_贸易总额)
    (67,382 missing values generated)

201 . gen 人均_FDI_diff=log(abs(人均_FDI-人均_FDI2))
```

```

(87,090 missing values generated)

202 . gen gdp_diff=人均地区生产总值元 - 人均地区生产总值元2
(26,657 missing values generated)

203 . replace gdp_diff=log(abs(gdp_diff))
(80,098 real changes made)

204 .
205 . global x "distance mean_remote altitude_diff river tri_diff"

206 .
207 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数2_inv
> ,robust

208 . est store m1

209 .
210 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数2_inv
> 公路密度 公路密度2,robust

211 . est store m2

212 .
213 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数2_inv
> highway,robust

214 . est store m3

215 .
216 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数2_inv
> 方言距离2015 ,robust

217 . est store m4

218 .
219 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数2_inv
> ln_买入地向卖出地迁徙指数 ln_卖出地向买入地迁徙指数 ,robust

220 . est store m5

221 .
222 . esttab m1 m2 m3 m4 m5, b(3) se(3) r2(3) keep(本地偏好指数_inv 本地偏好指数2_inv 边界 mean_border_num $x 人均地区生产总
> 值元 人均地区生产总值元2 公路密度 公路密度2 highway 方言距离2015 ln_买入地向卖出地迁徙指数 ln_卖出地向买入地迁徙指数) st
> arlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) 贸易总额_~边	(2) 贸易总额_~边	(3) 贸易总额_~边	(4) 贸易总额_~边	(5) 贸易总额_~边
边界	-1.224*** (0.045)	-1.315*** (0.055)	-1.200*** (0.045)	-1.159*** (0.049)	-0.187*** (0.038)
mean_borde-m	8.274*** (1.197)	-4.764** (2.009)	8.123*** (1.197)	17.922*** (1.284)	0.347 (1.002)
distance	-1.530*** (0.023)	-1.358*** (0.031)	-1.513*** (0.023)	-1.488*** (0.026)	1.156*** (0.031)
mean_remote	-1.380*** (0.088)	0.907*** (0.157)	-1.404*** (0.088)	-1.334*** (0.099)	-1.236*** (0.074)
altitude_d-f	-0.218*** (0.009)	-0.078*** (0.013)	-0.217*** (0.009)	-0.158*** (0.010)	-0.097*** (0.008)
river	-0.154*** (0.026)	0.200*** (0.035)	-0.153*** (0.026)	-0.156*** (0.027)	-0.438*** (0.022)
tri_diff	0.014 (0.010)	0.011 (0.014)	0.014 (0.010)	0.029*** (0.010)	0.008 (0.008)
人均地区~元	2.345*** (0.020)	2.126*** (0.027)	2.340*** (0.020)	2.376*** (0.021)	1.526*** (0.018)
人均地区生~2	2.639*** (0.021)	2.456*** (0.028)	2.635*** (0.021)	2.619*** (0.022)	1.858*** (0.019)
本地~数_inv	-0.123** (0.055)	-2.544*** (0.241)	-0.126** (0.056)	-2.172*** (0.171)	-1.623*** (0.147)

本地偏~2_inv	<b>-0.574***</b> (0.104)	<b>-3.680***</b> (0.252)	<b>-0.577***</b> (0.104)	<b>-3.608***</b> (0.211)	<b>-0.431***</b> (0.094)
公路密度		<b>0.437***</b> (0.037)			
公路密度2		<b>0.784***</b> (0.036)			
highway			<b>0.346***</b> (0.056)		
方言距离2015				<b>-0.154***</b> (0.030)	
ln_买入地~数					<b>0.494***</b> (0.013)
ln_卖出地~数					<b>0.613***</b> (0.013)
N	<b>64508</b>	<b>27940</b>	<b>64508</b>	<b>50703</b>	<b>63971</b>
R-sq	<b>0.468</b>	<b>0.524</b>	<b>0.468</b>	<b>0.496</b>	<b>0.605</b>

Standard errors in parentheses  
\* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```

223 .
224 . *****表6 本地偏好与贸易成本的回归结果*****
225 . qui reg barrier_贸易总额_ln 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数
    > 2_inv,robust

226 . est store m1

227 .
228 . qui reg barrier_贸易总额_ln 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数
    > 2_inv 公路密度 公路密度2 ,robust

229 . est store m2

230 .
231 . qui reg barrier_贸易总额_ln 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数
    > 2_inv highway ,robust

232 . est store m3

233 .
234 . qui reg barrier_贸易总额_ln 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数
    > 2_inv 方言距离2015 ,robust

235 . est store m4

236 .
237 . qui reg barrier_贸易总额_ln 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数
    > 2_inv ln_买入地向卖出地迁徙指数 ln_卖出地向买入地迁徙指数 ,robust

238 . est store m5

239 .
240 . esttab m1 m2 m3 m4 m5, b(3) se(3) r2(3) keep(本地偏好指数_inv 本地偏好指数2_inv 边界 mean_border_num $x 人均地区生产总
    > 值元 人均地区生产总值元2 公路密度 公路密度2 highway 方言距离2015 ln_买入地向卖出地迁徙指数 ln_卖出地向买入地迁徙指数) st
    > arlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) barrier_贸~n	(2) barrier_贸~n	(3) barrier_贸~n	(4) barrier_贸~n	(5) barrier_贸~n
边界	<b>2.481***</b> (0.106)	<b>2.741***</b> (0.132)	<b>2.455***</b> (0.105)	<b>2.274***</b> (0.113)	<b>0.969***</b> (0.087)
mean_borde-m	<b>-11.042***</b> (2.612)	<b>11.108**</b> (4.669)	<b>-10.871***</b> (2.612)	<b>-19.471***</b> (2.833)	<b>-0.166</b> (2.212)
distance	<b>2.881***</b> (0.054)	<b>2.543***</b> (0.074)	<b>2.862***</b> (0.055)	<b>2.772***</b> (0.061)	<b>-0.931***</b> (0.063)
mean_remote	<b>-0.628***</b> (0.196)	<b>-3.198***</b> (0.355)	<b>-0.601***</b> (0.196)	<b>-0.130</b> (0.225)	<b>-0.484***</b> (0.162)

altitude_d~f	0.362*** (0.020)	0.187*** (0.028)	0.361*** (0.020)	0.306*** (0.021)	0.204*** (0.017)
river	-0.075 (0.057)	-0.431*** (0.079)	-0.075 (0.057)	0.020 (0.061)	0.375*** (0.049)
tri_diff	-0.073*** (0.022)	-0.019 (0.030)	-0.074*** (0.022)	-0.064*** (0.023)	-0.013 (0.019)
人均地区~元	-2.723*** (0.047)	-2.450*** (0.066)	-2.719*** (0.047)	-2.702*** (0.049)	-1.711*** (0.041)
人均地区生~2	-2.800*** (0.044)	-2.541*** (0.059)	-2.794*** (0.045)	-2.858*** (0.047)	-1.554*** (0.041)
本地~数_inv	2.570*** (0.528)	8.081*** (0.536)	2.573*** (0.529)	4.691*** (0.438)	3.310*** (0.329)
本地偏~2_inv	3.416*** (0.496)	4.335*** (0.656)	3.425*** (0.496)	1.881*** (0.513)	3.538*** (0.429)
公路密度		-0.359*** (0.083)			
公路密度2		-1.113*** (0.082)			
highway			-0.366*** (0.131)		
方言距离2015				0.437*** (0.068)	
ln_买入地~数					-0.836*** (0.029)
ln_卖出地~数					-0.741*** (0.028)
N	26966	11683	26966	21651	26870
R-sq	0.452	0.517	0.452	0.466	0.593

Standard errors in parentheses  
\* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```
241 .
242 .
243 . *****表7 本地偏好对贸易的转移和挤出效应*****
244 .
245 . *delta=5, 贸易量
246 . qui reg 贸易总额_双边 边界 mean_border_num $x 本地偏好指数_inv 本地偏好指数2_inv 人均地区生产总值元 人均地区生产总值元2
> w_5_本地偏好指数 w_5_本地偏好指数2,robust

247 . est store mm1

248 .
249 . *delta=3 , 贸易量
250 . qui reg 贸易总额_双边 边界 mean_border_num $x 本地偏好指数_inv 本地偏好指数2_inv 人均地区生产总值元 人均地区生产总值元2
> w_3_本地偏好指数 w_3_本地偏好指数2,robust

251 . est store mm2

252 .
253 . *delta=7, 贸易量
254 . qui reg 贸易总额_双边 边界 mean_border_num $x 本地偏好指数_inv 本地偏好指数2_inv 人均地区生产总值元 人均地区生产总值元2
> w_7_本地偏好指数 w_7_本地偏好指数2,robust

255 . est store mm3

256 .
257 . *delta=5, 贸易成本
258 . qui reg barrier_贸易总额_ln 边界 mean_border_num $x 本地偏好指数_inv 本地偏好指数2_inv 人均地区生产总值元 人均地区生产
> 总值元2 w_5_本地偏好指数 w_5_本地偏好指数2 ,robust

259 . est store m2_test_1_1

260 .
261 . *delta=3 , 贸易成本
```

```

262 . qui reg barrier_贸易总额_ln 边界 mean_border_num $x 本地偏好指数_inv 本地偏好指数2_inv 人均地区生产总值元 人均地区生产
> 总值元2 w_3_本地偏好指数 w_3_本地偏好指数2 ,robust

263 . est store m2_test_1_2

264 .
265 . *delta=7, 贸易成本
266 . qui reg barrier_贸易总额_ln 边界 mean_border_num $x 本地偏好指数_inv 本地偏好指数2_inv 人均地区生产总值元 人均地区生产
> 总值元2 w_7_本地偏好指数 w_7_本地偏好指数2 ,robust

267 . est store m2_test_1_3

268 .
269 . esttab mm1 mm2 mm3 m2_test_1_1 m2_test_1_2 m2_test_1_3, b(3) se(3) r2(3) keep(边界 mean_border_num w_5_本地偏好指数 w_5
> _本地偏好指数2 w_3_本地偏好指数 w_3_本地偏好指数2 w_7_本地偏好指数 w_7_本地偏好指数2 人均地区生产总值元 人均地区生产总
> 值元2 本地偏好指数_inv 本地偏好指数2_inv) starlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) 贸易总额_~边	(2) 贸易总额_~边	(3) 贸易总额_~边	(4) barrier_贸~n	(5) barrier_贸~n	(6) barrier_贸~n
边界	-1.221*** (0.045)	-1.222*** (0.045)	-1.220*** (0.045)	2.479*** (0.106)	2.481*** (0.106)	2.478*** (0.106)
mean_borde-m	8.205*** (1.196)	8.261*** (1.196)	8.181*** (1.195)	-10.431*** (2.612)	-10.670*** (2.612)	-10.315*** (2.612)
本地~数_inv	-0.123** (0.056)	-0.135** (0.056)	-0.113** (0.056)	2.715*** (0.550)	2.675*** (0.546)	2.725*** (0.552)
本地偏~2_inv	-0.555*** (0.103)	-0.552*** (0.103)	-0.555*** (0.103)	3.319*** (0.499)	3.349*** (0.498)	3.320*** (0.498)
人均地区~元	2.367*** (0.020)	2.365*** (0.020)	2.368*** (0.020)	-2.761*** (0.047)	-2.744*** (0.047)	-2.770*** (0.047)
人均地区生~2	2.629*** (0.021)	2.623*** (0.021)	2.633*** (0.021)	-2.789*** (0.045)	-2.787*** (0.045)	-2.792*** (0.045)
w_5_本地~数	1.385*** (0.161)			-2.590*** (0.548)		
w_5_本地偏~2	-0.652*** (0.165)			0.601** (0.303)		
w_3_本地~数		1.614*** (0.207)			-1.848*** (0.667)	
w_3_本地偏~2		-1.296*** (0.215)			0.782** (0.393)	
w_7_本地~数			1.386*** (0.152)			-2.946*** (0.517)
w_7_本地偏~2			-0.395** (0.155)			0.518* (0.285)
N	64508	64508	64508	26966	26966	26966
R-sq	0.469	0.469	0.469	0.453	0.452	0.453

Standard errors in parentheses  
\* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```

270 .
271 . *****表III1 “排外”感知与城市本地偏好指数*****
272 . *城市层面的数据集
273 . preserve

274 . duplicates drop cityid 接受程度 排外程度 本地偏好指数_inv 人均地区生产总值元 年底总人口户籍人口万人 公路密度,force

Duplicates in terms of cityid 接受程度 排外程度 本地偏好指数_inv 人均地区生产总值元 年底总人口户籍人口万人 公路密度

(106,418 observations deleted)

275 . qui reg 接受程度 本地偏好指数_inv ,robust

276 . est store m1

277 . qui reg 排外程度 本地偏好指数_inv ,robust

```

```

278 . est store m2

279 . qui reg 接受程度 本地偏好指数_inv 人均地区生产总值元 公路密度 年底总人口户籍人口万人 政府支出比例 ,robust

280 . est store m3

281 . qui reg 排外程度 本地偏好指数_inv 人均地区生产总值元 公路密度 年底总人口户籍人口万人 政府支出比例 ,robust

282 . est store m4

283 . restore

284 .
285 . esttab m1 m2 m3 m4, b(3) se(4) r2(3) keep(年底总人口户籍人口万人 本地偏好指数_inv 人均地区生产总值元 公路密度 政府支出
> 比例) starlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) 接受程度	(2) 排外程度	(3) 接受程度	(4) 排外程度
本地~数_inv	<b>-0.420***</b> (0.1549)	<b>0.385**</b> (0.1832)	<b>-0.506**</b> (0.2449)	<b>0.670**</b> (0.2804)
人均地区~元			<b>-0.107**</b> (0.0474)	<b>0.100*</b> (0.0526)
公路密度			<b>0.017</b> (0.0356)	<b>-0.033</b> (0.0367)
年底总人~人			<b>-0.011</b> (0.0328)	<b>0.051</b> (0.0339)
政府支出比例			<b>-0.293</b> (0.2720)	<b>0.343</b> (0.3437)
N	250	250	145	145
R-sq	0.022	0.018	0.075	0.079

Standard errors in parentheses  
\* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```

286 .
287 .
288 .
289 . *****表 IV1 本地偏好与地级市贸易的稳健性检验 A*****
290 . gen 直辖市=0

291 . replace 直辖市=1 if cityname=="北京市" | cityname=="上海市" | cityname=="重庆市" | cityname=="天津市" | cityname2=="北京
> 市" | cityname2=="上海市" | cityname2=="重庆市" | cityname2=="天津市"
(2,675 real changes made)

292 .
293 . gen 一线城市=0

294 . replace 一线城市=1 if cityname=="北京市" | cityname=="上海市" | cityname=="广州市" | cityname=="深圳市" | cityname2=="北
> 京市" | cityname2=="上海市" | cityname2=="广州市" | cityname2=="深圳市"
(2,675 real changes made)

295 .
296 . gen 新一线城市=0

297 . replace 新一线城市=1 if cityname=="成都市" | cityname=="重庆市" | cityname=="杭州市" | cityname=="西安市" | cityname=="
> 武汉市" | cityname=="苏州市" | cityname=="郑州市" | cityname=="南京市" | cityname=="天津市" | cityname=="长沙市" | cityn
> ame=="东莞市" | cityname=="宁波市" | cityname=="佛山市" | cityname=="合肥市" | cityname=="青岛市" | cityname2=="成都市"
> | cityname2=="重庆市" | cityname2=="杭州市" | cityname2=="西安市" | cityname2=="武汉市" | cityname2=="苏州市" | cityname
> 2=="郑州市" | cityname2=="南京市" | cityname2=="天津市" | cityname2=="长沙市" | cityname2=="东莞市" | cityname2=="宁波市
> " | cityname2=="佛山市" | cityname2=="合肥市" | cityname2=="青岛市"
(9,855 real changes made)

298 .
299 . gen 计划单列市=0

300 . replace 计划单列市=1 if cityname=="深圳市" | cityname=="厦门市" | cityname=="宁波市" | cityname=="青岛市" | cityname=="
> 大连市" | cityname2=="深圳市" | cityname2=="厦门市" | cityname2=="宁波市" | cityname2=="青岛市" | cityname2=="大连市"
(3,332 real changes made)

301 .
302 . gen 省会城市=0

```

```

303 . replace 省会城市=1 if cityname=="济南市" | cityname=="石家庄市" | cityname=="长春市" | cityname=="哈尔滨市" | cityname==
> "沈阳市" | cityname=="呼和浩特市" | cityname=="乌鲁木齐市" | cityname=="兰州市" | cityname=="银川市" | cityname=="太原市"
> " | cityname=="西安市" | cityname=="郑州市" | cityname=="合肥市" | cityname=="南京市" | cityname=="杭州市" | cityname=="
> 福州市" | cityname=="广州市" | cityname=="南昌市" | cityname=="海口市" | cityname=="南宁市" | cityname=="贵阳市" | cityn
> ame=="长沙市" | cityname=="武汉市" | cityname=="成都市" | cityname=="昆明市" | cityname=="拉萨市" | cityname=="西宁市" |
> cityname2=="济南市" | cityname2=="石家庄市" | cityname2=="长春市" | cityname2=="哈尔滨市" | cityname2=="沈阳市" | cityna
> me2=="呼和浩特市" | cityname2=="乌鲁木齐市" | cityname2=="兰州市" | cityname2=="银川市" | cityname2=="太原市" | cityname
> 2=="西安市" | cityname2=="郑州市" | cityname2=="合肥市" | cityname2=="南京市" | cityname2=="杭州市" | cityname2=="福州市"
> " | cityname2=="广州市" | cityname2=="南昌市" | cityname2=="海口市" | cityname2=="南宁市" | cityname2=="贵阳市" | cityna
> me2=="长沙市" | cityname2=="武汉市" | cityname2=="成都市" | cityname2=="昆明市" | cityname2=="拉萨市" | cityname2=="西宁
> 市"
(17,398 real changes made)

304 .
305 .
306 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值2 本地偏好指数_inv 本地偏好指数2_inv
> if 一线城市==0 ,robust

307 . est store m1_test_1_1

308 .
309 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值2 本地偏好指数_inv 本地偏好指数2_inv
> if (新一线城市==0 | 一线城市==0),robust

310 . est store m1_test_1_2

311 .
312 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值2 本地偏好指数_inv 本地偏好指数2_inv
> if (省会城市==0 | 直辖市==0),robust

313 . est store m1_test_1_3

314 .
315 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值2 本地偏好指数_inv 本地偏好指数2_inv
> if 计划单列市==0,robust

316 . est store m1_test_1_4

317 .
318 . *制造业贸易量
319 . qui reg 贸易金额_工业品_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值2 本地偏好指数_inv 本地偏好指
> 数2_inv ,robust

320 . est store m1_test_1_5

321 .
322 . esttab m1_test_1_1 m1_test_1_2 m1_test_1_3 m1_test_1_4 m1_test_1_5, b(3) se(4) r2(3) keep(边界 mean_border_num $x 人均地
> 区生产总值元 人均地区生产总值2 本地偏好指数_inv 本地偏好指数2_inv) starlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) 贸易总额_~边	(2) 贸易总额_~边	(3) 贸易总额_~边	(4) 贸易总额_~边	(5) 贸~业品_双边
边界	-1.214*** (0.0452)	-1.225*** (0.0449)	-1.216*** (0.0449)	-1.240*** (0.0461)	-1.074*** (0.0411)
mean_borde~m	0.132 (1.2667)	7.966*** (1.2005)	6.343*** (1.1989)	9.159*** (1.2416)	6.875*** (1.0566)
distance	-1.583*** (0.0232)	-1.533*** (0.0227)	-1.537*** (0.0227)	-1.555*** (0.0234)	-1.423*** (0.0203)
mean_remote	-1.307*** (0.0888)	-1.360*** (0.0878)	-1.368*** (0.0878)	-1.388*** (0.0895)	-0.872*** (0.0724)
altitude_d~f	-0.208*** (0.0095)	-0.217*** (0.0093)	-0.217*** (0.0093)	-0.218*** (0.0096)	-0.221*** (0.0082)
river	-0.190*** (0.0260)	-0.153*** (0.0257)	-0.160*** (0.0256)	-0.149*** (0.0264)	-0.177*** (0.0226)
tri_diff	0.007 (0.0099)	0.013 (0.0098)	0.011 (0.0098)	0.015 (0.0101)	0.053*** (0.0083)
人均地区~元	2.231*** (0.0208)	2.334*** (0.0201)	2.328*** (0.0200)	2.278*** (0.0212)	2.157*** (0.0169)
人均地区生~2	2.538***	2.628***	2.622***	2.599***	2.465***

	(0.0219)	(0.0209)	(0.0209)	(0.0223)	(0.0169)
本地~数_inv	-0.145** (0.0563)	-0.126** (0.0555)	-0.127** (0.0555)	-0.116** (0.0565)	-0.135*** (0.0469)
本地偏~2_inv	-0.611*** (0.1053)	-0.577*** (0.1035)	-0.577*** (0.1035)	-0.574*** (0.1063)	-0.263*** (0.0925)
N	62464	64412	64370	61958	61832
R-sq	0.448	0.465	0.465	0.450	0.504

Standard errors in parentheses

\* p&lt;0.1, \*\* p&lt;0.05, \*\*\* p&lt;0.01

```

323 .
324 .
325 . *****表IV2 本地偏好与地级市贸易的稳健性检验B*****
326 . *Scc为产业结构相似度
327 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数2_inv
> Scc ,robust

328 . est store m1_Scc

329 .
330 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数2_inv
> gdp_diff,robust

331 . est store m1_gdp_diff

332 .
333 . qui reg 贸易总额_双边 边界 mean_border_num $x 人均地区生产总值元 人均地区生产总值元2 本地偏好指数_inv 本地偏好指数2_inv
> 人均_FDI_diff,robust

334 . est store m1_FDI_diff

335 .
336 . esttab m1_Scc m1_gdp_diff m1_FDI_diff, b(4) se(4) r2(4) keep(本地偏好指数_inv 本地偏好指数2_inv 边界 mean_border_num $x
> 人均地区生产总值元 人均地区生产总值元2 Scc gdp_diff 人均_FDI_diff) starlevel(* 0.1 ** 0.05 *** 0.01)

```

	(1) 贸易总额_~边	(2) 贸易总额_~边	(3) 贸易总额_~边
边界	-1.1968*** (0.0452)	-1.2309*** (0.0449)	-1.0621*** (0.0543)
mean_borde~m	-7.9001*** (1.2669)	8.1220*** (1.1959)	32.7230*** (2.1163)
distance	-1.5728*** (0.0231)	-1.5393*** (0.0228)	-1.2533*** (0.0327)
mean_remote	-1.1697*** (0.0885)	-1.3579*** (0.0880)	1.6441*** (0.1498)
altitude_d~f	-0.1959*** (0.0094)	-0.2209*** (0.0092)	-0.2815*** (0.0157)
river	-0.2125*** (0.0258)	-0.1613*** (0.0257)	0.1249*** (0.0395)
tri_diff	-0.0058 (0.0098)	0.0131 (0.0097)	0.1339*** (0.0163)
人均地区~元	2.2683*** (0.0203)	2.3280*** (0.0200)	1.7298*** (0.0347)
人均地区生~2	2.5716*** (0.0212)	2.6223*** (0.0210)	1.7718*** (0.0350)
本地~数_inv	-0.1684*** (0.0560)	-0.1436*** (0.0557)	-0.3518*** (0.0605)
本地偏~2_inv	-0.6115*** (0.1044)	-0.5974*** (0.1037)	-0.5485*** (0.1110)
Scc	-0.0004** (0.0002)		

```

gdp_diff                0.0860***
                        (0.0093)

人均_FDI_d~f                0.1215***
                              (0.0121)

```

N	62476	64508	14946
R-sq	0.4566	0.4685	0.5073

Standard errors in parentheses  
 \* p<0.1, \*\* p<0.05, \*\*\* p<0.01

```

337 .
338 . *****图A1 数据可靠性验证*****
339 . use "/Users/csh/Desktop/数据文件/贸易数据相关性验证.dta",clear

340 .
341 . twoway (scatter ln_贸易总量 ln_贸易总量_发票,mc(gs6) graphregion(color(gs14)) ytitle("省到省投入产出表金额，取Log") xtit
> le("省到省发票金额，取Log") title("省到省数据对比：对数值")) (lfit ln_贸易总量 ln_贸易总量_发票, lpattern(dash) lc(black
> ))

342 .
343 . twoway (scatter ln_贸易总量_工业品 ln_贸易总量_工业品_发票,mc(gs6) graphregion(color(gs14)) ytitle("省到省投入产出表金额
> ，取Log") xtitle("省到省发票金额，取Log") title("省到省数据对比：对数值（工业品）")) (lfit ln_贸易总量_工业品 ln_贸易总
> 量_工业品_发票, lpattern(dash) lc(black))

344 .
345 . twoway (scatter ln_贸易总量_水电燃气 ln_贸易总量_水电燃气_发票,mc(gs6) graphregion(color(gs14)) ytitle("省到省投入产出表
> 金额，取Log") xtitle("省到省发票金额，取Log") title("省到省数据对比：对数值（电热水燃气产品）")) (lfit ln_贸易总量_水电
> 燃气 ln_贸易总量_水电燃气_发票, lpattern(dash) lc(black))

346 .
347 .
348 .
349 .
350 .
    end of do-file

351 .

```