

基于 R 的文献复刻：利用中国微观数据库

黃建祺

2023-03-17

目录

前言	3
0.1 缘起	3
0.2 为何是 R	3
0.3 如何食用	3
0.4 R 入门	4
1 土地流转研究	7
1.1 载入包	7
1.2 导入数据	7
1.3 查看变量标签	8
1.4 数据规整	23
1.5 模型建立	29
2 文献复刻:《劳动力流动如何影响农户借贷》	34
2.1 文献回顾	34
2.2 数据处理	34
2.3 统计描述	34
2.4 模型设定	34
3 宗族文化	35
3.1 数据载入	35
3.2 可视化	38

目录	2
3.3 地图	38
3.4 其他数据源	43
3.5 重新再利用	45
4 CHARLS	52
4.1 数据导入	52
5 文献复刻: 《新型农村社会养老保险政策效果评估》	54
5.1 数据导入	54
6 CHFS	55
6.1 数据读取	55
6.2 构建变量	56
7 其他来源数据	63
7.1 方言数据	63
7.2 夜间灯光数据	66
7.3 政治数据	66
8 论文复刻 Climate risks and market efficiency	67
8.1 Data	67
9 文献复刻: The Long-term Effects of Africa's Slave Trades	74
9.1 文献回顾	74
9.2 数据来源	74
9.3 计算最近的贸易距离	76
10 后记	82

前言

0.1 缘起

突然想要研究某一个中国的微观数据库，但又无奈苯人掌握 Stata 技法有限，想要再去学有点学不动了。忽然一次在网上看到林敏杰老师的《CFPS 之 R 语言学习笔记》瞬间为我打开了一扇窗，所以尝试使用 R 来实现一些在实证经济学的一些应用或称为复刻，另外加上一丁点的探索。之前也看到国外的在利用 **shiny** 搭建的网站¹做的文献复刻，因此就有一些用 R 来做的冲动。

0.2 为何是 R

为何选择使用 R 呢。现在在复刻文献的主流方法是使用 Stata，同时在一些 wiki 或者期刊网站、数据网站² 上大部分是使用 stata 来实现复刻或者源代码的。但目前，一些经济学、政治学学者开始在利用 R 做基本的计量分析，甚至开发专门的 R 包来实现一些高级计量方法，在 R 中有很多超越于 Stata 的优势。

0.3 如何食用

这本书是基于中国的几大微观数据库及相关的顶刊上的文章为主要内容写成的自我技术修炼笔记，为最大化他的社会效益，以 **bookdown** 形式生成。希望能够对你有所帮助。

在经济学研究中，近些年来对于微观数据的使用变得尤为重要，尤其是大型的微观数据库的使用。但常用的处理方法往往是选择在 Stata 中进行操作。但在另一方面，R 相对于 Stata 的使用有其独特的优势，尤其是在使用多个数据框操作时候，在今天的一篇文章中很难就单单一个数据源就能完成一篇好的文章写作，因此对于不同数据源进行交互性操作愈发重要。因此有必要使用 R 来对数据进行相应的操作。同时 R 也是支持于.dta 数据的读取。

同时再深了讲在不同语言之间的比较，Aruoba and Fernández-Villaverde (2015) 有对比不同的编程语言的运行速度，同时需要强调的是 C++/Fortran/Java 对比其他的语言是更难学习的，由其本身的特性所决定的，所以对于经济学家来说没有那么多的时间成本学习前三种语言，但可以作为一个参照。最后发现 Julia 的表现尤其突出，

¹<https://ejd.econ.mathematik.uni-ulm.de/>

²这篇知乎总结了所有可以发现代码的地方

甚至超过于 Java，远甩身后的 Matlab 好几米；Python 与 R 基本上持平，但一用上 Pypy 编译器立马加速；最慢的是 Mathematica（虽然我也没用过）³

Julia 的强劲表现让很多经济学家极力推崇⁴ 相关的学习材料在Quantecon 就很丰富了。

但我们这里并不需要大量的数据，因此对于编译速度的要求并不高，RStudio 能够满足日常的基本需要。

0.4 R 入门

关于 R 的入门这里不多介绍，bookdown中有大量关于 R 的入门书籍。这里可以做出一定的推荐：

英文比较好的话：

- 官方 manual
- R for Data Science
- modern dive
- Big book of R：和字典一样厚，作者至今还在更新。

看中文更有优势的话：

- R 语言教程
- 数据科学中的 R 语言

当然因为这里主要介绍一些经济学论文和经济学方法，不可避免要学习和使用计量及 R 上的应用。同样有很多出色的线上教材。

- Introduction to Econometric with R
- Causal Inference: The Mixtape：有三种语言的代码任你选择。

需要注意的是在 R 中进行数据处理，必然无法避开学习和使用 **tidyverse**，因为这才是数据科学学习 R 的优势之处。

这篇所需要使用的 R 包：使用 **pacman** 免去验证是否安装的烦恼。

```
if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

³文章的代码可以在 github 仓库中找到

⁴以诺奖得主、宏观学者 Sargent 为代表

Table 1: Average and Relative Run Time (Seconds)

Language	Mac			Windows	
	Version/Compiler	Time	Rel. Time	Version/Compiler	Time
C++	GCC-4.9.0	0.73	1.00	Visual C++ 2010	0.73
	Intel C++ 14.0.3	1.00	1.38	Intel C++ 14.0.2	0.93
	Clang 5.1	1.00	1.38	GCC-4.8.2	1.10
Fortran	GCC-4.9.0	0.76	1.05	GCC-4.8.1	1.70
	Intel Fortran 14.0.3	0.95	1.30	Intel Fortran 14.0.2	0.88
Java	JDK8u5	1.95	2.69	JDK8u5	1.50
Julia	0.2.1	1.92	2.64	0.2.1	2.00
Matlab	2014a	7.91	10.88	2014a	6.70
Python	Pypy 2.2.1	31.90	43.86	Pypy 2.2.1	34.00
	CPython 2.7.6	195.87	269.31	CPython 2.7.4	117.00
R	3.1.1, compiled	204.34	280.90	3.1.1, compiled	184.00
	3.1.1, script	345.55	475.10	3.1.1, script	371.00
Mathematica	9.0, base	588.57	809.22	9.0, base	473.00
Matlab, Mex	2014a	1.19	1.64	2014a	0.90
Rcpp	3.1.1	2.66	3.66	3.1.1	4.00
Python	Numba 0.13	1.18	1.62	Numba 0.13	1.10
	Cython	1.03	1.41	Cython	1.80
Mathematica	9.0, idiomatic	1.67	2.29	9.0, idiomatic	2.20

图 1: 图来自上述文章

```
p_load(tidyverse,  
       purrr,  
       haven,  
       visdat,  
       sf,  
       units,  
       lwgeom,  
       rmapshaper,  
       tictoc)
```

Chapter 1

土地流转研究

数据来源：CFPS

本章主要参考四川师范大学王敏杰老师的研究笔记

1.1 载入包

```
library(tidyverse)
library(purrr)
library(haven)
library(visdat)
```

1.2 导入数据

```
cfps2010family <- read_dta("data/cfps/2010/cfps2010famecon_202008.dta")
cfps2010family %>%
  select(fid, urban, starts_with("fk201_a")) %>%
  glimpse()

## Rows: 14,797
## Columns: 8
## $ fid      <dbl+lbl> 110001, 110003, 110005, 110006, 110007, 110009, 110010, ~
## $ urban    <dbl+lbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

```
## $ fk201_a_1 <dbl+lbl> -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, ~
## $ fk201_a_2 <dbl+lbl> -8.0, -8.0, -8.0, -8.0, 2.5, -8.0, -8.0, -8.0, -8.0, -8~
## $ fk201_a_3 <dbl+lbl> -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, ~
## $ fk201_a_4 <dbl+lbl> -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, ~
## $ fk201_a_5 <dbl+lbl> -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, ~
## $ fk201_a_6 <dbl+lbl> -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, -8, ~
```

1.3 查看变量标签

对于原有数据，都是存在一个标签来显示原始的问题形式，因此我们可以先查看我们想要找的问题的标签是否对应。先创建一个 `get_var_label` 的函数。

```
library(purrr)
get_var_label <- function(dta) {
  labels <- map(dta, function(x) attr(x, "label"))
  data_frame(
    name = names(labels),
    label = as.character(labels)
  )
}
```

根据观察原有标签，我们可知 `fk201_a_n` 的变量都是拥有的农业资产，`fk202_a_n`、`fk203_a_n` 和 `fk204_a_n` 分别是经营、转租入和转租出多少农业资产。

```
cfps2010family %>%
  select(urban, starts_with("fk201_a")) %>%
  get_var_label()

## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## i Please use `tibble()` instead.

## # A tibble: 7 x 2
##   name      label
##   <chr>     <chr>
## 1 urban    基于国家统计局资料的城乡分类变量
## 2 fk201_a_1 您家拥有多少亩水田
## 3 fk201_a_2 您家拥有多少亩旱地
## 4 fk201_a_3 您家拥有多少亩林地
```

```
## 5 fk201_a_4 您家拥有多少亩果园  
## 6 fk201_a_5 您家拥有多少亩草场  
## 7 fk201_a_6 您家拥有多少亩池塘  
  
cfps2010family %>%  
  select(urban, starts_with("fk201_a")) %>%  
  map(~ count(data.frame(x = .x), x))  
  
## $urban  
##   x     n  
## 1 0 7694  
## 2 1 7103  
##  
## $fk201_a_1  
##      x     n  
## 1 -8.0 11656  
## 2 -1.0    3  
## 3  0.0   20  
## 4  0.1    6  
## 5  0.2   14  
## 6  0.3   36  
## 7  0.4   36  
## 8  0.5   74  
## 9  0.6   54  
## 10 0.7   49  
## 11 0.8   52  
## 12 0.9   14  
## 13 1.0  346  
## 14 1.1   18  
## 15 1.2   61  
## 16 1.3   27  
## 17 1.4   27  
## 18 1.5  144  
## 19 1.6   28  
## 20 1.7   24  
## 21 1.8   45  
## 22 1.9   14  
## 23 2.0  462
```

```
## 24    2.1    15
## 25    2.2    17
## 26    2.3    20
## 27    2.4    32
## 28    2.5    94
## 29    2.6    15
## 30    2.7    23
## 31    2.8    25
## 32    2.9     5
## 33    3.0   326
## 34    3.1     3
## 35    3.2    17
## 36    3.3     9
## 37    3.4    10
## 38    3.5    48
## 39    3.6    17
## 40    3.7     5
## 41    3.8    11
## 42    3.9     2
## 43    4.0   228
## 44    4.1     4
## 45    4.2    10
## 46    4.3     6
## 47    4.4     9
## 48    4.5    22
## 49    4.6     2
## 50    4.7     3
## 51    4.8     4
## 52    4.9     1
## 53    5.0   143
## 54    5.2     3
## 55    5.3     1
## 56    5.4     6
## 57    5.5    14
## 58    5.6     1
## 59    5.7     4
## 60    5.8     1
## 61    6.0    96
```

```
## 62     6.1      1
## 63     6.3      1
## 64     6.5      9
## 65     6.6      4
## 66     7.0     53
## 67     7.2      2
## 68     7.3      1
## 69     7.4      1
## 70     7.5      4
## 71     7.6      1
## 72     7.8      2
## 73     7.9      1
## 74     8.0     69
## 75     8.1      2
## 76     8.3      1
## 77     8.4      2
## 78     8.5      2
## 79     8.8      1
## 80     9.0     19
## 81    10.0     48
## 82    10.2      1
## 83    10.5      1
## 84    10.8      3
## 85    11.0     15
## 86    11.5      1
## 87    11.6      1
## 88    11.7      1
## 89    12.0     19
## 90    12.6      2
## 91    13.0      7
## 92    13.2      1
## 93    13.5      1
## 94    14.0      4
## 95    14.5      1
## 96    15.0     14
## 97    15.4      1
## 98    16.0      6
## 99    16.4      1
```

```
## 100 17.0      1
## 101 18.0      2
## 102 19.0      2
## 103 20.0      5
## 104 21.0      2
## 105 22.0      1
## 106 24.0      4
## 107 25.0      3
## 108 26.0      2
## 109 30.0      2
## 110 37.0      1
## 111 40.0      2
## 112 60.0      2
## 113 63.0      1
## 114 122.0     1
##
## $fk201_a_2
##          x    n
## 1      -8.0 8552
## 2      -1.0  12
## 3       0.0  42
## 4       0.1  31
## 5       0.2  67
## 6       0.3  74
## 7       0.4  39
## 8       0.5 178
## 9       0.6  64
## 10      0.7  50
## 11      0.8  47
## 12      0.9  11
## 13      1.0 503
## 14      1.1  10
## 15      1.2  45
## 16      1.3  19
## 17      1.4  26
## 18      1.5 132
## 19      1.6  27
## 20      1.7  16
```

```
## 21     1.8    33
## 22     1.9     4
## 23     2.0   542
## 24     2.1    12
## 25     2.2    18
## 26     2.3    11
## 27     2.4    17
## 28     2.5    96
## 29     2.6     7
## 30     2.7    13
## 31     2.8    23
## 32     2.9     3
## 33     3.0   479
## 34     3.1     2
## 35     3.2    22
## 36     3.3    13
## 37     3.4    17
## 38     3.5    44
## 39     3.6    27
## 40     3.7    10
## 41     3.8    12
## 42     3.9     7
## 43     4.0   384
## 44     4.1     5
## 45     4.2    17
## 46     4.3     3
## 47     4.4     9
## 48     4.5    70
## 49     4.6     4
## 50     4.7     8
## 51     4.8    16
## 52     4.9     3
## 53     5.0   357
## 54     5.1     1
## 55     5.2     8
## 56     5.3     5
## 57     5.4    12
## 58     5.5    20
```

```
## 59      5.6    14
## 60      5.7     5
## 61      5.8     4
## 62      5.9     2
## 63      6.0   337
## 64      6.1     2
## 65      6.2     5
## 66      6.3     5
## 67      6.4    11
## 68      6.5    17
## 69      6.6     4
## 70      6.7     4
## 71      6.8     9
## 72      6.9     3
## 73      7.0   220
## 74      7.2    11
## 75      7.3     1
## 76      7.4     5
## 77      7.5   23
## 78      7.6     5
## 79      7.7     5
## 80      7.8   15
## 81      8.0   220
## 82      8.1     5
## 83      8.2     5
## 84      8.4     7
## 85      8.5   13
## 86      8.6     1
## 87      8.7     3
## 88      8.8     8
## 89      9.0   128
## 90      9.2     4
## 91      9.3     1
## 92      9.4     3
## 93      9.5    10
## 94      9.6    10
## 95      9.7     2
## 96      9.8     3
```

```
## 97     9.9     1
## 98    10.0   291
## 99    10.4     1
## 100   10.5     6
## 101   10.6     1
## 102   10.7     1
## 103   10.8     6
## 104   11.0    73
## 105   11.2     1
## 106   11.3     1
## 107   11.5    11
## 108   11.6     1
## 109   11.7     2
## 110   11.8     4
## 111   11.9     1
## 112   12.0   162
## 113   12.4     1
## 114   12.5     7
## 115   12.6     1
## 116   12.7     1
## 117   12.8     2
## 118   13.0    67
## 119   13.2     2
## 120   13.5     2
## 121   13.7     1
## 122   13.8     3
## 123   14.0    65
## 124   14.1     1
## 125   14.5     2
## 126   14.7     1
## 127   14.8     1
## 128   14.9     1
## 129   15.0   116
## 130   15.2     1
## 131   15.3     1
## 132   15.4     1
## 133   15.5     1
## 134   15.6     3
```

```
## 135 15.7 1
## 136 16.0 57
## 137 16.8 1
## 138 16.9 2
## 139 17.0 32
## 140 17.2 1
## 141 17.4 2
## 142 17.5 3
## 143 17.6 1
## 144 17.9 1
## 145 18.0 46
## 146 18.1 1
## 147 18.5 2
## 148 18.6 1
## 149 18.7 1
## 150 19.0 12
## 151 19.2 1
## 152 19.5 1
## 153 20.0 123
## 154 20.2 1
## 155 20.8 1
## 156 21.0 20
## 157 21.5 2
## 158 22.0 25
## 159 22.4 1
## 160 22.5 1
## 161 23.0 17
## 162 24.0 22
## 163 25.0 25
## 164 25.7 1
## 165 26.0 7
## 166 26.9 1
## 167 27.0 7
## 168 27.5 1
## 169 28.0 9
## 170 29.0 5
## 171 29.9 1
## 172 30.0 45
```

```
## 173 31.6    1
## 174 32.0    4
## 175 33.0    3
## 176 33.2    1
## 177 33.5    1
## 178 34.0    1
## 179 34.1    1
## 180 34.4    1
## 181 34.5    1
## 182 35.0    6
## 183 35.9    1
## 184 36.0    4
## 185 40.0   14
## 186 45.0    2
## 187 47.0    2
## 188 50.0    5
## 189 52.0    1
## 190 55.0    1
## 191 56.0    1
## 192 60.0    4
## 193 70.0    1
## 194 90.0    1
## 195 100.0   3
## 196 160.0   1
## 197 200.0   1
## 198 225.0   1
##
## $fk201_a_3
##          x     n
## 1      -8.0 13699
## 2      -1.0    5
## 3       0.0   13
## 4       0.1   10
## 5       0.2   10
## 6       0.3   17
## 7       0.4    6
## 8       0.5   30
## 9       0.6    6
```

```
## 10      0.7      6
## 11      0.8     12
## 12      0.9      1
## 13      1.0    143
## 14      1.1      2
## 15      1.2      3
## 16      1.3      2
## 17      1.4      4
## 18      1.5     23
## 19      1.6      4
## 20      1.7      1
## 21      1.8      3
## 22      2.0    103
## 23      2.1      1
## 24      2.3      1
## 25      2.4      4
## 26      2.5      7
## 27      2.6      1
## 28      2.7      2
## 29      2.8      2
## 30      3.0    119
## 31      3.1      1
## 32      3.2      1
## 33      3.4      1
## 34      3.5      1
## 35      3.6      1
## 36      3.7      1
## 37      3.8      2
## 38      3.9      1
## 39      4.0     54
## 40      4.3      1
## 41      4.4      1
## 42      4.5      3
## 43      4.8      2
## 44      5.0     61
## 45      5.2      1
## 46      5.3      1
## 47      6.0     33
```

```
## 48      6.4      2
## 49      7.0     25
## 50      7.2      1
## 51      7.5      6
## 52      7.8      1
## 53      8.0     28
## 54      8.1      1
## 55      8.3      1
## 56      8.5      1
## 57      9.0      8
## 58      9.6      1
## 59     10.0     61
## 60     10.3      1
## 61     11.0      4
## 62     11.2      1
## 63     11.5      2
## 64     11.7      1
## 65     12.0     13
## 66     13.0      5
## 67     13.5      1
## 68     14.0      6
## 69     14.7      1
## 70     15.0     24
## 71     16.0      4
## 72     16.5      1
## 73     16.8      1
## 74     17.0      1
## 75     17.5      1
## 76     18.0      1
## 77     19.0      3
## 78     20.0     40
## 79     21.0      3
## 80     22.0      4
## 81     22.9      1
## 82     23.0      2
## 83     24.0      4
## 84     25.0      4
## 85     26.0      1
```

```
## 86    27.0    2
## 87    27.4    1
## 88    29.0    1
## 89    30.0   31
## 90    34.0    1
## 91    35.0    5
## 92    36.0    1
## 93    40.0   10
## 94    41.0    1
## 95    44.0    1
## 96    45.0    2
## 97    49.7    1
## 98    50.0   18
## 99    53.0    1
## 100   56.0    1
## 101   60.0   10
## 102   68.0    1
## 103   70.0    6
## 104   75.0    1
## 105   80.0    8
## 106   90.0    2
## 107  100.0    7
## 108  102.0    1
## 109  120.0    1
## 110  150.0    2
## 111  160.0    1
## 112  200.0    3
## 113  300.0    1
## 114  370.0    1
## 115 1000.0    3
##
## $fk201_a_4
##      x      n
## 1    -8.0 14151
## 2    -1.0     6
## 3     0.0    19
## 4     0.1     6
## 5     0.2     9
```

```
## 6    0.3    10
## 7    0.4     4
## 8    0.5    33
## 9    0.6     2
## 10   0.7     4
## 11   0.8     6
## 12   0.9     2
## 13   1.0   108
## 14   1.1     1
## 15   1.2     4
## 16   1.4     1
## 17   1.5    23
## 18   1.6     4
## 19   1.7     3
## 20   1.8     3
## 21   2.0   102
## 22   2.3     1
## 23   2.4     1
## 24   2.5     9
## 25   2.7     4
## 26   2.8     2
## 27   3.0    76
## 28   3.2     1
## 29   3.5     6
## 30   3.6     1
## 31   3.7     4
## 32   3.8     1
## 33   3.9     1
## 34   4.0    53
## 35   4.2     1
## 36   4.5     6
## 37   4.8     1
## 38   5.0    27
## 39   5.1     1
## 40   5.3     1
## 41   5.5     2
## 42   6.0    26
## 43   7.0    10
```

```
## 44    7.4      1
## 45    8.0      10
## 46   10.0      24
## 47   12.0      4
## 48   15.0      3
## 49   20.0      7
## 50   21.0      1
## 51   25.0      1
## 52   30.0      1
## 53   32.0      1
## 54   35.0      1
## 55   40.0      2
## 56   50.0      2
## 57   80.0      1
## 58 100.0      1
## 59 400.0      1
##
## 
## $fk201_a_5
##      x      n
## 1 -8.0 14789
## 2   1.0      4
## 3   2.0      2
## 4   3.0      1
## 5   3.3      1
##
## 
## $fk201_a_6
##      x      n
## 1  -8.0 14706
## 2   0.0     10
## 3   0.1      1
## 4   0.2      3
## 5   0.3      3
## 6   0.4      1
## 7   0.5      7
## 8   0.6      1
## 9   0.7      1
## 10  0.8      2
## 11  1.0      9
```

```
## 12 1.5      2
## 13 1.6      1
## 14 2.0      7
## 15 2.5      1
## 16 3.0      5
## 17 4.0      2
## 18 4.7      1
## 19 5.0      4
## 20 6.0      5
## 21 6.3      1
## 22 7.0      1
## 23 8.4      1
## 24 13.0     2
## 25 14.0     1
## 26 16.0     1
## 27 20.0     4
## 28 22.0     2
## 29 23.0     2
## 30 24.0     1
## 31 25.0     3
## 32 30.0     1
## 33 33.0     1
## 34 38.0     1
## 35 50.0     2
## 36 63.0     1
```

这里使用了 `map` 函数来构建一个映射，映射到一个累加求和，第一张表是农业户口和城镇户口的数量对比，后面的表都是密度分布。

1.4 数据规整

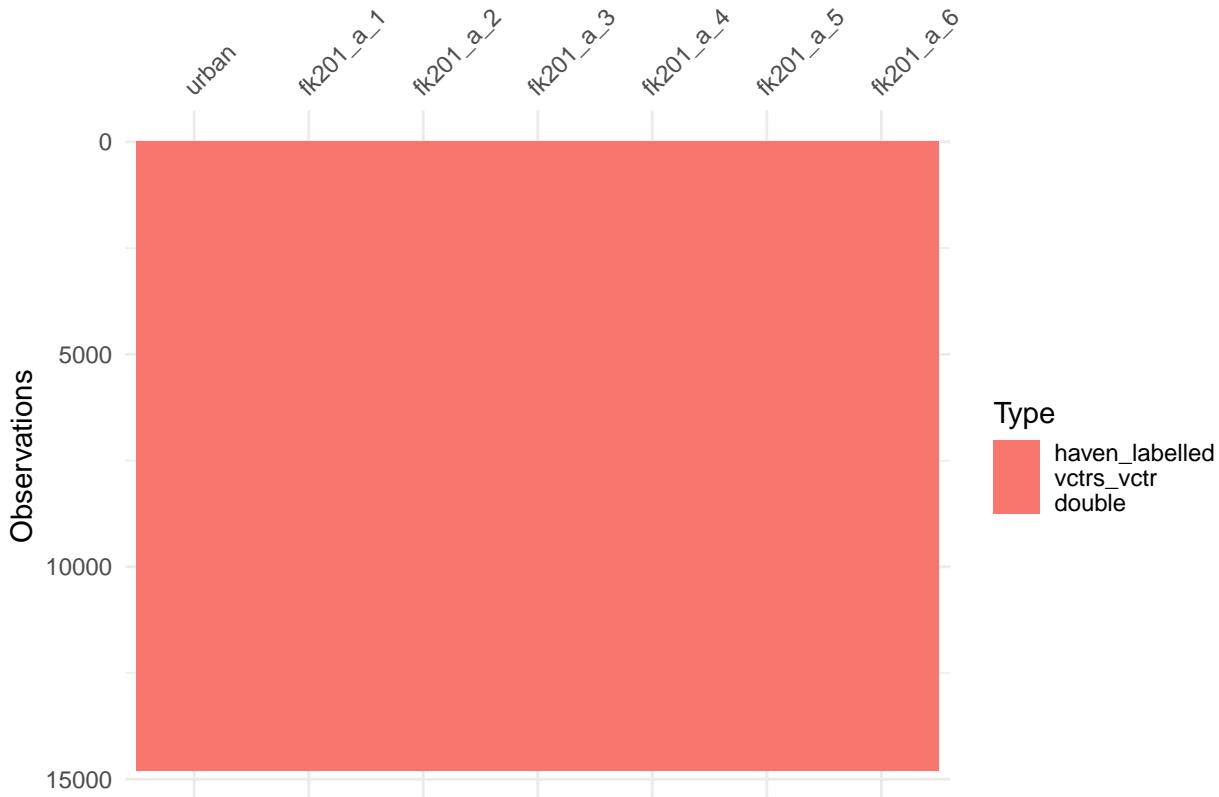
```
library(nanar)
cfps2010family %>%
  select(urban, starts_with("fk201_a")) %>%
  miss_var_summary()

## # A tibble: 7 x 3
```

```
##   variable n_miss pct_miss
##   <chr>     <int>    <dbl>
## 1 urban        0        0
## 2 fk201_a_1    0        0
## 3 fk201_a_2    0        0
## 4 fk201_a_3    0        0
## 5 fk201_a_4    0        0
## 6 fk201_a_5    0        0
## 7 fk201_a_6    0        0
```

基本上是没有缺失数据。

```
library(visdat)
cfps2010family %>%
  select(urban, starts_with("fk201_a")) %>%
  vis_dat()
```



为防止包之间的函数冲突，使用 `conflicted` 来 prefer 到 `dplyr` 中的 `filter`。

```

library(conflicted)
conflict_prefer("filter", "dplyr")

cfps2010family %>%
  select(urban, starts_with("fk2_s"))%>%
  filter(urban == 0)

## # A tibble: 7,694 x 6
##   urban   fk2_s_1   fk2_s_2   fk2_s_3   fk2_s_4   fk2_s_5
##   <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl>
## 1 0 [乡村]    4 [果园]   -8 [不适用]  -8 [不适用]  -8 [不适用]
## 2 0 [乡村]    2 [旱地]   -8 [不适用]  -8 [不适用]  -8 [不适用]
## 3 0 [乡村]    4 [果园]   -8 [不适用]  -8 [不适用]  -8 [不适用]
## 4 0 [乡村]    4 [果园]   -8 [不适用]  -8 [不适用]  -8 [不适用]
## 5 0 [乡村]   -8 [不适用] -8 [不适用]  -8 [不适用]  -8 [不适用]
## 6 0 [乡村]   -8 [不适用] -8 [不适用]  -8 [不适用]  -8 [不适用]
## 7 0 [乡村]   -8 [不适用] -8 [不适用]  -8 [不适用]  -8 [不适用]
## 8 0 [乡村]    6 [池塘]   -8 [不适用]  -8 [不适用]  -8 [不适用]
## 9 0 [乡村]    4 [果园]   -8 [不适用]  -8 [不适用]  -8 [不适用]
## 10 0 [乡村]  -8 [不适用] -8 [不适用]  -8 [不适用]  -8 [不适用]
## # ... with 7,684 more rows

```

先找出有经营土地的家户：并不考虑是否是自己拥有还是转租入。

```

a <- cfps2010family %>%
  select(fid,urban, starts_with("fk201_a")) %>%
  filter_at(vars(starts_with("fk201_a")), any_vars(. > 0))

## # A tibble: 7,688 x 8
##   fid   urban   fk201_a_1   fk201_a_2 fk201_~1 fk201_~2 fk201_~3 fk201_~4
##   <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lb> <dbl+lb> <dbl+lb> <dbl+lb>
## 1 110007    1 [城镇]   -8 [不适用]  2.5      ~ -8 [不~ -8 [不~ -8 [不~
## 2 120033    1 [城镇]   -8 [不适用]  -8 [不适~ -8 [不~  0.900 ~ -8 [不~ -8 [不~
## 3 120073    0 [乡村]   -8 [不适用]  -8 [不适~ -8 [不~  5.10   ~ -8 [不~ -8 [不~
## 4 120074    0 [乡村]   -8 [不适用]  1.80     ~ -8 [不~ -8 [不~ -8 [不~ -8 [不~
## 5 120076    0 [乡村]   -8 [不适用]  -8 [不适~ -8 [不~  6       ~ -8 [不~ -8 [不~
## 6 120080    0 [乡村]   -8 [不适用]  -8 [不适~ -8 [不~ -8 [不~ -8 [不~  38      ~

```

```

## 7 120081 0 [乡村] -8 [不适用] -8 [不适~ -8 [不~ 4 ~ -8 [不~ -8 [不~
## 8 120084 0 [乡村] -8 [不适用] -8 [不适~ -8 [不~ 1.5 ~ -8 [不~ -8 [不~
## 9 120087 0 [乡村] -8 [不适用] -8 [不适~ -8 [不~ 3 ~ -8 [不~ -8 [不~
## 10 120088 0 [乡村] -8 [不适用] 1.5 ~ -8 [不~ -8 [不~ -8 [不~ -8 [不~
## # ... with 7,678 more rows, and abbreviated variable names 1: fk201_a_3,
## #   2: fk201_a_4, 3: fk201_a_5, 4: fk201_a_6

```

再将负值转变为 0。

```
a %>% mutate_at(vars(starts_with("fk201_a")), funs(replace(., . < 0, 0)))
```

```

## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))

## # A tibble: 7,688 x 8
##   fid      urban    fk201_a_1 fk201_a_2 fk201_a_3 fk201_a_4 fk201_a_5 fk201~1
##   <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl>
## 1 110007    1 [城镇]    0        2.5     0        0        0        0
## 2 120033    1 [城镇]    0        0        0        0.900   0        0
## 3 120073    0 [乡村]    0        0        0        5.10    0        0
## 4 120074    0 [乡村]    0        1.80    0        0        0        0
## 5 120076    0 [乡村]    0        0        0        6        0        0
## 6 120080    0 [乡村]    0        0        0        0        0        38
## 7 120081    0 [乡村]    0        0        0        4        0        0
## 8 120084    0 [乡村]    0        0        0        1.5     0        0
## 9 120087    0 [乡村]    0        0        0        3        0        0
## 10 120088   0 [乡村]    0        1.5     0        0        0        0
## # ... with 7,678 more rows, and abbreviated variable name 1: fk201_a_6

```

1.4.1 农业生产效率

```
a <- cfps2010family %>%
  select(fid, urban, starts_with("fk201_a"), fk3, fk4, fe1) %>%
  mutate(revenue = fk3 - fk4) %>%
  mutate_at(vars(starts_with("fk201_a")), funs(replace(., . < 0, 0))) %>%
  mutate_at(vars("revenue"), funs(replace(., . < 0, 0))) %>%
  dplyr::filter(revenue > 0)
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()` : tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(.., trim = .2), ~ median(.., na.rm = TRUE))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()` : tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(.., trim = .2), ~ median(.., na.rm = TRUE))
```

```
a
```

```
## # A tibble: 6,093 x 12
##   fid      urban   fk201~1 fk201~2 fk201~3 fk201~4 fk201~5 fk201~6 fk3     fk4
##   <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl> <dbl>
## 1 110007    1 [城~ 0       2.5     0       0       0       0       1800    500
## 2 120033    1 [城~ 0       0       0       0.900   0       0       16000   1500
## 3 120074    0 [乡~ 0       1.80    0       0       0       0       2100    900
## 4 120075    0 [乡~ 0       0       0       0       0       0       15000   5500
## 5 120076    0 [乡~ 0       0       0       6       0       0       30000   12000
## 6 120081    0 [乡~ 0       0       0       4       0       0       32000   6000
## 7 120084    0 [乡~ 0       0       0       1.5     0       0       7000    4000
## 8 120087    0 [乡~ 0       0       0       3       0       0       25000   5000
```

```
## 9 120088 0 [乡~ 0 1.5 0 0 0 0 5300 300
## 10 120090 0 [乡~ 0 0 0 4.5 0 0 26000 6000
## # ... with 6,083 more rows, 2 more variables: fe1 <dbl+lbl>, revenue <dbl>, and
## # abbreviated variable names 1: fk201_a_1, 2: fk201_a_2, 3: fk201_a_3,
## # 4: fk201_a_4, 5: fk201_a_5, 6: fk201_a_6
```

一个有效的建议是在对原始数据进行操作时候，尽量保证原始数据的不变，再通过`%>%`进行传导到新的数据框中。我们计算农业生产效率的方法有很多这里主要参考的是一些主流的做法：将单位面积纯利润作为效率的衡量指标

```
a%>%
  mutate(landsum = rowSums(.[2:7]))%>%
  filter(landsum>0)%>%
  mutate(rates = revenue/landsum)->a1
a1

## # A tibble: 6,048 x 14
##   fid      urban fk201~1 fk201~2 fk201~3 fk201~4 fk201~5 fk201~6 fk3     fk4
##   <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl> <dbl>
## 1 110007 1 [城~ 0 2.5 0 0 0 0 1800 500
## 2 120033 1 [城~ 0 0 0 0.900 0 0 16000 1500
## 3 120074 0 [乡~ 0 1.80 0 0 0 0 2100 900
## 4 120076 0 [乡~ 0 0 0 6 0 0 30000 12000
## 5 120081 0 [乡~ 0 0 0 4 0 0 32000 6000
## 6 120084 0 [乡~ 0 0 0 1.5 0 0 7000 4000
## 7 120087 0 [乡~ 0 0 0 3 0 0 25000 5000
## 8 120088 0 [乡~ 0 1.5 0 0 0 0 5300 300
## 9 120090 0 [乡~ 0 0 0 4.5 0 0 26000 6000
## 10 120091 0 [乡~ 0 0 0 4 0 0 30000 5000
## # ... with 6,038 more rows, 4 more variables: fe1 <dbl+lbl>, revenue <dbl>,
## # landsum <dbl>, rates <dbl>, and abbreviated variable names 1: fk201_a_1,
## # 2: fk201_a_2, 3: fk201_a_3, 4: fk201_a_4, 5: fk201_a_5, 6: fk201_a_6
```

1.4.2 流动人口

我们可以用外出打工在家庭人口中的占比来测算流动率。

E1 E1"是否有人外出工作"

过去一年，您家是否有人外出工作？

- 1. 有
- 3. 无【跳至 E2】
- 5. 去年尚未成家【跳至 E2】

F1: (1) “外出”指不在自己户口和/或家庭常住地工作，农村通常指县/县级市以外，城市通常指本市以外。

(2) “外出工作”指非永久性离开家庭所在县（市）的就业，如农村人口外出打工。

【CAPI】如果 E1 选择“1”则进入【外出工作模块】，否则跳至 E2。

```
library(conflicted)
conflict_prefer('filter',"dplyr")
a1%>%
  filter(fe1!=5)->a2
a2$fe1[a2$fe1==3] <- 0
a2%>%
  select(rates,fe1)

## # A tibble: 6,017 x 2
##      rates fe1
##      <dbl> <dbl+lbl>
## 1    371. 0
## 2   7632. 0
## 3    667. 0
## 4   3000  1 [有]
## 5   6500  0
## 6   2000  0
## 7   6667. 0
## 8   3333. 0
## 9   4444. 0
## 10  6250  0
## # ... with 6,007 more rows
```

1.5 模型建立

我们试图考察关于流动人口与农业生产效率之间的关系：

```

reg <- lm(data = a2, fe1~rates)
summary(reg)

## 
## Call:
## lm(formula = fe1 ~ rates, data = a2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.3873 -0.3869 -0.3862  0.6130  0.9042 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 3.873e-01 6.329e-03   61.2   <2e-16 ***
## rates      -8.330e-07 6.409e-07    -1.3    0.194    
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.4869 on 6015 degrees of freedom
## Multiple R-squared:  0.0002807, Adjusted R-squared:  0.0001145 
## F-statistic: 1.689 on 1 and 6015 DF,  p-value: 0.1938

```

不过不显著。。。不过系数上看是一个较为合理的存在（效率上升，抑制外出）。对于一个想要看星星的 reg monkey 来说极其苦恼。我们可以考虑换一个变量：一篇 2016 年在《中国农村经济》的文章研究“非农就业、土地流转与农业生产效率变化”利用的是非农就业来考察劳动生产率（同样也是用单位土地的农产品收入来测算）就较为显著，主要的差别在于非农就业数量来测度，并非一个虚拟变量。还有一个可能是在先前的数据处理中存在一定的问题，比如是否将未从事农业活动的家户过滤进来。

```

cfps2010family%>%
  select(fid,familysize,starts_with("fu1_s"))%>%
  mutate_at(vars(starts_with("fu1_s")), funs(replace(., . < 0, 0)))%>%
  mutate_at(vars(starts_with("fu1_s")), funs(replace(., . >=1, 1)))>b1

## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
## 
## # Simple named list: list(mean = mean, median = median)
## 

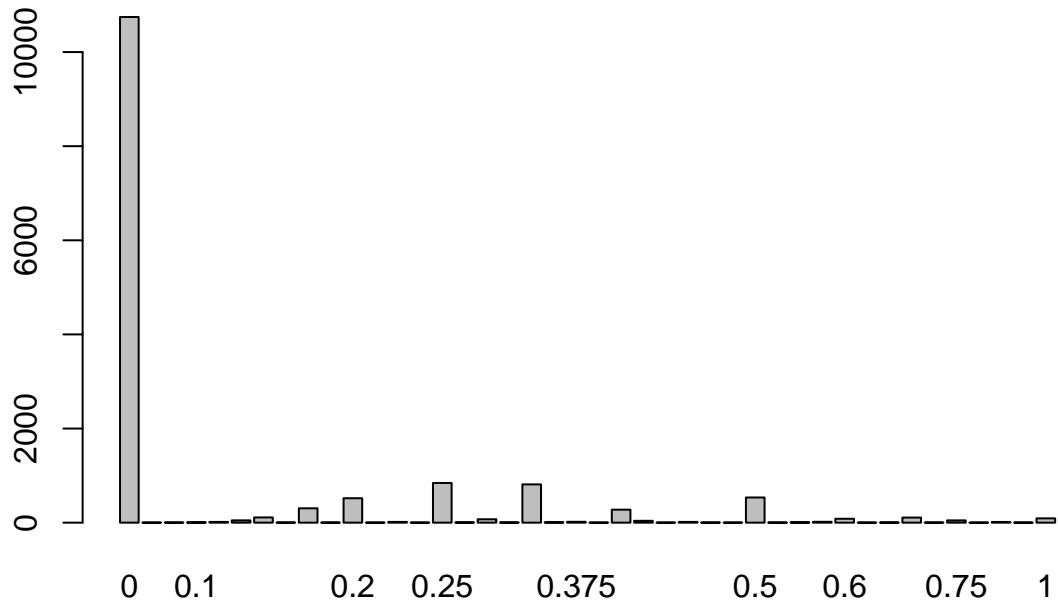
```

```
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))

## Warning: `fun` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

并不建议一次性将所有变换都做完，之后再检查是非常痛苦的。。。

```
b1%>%
  mutate(mig = rowSums(.[3:14]))%>%
  select(fid,familysize,mig)%>%
  mutate(mig_rate = mig/familysize)%>%
  filter(mig_rate<=1&mig_rate>=0)->b2# 剔除异常值
barplot(table(b2$mig_rate))
```



```
dim(b2)

## [1] 14795      4

b3 <- merge(a2,b2,by="fid")
reg2 <- lm(data = b3,mig_rate~rates)
summary(reg2)

##
## Call:
## lm(formula = mig_rate ~ rates, data = b3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.1249 -0.1248 -0.1247  0.1251  0.8760 
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1.249e-01 2.384e-03 52.40 <2e-16 ***  
## rates       -2.029e-07 2.415e-07 -0.84   0.401  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.1834 on 6015 degrees of freedom  
## Multiple R-squared: 0.0001174, Adjusted R-squared: -4.887e-05  
## F-statistic: 0.706 on 1 and 6015 DF, p-value: 0.4008
```

p 值比之前还更大了。。。上述提到的文章的核心解释变量是非农占家庭劳动力比例，但目前还不知咋构建的。。。想到了再补上去。

Chapter 2

文献复刻：《劳动力流动如何影响农户借贷》

2.1 文献回顾

这篇文章主要发现劳动力流动导致农户借出的概率和金额显著增加。

- 核心的被解释变量为家庭是否有借给亲戚、朋友等的借出款项和为家庭人均借出金额的对数值。降低极端值影响，进行上下 2% 缩尾处理。
- 劳动力流动：是否有劳动力流动以及家庭劳动力流动人数。
- 控制变量：

2.2 数据处理

加载 cfps2018 数据：

2.3 统计描述

2.4 模型设定

Chapter 3

宗族文化

对于宗族文化的研究近些年一直是较为火热的研究热点 (Cao et al., 2022),(潘越 et al., 2019),(张心仪 et al., 2021),(陈斌开 & 陈思宇, 2018),(张川川 & 马光荣, 2017),(Zhang, 2020) 和 (Fan et al., 2023)。但对于宗族文化的测度方法又各具差异, 比如 Zhang (2020), Cao et al. (2022) 和 Fan et al. (2023) 都是使用上海古籍出版社的县地方族谱数据来测量宗族文化, 张川川 and 马光荣 (2017) 使用的是 CFPS 的数据来测量; 张心仪 et al. (2021) 使用的是地方的前三姓氏来作为度量, 数据来源是 2005 年的 1% 人口抽样调查数据。数据质量上, 直观感受是上海古籍出版社的数据会优于其他几个。

3.1 数据载入

如何在 R 中没有任何资源的前提下进行关于宗族文化的测度, 我先试了做法最为简便的, 城市的前三姓氏我翻遍了所有变量都没找到姓氏的变量; 后面又看了下上海古籍出版社数据, 数据量太大, 估计需要爬虫等黑科技, 遂又放弃, 之后只有选择 CFPS, 之后也并不很顺利。

```
library(haven)
```

A3 CA3"设施拥有情况"您村/居地界内是否有以下设施？【可多选】

访员注意：不论所有权是否属于村/居，只要在地界范围内就算有。

1. 小商店/小卖部/百货店
2. 幼儿园
3. 小学
4. 医院/医疗点
5. 药店
6. 庙宇/道观
- 7. 家族祠堂**
8. 教堂/清真寺
9. 老年活动场所/老年社区服务机构
10. 敬老院/养老院
11. 体育运动场所
12. 儿童游乐场所
13. 村/居务公告栏
14. 举报箱
15. 社区网站

F1: (1) “**家族祠堂**”是指家族公共聚会的场所，也是家族供奉祖先牌位的地方。

(2) “**儿童游乐场所**”是指具有儿童玩耍设施（如滑梯）的场所。

【CAPI】针对 A3 选择的除“13”外的选项，分别提问 A301。

A301 CA301"拥有数量"您村/居地界内有多少个“*** (A3 选项)"？_____ 1..1000 个

【CAPI】针对 A3 选择“13”，提问 A302。

A302 CA302"公告栏内容"您村/居务公告栏张贴以下哪几方面的内容？【可多选】

1. 医保相关
2. 低保相关
3. 计划生育相关
4. 财务相关

```
cfps2010comm <- read_dta("/Users/a182501/rproject/cfps/data/社区数据 cfps/cfps2010comm_201906.dta")
cfps2010comm%>%
  select(cid, provcd, countyid, cyear, cmonth, ca3_s_6, ca3_s_7) -> df1
head(df1)
```

```
## # A tibble: 6 x 7
##   cid      provcd    countyid  cyear     cmonth ca3_s_6      ca3_s_7
##   <dbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl> <dbl+lbl> <dbl+lbl>
## 1 13200    12 [天津市]    79    2010        10 -8 [不适用] ~ -8 [不~
## 2 13190    12 [天津市]    79    2010        10  9 [老年活动场所/~ 13 [村/~
## 3 12780    14 [山西省]    69    2010        10  8 [教堂/清真寺] ~ 10 [敬~
## 4 21340    44 [广东省]   116    2010        10  9 [老年活动场所/~ 13 [村/~
## 5 12260    23 [黑龙江省]  56    2010        9 -8 [不适用] ~ -8 [不~
## 6 21640    44 [广东省]   123    2010        10  7 [家族祠堂] ~ 11 [体~
```

```
dim(df1)
```

```
## [1] 635    7
```

```
cfps2010comm$ca3_s_6[which(df1$ca3_s_6== -8)] = 0
cfps2010comm$ca3_s_7[which(df1$ca3_s_7== -8)] = 0
table(cfps2010comm$ca3_s_7)
```

```

## 
##   0    2    7    8    9   10   11   12   13   14   15
## 262    2   14     6   41   26   66   26  106   68   18

table(cfps2010comm$ca3_s_6)

## 
##   0    2    3    5    6    7    8    9   10   11   12   13   14   15
## 172    1    1    1   47   23   23   92   24   58   21   94   72    6

na.omit(df1)

## # A tibble: 635 x 7
##       cid      provcd      countyid    cyear   cmonth ca3_s_6      ca3_s_7
##       <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 13200      12 [天津市]      79      2010      10 -8 [不适用] ~ -8 [不~
## 2 13190      12 [天津市]      79      2010      10  9 [老年活动场所~ 13 [村/~
## 3 12780      14 [山西省]      69      2010      10  8 [教堂/清真寺]~ 10 [敬~
## 4 21340      44 [广东省]     116      2010      10  9 [老年活动场所~ 13 [村/~
## 5 12260      23 [黑龙江省]     56      2010      9 -8 [不适用] ~ -8 [不~
## 6 21640      44 [广东省]     123      2010      10  7 [家族祠堂] ~ 11 [体~
## 7 21730      44 [广东省]     126      2010      10 -8 [不适用] ~ -8 [不~
## 8 22523      62 [甘肃省]     145      2010      9 -8 [不适用] ~ -8 [不~
## 9 10930      52 [贵州省]      24      2010      10 12 [儿童游乐场所~ 13 [村/~
## 10 10100      34 [安徽省]      3      2010      10  6 [庙宇/道观] ~ 10 [敬~

## # ... with 625 more rows

dim(df1)

## [1] 635    7

```

根据社区问卷手册，我们可以指导

```

library(dplyr)
cfps2010comm%>%
  group_by(provcd)%>%
  dplyr::summarise(x1=sum(ca3_s_6),x2=sum(ca3_s_7))->df2
df2

```

```
## # A tibble: 25 x 3
##   provcd      x1     x2
##   <dbl+lbl>    <dbl> <dbl>
## 1 11 [北京市]    27    14
## 2 12 [天津市]    9    13
## 3 13 [河北省]   216   178
## 4 14 [山西省]   207   220
## 5 21 [辽宁省]   525   511
## 6 22 [吉林省]   97    112
## 7 23 [黑龙江省] 159   155
## 8 31 [上海市]   495   386
## 9 32 [江苏省]   127   117
## 10 33 [浙江省]  111   126
## # ... with 15 more rows
```

3.2 可视化

```
library(ggplot2)
ggplot(df2) +
  geom_point(aes(x=x1,y=x2)) +
  geom_smooth(method = 'lm',aes(x=x1,y=x2))

## `geom_smooth()` using formula = 'y ~ x'
```

3.3 地图

将论文图表绘制在图上。

```
d <- attributes(df2$provcd)$labels
d <- as.data.frame(d)
d2 <- rownames(d)
d3 <- cbind(d,d2)
colnames(d3) <- c("provcd","label")
d4 <- merge(df2,d3,by = "provcd")
d4
```

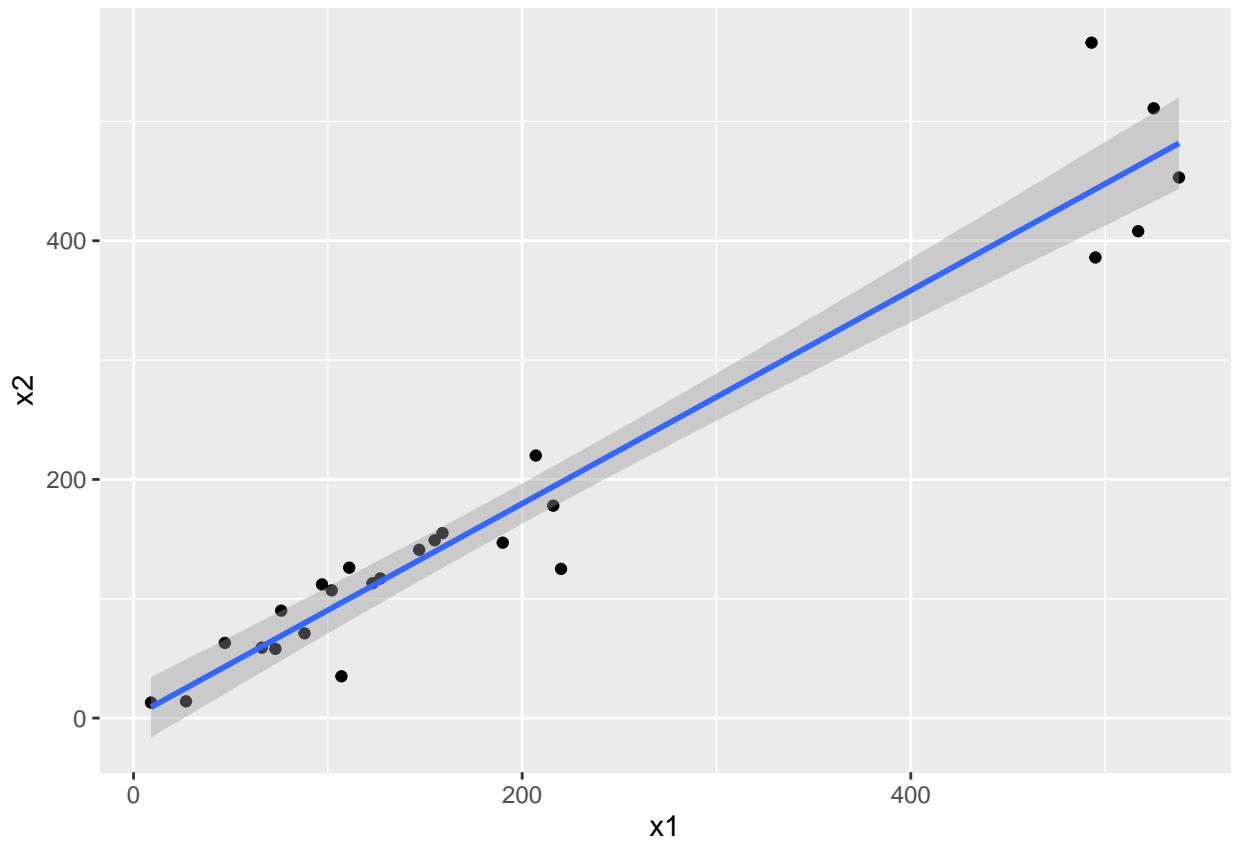


图 3.1: 祠堂与族谱

```

##   provcd  x1  x2      label
## 1       11  27  14    北京市
## 2       12   9  13    天津市
## 3      13 216 178    河北省
## 4      14 207 220    山西省
## 5      21 525 511    辽宁省
## 6      22  97 112    吉林省
## 7      23 159 155    黑龙江省
## 8      31 495 386    上海市
## 9      32 127 117    江苏省
## 10     33 111 126    浙江省
## 11     34 123 113    安徽省
## 12     35  47  63    福建省
## 13     36 102 107    江西省
## 14     37 220 125    山东省
## 15     41 538 453    河南省
## 16     42  76  90    湖北省
## 17     43 190 147    湖南省
## 18     44 493 566    广东省
## 19     45 107  35  广西壮族自治区
## 20     50  66  59    重庆市
## 21     51 155 149    四川省
## 22     52  88  71    贵州省
## 23     53 147 141    云南省
## 24     61  73  58    陕西省
## 25     62 517 408    甘肃省

```

json 数据来源于阿里 DataV 数据可视化平台，能够在多个行政层级绘制中国地图。

```

library(echarts4r.maps)
library(echarts4r)
colnames(d4) <- c("provcd", "value1", "value2", "region")
china_map <- jsonlite::read_json("rep.json")
d4 %>%
  e_charts(region)%>%
  e_map_register("China2", china_map) %>%
  e_map(value1, map = "China2") %>%
  e_visual_map(value1)

```

```
d4 %>%
  e_charts(region)%>%
  e_map_register("China2", china_map) %>%
  e_map(value2, map = "China2") %>%
  e_visual_map(value2)
```

上面的数据还是挺让人吃惊的，一般会认为宗族文化会在南方更为发达，包括修建祠堂上，我们通过图 3.1 中知道祠堂与家谱是基本上在省层面是正相关的，但地域上呈现了较大的差异。可能是与抽样方法有关，需要进一步的处理。

3.4 其他数据源

目前学界用的较为广泛的是通过上海家族族谱来测算宗族文化，也就是看一个地方的族谱的密度来作为宗族文化的代理变量，代表性学者有浙大的张川川老师，他目前发表的关于宗族文化的论文有 (Cao et al., 2022), (Zhang, 2020), (张川川 & 马光荣, 2017)。很巧，他和合作者Yiqin Xu和博士生曹家瑞在 JDE 刊发的论文有replicate file(可直接下载)

但图中的图是使用 ArcGIS 来实现的，这里试图通过 R 来进行复刻。

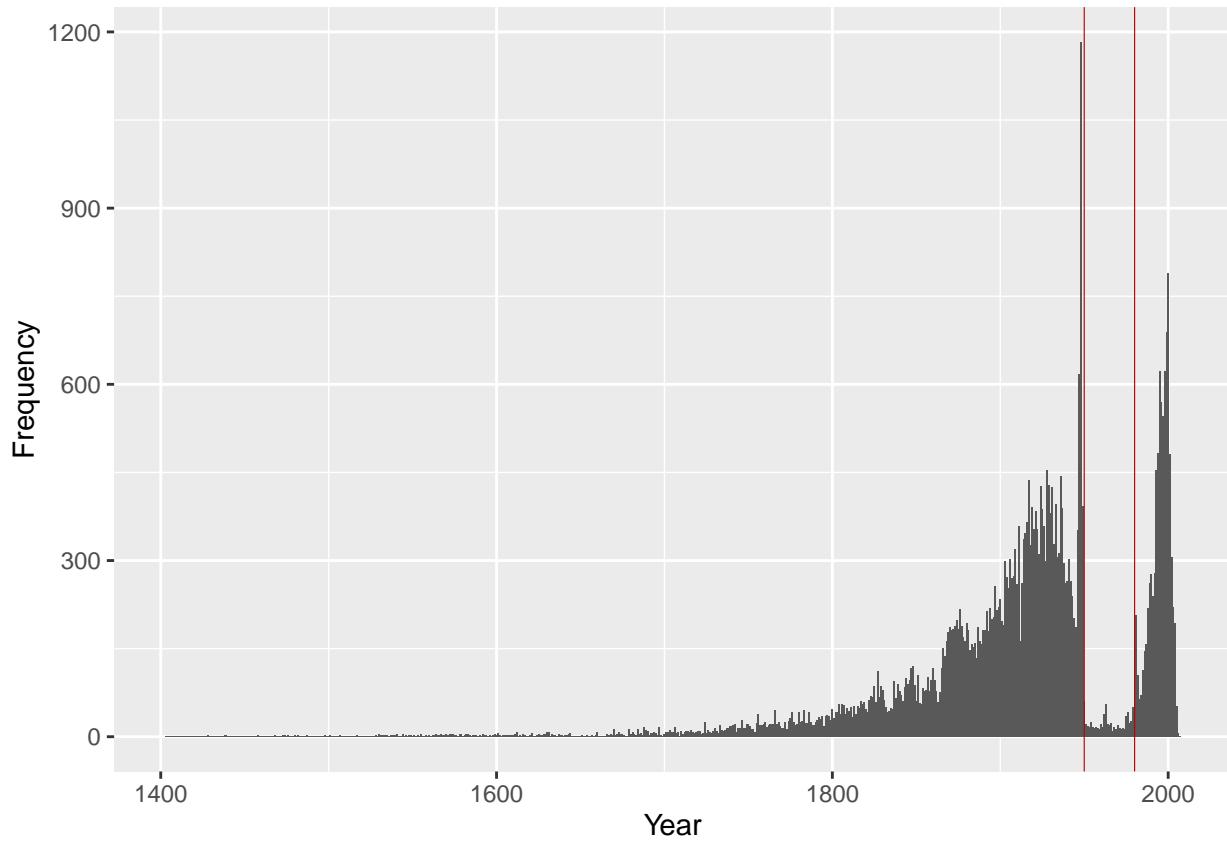
```
clan_gbook <- read_dta("/Users/a182501/stata-replicat/replication/datafiles/gbooks_byyear.dta")
head(clan_gbook)

## # A tibble: 6 x 2
##   year year_imp
##   <dbl>    <dbl>
## 1 970     970
## 2 1430    1430
## 3 1800    1800
## 4 1880    1880
## 5 1890    1890
## 6 1900    1900

p <- clan_gbook%>%
  dplyr::filter(year<=2010&year>=1400)%>%
  ggplot()+
  geom_histogram(aes(year_imp), binwidth = 1)

p+geom_vline(aes(xintercept=1950), colour="#BB0000", size = 0.2)+
  geom_vline(aes(xintercept=1980), colour="#BB0000", size = 0.2)+xlab("Year")+ylab("Frequency")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```



在统计上的大小与原作者给出的频率有一定的差异，不过形状是相同的。

```
library(mapchina)
library(sysfonts)
library(showtextdb)
library(showtext)
library(sf)
library(haven)
clan_distr <- read_dta("/Users/a182501/stata-replicat/replication/datafiles/clan_distr.dta")
arrange(clan_distr,provcd)
```

```
## # A tibble: 1,145 x 3
##   provcd lnzupunum50 countycode
##       <dbl>        <dbl>      <dbl>
## 1       13        0.0421       1
## 2       13        0.0000       3
## 3       13        0.0000       9
## 4       13        0.0000      18
```

```

##   5     13    0.105      23
##   6     13    0.203      26
##   7     13    0.310      42
##   8     13    0.160      44
##   9     13    0.0281     48
##  10     13     0         70
## # ... with 1,135 more rows

head(china)

## Simple feature collection with 6 features and 13 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: 115.4248 ymin: 39.44473 xmax: 116.8805 ymax: 41.05936
## Geodetic CRS: WGS 84
## # A tibble: 6 x 14
##   Code_~1 Code_~2 Code_~3 Name_~4 Name_~5 Name_~6 Pinyin Pop_2~7 Pop_2~8 Pop_2~9
##   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <dbl>   <dbl>   <dbl>
## 1 110101  1101    11     北京市  <NA>    东城区 Dōngc~  881763  919253   NA
## 2 110102  1101    11     北京市  <NA>    西城区 Xīché~ 1232823 1243315   NA
## 3 110114  1101    11     北京市  <NA>    昌平区 Chāng~  614821  1660501   NA
## 4 110115  1101    11     北京市  <NA>    大兴区 Dàxīn~  671444  1365112   NA
## 5 110111  1101    11     北京市  <NA>    房山区 Fángs~  814367  944832   NA
## 6 110116  1101    11     北京市  <NA>    怀柔区 Huáir~  296002  372887   NA
## # ... with 4 more variables: Pop_2018 <dbl>, Area <dbl>, Density <dbl>,
## #   geometry <MULTIPOLYGON [°]>, and abbreviated variable names 1: Code_County,
## #   2: Code_Perfecture, 3: Code_Province, 4: Name_Province, 5: Name_Perfecture,
## #   6: Name_County, 7: Pop_2000, 8: Pop_2010, 9: Pop_2017

```

3.5 重新再利用

尽管我们没有关于县级层面的宏观经济等控制变量，但我们可以将获得的数据反向匹配给到个人，看宗族祠堂对于个人的影响是如何呈现的。

我们这里选择使用 cfps2014 年的数据，处理方法基本上与 2010 年一致。

```

cfps2014comm <- read_dta("/Users/a182501/rproject/cfps/data/cfps/2014/cfps2014comm_201906.dta")
cfps2014comm$ca3_s_6[which(cfps2014comm$ca3_s_6==8)] <- 0

```

```
cfps2014comm$ca3_s_7[which(cfps2014comm$ca3_s_7==8)] <- 0
#View(cfps2014comm) 随时观察变量
cfps2014comm%>%
  select(cid14,cid10,ca3_s_6,ca3_s_7)%>comm14
head(comm14)
```

```
## # A tibble: 6 x 4
##   cid14     cid10    ca3_s_6    ca3_s_7
##   <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl>
## 1 118100     11810      0         0
## 2 118200     11820      0         0
## 3 212300     21230      7 [通公路] 12 [实施村/居直接选举]
## 4 209100     20910      0         0
## 5 118300     11830      0         0
## 6 118400     11840      0         0
```

调用家户数据

```
famconf14 <- read_dta("/Users/a182501/rproject/cfps/data/cfps/2014/cfps2014famconf_170630.dta")
head(famconf14)
```

```
## # A tibble: 6 x 307
##   fid14     fid12          fid10      provcd14 count~1 cid14  urban14 pid
##   <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lb> <dbl+l> <dbl+> <dbl+l> <dbl+>
## 1 100051     -8 [不适用]     -8 [不适~ 11 [北~ 45     624942 1 [城~ 1.00e8
## 2 100051     -8 [不适用]     -8 [不适~ 11 [北~ 45     624942 1 [城~ 1.00e8
## 3 100051     110043        110043     ~ 11 [北~ 45     624942 1 [城~ 1.10e8
## 4 100125     110147        110147     ~ 11 [北~ 170    564346 1 [城~ 1.10e8
## 5 100160     120009        120009     ~ 12 [天~ 79     131700 1 [城~ 1.20e8
## 6 100286     130005        130005     ~ 13 [河~ 237    161210 1 [城~ 1.30e8
## # ... with 299 more variables: code_a_p <dbl+lbl>, tb2_a_p <dbl+lbl>,
## #   tb1y_a_p <dbl+lbl>, tb1m_a_p <dbl+lbl>, tb1a_a_p <dbl+lbl>,
## #   tb3_a14_p <dbl+lbl>, tb4_a14_p <dbl+lbl>, alive_a14_p <dbl+lbl>,
## #   ta4y_a14_p <dbl+lbl>, ta4m_a14_p <dbl+lbl>, ta401_a14_p <chr>,
## #   qa301_a14_p <dbl+lbl>, qa302_a14_p <dbl+lbl>, tb6_a14_p <dbl+lbl>,
## #   tb601_a14_p <dbl+lbl>, co_a14_p <dbl+lbl>, outpers_where14_p <dbl+lbl>,
## #   tb602acode_a14_p <dbl+lbl>, cfps2014_interv_p <dbl+lbl>, ...
```

使用左连接 `left_join` 以保留我们的家户信息，用村居样本代码 `cid14` 来进行匹配。

```
library(visdat)

famcon14_clan <- left_join(famconf14,comm14,by="cid14")
dim(famcon14_clan)

## [1] 57734    310

dim(famconf14)

## [1] 57734    307

#View(famcon14_clan)
famcon14_clan %>%
  select(cid14,cid10,ca3_s_6,ca3_s_7,tb2_a_p,tb1y_a_p,cfps2014_interv_p,tb4_a14_p,urban14,tb4_a14_f)%
  dplyr::filter(urban14==0)%>%
  dplyr::filter(tb4_a14_p!=8)%>%
  dplyr::filter(tb4_a14_p!=9)%>%
  dplyr::filter(!is.na(cid10))%>%
  mutate(age=2014-tb1y_a_p)%>%

  mutate(eduyear = case_when(
    tb4_a14_p==8 ~ 23,
    tb4_a14_p==7 ~ 19,
    tb4_a14_p==6 ~ 16,
    tb4_a14_p==5 ~ 15,
    tb4_a14_p==4 ~ 12,
    tb4_a14_p==3 ~ 9,
    tb4_a14_p==2 ~ 6,
    tb4_a14_p==1 ~ 0
  ))%>%
  mutate(temple_dummy = case_when(
    ca3_s_6==0~0,
    ca3_s_6>0~1
  ))%>%
  mutate(genealogy_dummy = case_when(
    ca3_s_7 == 0~0,
    ca3_s_7 > 0~1
  ))%>%
```

```

mutate(eduyear_fa = case_when(
  tb4_a14_f==8 ~ 23,
  tb4_a14_f==7 ~ 19,
  tb4_a14_f==6 ~ 16,
  tb4_a14_f==5 ~ 15,
  tb4_a14_f==4 ~ 12,
  tb4_a14_f==3 ~ 9,
  tb4_a14_f==2 ~ 6,
  tb4_a14_f==1 ~ 0
))-> facon14_clan_clean

dim(facon14_clan_clean)

## [1] 29190     15

sum(table(facon14_clan_clean$eduyear))

## [1] 27780

table(facon14_clan_clean$eduyear_fa)

## 
##      0      6      9     12     15     16     19     23 
## 9139 6435 5176 1850  284   109    4     1 

table(facon14_clan_clean$temple_dummy)

## 
##      0      1 
## 28033 1157 

colnames(facon14_clan_clean)

## [1] "cid14"           "cid10"          "ca3_s_6"        
## [4] "ca3_s_7"         "tb2_a_p"        "tb1y_a_p"      
## [7] "cfps2014_interv_p" "tb4_a14_p"    "urban14"      
## [10] "tb4_a14_f"       "age"            "eduyear"      
## [13] "temple_dummy"    "genealogy_dummy" "eduyear_fa"

```

```

reg_temple <- lm(data = facon14_clan_clean, eduyear~temple_dummy+age+tb2_a_p) # 控制母亲的受教育水平/父亲
summary(reg_temple)

## 
## Call:
## lm(formula = eduyear ~ temple_dummy + age + tb2_a_p, data = facon14_clan_clean)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -5.9472 -4.8059  0.3229  3.3196 14.2842 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.6291262  0.0459993 100.635 <2e-16 ***
## temple_dummy -0.2351321  0.1448925 -1.623   0.105    
## age          0.0033346  0.0003746  8.903 <2e-16 ***  
## tb2_a_p      0.9879641  0.0534791 18.474 <2e-16 ***  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.714 on 27776 degrees of freedom
##   (1410 observations deleted due to missingness)
## Multiple R-squared:  0.0123, Adjusted R-squared:  0.0122 
## F-statistic: 115.3 on 3 and 27776 DF,  p-value: < 2.2e-16

reg_genealogy <- lm(data = facon14_clan_clean, eduyear~genealogy_dummy+age+tb2_a_p)
summary(reg_genealogy)

## 
## Call:
## lm(formula = eduyear ~ genealogy_dummy + age + tb2_a_p, data = facon14_clan_clean)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -5.9464 -4.8052  0.3202  3.3202 14.2848 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 4.6291262  0.0459993 100.635 <2e-16 ***
## genealogy_dummy -0.2351321  0.1448925 -1.623   0.105    
## age          0.0033346  0.0003746  8.903 <2e-16 ***  
## tb2_a_p      0.9879641  0.0534791 18.474 <2e-16 ***  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
```

```

## (Intercept) 4.6285474 0.0459432 100.745 <2e-16 ***
## genealogy_dummy -0.2627547 0.1583737 -1.659 0.0971 .
## age 0.0033325 0.0003746 8.897 <2e-16 ***
## tb2_a_p 0.9879593 0.0534790 18.474 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.714 on 27776 degrees of freedom
## (1410 observations deleted due to missingness)
## Multiple R-squared: 0.01231, Adjusted R-squared: 0.0122
## F-statistic: 115.4 on 3 and 27776 DF, p-value: < 2.2e-16

```

还是对族谱的回归会微弱显著，不过都是负的系数，和一些学者之前研究的结论有一些差别，但可能是在这里控制变量控制的不够，可能存在内生性问题，比如遗漏一些关键的控制变量，受访者的智力水平、家庭规模，父亲的政治背景、教育理念、文化资本等。

不过个人认为这里可以使用族谱的数量，并不需要将其转变为虚拟变量。

```

reg_genealogy_cont <- lm(data = facon14_clan_clean, eduyear ~ ca3_s_7 + age + tb2_a_p)
summary(reg_genealogy_cont)

```

```

##
## Call:
## lm(formula = eduyear ~ ca3_s_7 + age + tb2_a_p, data = facon14_clan_clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.9476 -4.8062  0.3191  3.3191 14.2838
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.6294935 0.0459205 100.815 <2e-16 ***
## ca3_s_7     -0.0323995 0.0169295 -1.914 0.0557 .
## age         0.0033338 0.0003746  8.901 <2e-16 ***
## tb2_a_p     0.9880811 0.0534782 18.476 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.713 on 27776 degrees of freedom

```

```
## (1410 observations deleted due to missingness)
## Multiple R-squared:  0.01234,    Adjusted R-squared:  0.01223
## F-statistic: 115.7 on 3 and 27776 DF,  p-value: < 2.2e-16
```

Chapter 4

CHARLS

CHARLS是中国中老年人调查数据，由北大发起的关于中国中老年人的社会调查。

4.1 数据导入

```
library(haven)

getwd()

## [1] "/Users/a182501/rproject/rrp"

charls2018cogn <- read_dta("data/charls/2018/Cognition.dta")
head(charls2018cogn)

## # A tibble: 6 x 219
##   ID      house~1 commu~2 dc001~3 dc002~4 dc003~5 dc005~6 dc006~7 dc007~8 dc008~9
##   <chr>    <chr>    <chr>    <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl> <dbl+lbl>
## 1 09400~ 094004~ 0940041 1 [1 C~ 1 [1 C~ 5 [5 E~ 1 [1 C~ 1 [1 C~ 1 [1 C~ 1 [1 C~
## 2 09400~ 094004~ 0940041 1 [1 C~ 1 [1 C~
## 3 09400~ 094004~ 0940041 1 [1 C~ 1 [1 C~
## 4 09400~ 094004~ 0940041 1 [1 C~ 1 [1 C~
## 5 09400~ 094004~ 0940041 1 [1 C~ 1 [1 C~ 5 [5 E~ 1 [1 C~ 1 [1 C~ 1 [1 C~ 1 [1 C~
## 6 09400~ 094004~ 0940041 1 [1 C~ 1 [1 C~
## # ... with 209 more variables: dc009_w4 <dbl+lbl>, dc010_w4 <dbl+lbl>,
```

```
## #  dc012_w4 <dbl+lbl>, dc004 <dbl+lbl>, dc013_w4_1_s1 <dbl+lbl>,
## #  dc013_w4_1_s2 <dbl+lbl>, dc013_w4_1_s3 <dbl+lbl>, dc013_w4_1_s4 <dbl+lbl>,
## #  dc013_w4_1_s97 <dbl+lbl>, dc013_w4_2_s1 <dbl+lbl>, dc013_w4_2_s2 <dbl+lbl>,
## #  dc013_w4_2_s3 <dbl+lbl>, dc013_w4_2_s4 <dbl+lbl>, dc013_w4_2_s97 <dbl+lbl>,
## #  dc013_w4_3_s1 <dbl+lbl>, dc013_w4_3_s2 <dbl+lbl>, dc013_w4_3_s3 <dbl+lbl>,
## #  dc013_w4_3_s4 <dbl+lbl>, dc013_w4_3_s97 <dbl+lbl>, ...

library(purrr)
get_var_label <- function(dta) {
  labels <- map(dta, function(x) attr(x, "label"))
  data_frame(
    name = names(labels),
    label = as.character(labels)
  )
}
```

Chapter 5

文献复刻：《新型农村社会养老保险政策效果评估》

这篇文章是使用断点回归和 DID 的方法，

实际上是利用领取养老金的年龄规则，只有年满 60 周岁的参保人员才能领取

因变量：家户总收入、家户人均收入、个人收入、个人非劳动收入；

5.1 数据导入

Chapter 6

CHFS

CHFS是西南财经大学组织的中国家庭金融调查。中国家庭金融调查采用三阶段、分层、与人口规模成比例 (PPS) 的抽样方法，通过科学抽样、现代调查技术和调查管理手段，收集中国家庭金融微观信息，为国内外研究者提供研究中国家庭金融问题的高质量微观数据。CHFS 样本覆盖全国 29 个省，262 个县，总共包含 28000 多户家庭的资产负债、收入与支出、保险与保障，家庭人口特征及就业等方面详细信息的大型微观数据。

6.1 数据读取

从其官网下载 2019 年的调查数据

```
library(haven)
chfs_hh2019 <- read_dta("data/chfs/CHFS_2019/chfs2019_hh_202112.dta")

library(purrr)
get_var_label <- function(dta) {
  labels <- map(dta, function(x) attr(x, "label"))
  tibble(
    name = names(labels),
    label = as.character(labels)
  )
}
```

6.2 构建变量

6.2.1 社会互动

- Du et al. (2014)
- 社会互动相关的礼金支出、外出就餐支出、娱乐支出、通讯支出、交通支出、旅游探亲支出、兄弟姐妹数量、与父母通话次数 8 个变量。

6.2.2 金融知识

在 CHFS 中第五部分就是关于金融知识、底层治理与主观评价。

2. 金融知识的度量

沿用以往文献的做法,在“利率计算、通货膨胀理解和投资风险”三个衡量受访者的金融知识水平的提取变量的基础上(Rooij et al. , 2011; Lusardi and Mitchell, 2011; 尹志超等,2014),我们增加“一般性金融知识”、“专业性金融知识”和“投资风险计算”三个问题变量,选择 CHFS 问卷中“是否关注金融、经济信息”代表一般性金融知识,“是否上过金融课程”代表专业性金融知识(Romer, 1986),“投资风险是否计算正确”衡量家庭投资风险的计算能力(Lusardi, 2012),连同利率计算、通货膨胀预期和投资风险选择共六个问题^③,考虑金融知识指标提取变量都是哑变量的形式,我们采取因子分析中的极大似然法提取知识因子、计算因子和预期因子。

```
chfs_hh2013 <- read_dta("data/chfs/CHFS_2013/chfs2013_hh_20191120_version14.dta")
chfs_hh2013 %>% get_var_label()
```

```
## # A tibble: 2,184 x 2
##   name      label
##   <chr>     <chr>
## 1 hhid_2011 household id in 2011
## 2 hhid_2013 household id in 2013
## 3 a2000a    居住在一起家庭成员个数
## 4 a2000b    外出家庭成员个数
## 5 a1001     地址是否正确
## 6 a1002     是否居民住宅
## 7 a1003     住宅户数
## 8 a1007     主要经济活动
```

```
## 9 a1008      是否常住本市
## 10 a1009     有无外国国籍
## # ... with 2,174 more rows

dim(chfs_hh2013)

## [1] 28141 2184

chfs_hh2013%>%
  select(starts_with("a400"))->df2_a
df2_na <- na.omit(df2_a)
dim(df2_na)

## [1] 26922      7

df2_na%>%
  get_var_label()

## # A tibble: 7 x 2
##   name    label
##   <chr>   <chr>
## 1 a4002a  对经济、金融信息的关注度
## 2 a4002b  上学期间是否上过经济、金融课程
## 3 a4003  投资风险态度
## 4 a4004a  100元存5年的本息和
## 5 a4005a  100元存款一年后的购买力
## 6 a4006a  100%得4000和50%得10000的偏好
## 7 a4007aa 单买一只股票和买一只股票基金的风险
```

参考 张号栋 and 尹志超 (2016) 的做法：

③ CHFS 问卷中涉及到的 6 个问题分别是【问题一】是“您平时对经济、金融方面的信息关注程度如何？1. 非常关注 2. 很关注 3. 一般 4. 很好关注 5. 从不关注”，我们将“非常关注、很关注和一般”赋值为 1，否则赋值为 0。【问题二】是“在您上学期间，是否上过经济或金融类的课程？1. 是 2. 否”，我们选择“是”赋值为 1，否则为 0。【问题三】是“您认为一般而言，单独买一只公司的股票是否比买一只股票基金风险更大？1. 是 2. 否 3. 没有听过股票 4. 没有听说过股票基金 5. 两者都没有听说过”，我们选择“是”赋值为 1，否则为 0。【问题四】是“假设您现在有 100 块钱，银行的年利率是 4%，如果您把这 100 元钱存 5 年定期，5 年后您获得的本金和利息为？1. 小于 120 元 2. 等于 120 元 3. 大于 120 元 4. 算不出来”，我们选择“等于 120 元”赋值为 1，否则为 0。【问题五】是“如果您有两张彩票供您选择，若您选择第一张，您有 100% 的机会获得 4000 元；若您选择第二张，您有 50% 的机会获得 10000 元，50% 的机会什么也没有，您愿选哪张？1. 第一张 2. 第二张”，我们选择“第二张”赋值为 1，否则为 0。【问题六】是“假设您现在有 100 块钱，银行的年利率是每年 5%，通货膨胀率是 3%，您的这 100 元钱存银行一年以后能够买到的东西将？1. 比一年前多 2. 跟一年前一样多 3. 比一年前少 4. 算不出来”，我们选择“比一年前多”赋值为 1，否则为 0。

```
df2_na%>%
  mutate(fa1=case_when(
    a4002a%in%c(1,2,3)~1,
    a4002a%in%c(4,5)~0
  ))%>%
  mutate(fa2 = case_when(
    a4002b==1~1,
    a4002b==2~0
  ))%>%
  mutate(fa3 = case_when(
    a4007aa==1~1,
    a4007aa!=1~0
  ))%>%
  mutate(fa4=case_when(
    a4004a==2~1,
    a4004a!=2~0
  ))%>%
  mutate(fa5=case_when(
    a4006a==2~1,
    a4006a==1~0
  ))%>%
  mutate(fa6=case_when(
    a4005a==1~1,
    a4005a!=1~0
  ))%>%
  select(starts_with("fa"))->df_fa
```

```
library(visdat)
df_fa %>%
  vis_dat()
```



```
mean_fa <- map_dbl(df_fa, mean)
mean_fa
```

```
##      fa1      fa2      fa3      fa4      fa5      fa6
## 0.37118342 0.08160612 0.30740658 0.15474333 0.26799643 0.16072357
```

得到的均值水平会比原文章都普遍较高一些。但因素 5 的偏差较大。（但本人又认为这个变量不太好，更像是一个风险测试的问题，与受访者的风险偏好有关，并没有一个所谓正确的答案）

```
1-mean_fa[5]
```

```
##      fa5
## 0.7320036
```

6.2.2.1 因子转换

```
library(psych) #KMO 和 Bartlette 检验所需包

## 
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
## 
##     %+%, alpha
```

```
KMO(df_fa)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = df_fa)
## Overall MSA = 0.65
## MSA for each item =
## fa1  fa2  fa3  fa4  fa5  fa6
## 0.64 0.64 0.66 0.69 0.70 0.70
```

都大于 0.6，勉强适合。

```
bartlett.test(df_fa)

## 
## Bartlett test of homogeneity of variances
## 
## data: df_fa
## Bartlett's K-squared = 10938, df = 5, p-value < 2.2e-16
```

p 值非常小，也验证可以做因子。

```
corr=cor(df_fa)
eig=eigen(corr)
(cx=(eig$va)/sum(eig$va)) # 贡献率

## [1] 0.2677825 0.1643641 0.1626099 0.1501068 0.1306874 0.1244494
```

```
(cx=cumsum(eig$va)/sum(eig$va)) # 累计方差贡献率

## [1] 0.2677825 0.4321466 0.5947564 0.7448632 0.8755506 1.0000000

fit <- factanal(df_fa, 3, rotation="promax") # 第 2 个参数是提取的因子个数
print(fit, digits=2, sort=TRUE) # 输出结果

##
## Call:
## factanal(x = df_fa, factors = 3, rotation = "promax")
##
## Uniquenesses:
## fa1 fa2 fa3 fa4 fa5 fa6
## 0.74 0.75 0.80 0.00 0.95 0.92
##
## Loadings:
##          Factor1 Factor2 Factor3
## fa4    1.00
## fa1      0.38    0.18
## fa2      0.49
## fa3      0.34    0.15
## fa5      0.22
## fa6      0.30
##
##          Factor1 Factor2 Factor3
## SS loadings   0.99    0.51    0.20
## Proportion Var 0.17    0.08    0.03
## Cumulative Var 0.17    0.25    0.28
##
## Factor Correlations:
##          Factor1 Factor2 Factor3
## Factor1    1.00    0.17    0.23
## Factor2    0.17    1.00    0.58
## Factor3    0.23    0.58    1.00
##
## The degrees of freedom for the model is 0 and the fit was 0
```

6.2.3 金融排斥

“是否有金融账户”来作为是否是存在金融排斥的代理变量。

[D1101] 目前，您家是否有人民币活期存款账户？

1. 有

2. 没有【跳至 D2101】

```
chfs_hh2013 %>%
  select(d1101)%>%
  dplyr::filter(!is.na(d1101))%>%
  mutate(cur= case_when(
    d1101 == 1~1,
    d1101 == 2~0
  ))->df3
mean(df3$cur)
```

```
## [1] 0.595189
```

也就是有人民币活期存款账户的有 59.51%。但根据文章中的定义：还需要对其余的金融账户进行筛选，通过逻辑运算实现。

1. 家庭金融排斥的界定

本文借鉴 Kempson and Whyley(1999) 将家庭在正规金融产品或服务方面受到约束作为金融排斥的定义，利用李涛等(2010) “是否拥有金融账户”哑变量形式度量金融排斥指标。具体而言，我们以“家庭是否具有正规金融账户”哑变量对家庭金融排斥进行度量，将家庭金融排斥分解成家庭投资类金融排斥和家庭融资类金融排斥。进一步，家庭投资类金融排斥细分为活期存款排斥、定期存款排斥、股票排斥、基金排斥、债券排斥、银行理财产品排斥、外汇排斥、商业保险排斥^①、金融衍生品排斥和黄金排斥；家庭融资类金融排斥细分为农业、工商业(以下简称农工商) 贷款排斥、住房贷款排斥、汽车贷款排斥、教育贷款排斥、信用卡排斥。据 CHFS 数据统计计算，我国 29. 6% 的家庭没有任何金融产品^②。

Chapter 7

其他来源数据

其他数据来源包括相关领域内的学者对于在其文章中的数据以及公开的数据集中的数据。比如哈佛大学的王教授收集了关于地方的数据

7.1 方言数据

```
library(mapchina)
library(sysfonts)
library(showtextdb)
library(showtext)
library(tidyverse)
library(sf)
```

徐现祥老师在其个人网站上公开了其地方的方言数据，也就是连享会 gitee 仓库有相关数据集合

```
library(haven)
df <- read_dta("data/China_dialect_diversity_index.dta")
head(df)
```

```
## # A tibble: 6 x 3
##   city           diversity1 diversity2
##   <dbl+lbl>      <dbl>      <dbl>
## 1 256 [阿勒泰地区]    0.650        2
## 2 75 [安康市]       0.475        2
```

```
## 3 74 [安庆市]      0.477      2
## 4 77 [安顺市]      0.570      1
## 5 76 [安阳市]      0.463      2
## 6 261 [鞍山市]     0.428      3
```

```
lab <- attributes(df$city)$labels
lab <- as.data.frame(lab)
lab_name <- rownames(lab)
lab_df <- cbind(lab, lab_name)
colnames(lab_df) <- c("city", "city_name")
df_lab <- merge(df, lab_df, by = "city")
head(df_lab)
```

```
##   city diversity1 diversity2 city_name
## 1    1      0.1071      1 七台河市
## 2    2      0.5067      1 三亞市
## 3    3      0.4763      2 三門峽市
## 4    4      0.0126      1 上海市
## 5    5      0.5792      4 上饒市
## 6    6      0.0603      1 东莞市
```

我们这里使用的是mapchina包，能够在不同的行政层级上绘制中国区划地图。

```
sf_use_s2(FALSE)
df1 <- china%>%
  dplyr::filter(is.na(china>Name_Perfecture))%>%
  mutate(Name_Perfecture=Name_Province)
```

```
df2 <- china%>%
  dplyr::filter(is.na(china>Name_Perfecture)==F)
```

```
df3 <- rbind(df1, df2)
dim(df3)
```

```
## [1] 2901 14
```

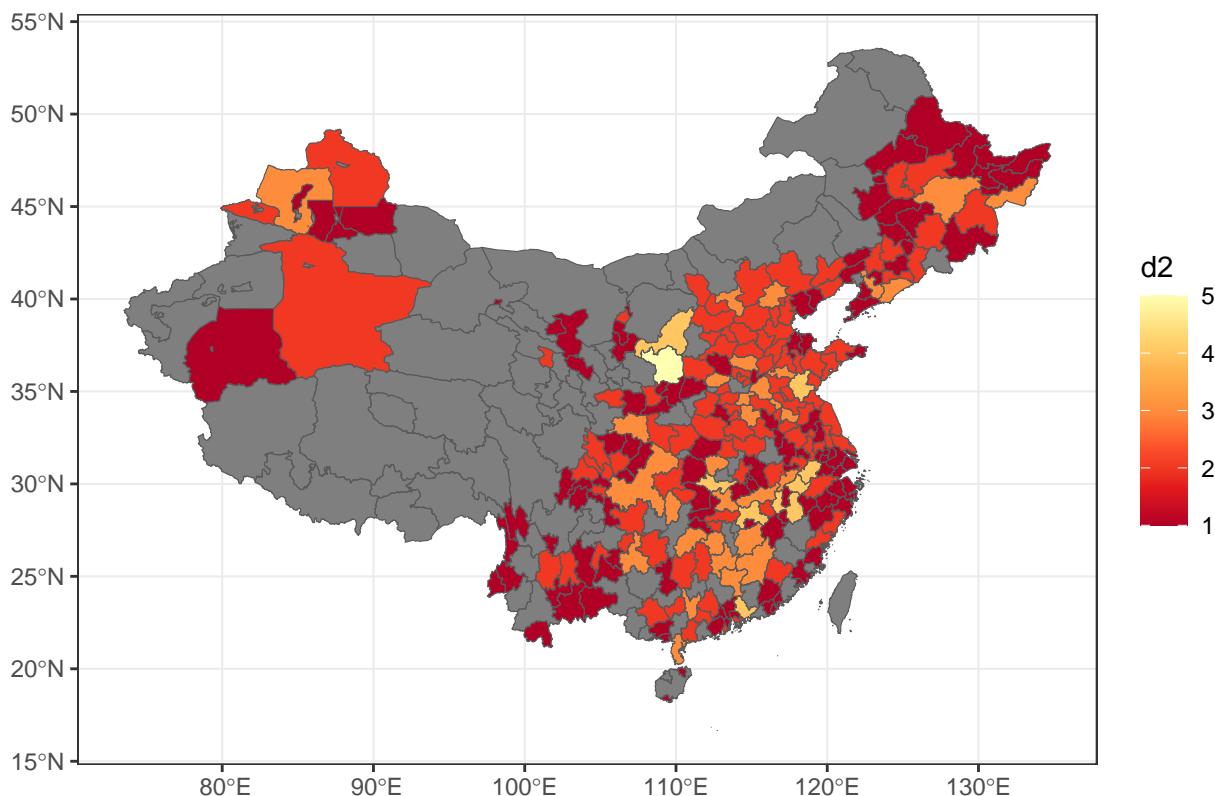
```
dim(china)
```

```
## [1] 2901 14
```

```
df3_perf <- df3 %>%
  group_by(Name_Perfecture) %>%
  summarise(geometry = st_union(geometry))
colnames(df_lab) <- c("city", "d1", "d2", "Name_Perfecture")
df_all <- left_join(df3_perf, df_lab, by = "Name_Perfecture")
```

地区的不同方言类别:

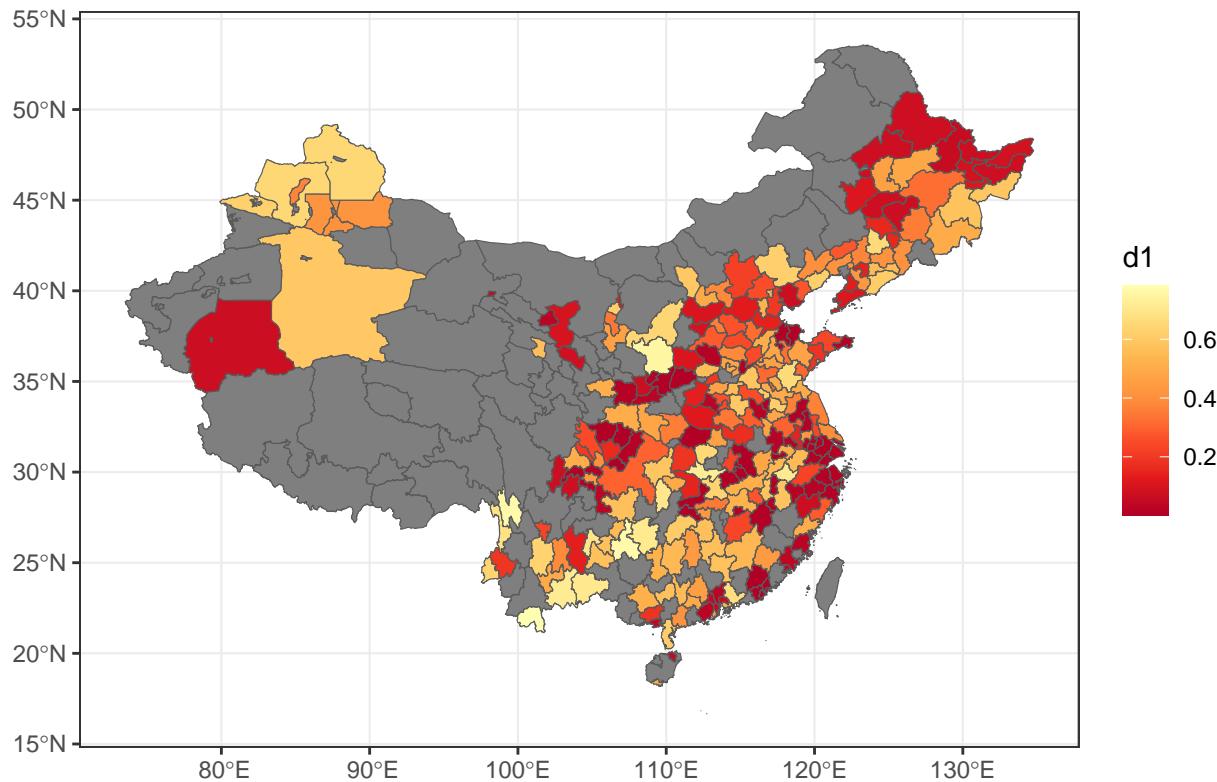
```
ggplot(data = df_all) +
  geom_sf(aes(fill = d2)) +
  scale_fill_distiller(palette = "YlOrRd") +
  theme_bw()
```



地方方言的多样性指数:

```
ggplot(data = df_all) +
  geom_sf(aes(fill = d1)) +
```

```
scale_fill_distiller(palette = "YlOrRd")+
theme_bw()
```



7.2 夜间灯光数据

我们想要复刻一下上面的案例，但缺乏城市 GDP 数据，但我又懒于去用学校 wind 或数据网站上爬。就想要看下国内有没有可以用的 API 可以获得相关数据。

7.2.1 获取方法

nightlight data

7.2.2 API

7.3 政治数据

Chapter 8

论文复刻 Climate risks and market efficiency

Hong et al. (2019) 这篇文章发表在 JoE 上，主要讲的是国家气候风险上升，使得国家的粮食价格得到下降，但并没有反映在实际的价格上，存在一定的非完全有效市场。

8.1 Data

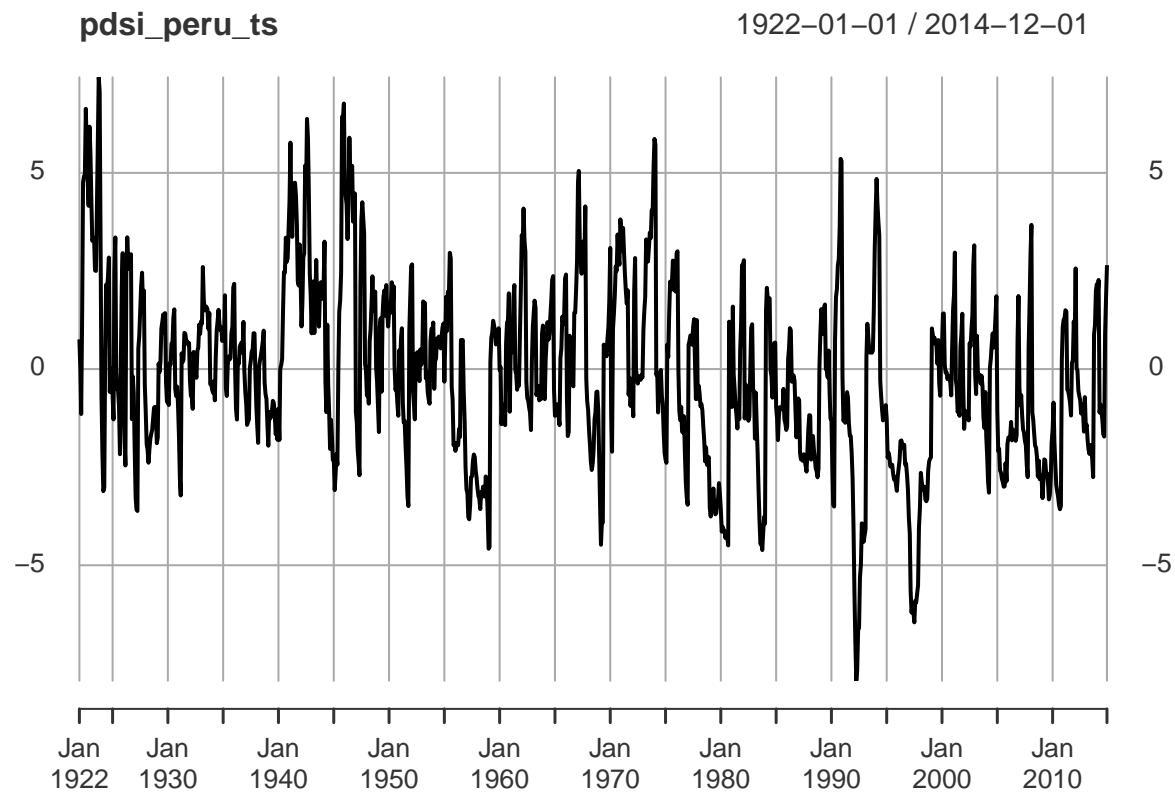
数据来自于 Harrison Hong 的homepage 同样也有 do file。

```
getwd()  
  
## [1] "/Users/a182501/rproject/rrp"  
  
library(haven)  
pdsi <- read.csv("data/honglixu_2019/PDSI_world.csv")
```

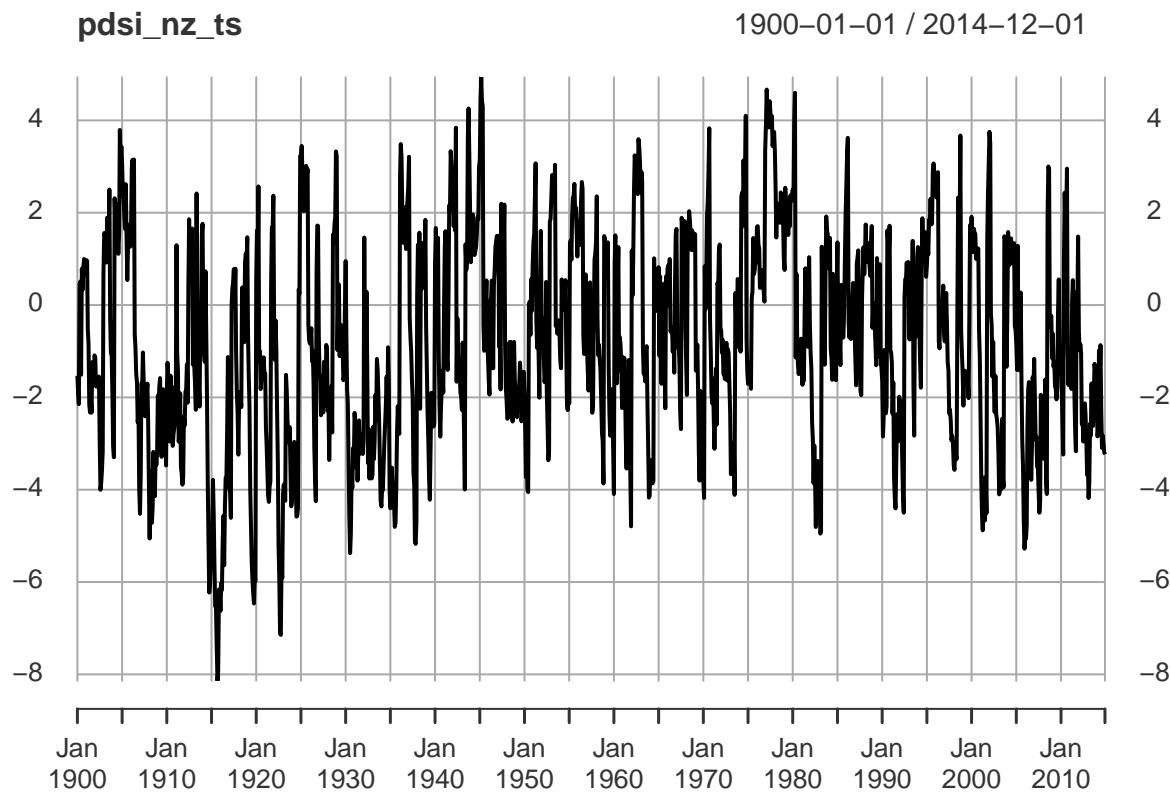
PDSI 数据也可以从NOAA中获取，不过需要注册，经过审核之后就可以免费下载。

```
pdsi%>%  
  select(Date,Peru)%>%  
  dplyr::filter(Peru!=999)->pdsi_peru  
library(xts)  
  
## Loading required package: zoo
```

```
##  
## Attaching package: 'zoo'  
  
## The following objects are masked from 'package:base':  
##  
##     as.Date, as.Date.numeric  
  
##  
## ##### WARNING #####  
## # We noticed you have dplyr installed. The dplyr lag() function breaks how #  
## # base R's lag() function is supposed to work, which breaks lag(my_xts). #  
## #  
## # Calls to lag(my_xts) that you enter or source() into this session won't #  
## # work correctly. #  
## #  
## # All package code is unaffected because it is protected by the R namespace #  
## # mechanism. #  
## #  
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning. #  
## #  
## # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #  
## # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #  
## # dplyr from breaking base R's lag() function. #  
## ##### WARNING #####  
  
##  
## Attaching package: 'xts'  
  
## The following objects are masked from 'package:dplyr':  
##  
##     first, last  
  
pdsi_peru_ts <- as.xts(pdsi_peru$Peru, as.Date(pdsi_peru$date))  
plot(pdsi_peru_ts)
```



```
pdsi%>%
  select(Date,New_Zealand)%>%
  dplyr::filter(New_Zealand!=999)->pdsi_nz
pdsi_nz_ts <- as.xts(pdsi_nz$New_Zealand,as.Date(pdsi_nz>Date))
plot(pdsi_nz_ts)
```



从数据中，我们可以看出有一定的趋势，尽管可能并不明显。我们就可以从试图去构建一个含有线性时间固定趋势的模型。

```
tt <- seq(length(pdsi_nz_ts))
reslm <- lm(c(pdsi_nz_ts) ~ tt); summary(reslm)

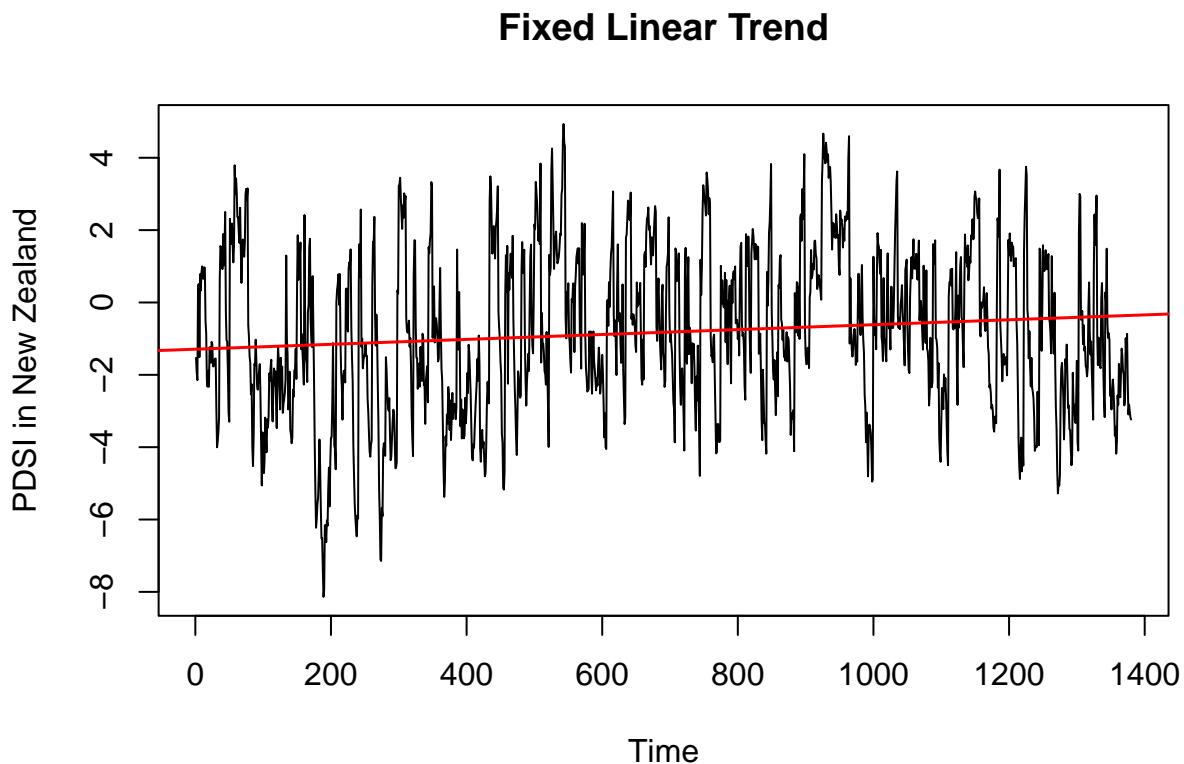
##
## Call:
## lm(formula = c(pdsi_nz_ts) ~ tt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -6.9739 -1.5492 -0.1941  1.7932  5.8579 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.2921454  0.1212183 -10.660 < 2e-16 ***
## tt          0.0006777  0.0001521    4.457 8.98e-06 ***
```

```

## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.25 on 1378 degrees of freedom
## Multiple R-squared: 0.01421, Adjusted R-squared: 0.0135
## F-statistic: 19.87 on 1 and 1378 DF, p-value: 8.982e-06

plot(tt, pdsi_nz_ts, type="l", xlab="Time", ylab="PDSI in New Zealand", main="Fixed Linear Trend")
abline(reslm, lwd=1.5, col="red")

```



但固定趋势的模型可能并不精确，比如上述的线性固定趋势模型的 R 方就较小，拟合性较差。因此在文章中就使用每次的来 b_{it} 来构建。

第一个回归方程就是想要先构建一个趋势 ($Trend_{it}$) 变量，通过 AR(1) 模型来实现

We have tried different specifications of estimating PDSI time trends including: (1) PDSI time trend estimated with the lagged PDSI and (2) PDSI time trend estimated with lagged PDSI and month dummies. The correlation between the baseline PDSI trend with these alternative measures of PDSI trends are very high, and our subsequent results are robust.

后面的数据都是用 1984-2019 年的。

```
end_date <- which(pdsi$date=="2014/12/1")
begin_date <- which(pdsi$date=="1984/1/1")
pdsi_1984 <- pdsi[begin_date:end_date,]
dim(pdsi_1984)
```

```
## [1] 372 45
```

```
#12*(2014-1984)+12
```

$$PDSI_{i,t} = a_i + b_i t + c_i PDSI_{i,t-1} + \epsilon_{i,t} \quad (1)$$

其中的 b_i 就是希望捕捉的趋势变量，每到一个 m 进行一次 rolling，可以获得从 1900（或者更早）到 m 时期的 $Trend_{i,m}$

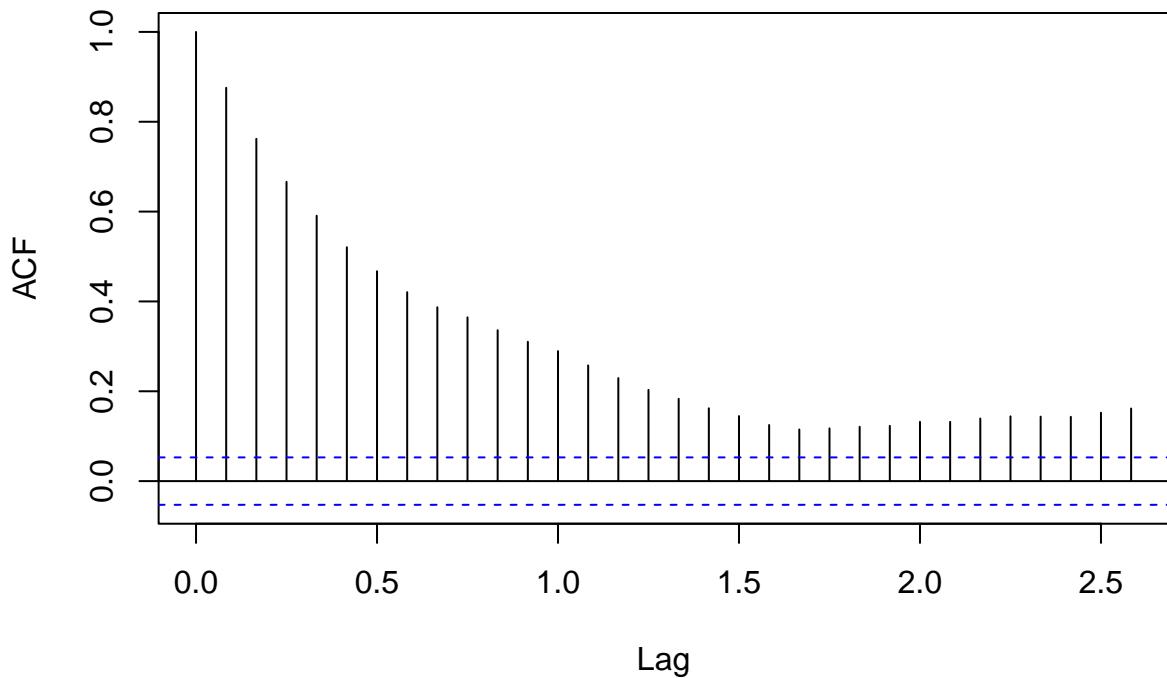
作者还将滞后 2, 3 期的回归附录放在Dropbox

```
pdsi%>%
  select(Date,Australia)%>%
  dplyr::filter(Australia!=999)->pdsi_au
head(pdsi_au)
```

```
##           Date   Australia
## 1 1900/1/1 -3.31573868
## 2 1900/2/1 -4.25730371
## 3 1900/3/1 -0.01355382
## 4 1900/4/1  1.12726247
## 5 1900/5/1  2.32717538
## 6 1900/6/1  2.16344261
```

```
au_ts <- ts(pdsi_au$Australia,
             start=c(1900,1), frequency=12)

acf(au_ts, main="")
```



```

ar(au_ts, method="mle")

##
## Call:
## ar(x = au_ts, method = "mle")
##
## Coefficients:
##       1
## 0.8759
##
## Order selected 1  sigma^2 estimated as  0.9384

```

对于澳大利亚的数据给出的 AR(1) 模型。

1 的可以转换为 $Y_t = a + b * t + \epsilon_t$, where $\epsilon_t = \phi\epsilon_{t-1} + \gamma_t$ a AR(1) process, where ϵ_t is a white noise.

数据中有 31 个国家, 也就是要做 31 次回归。。。太累了。

Chapter 9

文献复刻：The Long-term Effects of Africa's Slave Trades

Nunn (2008) 这篇文献可称为是在学习 IV 时候的经典文献，2008 年发表在 QJE。

9.1 文献回顾

9.2 数据来源

```
if (!require("pacman")) install.packages("pacman")

## Loading required package: pacman

pacman::p_load(
  sf, # vector data operations
  tidyverse, # data wrangling
  units,
  rmapshaper,
  lwgeom,
  tictoc,
  haven
)
```

```

#--- coast line ---#
coast <-
  sf::st_read(here::here("data/nunn_2008/input/10m-coastline/10m_coastline.shp")) %>%
  st_transform(3857)

## Reading layer `10m_coastline` from data source
##   `/Users/a182501/rproject/rrp/data/nunn_2008/input/10m-coastline/10m_coastline.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 4177 features and 3 fields
## Geometry type: MULTILINESTRING
## Dimension:      XY
## Bounding box:  xmin: -180 ymin: -85.22198 xmax: 180 ymax: 83.6341
## Geodetic CRS:  WGS 84

#--- African countries ---#
countries <-
  sf::st_read(here::here("data/nunn_2008/input/gadm36_africa/gadm36_africa.shp")) %>%
  st_transform(3857)

## Reading layer `gadm36_africa` from data source
##   `/Users/a182501/rproject/rrp/data/nunn_2008/input/gadm36_africa/gadm36_africa.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 54 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:  xmin: -25.3618 ymin: -34.83514 xmax: 63.50347 ymax: 37.55986
## Geodetic CRS:  WGS 84

#--- ethnic regions ---#
ethnic_regions <-
  sf::st_read(here::here("data/nunn_2008/input/Murdock_shapefile/borders_tribes.shp")) %>%
  st_transform(3857)

## Reading layer `borders_tribes` from data source
##   `/Users/a182501/rproject/rrp/data/nunn_2008/input/Murdock_shapefile/borders_tribes.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 843 features and 4 fields
## Geometry type: MULTIPOLYGON

```

```
## Dimension:      XY
## Bounding box:  xmin: -25.35875 ymin: -34.82223 xmax: 63.50018 ymax: 37.53944
## Geodetic CRS:  WGS 84

# lat/long for slave trade centers
trade_centers <- readxl::read_xls(here::here("data/nunn_2008/input/nunn2008.xls"))
```

9.3 计算最近的贸易距离

```
countries_simp <- rmapshaper::ms_simplify(countries)
```

```
(

g_countries <-
  ggplot(data = countries_simp) +
  geom_sf() +
  theme_void()
)
```

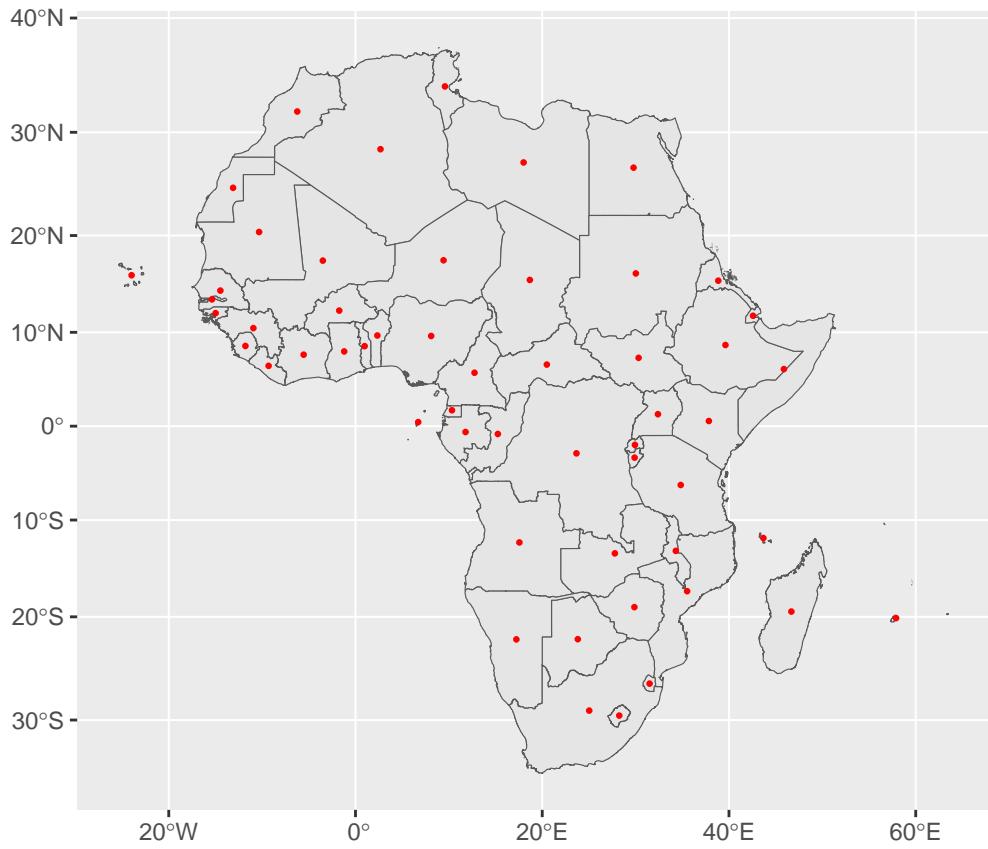


用 `st_centroid()` 来发现每一个国家的质心.

```
countries_centroid <- st_centroid(countries)

## Warning in st_centroid.sf(countries): st_centroid assumes attributes are
## constant over geometries of x

ggplot()+
  geom_sf(data = countries_simp)+
  geom_sf(data = countries_centroid,color='red',size =0.5)
```



```
(  
  coast_union <- st_union(coast)  
)  
  
## Geometry set for 1 feature  
## Geometry type: MULTILINESTRING  
## Dimension: XY  
## Bounding box: xmin: -20037510 ymin: -20261860 xmax: 20037510 ymax: 18428920  
## Projected CRS: WGS 84 / Pseudo-Mercator  
  
## MULTILINESTRING ((4926419 -2317631, 4926038 -23...
```

```
minum_dist_to_coast <- st_nearest_points(countries_centroid, coast_union)
```

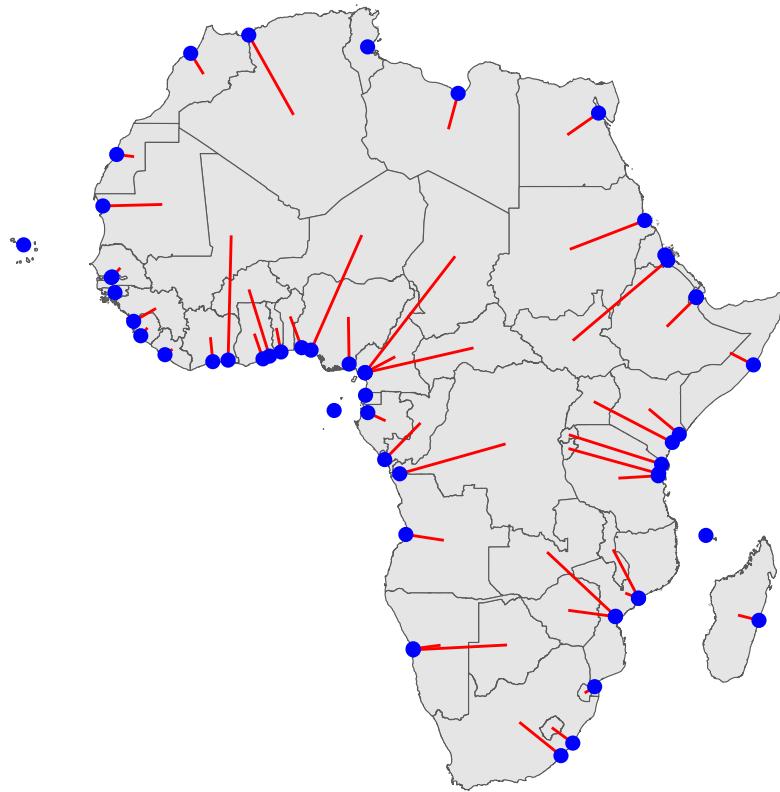
```
(  
  g_min_dist_line <-  
    ggplot() +
```

```
geom_sf(data = countries_simp) +  
  geom_sf(data = minum_dist_to_coast, color = "red") +  
  theme_void()  
)
```



```
closest_pt_on_coast <- lwgeom::st_endpoint(minum_dist_to_coast)
```

```
g_min_dist_line +  
  geom_sf(  
    data = closest_pt_on_coast,  
    color = "blue",  
    size = 2  
) +  
  theme_void()
```



```
countries_simp$nearest_pt <- closest_pt_on_coast
```

```
(  
  trade_centers_sf <-  
    trade_centers %>%  
    st_as_sf(coords = c("lon", "lat"), crs = 4326) %>%  
    st_transform(crs = 3857)  
)  
  
## Simple feature collection with 9 features and 2 fields  
## Geometry type: POINT  
## Dimension: XY  
## Bounding box: xmin: -9168273 ymin: -2617513 xmax: -4288027 ymax: 4418219  
## Projected CRS: WGS 84 / Pseudo-Mercator  
## # A tibble: 9 x 3  
##   name      fallingrain_name      geometry  
## * <chr>      <chr>          <POINT [m]>
```

```
## 1 Virginia      Virginia Beach      (-8458055 4418219)
## 2 Havana        Habana            (-9168273 2647748)
## 3 Haiti          Port au Prince    (-8051739 2100853)
## 4 Kingston       Kingston         (-8549337 2037549)
## 5 Dominica       Roseau           (-6835017 1723798)
## 6 Martinique     Fort-de-France   (-6799394 1644295)
## 7 Guyana         Georgetown      (-6473228 758755.9)
## 8 Salvador        Salvador da Bahia (-4288027 -1457447)
## 9 Rio de Janeiro Rio              (-4817908 -2617513)
```

```
ggplot() +
  geom_sf(data = trade_centers_sf, color = "red") +
  geom_sf(data = countries_simp, aes(geometry = geometry)) +
  theme_void()
```



Chapter 10

后记

在写的过程中，逐渐感受到的一大趋势是：编程对于经济学来说是一项重要的方法，或者说，编程语言已经在当下逐渐变得越来越容易上手，使得不同行业的人群开始使用编程，来规范化行业内部的标准。目前随着 chatgbt 的盛行和快速发展，编程的门槛又进一步降低。

但另一方面，又存在一定的担忧，这些可能简单的编程工作若不上手，把手浸入脏水中，很难摸出大鱼。我的意思是，数据处理、文献复刻是一些非常简单的工作，但只是好高骛远的做，脑子里空转也无法得到一些有用的结论。另一个较为深刻的感触是，数据难得，论文难写。

Bibliography

- Aruoba, S. B., & Fernández-Villaverde, J. (2015). A comparison of programming languages in macroeconomics. *Journal of Economic Dynamics and Control*, 58, 265–273. <https://www.sciencedirect.com/science/article/pii/S0165188915000883>
- Cao, J., Xu, Y., & Zhang, C. (2022). Clans and calamity: How social capital saved lives during China's Great Famine. *Journal of Development Economics*, 157, 102865. <https://doi.org/10.1016/j.jdeveco.2022.102865>
- Fan, H., Li, C., Xue, C., & Yu, M. (2023). Clan culture and patterns of industrial specialization in China. *Journal of Economic Behavior & Organization*, 207, 457–478. <https://doi.org/10.1016/j.jebo.2023.01.026>
- Hong, H., Li, F. W., & Xu, J. (2019). Climate risks and market efficiency. *Journal of Econometrics*, 208(1), 265–281. Retrieved March 16, 2023, from <https://linkinghub.elsevier.com/retrieve/pii/S0304407618301817>
- Nunn, N. (2008). The Long-Term Effects of Africa's Slave Trades *. *Quarterly Journal of Economics*, 123(1), 139–176. Retrieved March 16, 2023, from <https://academic.oup.com/qje/article-lookup/doi/10.1162/qjec.2008.123.1.139>
- Zhang, C. (2020). Clans, entrepreneurship, and development of the private sector in China. *Journal of Comparative Economics*, 48(1), 100–123. <https://doi.org/10.1016/j.jce.2019.08.008>
- 张号栋 & 尹志超. (2016). 金融知识和中国家庭的金融排斥——基于 CHFS 数据的实证研究. 金融研究, (7), 80–95. <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFD&dbname=CJFDLAST2016&filename=JRYJ201607006&v=413> citations(CNKI)[3-16-2023]<北大核心, CSSCI>.
- 张川川 & 马光荣. (2017). 宗族文化、男孩偏好与女性发展. 世界经济, 40(3), 122–143. <https://doi.org/10.19985/j.cnki.cassjwe.2017.03.007>
72 citations(CNKI)[3-12-2023]<北大核心, CSSCI>.
- 张心仪的, 孙伟增, & 陈思宇. (2021). 传统宗族文化是否影响城市犯罪率? 世界经济文汇, (2), 71–87. <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFD&dbname=CJFDLAST2021&filename=SZWH202102005&v=4> citations(CNKI)[3-12-2023]<北大核心, CSSCI>.

- 潘越, 翁若宇, 纪翔阁, & 戴亦一. (2019). 宗族文化与家族企业治理的血缘情结. 管理世界, 35(7), 116-135+203-204. <https://doi.org/10.19744/j.cnki.11-1235/f.2019.0096>
136 citations(CNKI)[3-12-2023]< 北大核心, CSSCI, AMI>.
- 陈斌开 & 陈思宇. (2018). 流动的社会资本——传统宗族文化是否影响移民就业? 经济研究, 53(3), 35-49. <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFD&dbname=CJFDLAST2018&filename=JJYJ201803004&v=122> citations(CNKI)[3-14-2023]< 北大核心, CSSCI>.