Lecture II: Data Visualization
Stata Training Course 2024 Fall

George Chou

School of Public Finance and Taxation, ZUEL

November 24th, 2024

Outline

Definition

Translating info into visuals, a technique known as data visualization, is helpful for making sense of any data set.

Process Flow

1. Determine the decision you want to make.
2. Identify the metrics that inform the decision.
3. Develop the story you want to tell.
4. Select the appropriate visual.
5. Add relevant elements to the visual.
6. Clearly label and review the visual.
7. Let a non-expert review the visual.

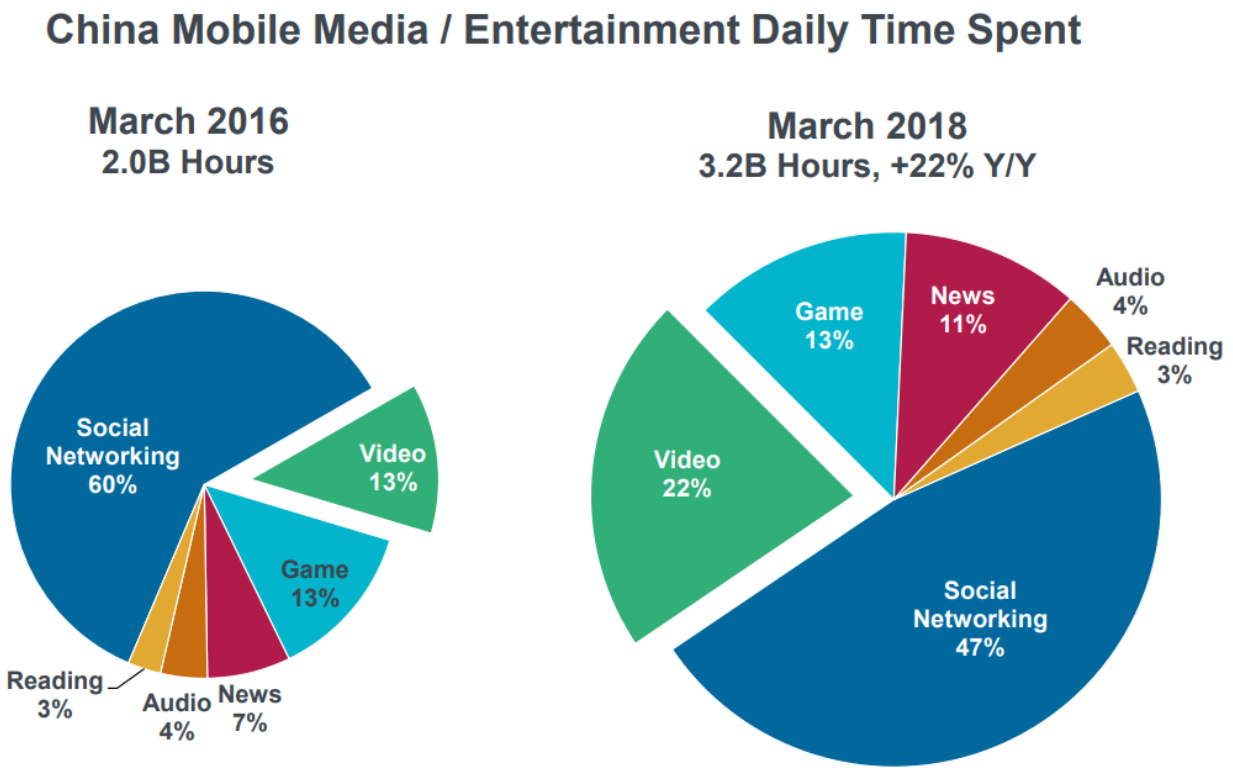
General Graph Grammar

- **graph** *graph_choice (plot_choice, plot_options) (the same...), graph_options*
- For example:

```
1 graph twoway (scatter medv nox, ///
2 ytitle(price) xtitle(pollution)), ///
3 title(A funny figure) ///
4 subtitle(Which is absolutely meaningless) ///
5 caption(Figure 1: Economic Science) ///
6 note(Notes: Don't be cheated by those people.)
```

Pie Chart

China Mobile Media / Entertainment Time Spent = +22% Y/Y…Mobile Video Growing Fastest



KLEINER PERKINS 2018 INTERNET TRENDS

Source: QuestMobile (3/18).

Hillhouse Capital

Pie Chart

- Characteristics: Display the proportion of each part to the whole, suitable for displaying categorical data.
- Applicable Data: Any data with categorical variables, such as market share, population distribution, etc.
- Basic Command: *graph pie, over(varname)*

Pie Chart

- Adding some plot options:

```
1 graph pie, over(grade) missing
2 graph pie, over(grade) noclockwise
3 graph pie, over(grade) angle0(0)
4 graph pie, over(grade) sort
5 graph pie, over(grade) sort descending
6 graph pie, over(grade) sort pie(3, explode)
```

Pie Chart

- Adding some labels:

```
1 graph pie, over(grade) plabel(_all percent)
2 graph pie, over(grade) plabel(_all sum)
3 graph pie, over(grade) plabel(_all name)
```

- Categorizing by another variable:

```
1 graph pie, over(grade) by(gender)
```

Graph Combination

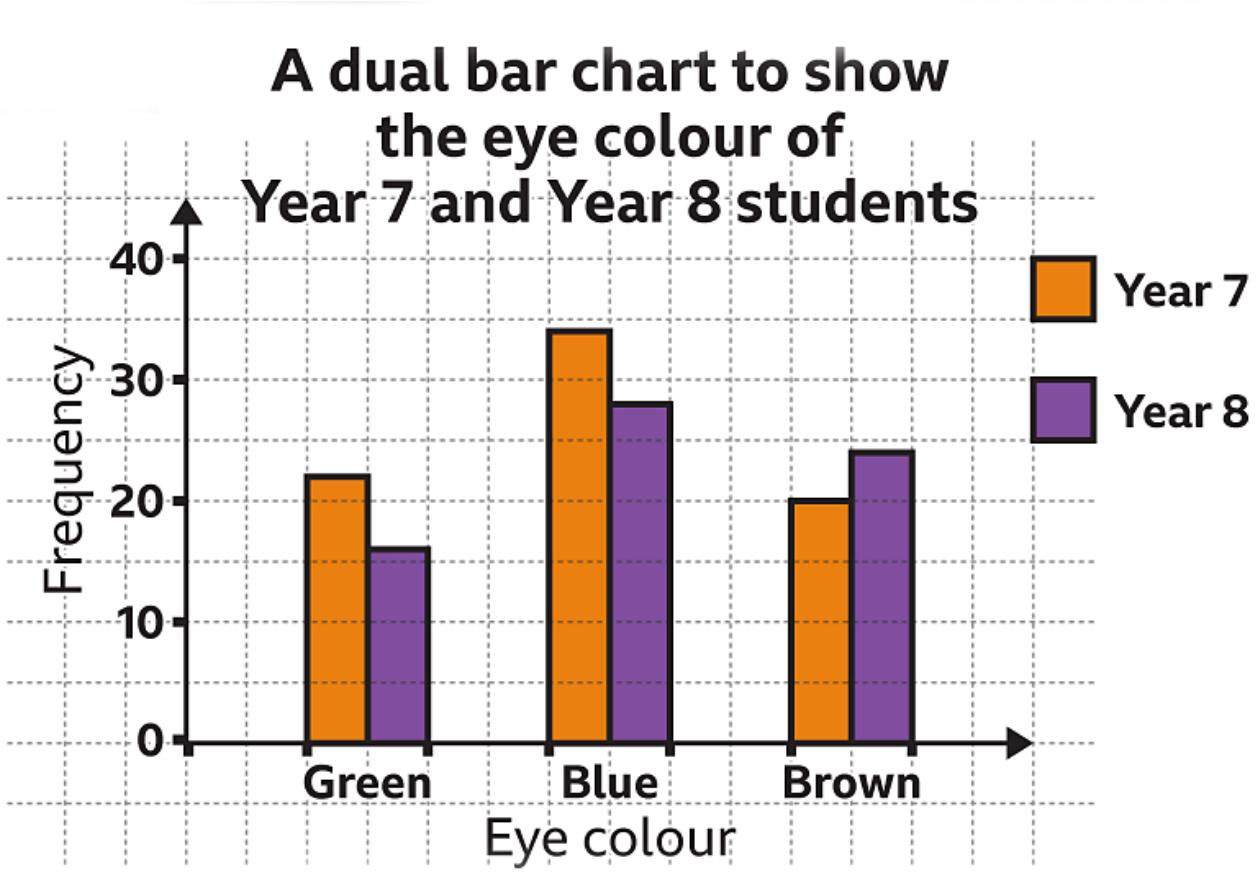
- Saving each graph temporarily:

```
1 graph pie, over(grade) ///
2 title(grade_pie) saving(grade_pie)
3 graph pie, over(gender) ///
4 title(gender_pie) saving(gender_pie)
```

- Using the command *graph combine*:

```
1 graph combine grade_pie.gph gender_pie.gph
```

Bar Chart


A dual bar chart to show
the eye colour of
Year 7 and Year 8 students
Frequency
40
30
20
10
0
Green
Blue
Brown
Eye colour
Year 7
Year 8

Bar Chart

- Characteristics: Display the numerical size of different categories, suitable for discrete variables.
- Applicable Data: Categorical data, such as sales of different products, GDP of different countries, etc.
- Basic Command: *graph (h)bar yvars, over(varname)*

Bar Chart

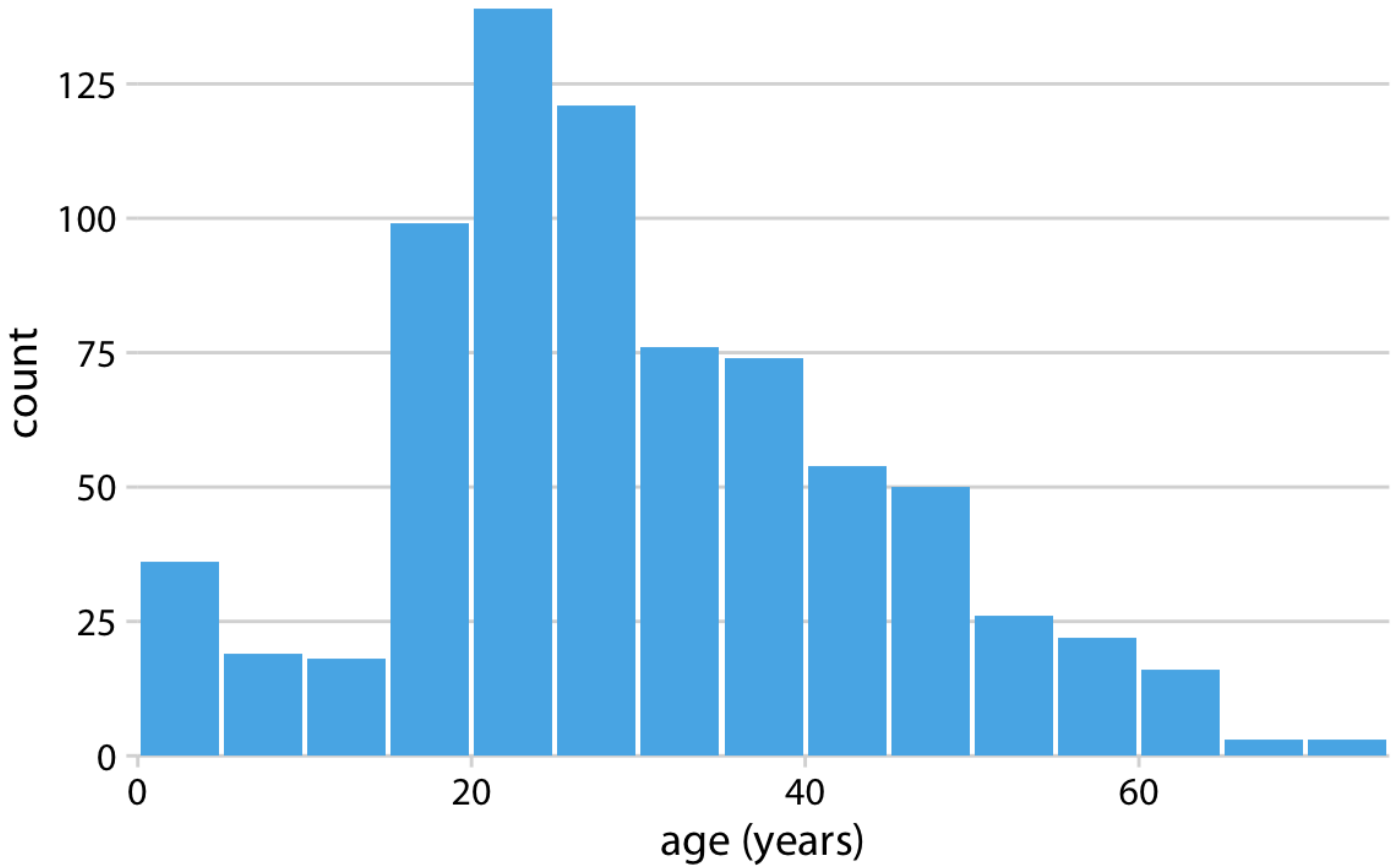
- Multiple categories:

```
1 graph bar (mean) qg11, over(year)
2 graph bar (mean) qg11, over(year) over(gender)
3 graph bar (mean) qg11, over(year) by(gender)
```

- Multiple variables:

```
1 graph bar medv lstat, over(rad)
2 graph bar medv lstat, stack over(rad)
```

Histogram



Histogram

- Characteristics: Display the distribution of continuous variables, showing the frequency or density of data through grouping (bin).
- Applicable Data: Continuous data, such as age, income, temperature, etc.
- Basic Command: *histogram varname*

Histogram

- Modifying the y-axis:

```
1 histogram tax, freq // abbr. frequency
2 histogram tax, frac // abbr. fraction
3 histogram tax, percent // 100*fraction
```

- Then modifying the x-axis:

```
1 histogram tax, bin(10)
2 histogram tax, width(50)
3 histogram tax, start(50)
```

Histogram

- Adding some other elements:

```
1 histogram tax, normal // normal distribution plot
2 histogram tax, kdensity // KDE plot
3 histogram tax, percent addlabels // value labels
```

Histogram

- There's a two-variables histogram:

```
twoway (histogram tax if charles==1, percent ///
fcolor(gs12) lcolor(gs12)
width(25) start(187)) ///
(histogram tax if charles==0, percent ///
fcolor(none) lcolor(black)
width(25) start(187)), ///
legend(label(1 "charles") ///
label(2 "non-charles")) ///
graphregion(color(white)) ///
title("This_is_a_advanced_histogram_example")
```

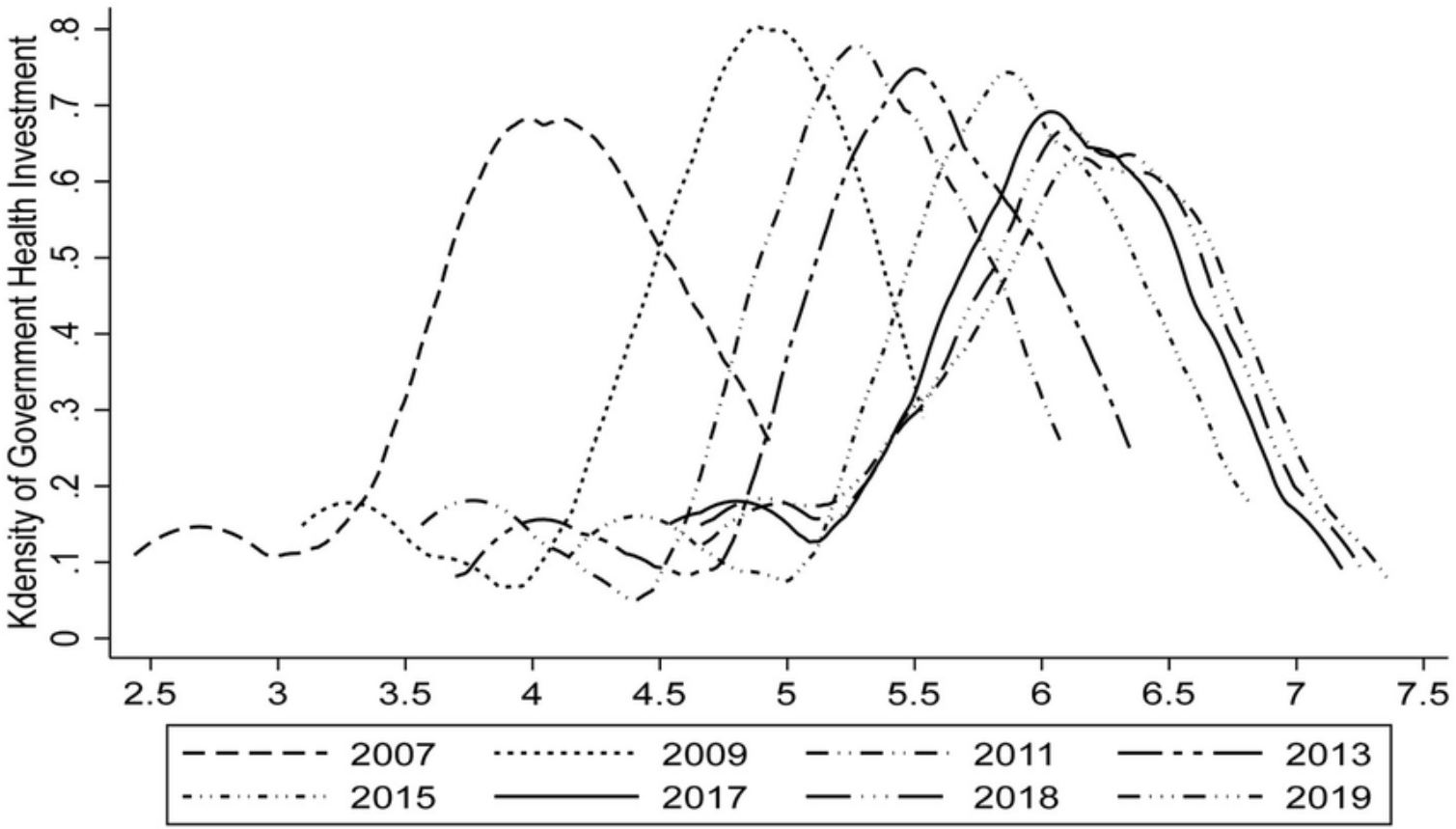
KDE Plot

- Let $(x_1, x_2, ..., x_n)$ be independent and identically distributed samples drawn from some univariate distribution with an unknown density f at any given point x. We are interested in estimating the shape of this function f which calls kernel density estimator.
- Often we use the normal kernel, and the KDE becomes:

$$\hat{f}_h(x) = \frac{1}{nh\sigma}\frac{1}{\sqrt{2\pi}}\sum_{i=1}^{n}\exp\left(-\frac{(x-x_i)^2}{2h^2\sigma^2}\right)$$

where K is the kernel (a non-negative function), and $h > 0$ is a smoothing parameter called the bandwidth.

KDE Plot


Kdensity of Government Health Investment
0
.1
.2
.3
.4
.5
.6
.7
.8
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
2007
2009
2011
2013
2015
2017
2018
2019

KDE Plot

- Characteristics: Smooth processing of the distribution of continuous variables, providing an estimate of the data distribution.
- Applicable Data: Continuous data, especially when the data volume is large, such as stock prices, height and weight, etc.
- Basic Command: *twoway (kdensity varname)*

KDE Plot

- Adding the area below the KDE plot. For example, using a specific value as the threshold:

```
kdensity price, gen(x h)
local threshold = 400
twoway (line h x) (area h x if x<`threshold')
```

- Or using the mean as the threshold:

```
sum tax, mean
local mean = r(mean)
kdensity price, gen(x h)
twoway (line h x) (area h x if x<`threshold')
```

Line Chart


Dow Jones Industrial Average
10000
9000
8000
7000
Jan 2009
Apr 2009
Jul 2009
Oct 2009
Jan 2010

Line Chart

- Characteristics: Display data trends over time, suitable for continuous time series data.
- Applicable Data: Time series data, such as stock prices, temperature changes, economic indicators, etc.
- Basic Command: *twoway line yvar xvar*

Line Chart

- Combining multiple lines:

```
1 twoway (line qg11 year) (line qg12 year)
2 twoway line qg11 qg12 year
```

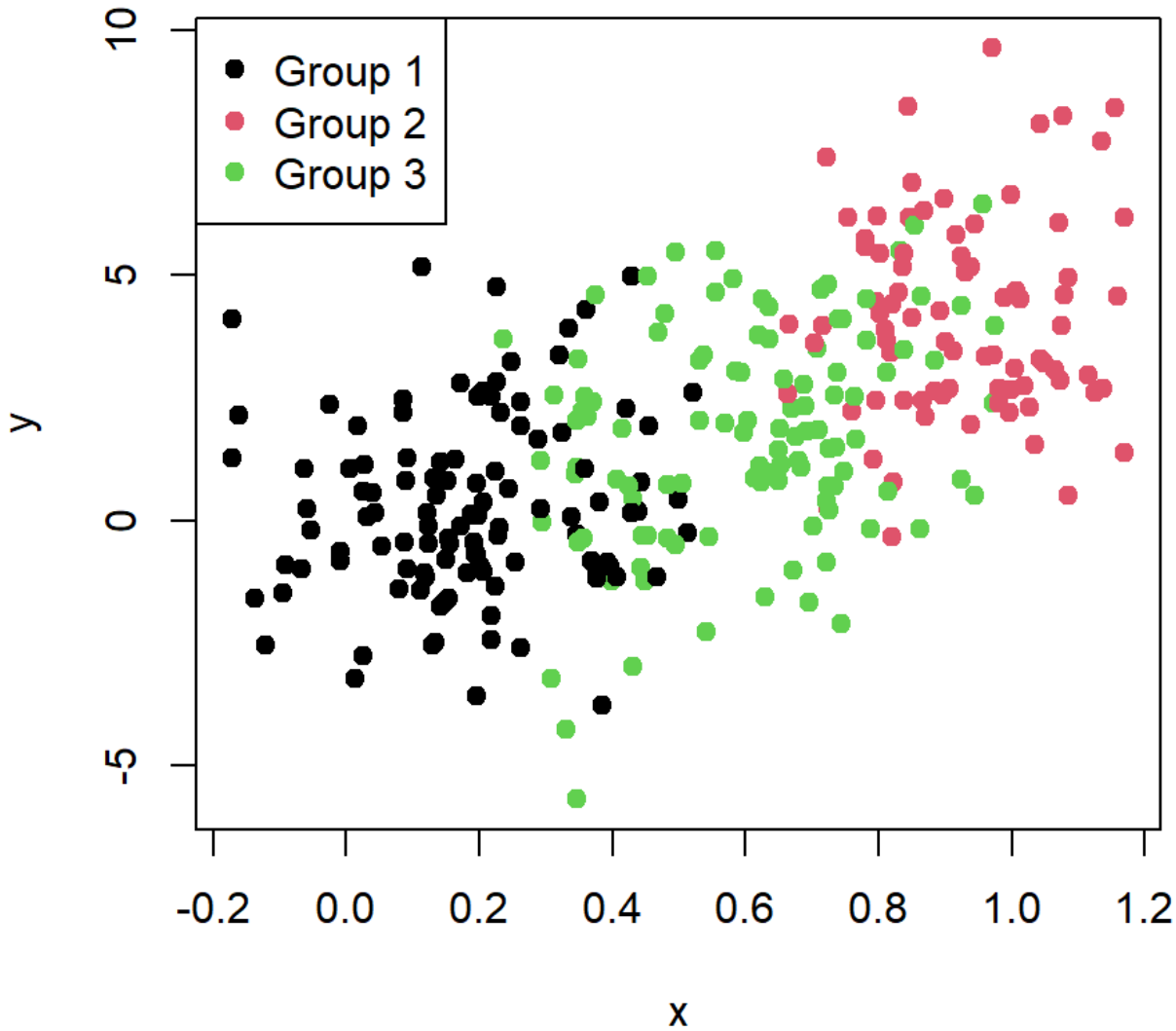
- Modifying the line options:

```
1 twoway line qg11 qg12 year, lpattern(solid dash)
2 twoway line qg11 qg12 year, lwidth(1.2 1.3)
3 twoway line qg11 qg12 year, lcolor(red blue)
```

- Using double axis:

```
1 twoway (line qg11 year, yaxis(1)) ///
2 (line qg401 year, yaxis(2))
```

Scatter Plot



Scatter Plot

- Characteristics: Display the relationship between two continuous variables, allowing observation of the correlation between variables.
- Applicable Data: Data of two continuous variables, such as height and weight, price and sales volume, etc.
- Basic Command: *twoway scatter yvar xvar*

Scatter Plot

- Adding the fitted lines:

```
1 twoway lfit medv nox
2 twoway qfit medv nox
```

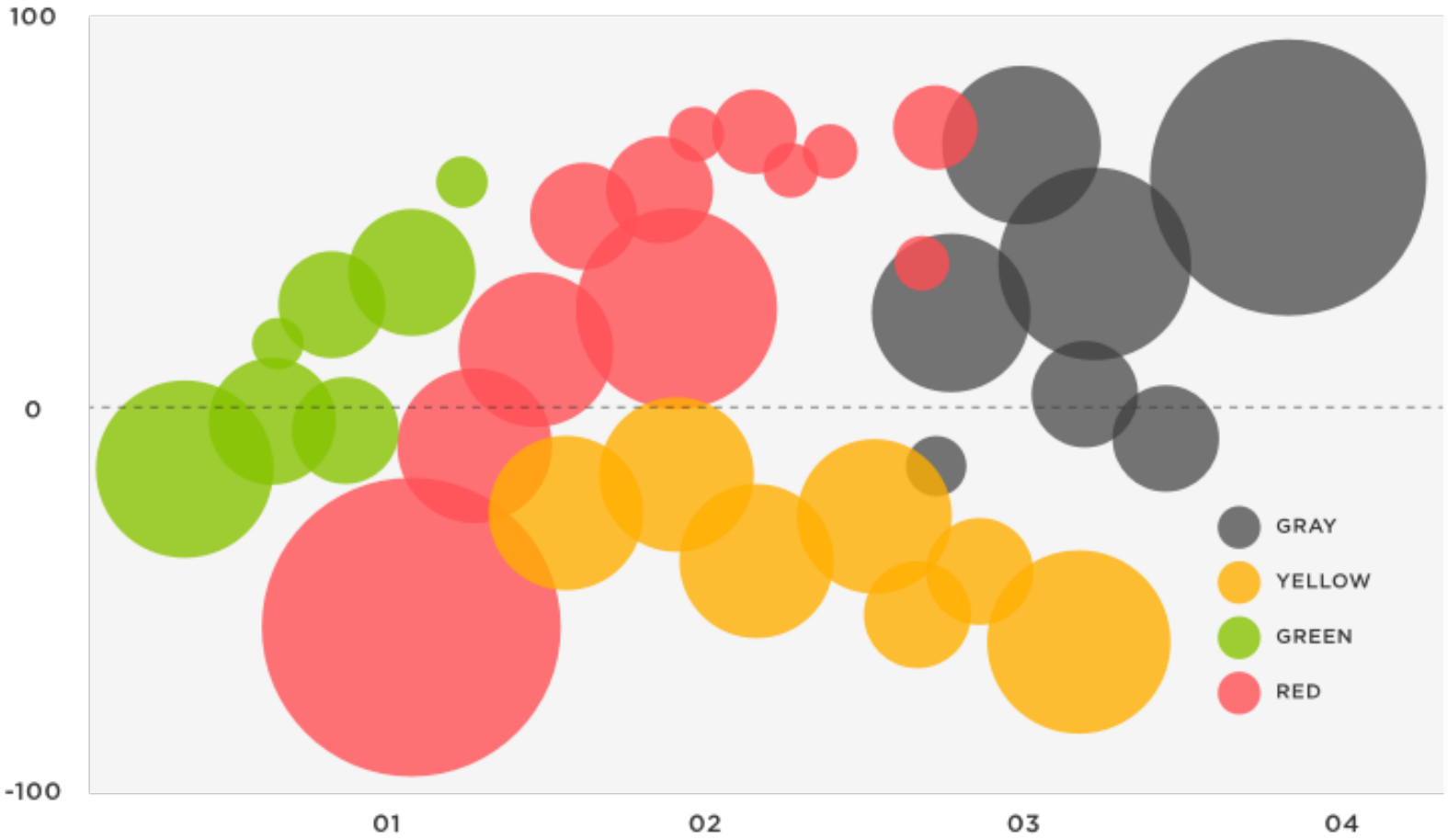
- Adding the 95% confidence interval:

```
1 twoway lfitci medv nox
2 twoway qfitci medv nox
```

- Combining these plots:

```
1 twoway (lfitci medv nox) (scatter medv nox)
```

Bubble Chart



Bubble Chart

- Characteristics: Represent the size of a third variable through the size of the bubble, suitable for displaying the relationship between three variables.
- Applicable Data: Data of three continuous variables, such as national GDP, population, and per capita GDP.
- For example:

```
gen medvrate = medv / room
label var medvrate "price_per_room"
scatter medvrate nox [w=room], ///
msymbol(circle_hollow)
```

Titles

- We will start with the example below which shows us how to change the title, subtitle, axis-title, caption and note in the graph:

```
graph twoway (lfitci medv nox) ///
(qfit medv nox) (scatter medv nox), ///
ytitle(price) xtitle(pollution) ///
title(A funny figure) ///
subtitle(Which is absolutely meaningless) ///
caption(Figure 1: Economic Science) ///
note(Notes: Don't be cheated by those people.)
```

Titles

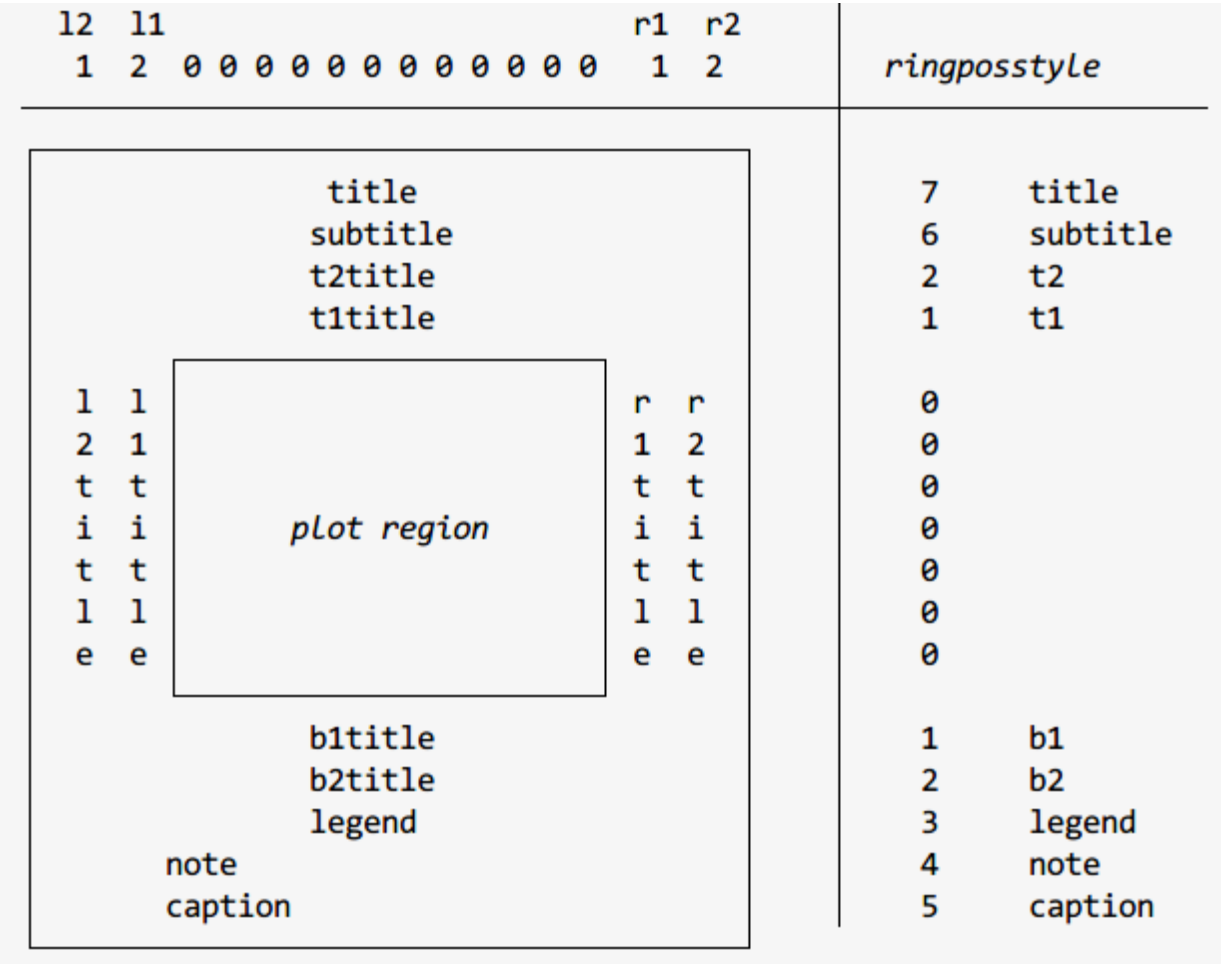


Figure: Default position of titles

Legends

- Changing the position of the legend:

```
1 graph twoway (lfitci medv nox) ///
2 (qfit medv nox) (scatter medv nox), ///
3 ytitle(price) xtitle(pollution) ///
4 legend(ring(0) pos(5))
```

Legends

- Leaving some specific legends:

```
graph twoway (lfitci medv nox) ///
(qfit medv nox) (scatter medv nox), ///
ytitle(price) xtitle(pollution) ///
legend(order(2 "linear_fit" 3 "quad_fit"))
```

Axis

- Changing the position of ticks:

```
graph twoway (lfitci medv nox) ///
(qfit medv nox) (scatter medv nox), ///
ytitle(price) xtitle(pollution) legend(off) ///
xlabel(, tposition(inside)) ///
ylabel(, tposition(inside))
```

- Changing the region of the axis:

```
graph twoway (lfitci medv nox) ///
(qfit medv nox) (scatter medv nox), ///
ytitle(price) xtitle(pollution) legend(off) ///
plotregion(margin(0))
```

Axis

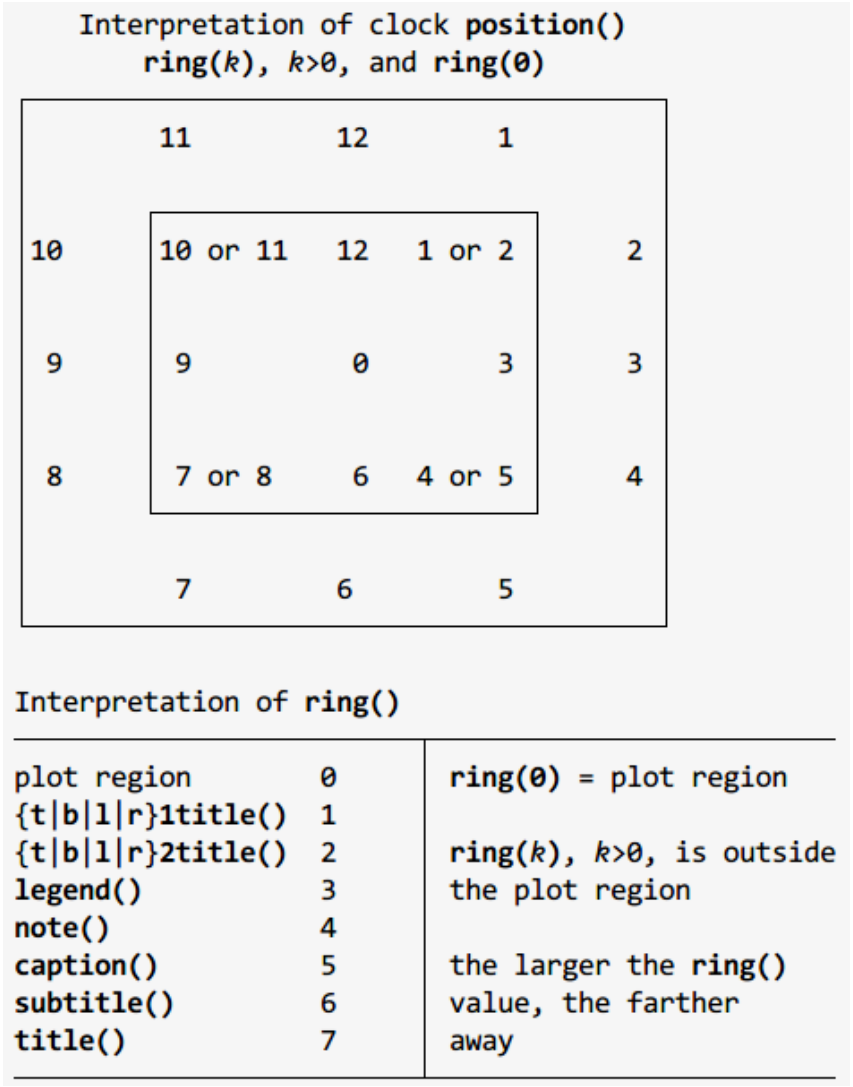


Figure: Interpretation of *ring* and *position*

Axis

- Extending the axis:

```
1 graph twoway (lfitci medv nox)
2 (qfit medv nox) (scatter medv nox), ///
3 ytitle(price) xtitle(pollution) legend(off) ///
4 xscale(extend) yscale(extend)
```

- Changing the scale:

```
1 graph twoway (lfitci medv nox) ///
2 (qfit medv nox) (scatter medv nox), ///
3 ytitle(price) xtitle(pollution) legend(off) ///
4 yscale(log)
```

Text

- Adding texts into the graph:

```
1 graph twoway (lfitci medv nox)
2 (qfit medv nox) (scatter medv nox), ///
3 ytitle(price) xtitle(pollution) legend(off) ///
4 text(30 0.8 "Class_is_over.", place(3))
```

Map at the Province-Level

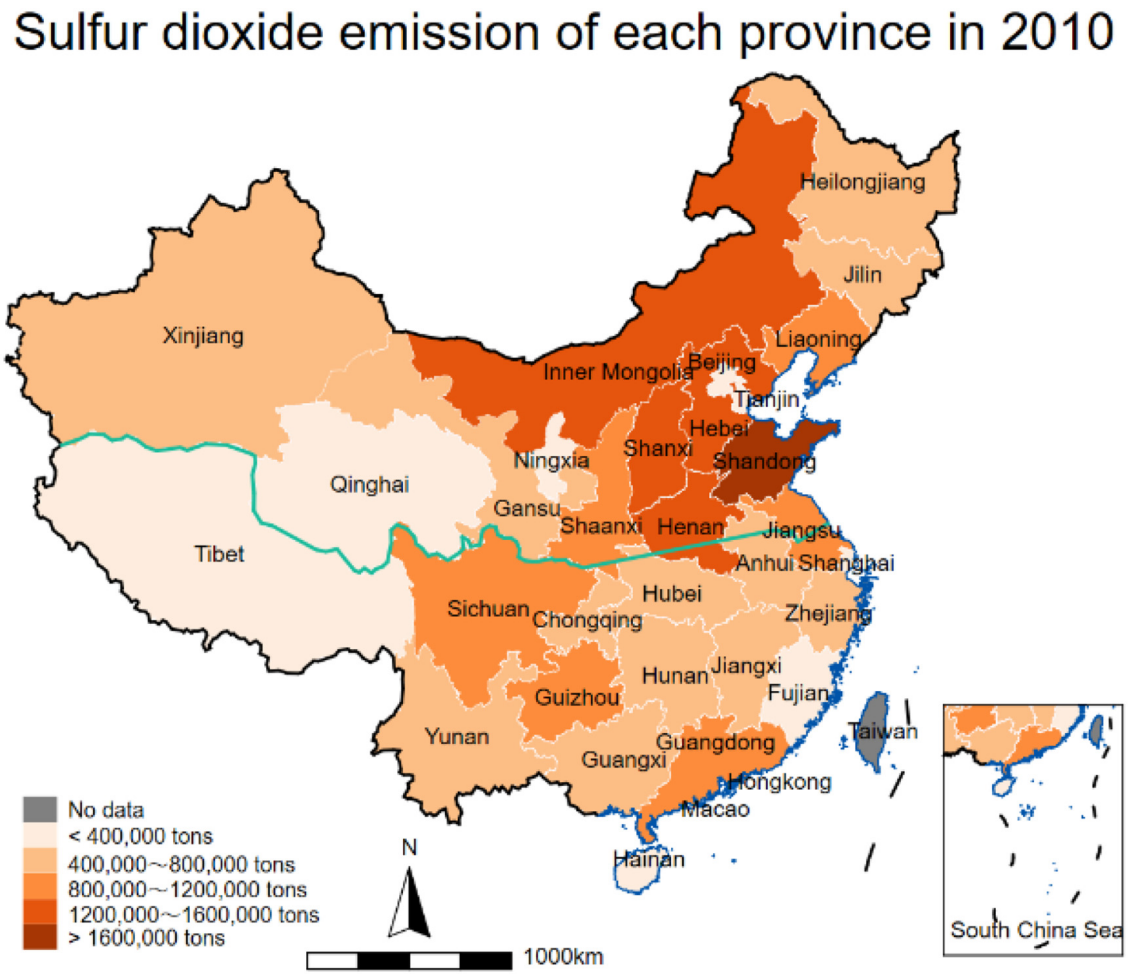


Figure: Map of China at the province level

Map at the Province-Level

```
use china_map_value, clear
grmap, activate
grmap Value using chinaprov40_coord, ///
    id(ID) osize(vvthin ...) ocolor(white ...) ///
    clmethod(custom) ///
    clbreaks(-1 0 40 80 120 160 200) ///
    fcolor(gray "254 237 222" "253 190 133" ///
    "253 141 60" "230 85 13" "166 54 3") ///
    legend(order(2 "No Data" 3 "<40" 4 "40~80" ///
    5 "80~120" 6 "120~160" 7 ">160")) ///
    graphr(margin(medium)) ///
```

Map at the Province-Level

```
    line(data(chinaprov40_line_coord.dta) ///
        select(
        keep if inlist(group, 1, 2, 3, 5, 6)) ///
        by(group) size(vvthin *1 *.5 *.5 *.5) ///
        pattern(solid ...) ///
        color(white black "0 85 170" ///
        black black) ///
        ) ///
    polygon(data(china_map_compass) ///
    fcolor(black) osize(vvthin)) ///
    label(data(china_map_label) ///
    x(X) y(Y) label(ename) ///
    length(20) size(*.6)) ///
    title("This_is_our_homeland")
graph export china_map_export.png, ///
replace width(1200)
```

Map at the City-Level

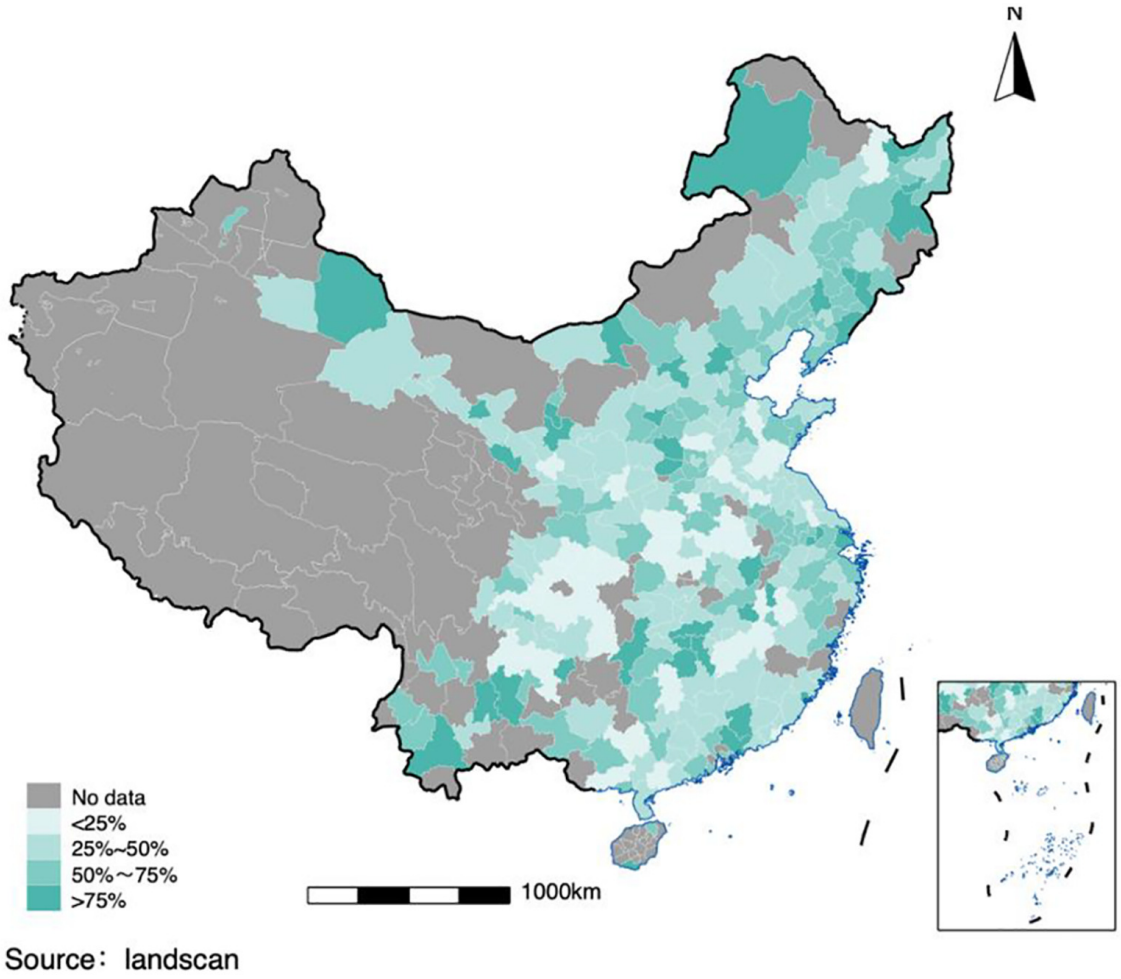


Figure: Map of China at the city level

Map at the City-Level

```
use "data_fig3.dta", clear
grmap, activate
grmap mono using chinacity2020_coord.dta, ///
    id(ID) osize(vvthin ...) ocolor(white ...) ///
    clmethod(custom) ///
    clbreaks(-1  0 25 50 75 100 ) ///
    fcolor("gs10" "224 242 241" "178 223 219" ///
    "128 203 196" "77 182 172" "38 166 154") ///
    legend(order(2 "No data" 3 "<25%" ///
    4 "25%~50%" 5 "50%~75%" 6 ">75%" )) ///
        graphr(margin(medium)) ///
```

Map at the City-Level

```
    line(data(chinacity2020_line_coord3.dta) ///
        by(group) size(vvthin *1 *.5 *.5 *.5) ///
        pattern(solid ...) ///
        select(drop if inlist(group, 4, 7)) ///
        color(white black "0 85 170" ///
        black black)) ///
    polygon(data(china_map_compass) ///
    fcolor(black) osize(vvthin)) ///
    label(data(china_map_label) ///
    x(X) y(Y) label(ename) ///
    length(20) size(*.6)) ///
    title("This_is_our_homeland")
graph export china_map_export.png, ///
replace width(1200)
```

References

- Wilke, Claus O. "Fundamentals of Data Visualization." *O'Reilly Media, Inc*, 2019.
- Wickham, Hadley, Danielle Navarro, and Thomas L. Pedersen. "ggplot2: Elegant Graphics for Data Analysis." *Springer*, 2016.
- Schwabish, Jonathan A. "An Economist's Guide to Visualizing Data." *Journal of Economic Perspectives*, 2014.
- Zhang, Dongyang. "Green Financial System Regulation Shock and Greenwashing Behaviors: Evidence from Chinese Firms." *Energy Economics*, 2022.
- He, Xiaoping, Yuxuan Yu, Shuo Jiang. "City Centrality, Population Density and Energy Efficiency." *Energy Economics*, 2023.

Questions are welcome.