Data Visualization and Diagramming

Academic Writing

Wang, Yu

dr.yuwang@outlook.com

Copyright © Wang Yu, 2024. This copy of slides is preliminary and intended for personal and educational use only. Redistribution or sharing, in whole or in part, without explicit permission from the author is strictly prohibited. A full list of references is provided at the end of the slides.

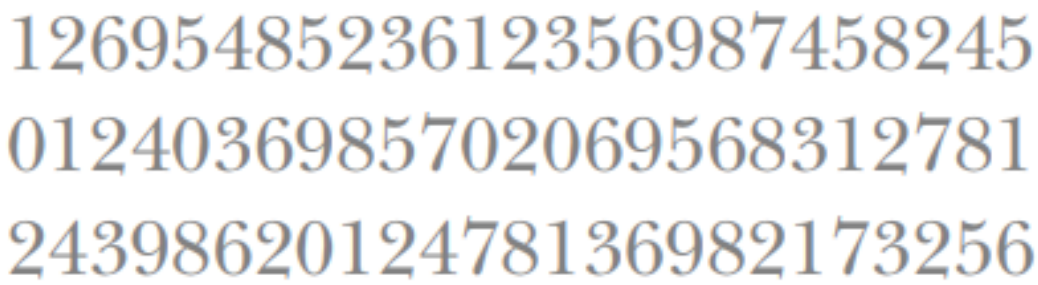
December 24, 2024

Overview

Table of Contents

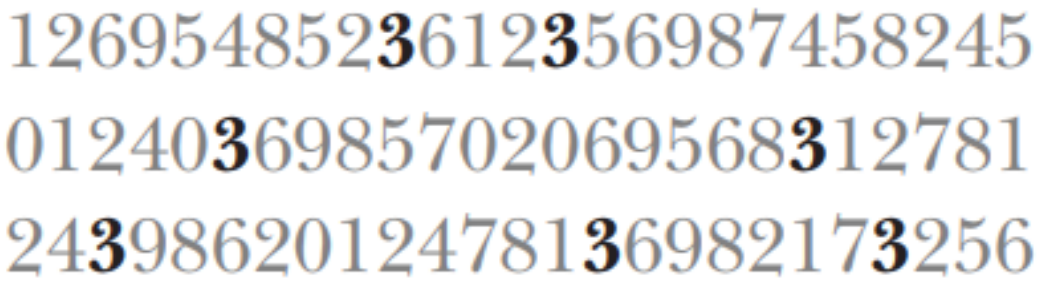
“A picture is worth a thousand words.”

How many the number 3?



“A picture is worth a thousand words.”

How many the number 3?



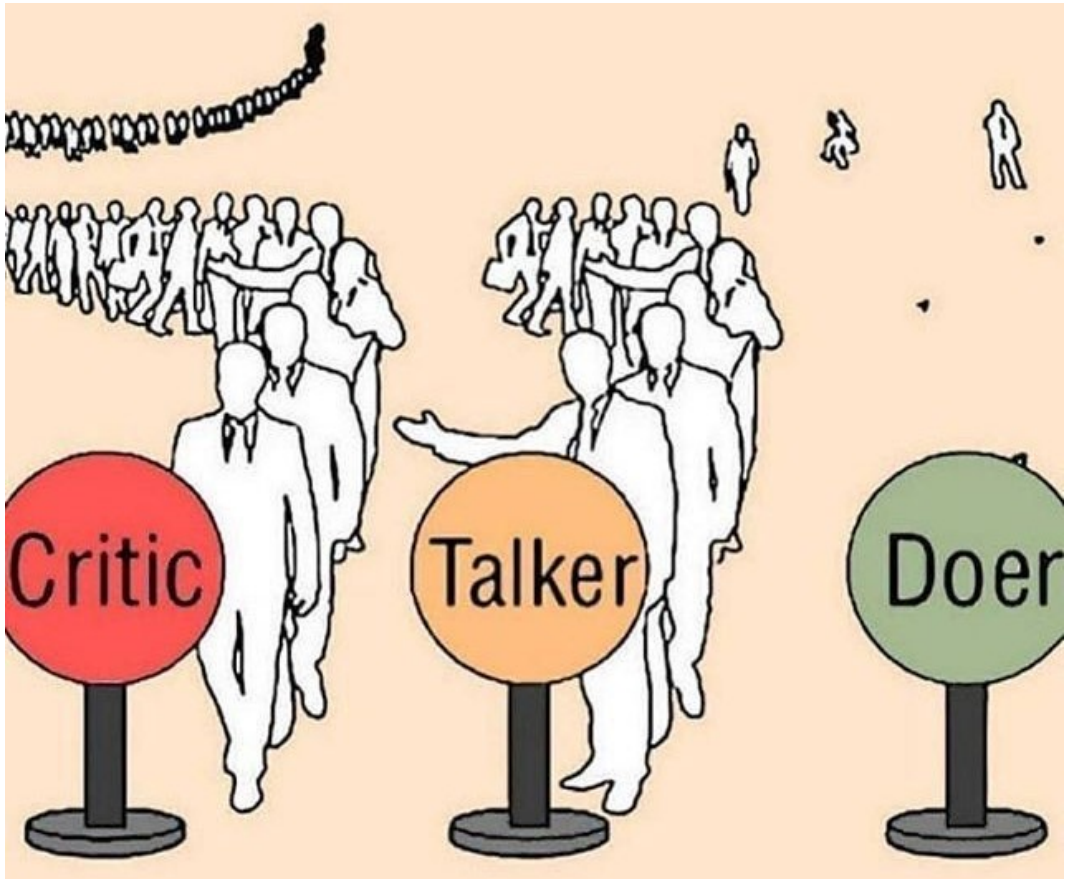
"A picture is worth a thousand words."

Two types of brain's visual processing:

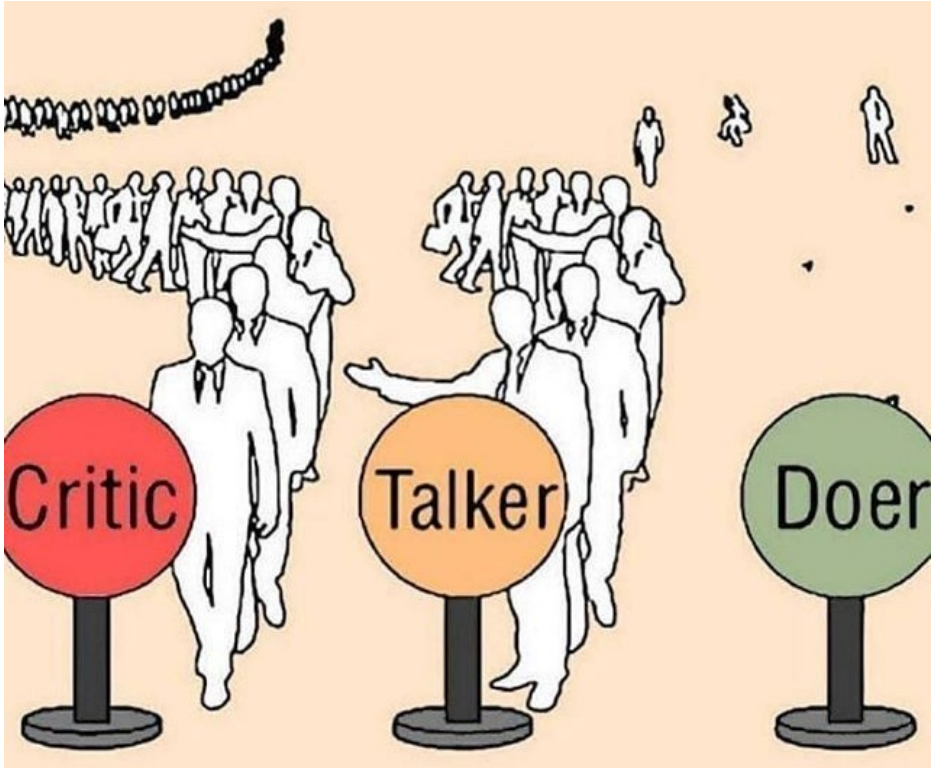
- Attentive processing is the conscious part of perception that allows us to perceive things serially;
- **Pre-attentive processing** allows the reader to perceive multiple basic visual elements simultaneously. It is done in parallel and is much faster.

"A picture is worth a thousand words."

Can you describe the picture with 30 words?



"A picture is worth a thousand words."



Picture superiority effect

- Pictures are superior to words in terms of recalling and recognizing information.

Three Basic Principles

Data visualization

- The graphical representation of information and data. It is a more efficient way of communication than words.

Three principles of visualizing data (Schwabish, 2014):

- 1) **Show the data, but not too much**

  The data are the most important part of the graph and should be presented in the clearest way possible;

- 2) **Integrate the text and the graph**

  Visualizations should be constructed to complement the text and at the same time to contain enough information to stand alone;

Three Basic Principles

Which figure below is more effective in communicating information?

Three Basic Principles

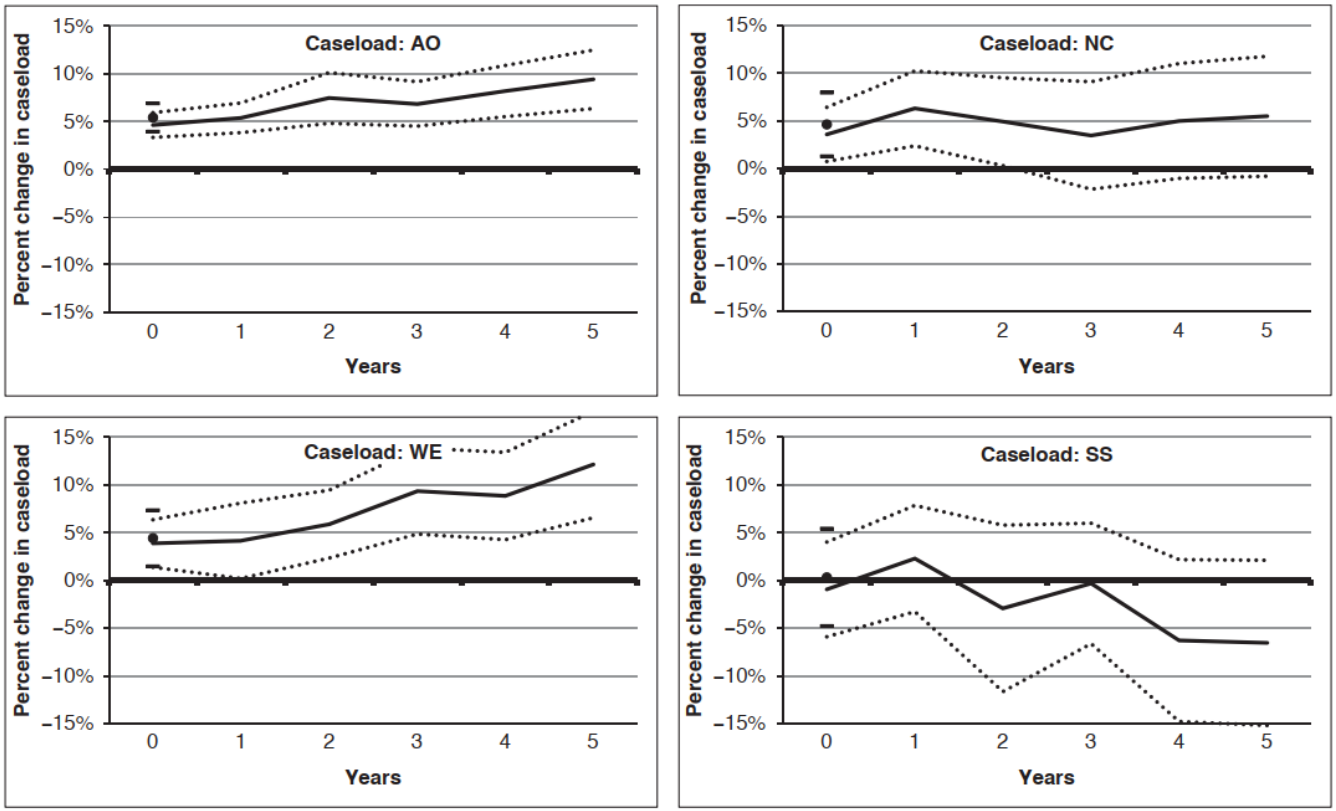
- 3) **Reduce the clutter**

  The use of unnecessary or distracting visual elements will reduce effectiveness.

Note: figures in the print version of economic journals are in greyscale (black and white), though some journals offer free reproduction of color figures in the online version.

Table of Contents

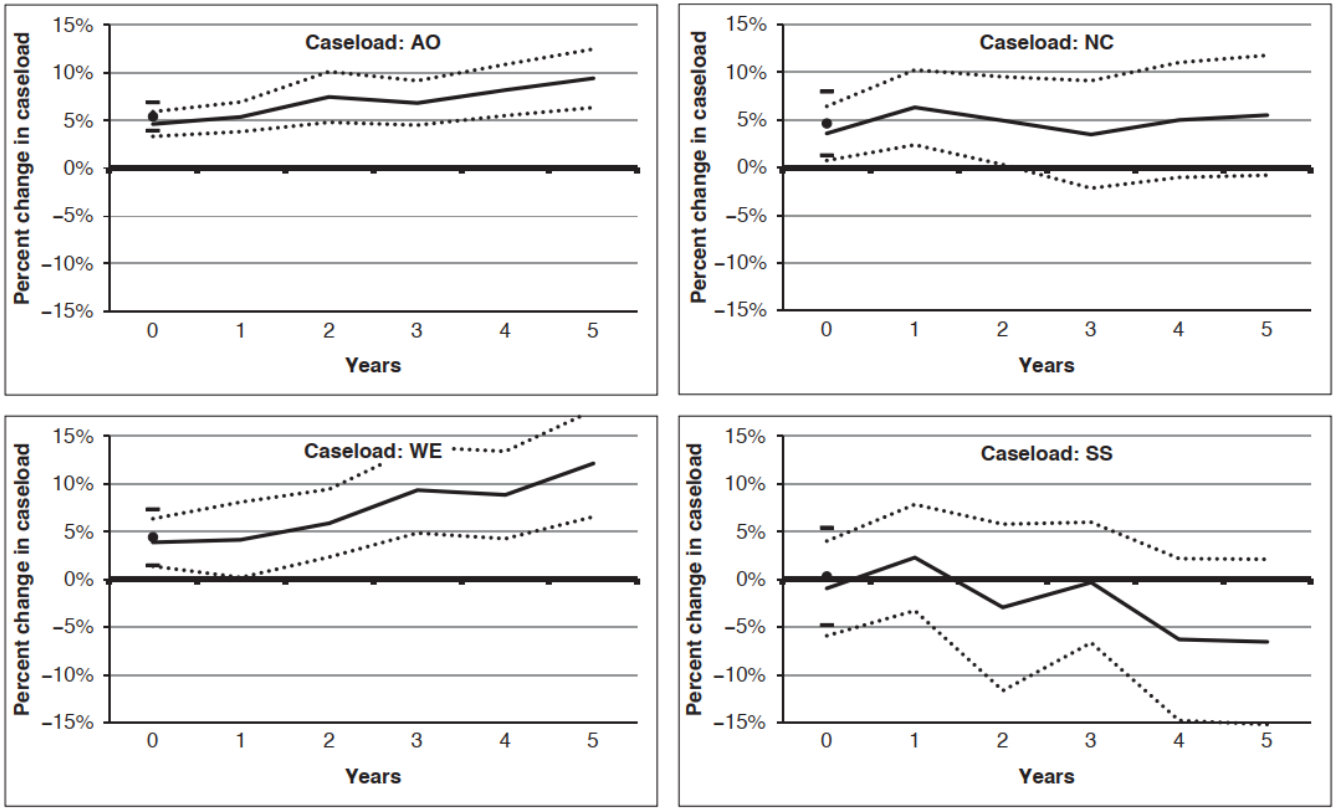
The Line Chart



Is this figure well designed? Consider "the three basic principles":

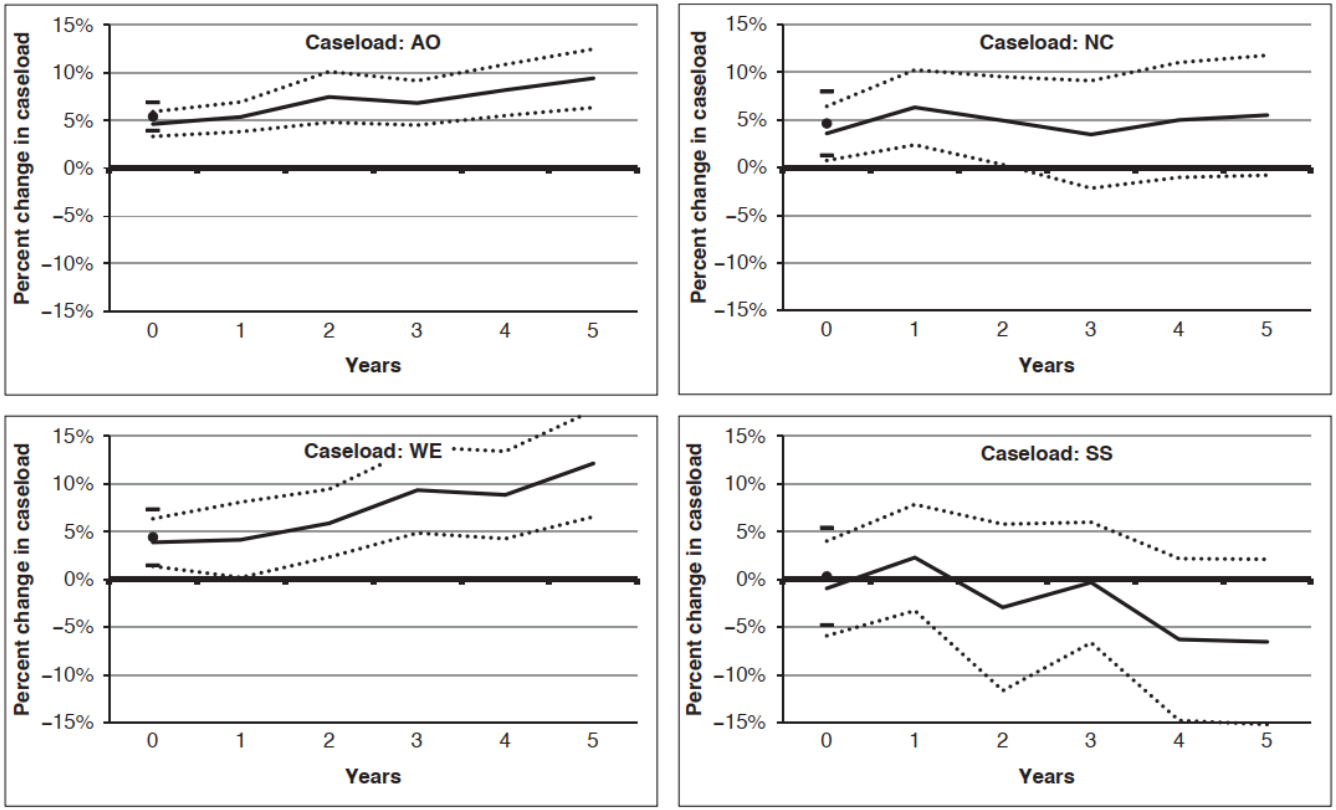
- Show the data, but not too much;
- Integrate the text and the graph;
- Reduce the clutter.

The Line Chart



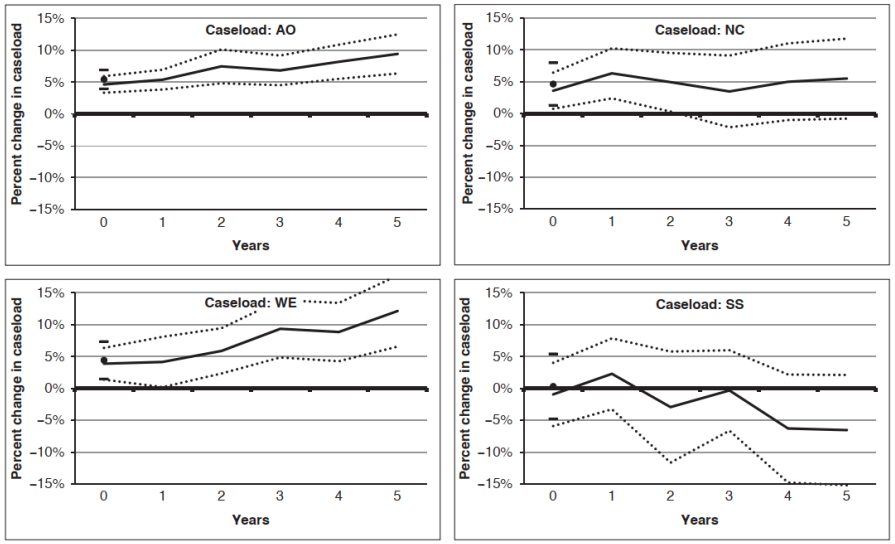
- 1) Fail to emphasize the data: the darkest and thickest line is the 0 percent gridline;
- 2) Fail to show all data points: the data values of the WE and SS charts exceed 15 percent data marker;

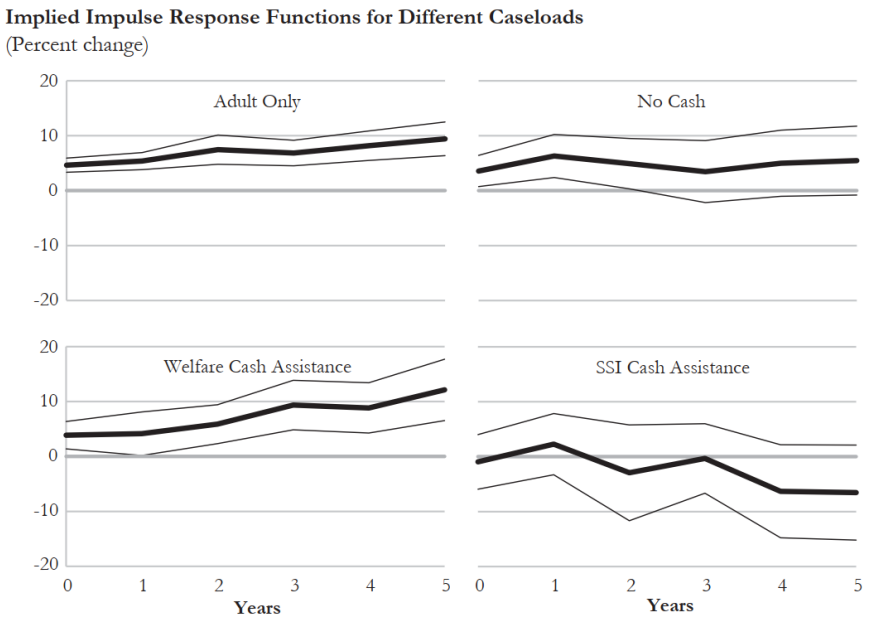
The Line Chart



- 3) Unnecessary clutter: the x-axis and y-axis labels, tick marks on the y-axes, and percentage signs are redundant;
- 4) Text and graphs are not integrated: what do AO, NC, WE, and SS mean?

The Line Chart (Left: Original, Right: Revised)

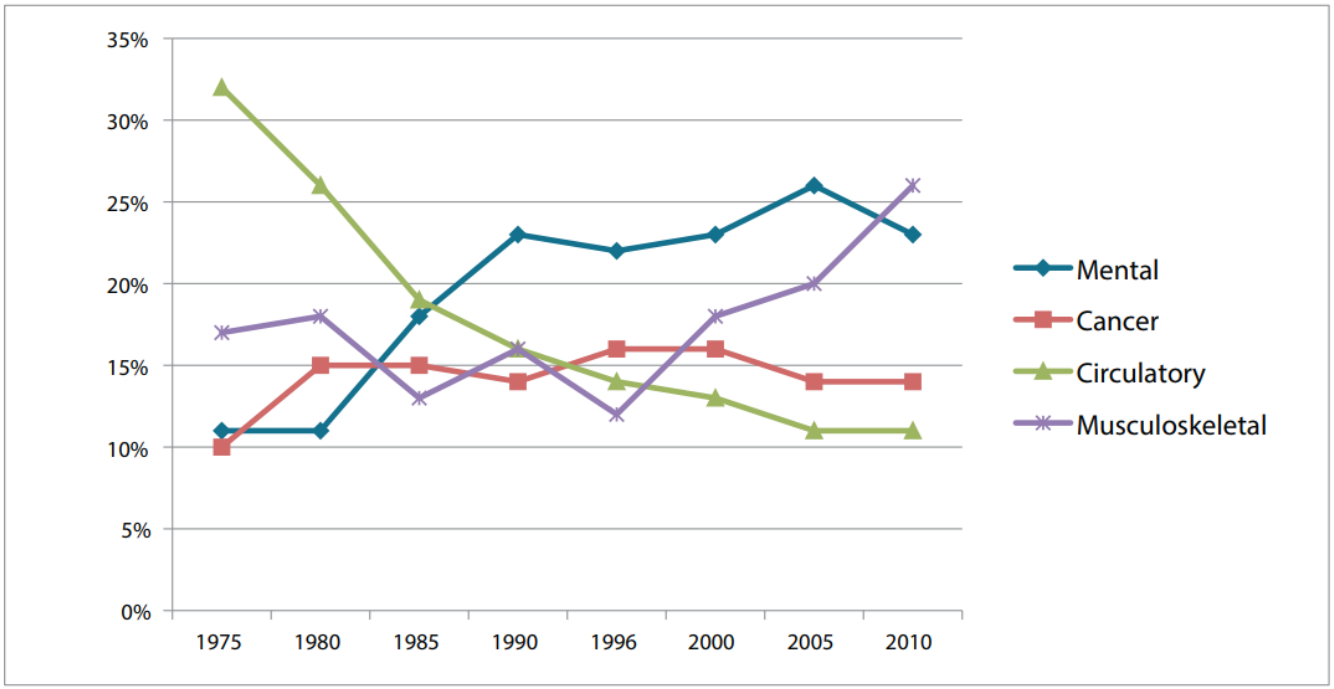




- Change the darkest line to show the data (the coefficient estimate);
- Lighten the gridlines, but leaving the 0 percent gridline slightly darker;
- Eliminate y-axis and x-axis labels to reduce clutter;
- Eliminate the percent signs and identify the unit below the title;
- Reposition the word "Caseload" into the title;
- Spell out AO, NC, WE, and SS abbreviations.

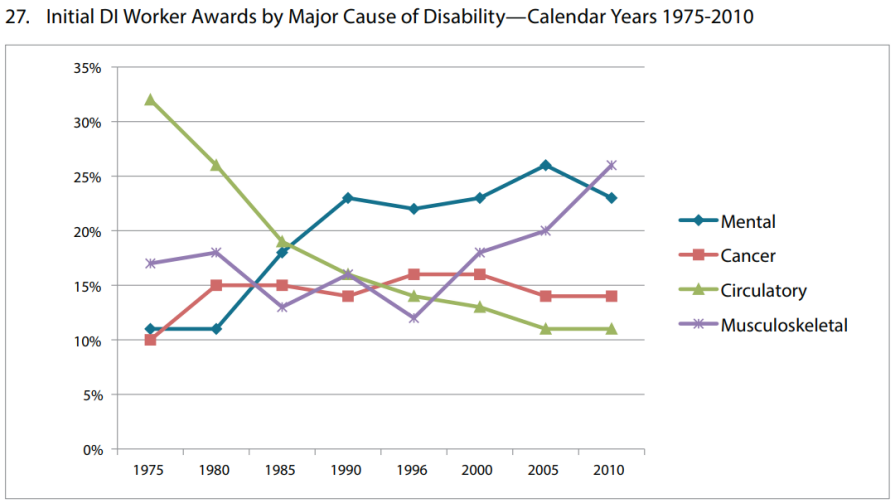
The Line Chart 2

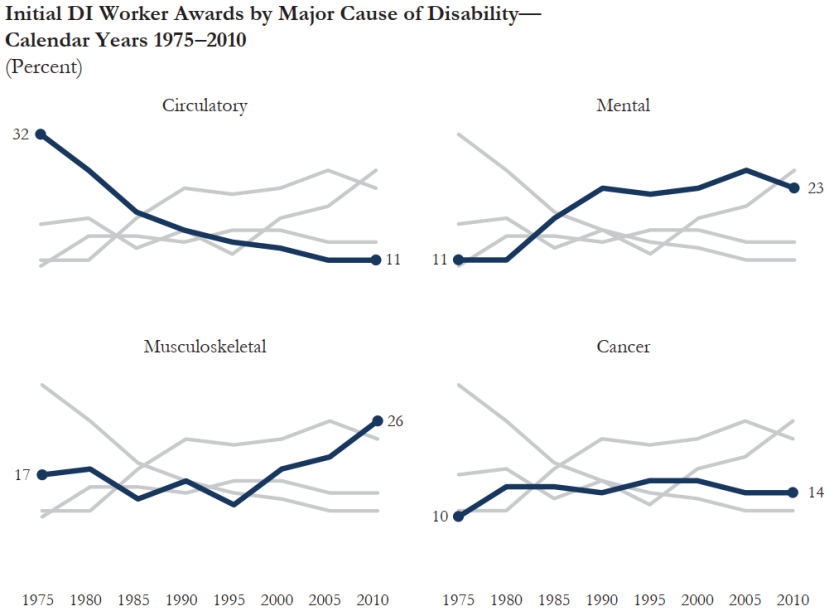
27. Initial DI Worker Awards by Major Cause of Disability—Calendar Years 1975-2010



- 1) Too many series in one chart;
- 2) Too many data markers;
- 3) Disconnection between the legend and the data.

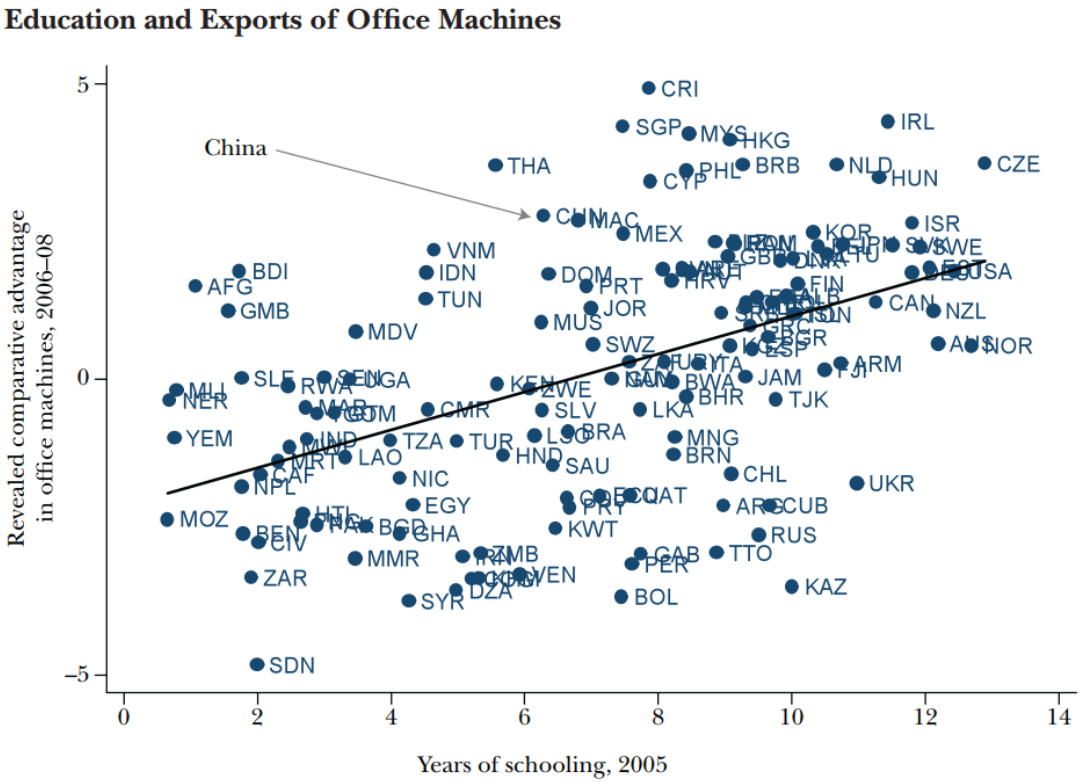
The Line Chart 2 (Left: Original, Right: Revised)





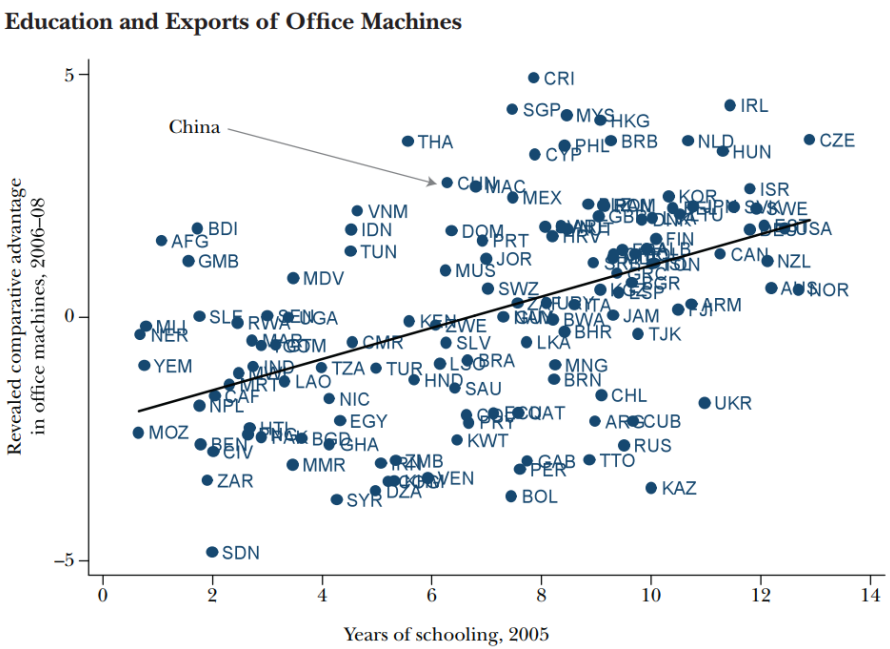
- Create smaller charts in series;
- Use titles instead of legends;
- Lighten the other line series to reduce clutter;
- Delete the y-axis and identify the unit below the title.

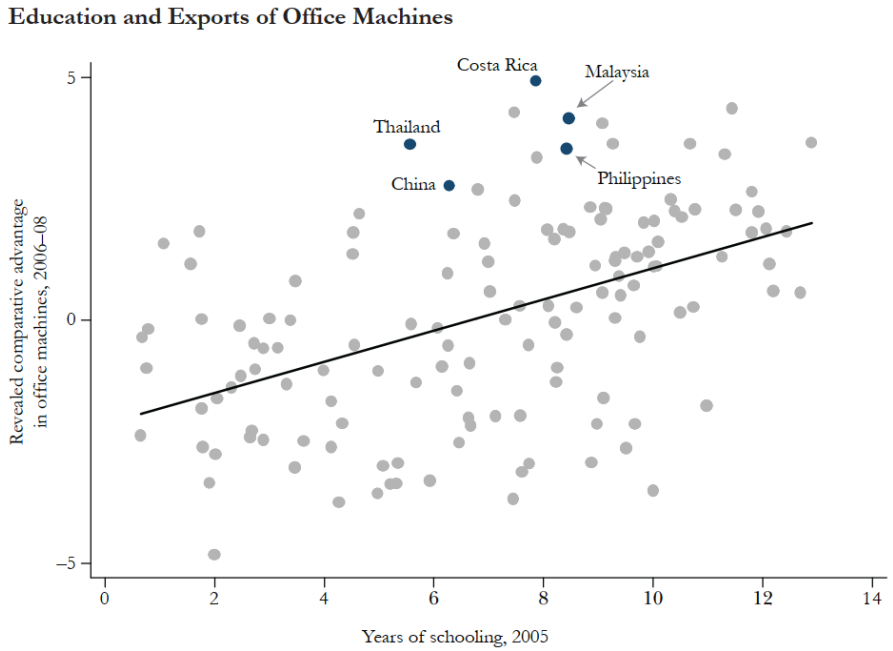
The Scatter Plot



- 1) Too many labels and dots;
- 2) Unfamiliarity and confusion with three-letter country codes.

The Scatter Plot (Left: Original, Right: Revised)



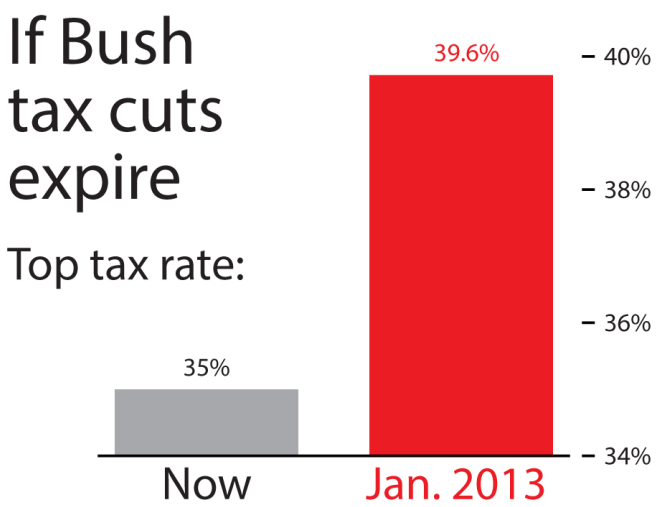


- Eliminate all data labels other than those mentioned in text;
- Make the five data points darker and lighten the other points;
- Replace country codes with country names.

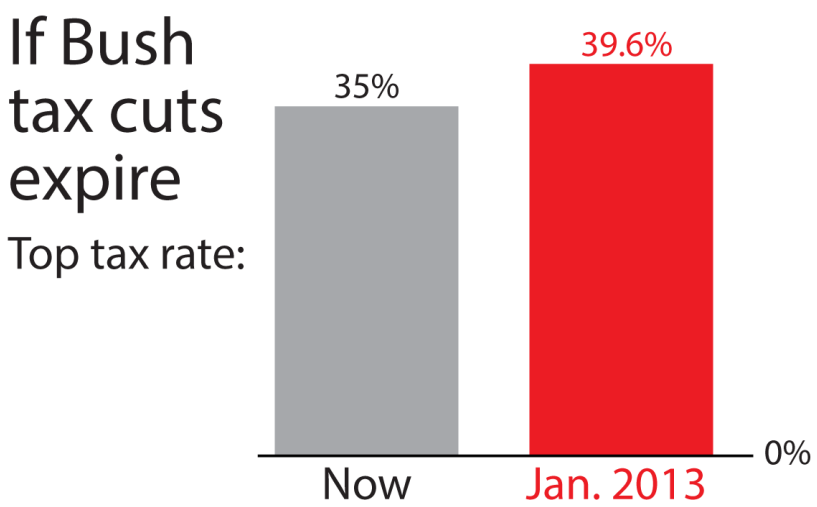
The Bar Chart

How do you graph a lie?

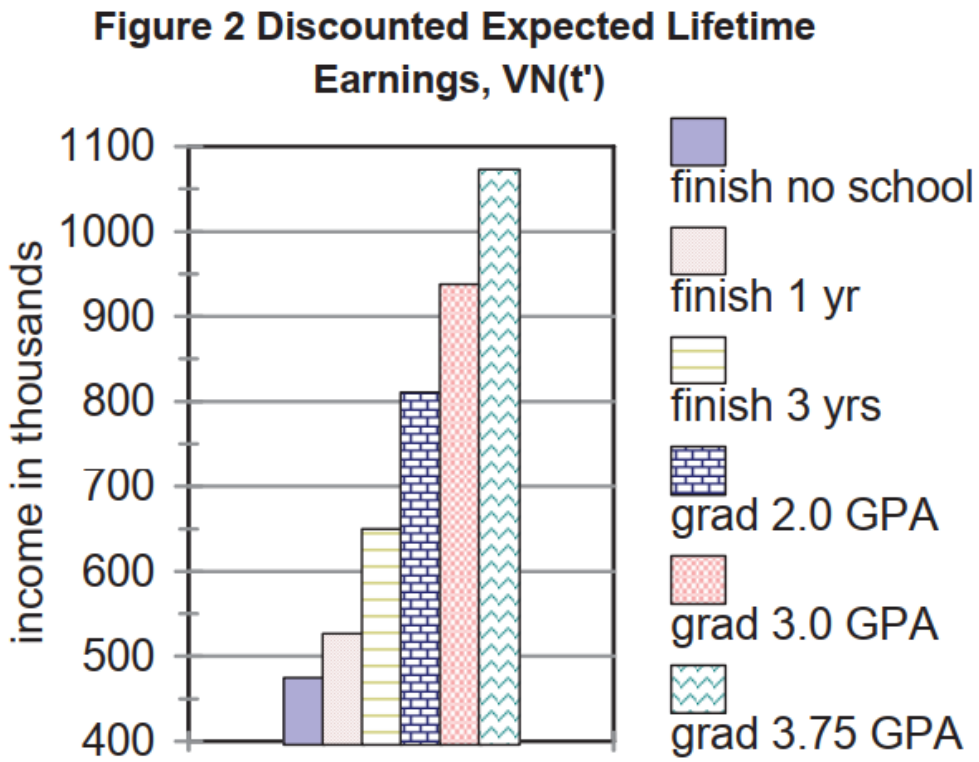
Misleading



More accurate

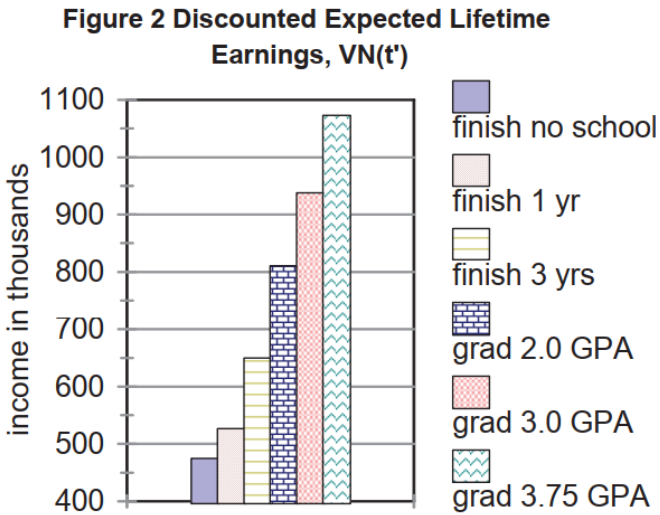


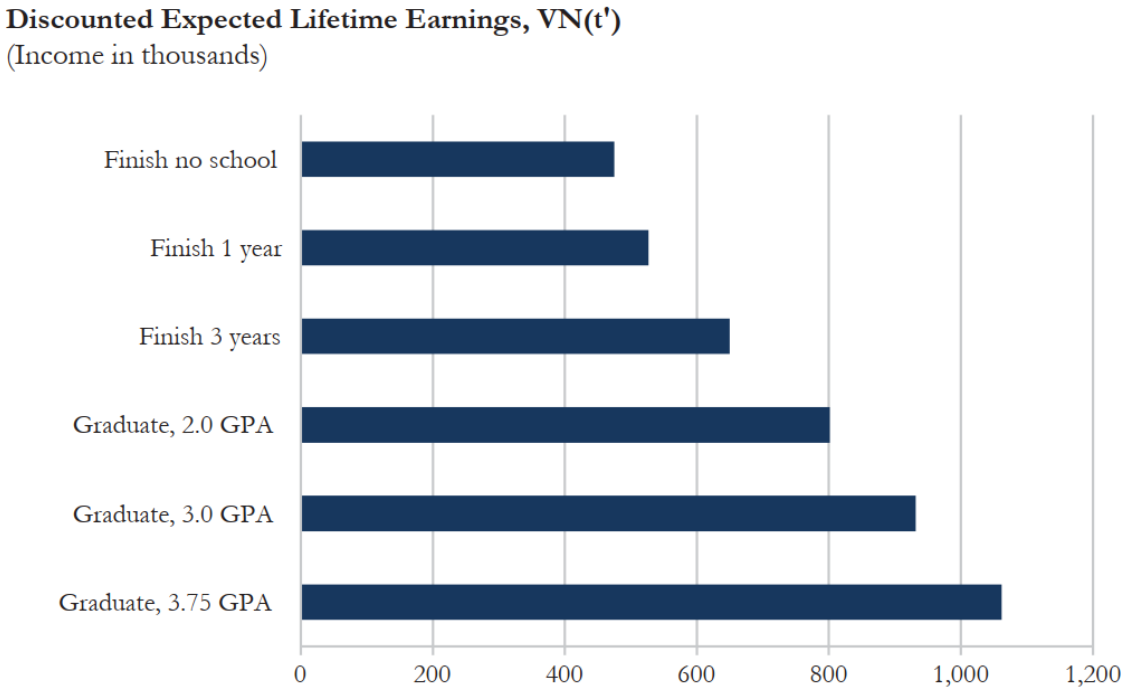
The Bar Chart



- 1) **Non-zero starting point (the differences between the bars are overemphasized);**
- 2) Unnecessary clutter;
- 3) Unnecessary colors.

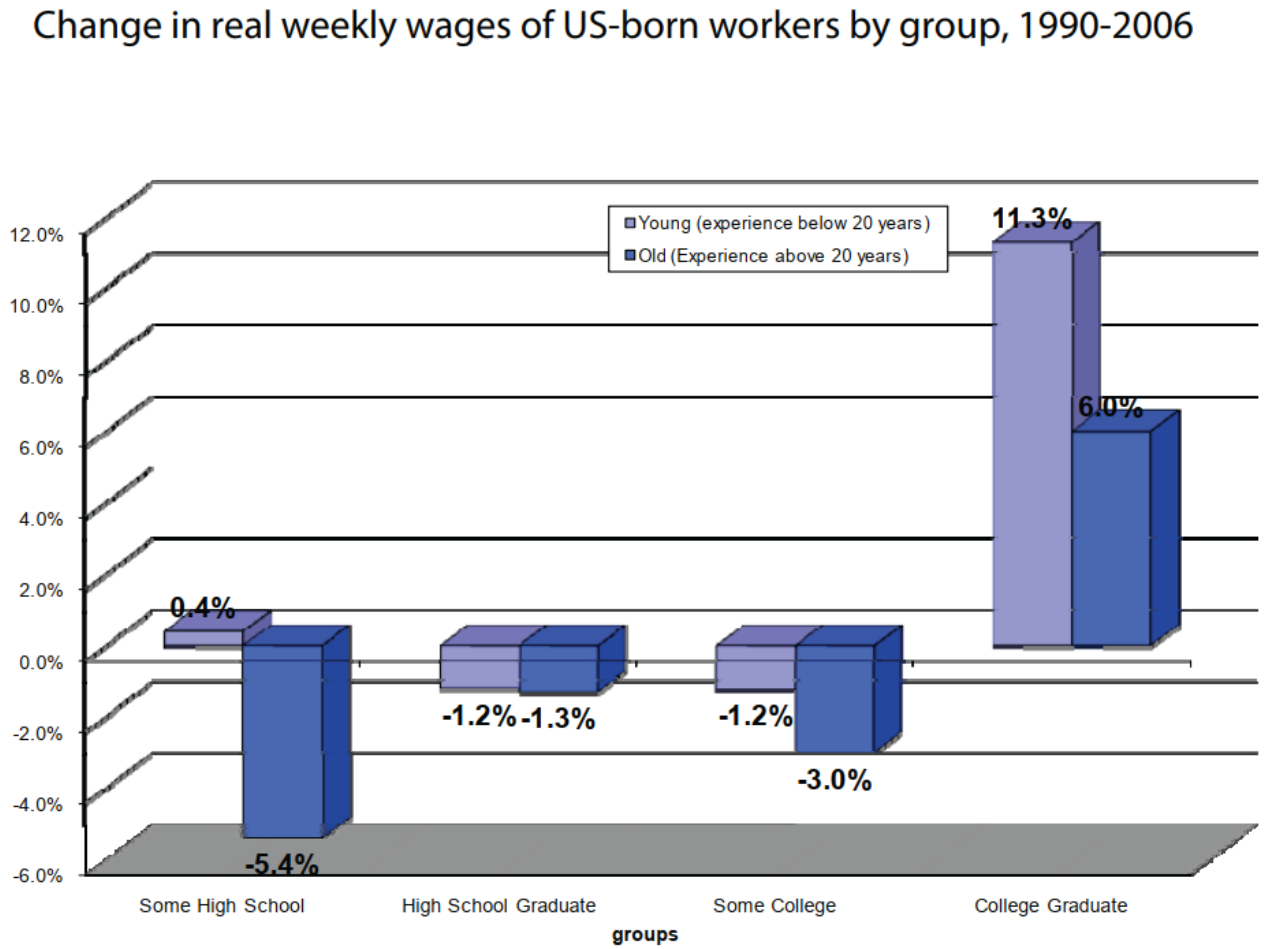
The Bar Chart (Left: Original, Right: Revised)





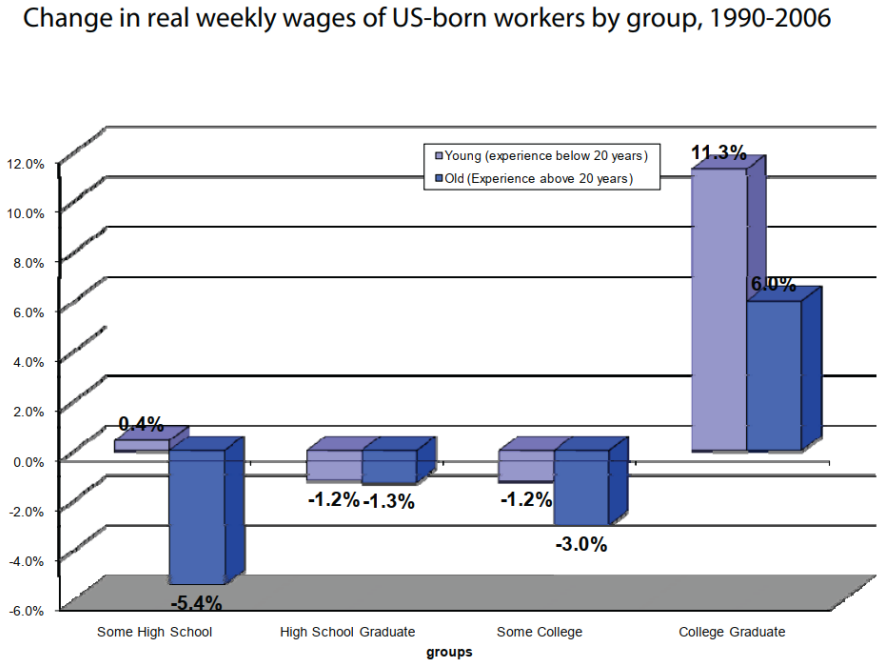
- Start the bars at zero;
- Rotate horizontally to make room for full labels;
- Eliminate bar clutter;
- Unify bar colors;
- Identify the unit below the title.

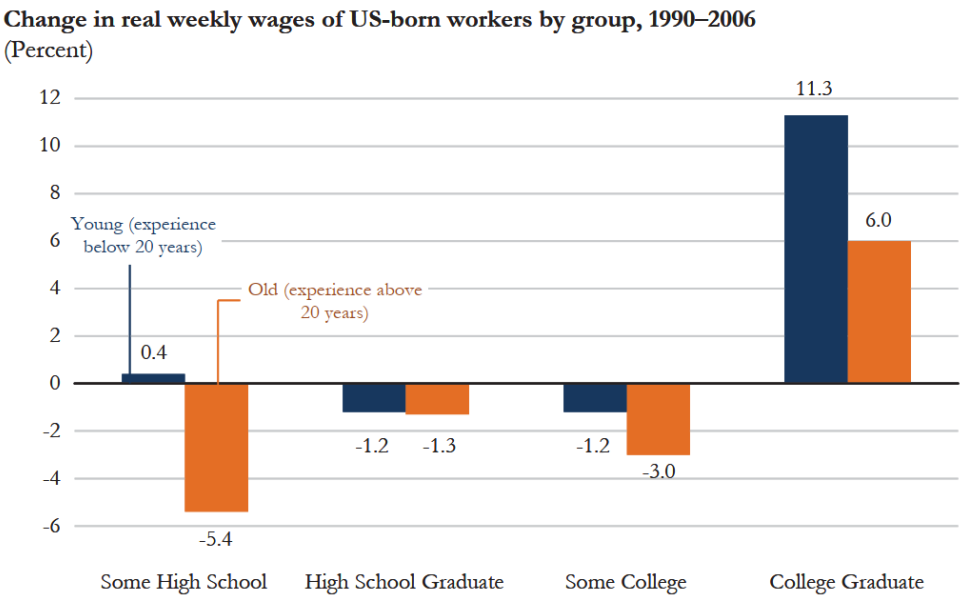
The 3D Chart



- Unnecessary 3rd dimension and clutter.

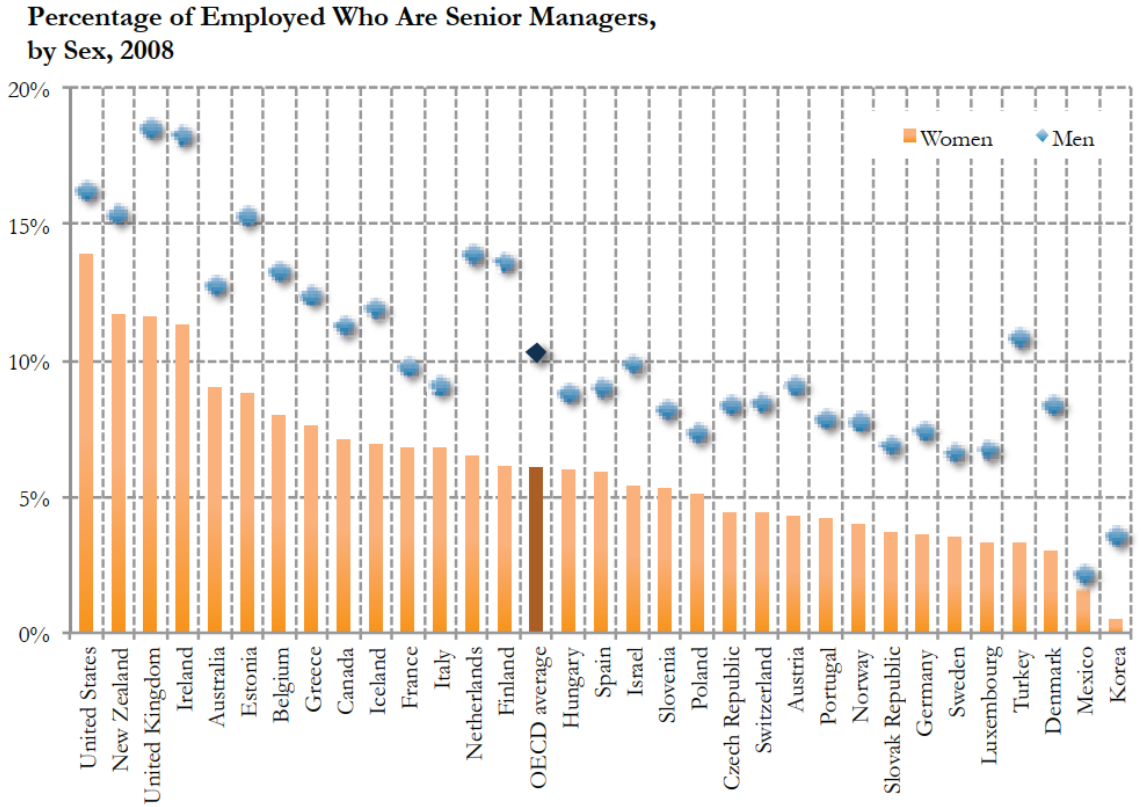
The 3D Chart (Left: Original, Right: Revised)





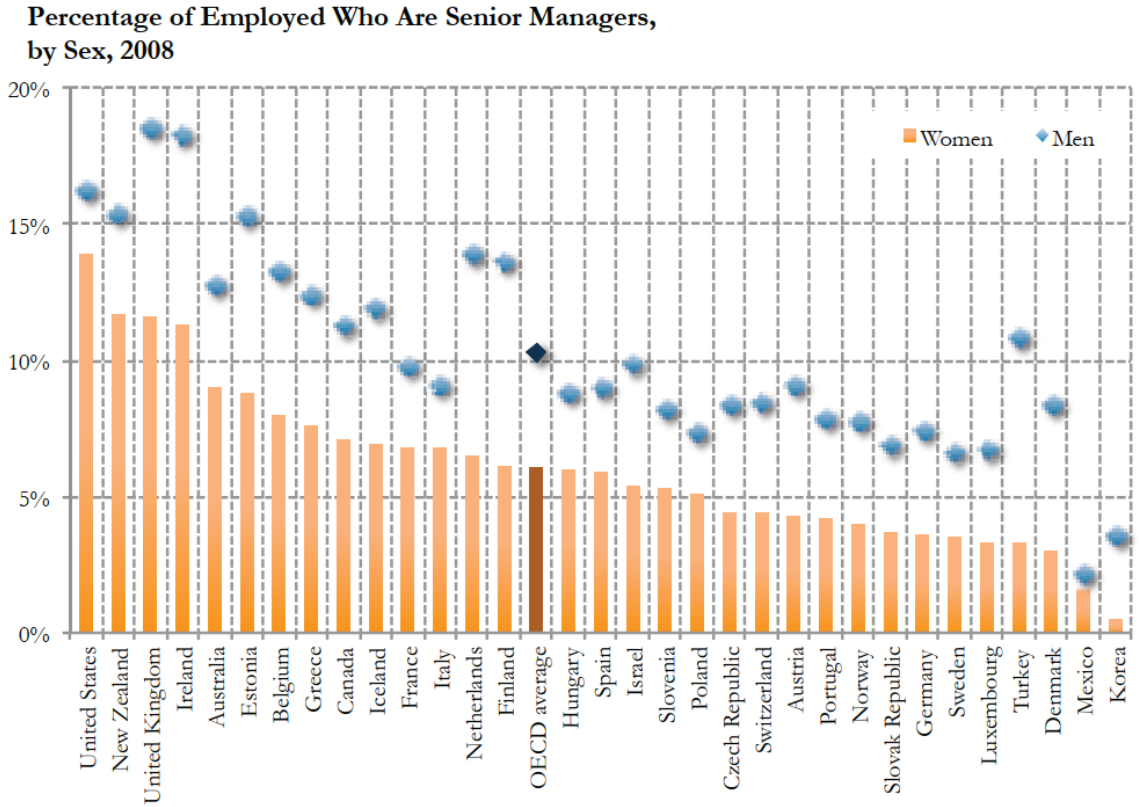
- Cancel the 3D treatment;
- Integrate the legend with the graph;
- Identify the unit below the title.

The Unbalanced Chart



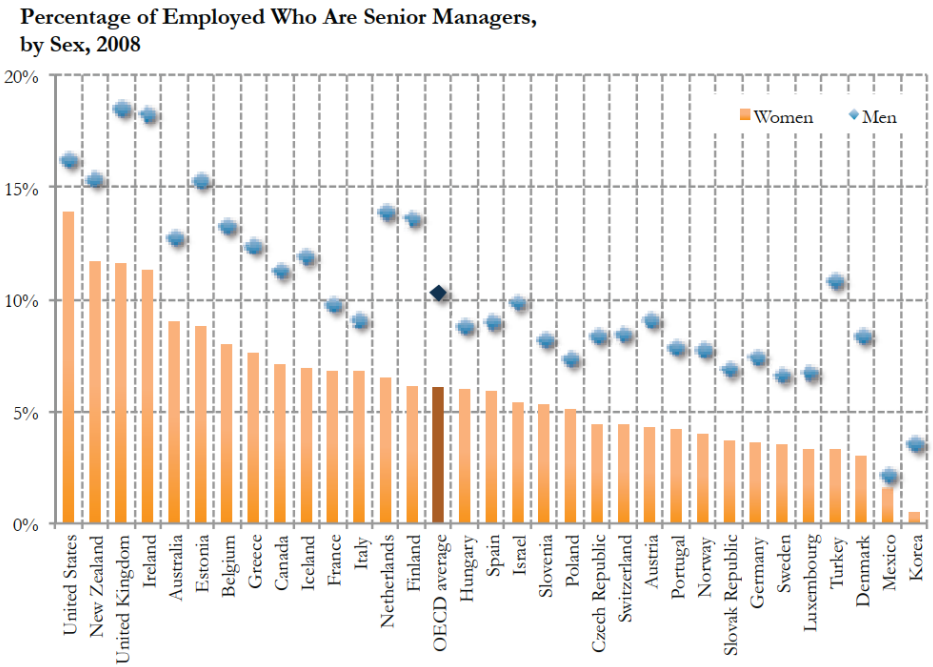
- 1) Two types of encoding (diamonds and bars) for the same type of data (overemphasizing women's data);
- 2) No visual connection between men's data points and women's;
- 3) Unnecessary color differences;

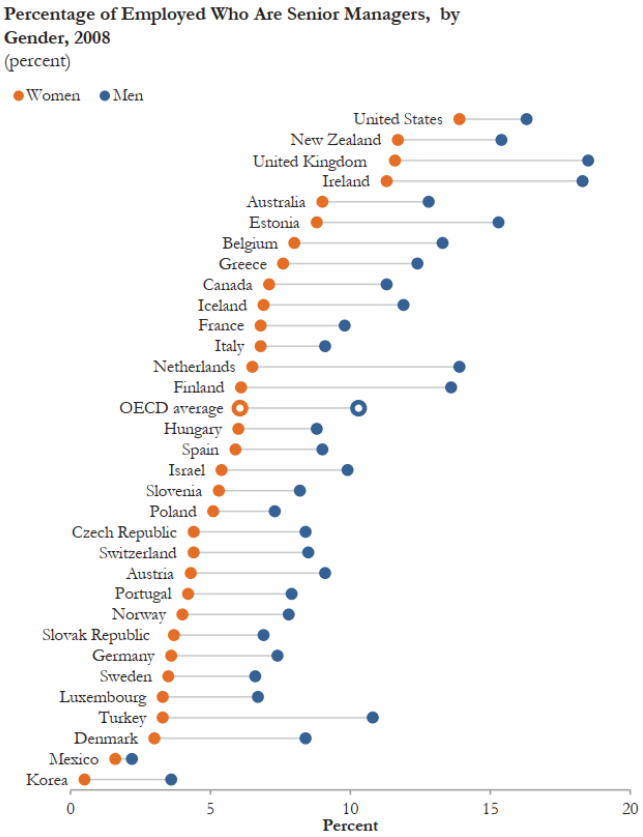
The Unbalanced Chart



- 4) Unnecessary gradient color shading for bars;
- 5) Too many heavy gridlines;
- 6) Redundant percentage signs on y-axis;
- 7) Vertical x-axis labels.

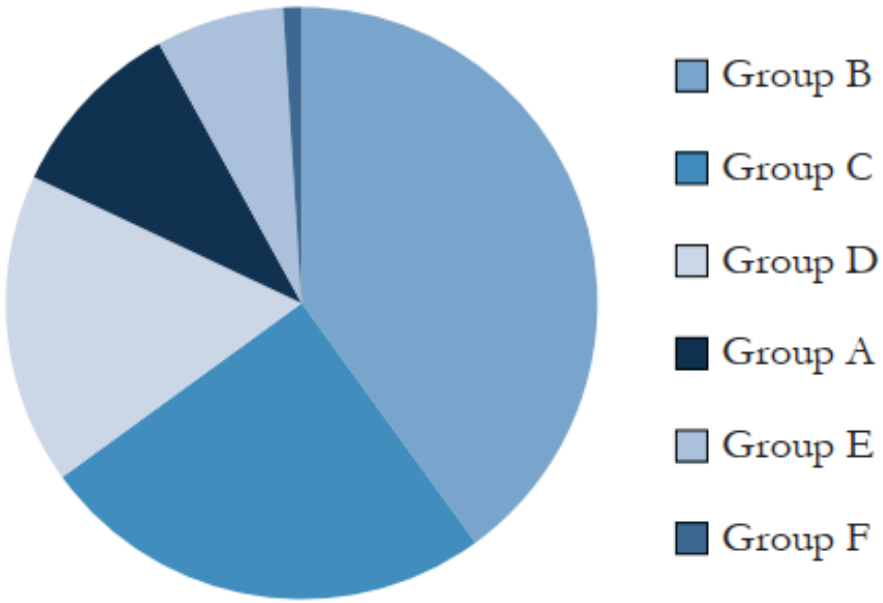
The Unbalanced Chart (Left: Original, Right: Revised)





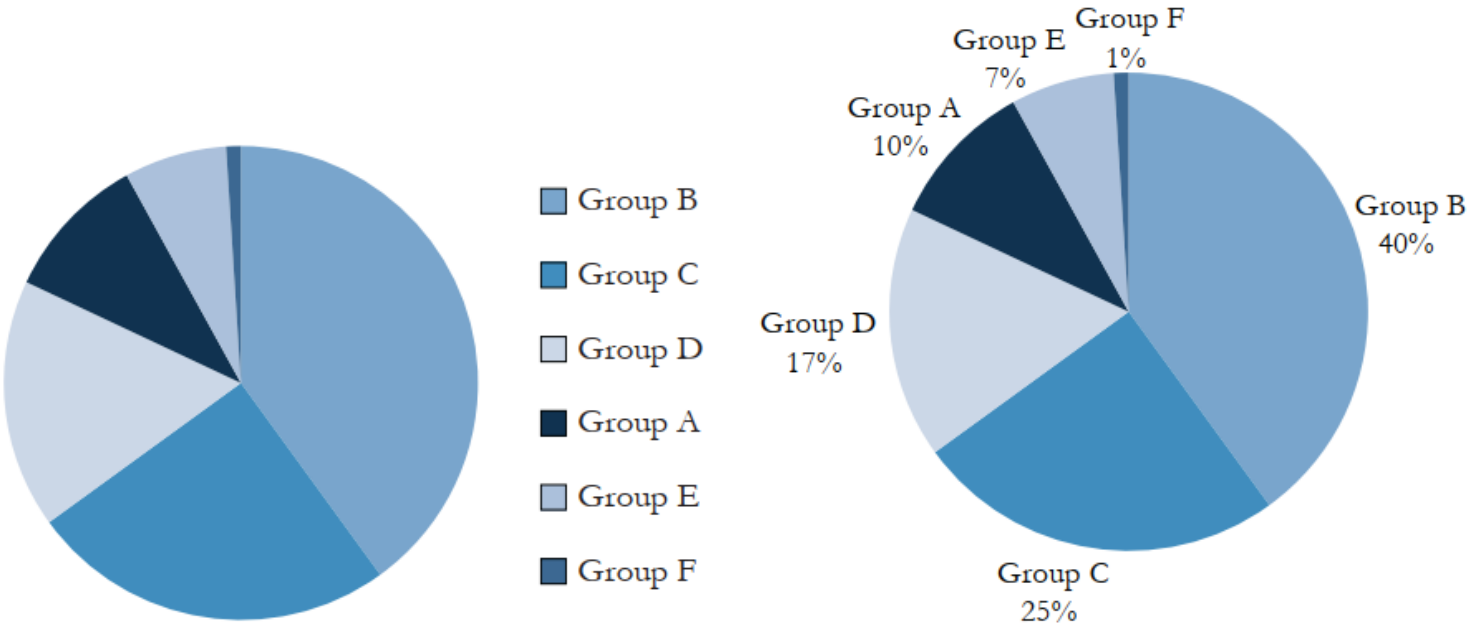
- Encode the data similarly for men and women for comparison;
- Integrate the title, units, and legend;
- Rotate country labels horizontally;
- Use unfilled circle for the average value.

The Pie Chart



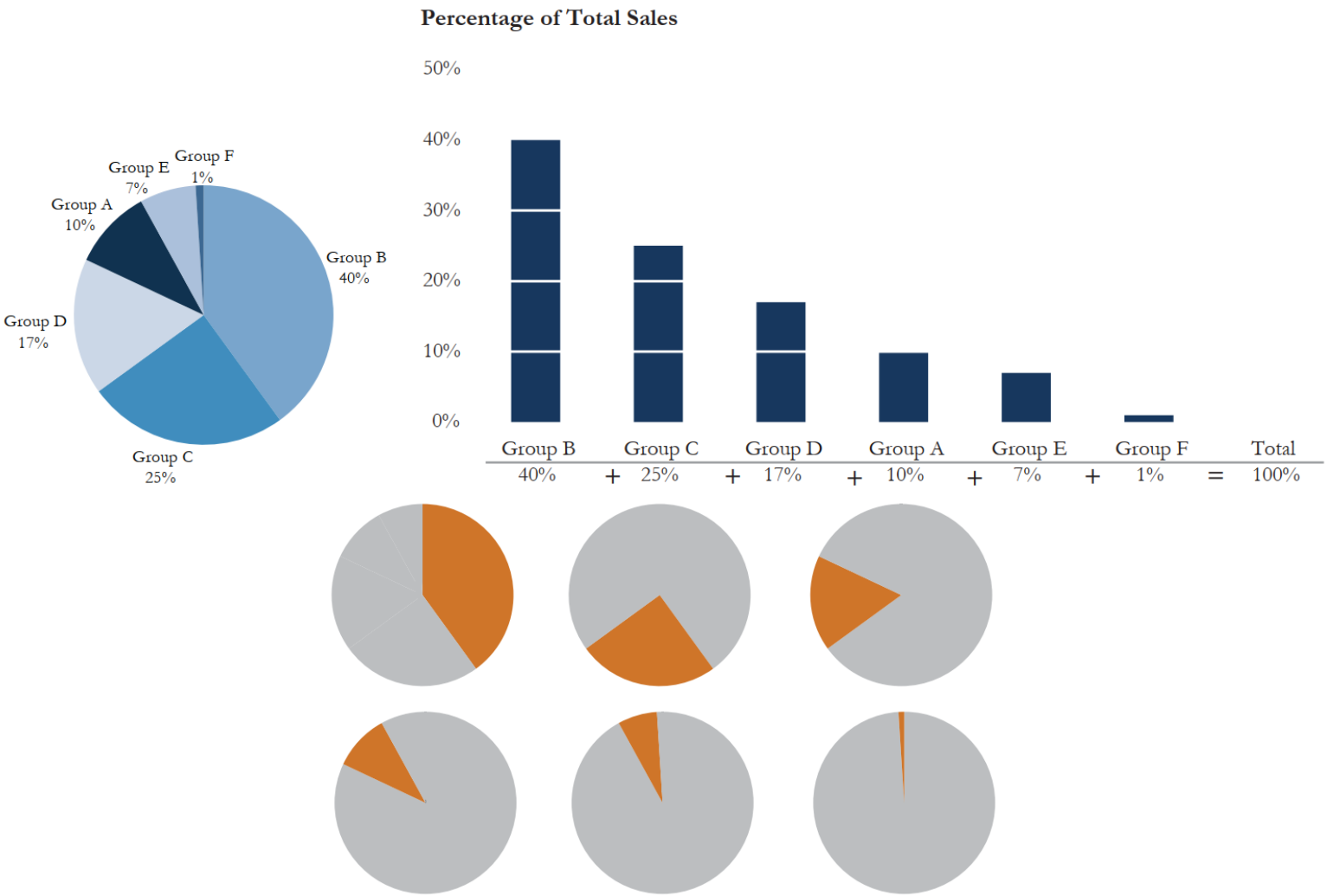
- Unknown shares of the pie.

The Pie Chart (Left: Original, Right: Revised)



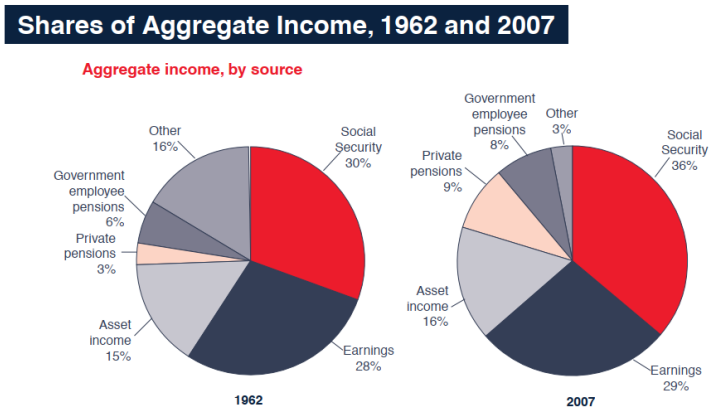
- Replace the legend with labels;
- Add percentage values to integrate the data.

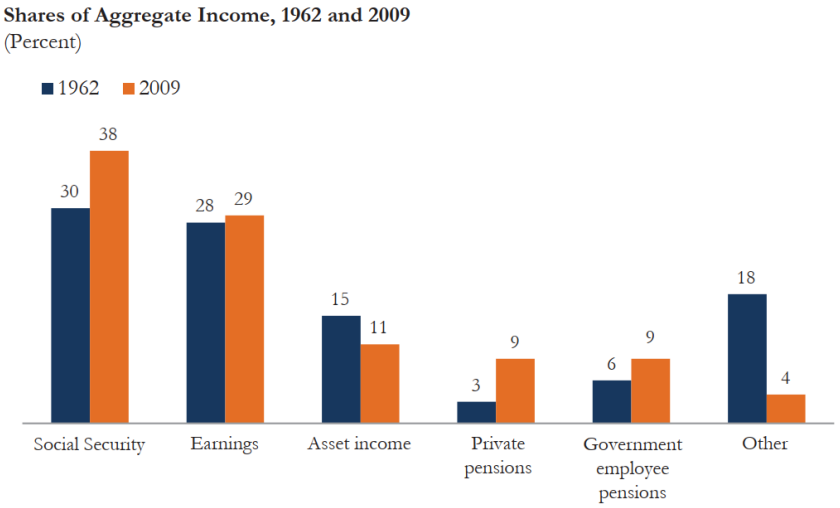
Alternatives for the Pie Chart

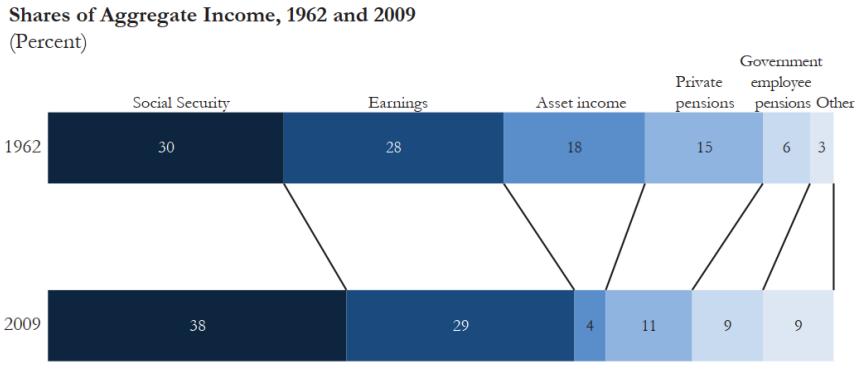


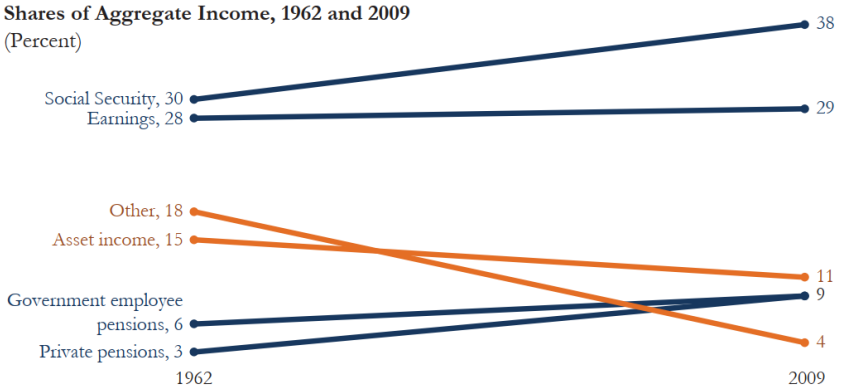
Pie chart vs. Bar chart vs. Part-to-whole mini-pie charts

Alternatives for the Paired Pie Chart









Paired pie chart vs. Paired bar chart vs. Stacked bar chart vs. Slope chart

Stata Cheat Sheet, Data Visualization

Data Visualization with Stata Cheat Sheet

For more info, see Stata's reference manual (stata.com)

Basic Plot Syntax:

graph <plot type> variables: y first $y_1\ y_2 \ldots y_n$ x **[in] [if],** plot-specific options <plot options> **– facet – by(var)** annotations **xline(**xint**) yline(**yint**) text(**y x "annotation"**)**

titles **title("**title**") subtitle("**subtitle**") xtitle("**x-axis title**") ytitle("**y axis title**")** axes **xscale(range(**low high**) log reverse off noline) yscale(**<options>**)**

custom appearance <marker, line, text, axis, legend, background options> **scheme(**s1mono**) play(**customTheme**) xsize(**5**) ysize(**4**) saving("**myPlot.gph**", replace)** plot size save

One Variable

sysuse auto, clear

Continuous

histogram mpg, **width(**5**) freq kdensity kdenopts(bwidth(**5**))**
histogram
bin(#) • width(#) • density • fraction • frequency • percent • addlabels addlabopts(<options>) • normal • normopts(<options>) • kdensity kdenopts(<options>)

kdensity mpg, **bwidth(**3**)**
smoothed histogram
bwidth • kernel(<options> ← main plot-specific options; see help for complete set
normal • normopts(<line options>)

Discrete

graph bar (count**), over(**foreign**, gap(***0.5**)) intensity(***0.5**)**
bar plot graph hbar draws horizontal bar charts
(asis) • (percent) • (count) • over(<variable>, <options: gap(*#) • relabel • descending • reverse>) • cw •missing • nofill • allcategories • percentages • stack • bargap(#) • intensity(*#) • valternate • xalternate

graph bar (percent**), over(**rep78**) over(**foreign**)**
grouped bar plot graph hbar ...
(asis) • (percent) • (count) • over(<variable>, <options: gap(*#) • relabel • descending • reverse>) • cw •missing • nofill • allcategories • percentages • stack • bargap(#) • intensity(*#) • valternate • xalternate

Discrete X, Continuous Y

graph bar (median**) price, over(**foreign**)** graph hbar ...
bar plot (asis) • (percent) • (count) • (stat: mean median sum min max ...) over(<variable>, <options: gap(*#) • relabel • descending • reverse sort(<variable>)>) • cw • missing • nofill • allcategories • percentages stack • bargap(#) • intensity(*#) • valternate • xalternate

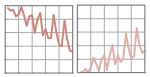
graph dot (mean**) length headroom, over(**foreign**) m(**1, ms(S)**)**
dot plot (asis) • (percent) • (count) • (stat: mean median sum min max ...) over(<variable>, <options: gap(*#) • relabel • descending • reverse sort(<variable>)>) • cw • missing • nofill • allcategories • percentages linegap(#) • marker(#, <options>) • linetype(dot | line | rectangle) dots(<options>) • lines(<options>) • rectangles(<options>) • rwidth

graph hbox mpg, **over(**rep78, descending**) by(**foreign**) missing**
box plot graph box draws vertical boxplots
over(<variable>, <options: total • gap(*#) • relabel • descending • reverse sort(<variable>)>) • missing • allcategories • intensity(*#) • boxgap(#) medtype(line | line | marker) • medline(<options>) • medmarker(<options>)

violplot price, **over(**foreign**)** ssc install violplot
violin plot over(<variable>, <options: total • missing>) • nofill • vertical • horizontal • obs • kernel(<options>) • bwidth(#) • barwidth(#) • dscale(#) • vgap(#) • pgap(#) • density(<options>) bar(<options>) • median(<options>) • obsopts(<options>)

Plot Placement

Juxtapose (Facet)



twoway scatter mpg price, **by(**foreign, norescale**)**
total • missing • colfirst • rows(#) • cols(#) • holes(<numlist>) compact • [no]edgelabel • [no]rescale • [no]yrescal • [no]xrescale [no]iyaxes • [no]ixaxes • [no]iytick • [no]ixtick • [no]iylabel [no]ixlabel • [no]iytitle • [no]ixtitle • imargin(<options>)

Superimpose

graph combine plot1.gph plot2.gph...
combine two or more saved graphs into a single plot

scatter y3 y2 y1 x, msymbol(i o i**) mlabel(**var3 var2 var1**)**
plot several y values for a single x value

graph twoway scatter mpg price in 27/74 || **scatter** mpg price /* */ if mpg < 15 & price > 12000 in 27/74, mlabel(make) m(i)
combine twoway plots using ||

Two+ Continuous Variables

graph matrix mpg price weight, **half**
scatterplot of each combination of variables
half • jitter(#) • jitterseed(#)
diagonal • [aweights(<variable>)]

twoway scatter mpg weight, **jitter(**7**)**
scatterplot
jitter(#) • jitterseed(#) • sort • cmissing(yes | no) connect(<options>) • [aweight(<variable>)]

twoway scatter mpg weight, **mlabel(**mpg**)**
scatterplot with labelled values
jitter(#) • jitterseed(#) • sort • cmissing(yes | no) connect(<options>) • [aweight(<variable>)]

twoway connected mpg price, **sort(**price**)**
scatterplot with connected lines and symbols
jitter(#) • jitterseed(#) • sort see also line
connect(<options>) • cmissing(yes | no)

twoway area mpg price, **sort(**price**)**
line plot with area shading
sort • cmissing(yes | no) • vertical, • horizontal base(#)

twoway bar price rep78
bar plot
vertical, • horizontal • base(#) • barwidth(#)

twoway dot mpg rep78
dot plot vertical, • horizontal • base(#) • ndots(#) dcolor(<color>) • dfcolor(<color>) • dlcolor(<color>) dsize(<markersize>) • dsymbol(<marker type>) dlwidth(<strokesize>) • dotextend(yes | no)

twoway dropline mpg price **in** 1/5
dropped line plot
vertical, • horizontal • base(#)

twoway rcapsym length headroom price
range plot (y_1 ÷ y_2) with capped lines
vertical • horizontal see also rcap

twoway rarea length headroom price, **sort**
range plot (y_1 ÷ y_2) with area shading
vertical • horizontal • sort
cmissing(yes | no)

twoway rbar length headroom price
range plot (y_1 ÷ y_2) with bars
vertical • horizontal • barwidth(#) • mwidth msize(<marker size>)

twoway pcspike wage68 ttl_exp68 wage88 ttl_exp88
Parallel coordinates plot **(sysuse nlswide1)**
vertical, • horizontal

twoway pccapsym wage68 ttl_exp68 wage88 ttl_exp88
Slope/bump plot **(sysuse nlswide1)**
vertical • horizontal • headlabel

Three Variables

twoway contour mpg price weight, **level(**20**) crule(**intensity**)**
3D contour plot
ccuts(#s) • levels(#) • minmax • crule(hue | chue | intensity | linear) • scolor(<color>) • ecolor (<color>) • ccolors(<colorlist>) • heatmap interp(thinplatespline | shepard | none)

regress price mpg trunk weight length turn, **nocons**
matrix regmat = **e(V)** ssc install plotmatrix
plotmatrix, mat(regmat**) color(**green**)**
heatmap mat(<variable>) • split(<options>) • color(<color>) • freq

Summary Plots

twoway mband mpg weight **|| scatter** mpg weight
plot median of the y values
bands(#)

binscatter weight mpg, **line(**none**)** ssc install binscatter
plot a single value (mean or median) for each x value
medians • nquantiles(#) • discrete • controls(<variables>) • linetype(lfit | qfit | connect | none) • aweight[<variable>]

Fitting Results

twoway lfitci mpg weight **|| scatter** mpg weight
calculate and plot linear fit to data with confidence intervals
level(#) • stdp • stdf • nofit • fitplot(<plottype>) • ciplot(<plottype>) • range(# #) • n(#) • atobs • estopts(<options>) • predopts(<options>)

twoway lowess mpg weight **|| scatter** mpg weight
calculate and plot lowess smoothing
bwidth(#) • mean • noweight • logit • adjust

twoway qfitci mpg weight, **alwidth(**none**) || scatter** mpg weight
calculate and plot quadratic fit to data with confidence intervals
level(#) • stdp • stdf • nofit • fitplot(<plottype>) • ciplot(<plottype>) • range(# #) • n(#) • atobs • estopts(<options>) • predopts(<options>)

Regression Results

regress price mpg headroom trunk length turn
coefplot, drop(_cons**) xline(**0**)** ssc install coefplot
Plot regression coefficients
baselevels • b(<options>) • at(<options>) • noci • levels(#) keep(<variables>) • drop(<variables>) • rename(<list>) horizontal • vertical • generate(<variable>)

regress mpg weight length turn
margins, eyex(weight**) at(**weight = (1800(200)4800)**)**
marginsplot, noci
Plot marginal effects of regression
horizontal • noci

Laura Hughes (lhughes@usaid.gov) • Tim Essam (tessam@usaid.gov) follow us @flaneuseks and @StataRGIS

inspired by RStudio's awesome Cheat Sheets (rstudio.com/resources/cheatsheets)

geocenter.github.io/StataTraining updated July 2019

Disclaimer: we are not affiliated with Stata. But we like it. CC BY 4.0

Stata Cheat Sheet, Ploting in Stata

Plotting in Stata

Customizing Appearance

For more info, see Stata's reference manual (stata.com)

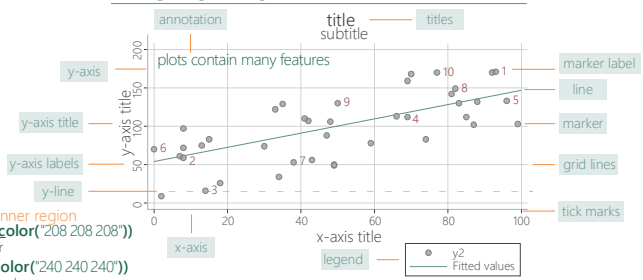
graph region
inner graph region
inner plot region
plot region

outer region
inner region

scatter price mpg, **graphregion(fcolor**("192 192 192") **ifcolor**("208 208 208"))
specify the fill of the background in RGB or with a Stata color

scatter price mpg, **plotregion(fcolor**("224 224 224") **ifcolor**("240 240 240"))
specify the fill of the plot background in RGB or with a Stata color

Anatomy of a Plot



Symbols

Syntax

marker
<marker options>

arguments for the plot objects (in green) go in the options portion of these commands (in orange)
for example:
scatter price mpg, **xline**(20, **lwidth**(vthick))

Color

mcolor("145 168 208") **mcolor**(none)
specify the fill and stroke of the marker in RGB or with a Stata color
mfcolor("145 168 208") **mfcolor**(none)
specify the fill of the marker

Size / Thickness

msize(medium) specify the marker size:

ehuge, huge, vhuge, vlarge, large, medlarge, medium, medsmall, small, vsmall, tiny, vtiny

Appearance

msymbol(Dh) specify the marker symbol:

O, o, Oh, oh, D, d, Dh, dh, T, t, Th, th, S, s, Sh, sh, +, X, p, none, i

Position

jitter(#) randomly displace the markers
jitterseed(#) set seed

Lines / Borders

Syntax

line
<line options>
xline(...)
yline(...)

marker
<marker options>
legend
legend(region(...))

axes
xscale(...)
yscale(...)

tick marks
grid lines
xlabel(...)
ylabel(...)

Color

lcolor("145 168 208") **lcolor**(none)
specify the stroke color of the line or border
marker **mlcolor**("145 168 208")
tick marks **tlcolor**("145 168 208")
grid lines **glcolor**("145 168 208")

Size / Thickness

lwidth(medthick)
specify the thickness (stroke) of a line:
marker **mlwidth**(thin)
tick marks **tlwidth**(thin)
grid lines **glwidth**(thin)

vvvthick, vvthick, vthick, thick, medthick, medium, medthin, thin, vthin, vvthin, vvvthin, none

Appearance

line, axes, grid lines **lpattern**(dash) **glpattern**(dash) specify the line pattern

solid, longdash, longdash_dot, dash, shortdash, shortdash_dot, dot, dash_dot, blank

axes **noline** axes **off** no axis/labels
tick marks **noticks** tick marks **tlength**(2)
grid lines **nogrid** **nogmin** **nogmax**

Position

tick marks **xlabel**(#10, **tposition**(crossing))
number of tick marks, position (outside | crossing | inside)

Text

Syntax

marker label
<marker options>
annotation
text(...)

titles
title(...)
subtitle(...)
xtitle(...)
ytitle(...)

axis labels
xlabel(...)
ylabel(...)
legend
legend(...)

Color

color("145 168 208") **color**(none)
specify the color of the text
marker label **mlabcolor**("145 168 208")
axis labels **labcolor**("145 168 208")
adjust transparency by adding %#
mcolor("145 168 208 %20")

Size / Thickness

size(medsmall) specify the size of the text:
marker label **mlabsize**(medsmall)
axis labels **labsize**(medsmall)

Text: vhuge, huge, vlarge, large, medlarge, medium, medsmall, small, vsmall, tiny, half_tiny, third_tiny, quarter_tiny, minuscule

Appearance

marker label **mlabel**(foreign)
label the points with the values of the foreign variable
axis labels **nolabels**
no axis labels
axis labels **format**(%12.2f)
change the format of the axis labels
legend **off**
turn off legend
legend **label**(# "label")
change legend label text

Position

marker label **mlabposition**(5)
label location relative to marker (clock position: 0 – 12)

Apply Themes

Schemes are sets of graphical parameters, so you don't have to specify the look of the graphs every time.

Using a Saved Theme

twoway scatter mpg price, **scheme**(customTheme)

help scheme entries Create custom themes by saving options in a .scheme file
see all options for setting scheme properties

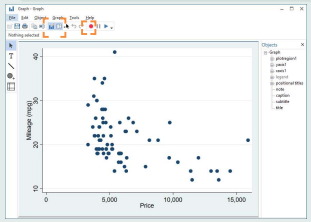
adopath ++ "~/<location>/StataThemes"
set path of the folder (StataThemes) where custom .scheme files are saved

set scheme customTheme, **permanently** set as default scheme
change the theme

net inst brewscheme, from("https://wbuchanan.github.io/brewscheme/") replace
install William Buchanan's package to generate custom schemes and color palettes (including ColorBrewer)

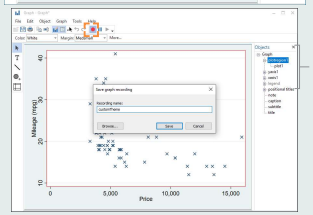
Using the Graph Editor

twoway scatter mpg price, **play**(graphEditorTheme)



Select the Graph Editor

Click Record



Double-click on symbols and areas on plot, or regions on sidebar to customize

Unclick Record

Save theme as a .grec file

Save Plots

graph twoway scatter y x, **saving**("myPlot.gph") **replace**
save the graph when drawing
graph save "myPlot.gph", **replace**
save current graph to disk
graph combine plot1.gph plot2.gph...
combine two or more saved graphs into a single plot
graph export "myPlot.pdf", **as**(.pdf)
export the current graph as an image file
see options to set size and resolution

Laura Hughes (lhughes@usaid.gov) • Tim Essam (tessam@usaid.gov)
follow us @flaneuseks and @StataRGIS

inspired by RStudio's awesome Cheat Sheets (rstudio.com/resources/cheatsheets)

geocenter.github.io/StataTraining updated July 2019

Disclaimer: we are not affiliated with Stata. But we like it.
CC BY 4.0

Exercise 1

Finish the Stata data visualization introduction "Data Visualization with Stata - Some Basic Graphs".

- https://people.umass.edu/biep640w/webpages/demonstrations.html

Exercise 2

Finish the Stata data visualization exercises "Intro to data visualization".

- Use the .do file revised and shared by me.
- https://dss.princeton.edu/training/Visual101.pdf

Table of Contents

Diagrams

A diagram is a symbolic representation of information using visualization techniques. There are mainly three types of diagrams:

- Quantitative diagrams;
- Conceptual or logical diagrams;
- Schematics.

Diagrams

1) Quantitative diagrams

Display a relationship between two variables (discrete or continuous):

- Line chart;
- Bar chart;
- Pie chart;
- Scatter plot;
- Histogram, etc.

They are often used in data visualization in empirical studies.

Diagrams

2) Conceptual (or logical) diagrams

Express relationships as connections between the elements or overlaps between the elements:

- Tree diagram;
- Network diagram;
- Flowchart;
- Venn diagram, etc.

They are often used in economic modelling in theoretical studies.

Diagrams

3) Schematic diagrams

Represent the elements by omitting all irrelevant details and using abstract graphic symbols rather than realistic pictures:

- Map, etc.

Tools

- 1) diagrams.net

  https://app.diagrams.net/

  "Security-first diagramming for teams."
- 2) Mathcha

  https://www.mathcha.io/

  "A fast way to write and share mathematics."
- 3) LaTeXpackage "TikZ"

  "TikZ is probably the most complex and powerful tool to create graphic elements in LaTeX."
- 4) LaTeXpackage "spmap"

  "Maps of the world and regions."

Tools

- 5) WebPlotDigitizer

  https://automeris.io/WebPlotDigitizer/

  "Web based tool to extract data from plots, images, and maps."
- 6) Mathpix Snip

  https://mathpix.com/

  "Convert images into LaTeX."
- 7) Jason Davies website

  https://www.jasondavies.com/

  Data visualisations (word cloud, word tree etc.)

Exercise 3

Reproduce the following tree diagram in the "diagrams.net":

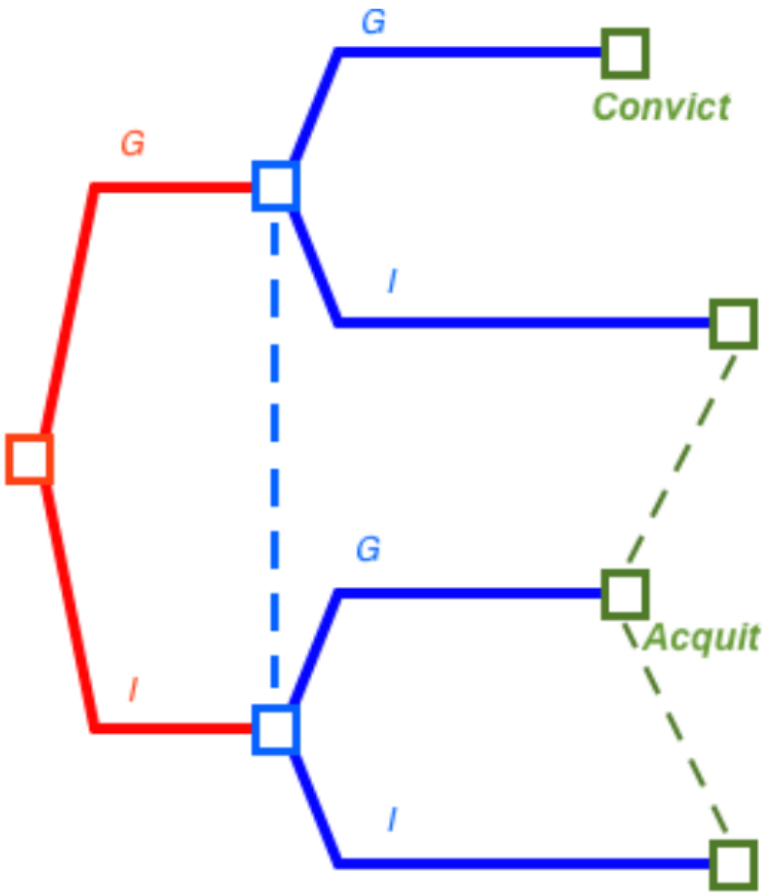
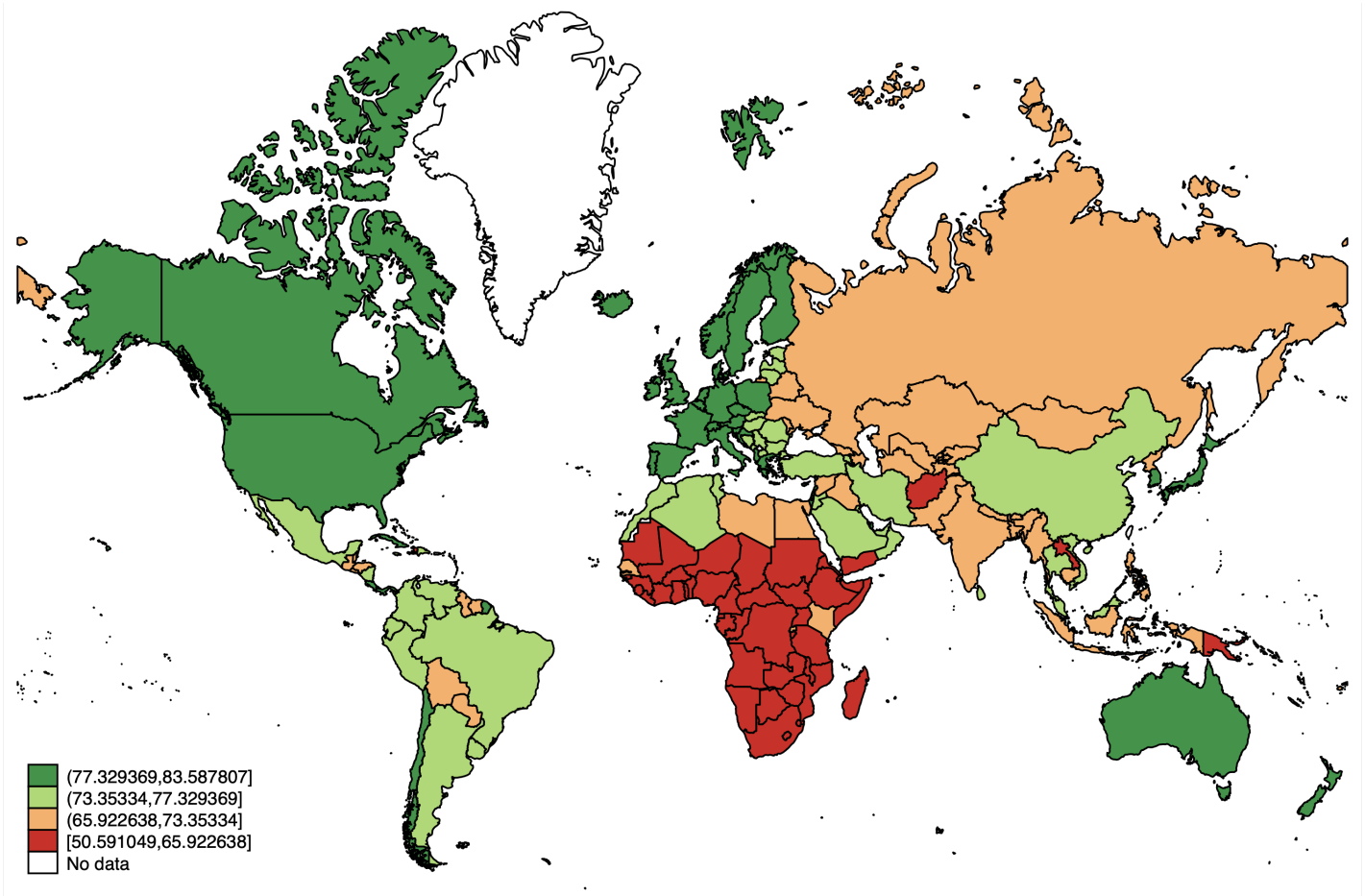


Figure 2

Exercise 4

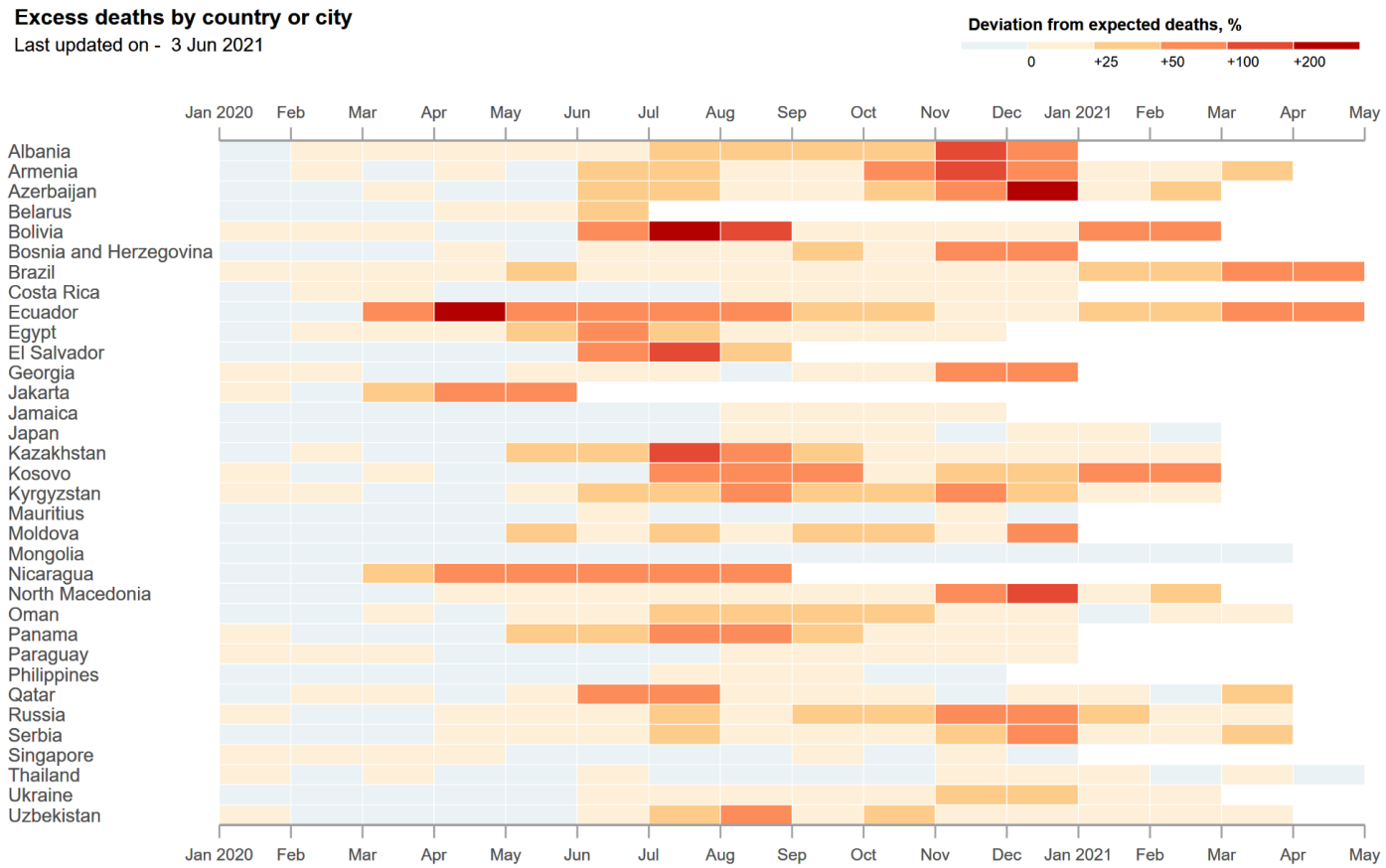
Finish the map graphing exercise with Stata *spmap* package:

- https://www.stathelp.se/en/spmap_world_en.html



Exercise 5

Challenge yourself with the graph replication: https://medium.com/the-stata-guide/graph-replication-the-economists-covid-19-excess-death-graph-c6b2bc65f760



References


Carol Bigelow. "Data Visualization with Stata - Some Basic Graphs". In: (2019).


Jonathan A Schwabish. "An economist's guide to visualizing data". In: *Journal of Economic Perspectives* 28.1 (2014), pp. 209–34.


Oscar Torres-Reyna. "Getting Started in Data Analysis using Stata". In: *Princeton: Princeton University* (2007).


Wikipedia contributors. *Diagram — Wikipedia, The Free Encyclopedia*. [Online; accessed 2-March-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Diagram&oldid=1008857925.