Programming, Data Management and Visualization

Module E: **Data analysis & visualization**

Alexander Ahammer

Department of Economics, Johannes Kepler University, Linz, Austria
Christian Doppler Laboratory Ageing, Health, and the Labor Market, Linz, Austria

β version, *more or less complete*
Last updated: Monday 20th January, 2020 (13:27)

Introduction

- By now you should be capable of basic data organization and programming commands, you should know how to transform and combine data, and how to save and report results (+ how to make fancy tables and graphs).

- Our last topic will be **data analysis and visualization,** we will learn …
  - how good graphs and tables look like,
  - how good graphs and tables are done in Stata, and finally
  - some selected topics (such as geographical maps and how to do them)

- I assume you have the basic statistical knowledge (e.g., what are moments of a distribution, types of distributions, joint distributions, regression theory, and so forth) — what I teach in Econometrics I is totally sufficient.

- There are three main references I use for this chapter: (esp. the last one)
  - Tufte, E. (2007), *The Visual Display of Quantitative Information*, Graphics Press.
  - Schwabish, J.A. (2014), An Economist's Guide to Visualizing Data, *Journal of Economic Perspectives*, 28(1), 209–234.
  - Martin Halla, *How to make good graphs and tables,* slide set. [download]

E.1

How to present data

How to present data

- How do **good graphs** look like?
- How do **good tables** look like?

Good graphs

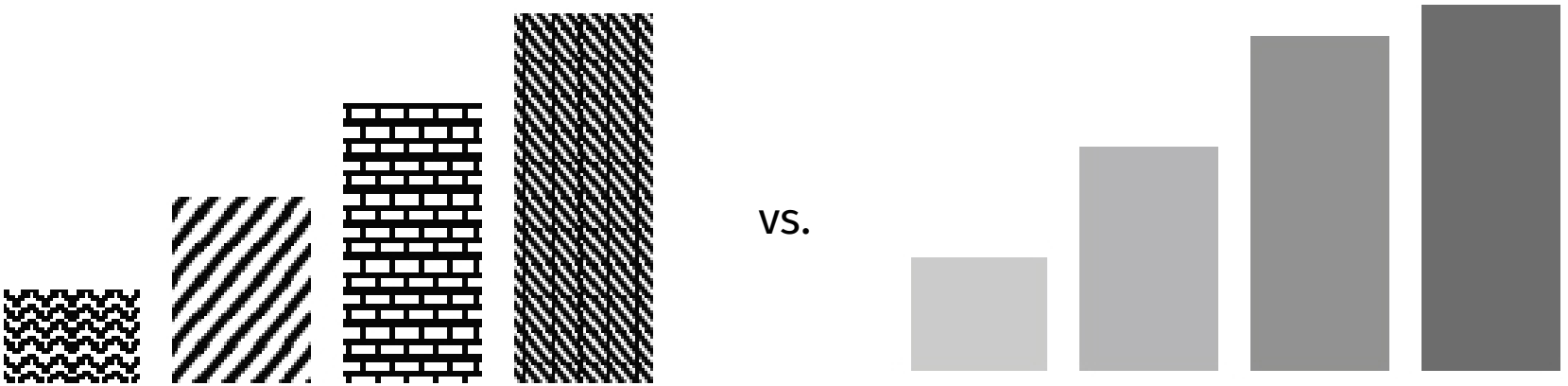
- There is a common theme in the references I provided before. They can be summarized as follows.
- **Garbage in—garbage out** $\longrightarrow$ good graphs *reveal* data, with as few theoretical/structural assumptions as possible.
  - *"Of course, statistical graphics, just like statistical calculations, are only as good as what goes into them. An ill-specified or preposterous model or a puny data set cannot be rescued by a graphic (or by calculation), no matter how clever or fancy."*
- Maximize **information–ink ratio,** reduce the clutter, and show the graph in the clearest way possible.
- Integrate the text and the graph $\longrightarrow$ graphs are constructed to **complement the text,** but should also contain enough information to **stand alone.**
- Standard graphs in Stata often don't fulfill these points. Download the `tufte` scheme from the SSC library.

Good graphs according to Tufte …

- **show the data** and **avoid distorting** what the data have to say
- induce the viewer to **think about the substance** rather than about methodology, graphic design, the technology of graphic production, or something else
- present **many numbers in a small space**
- make large data sets **coherent**
- encourage the eye to **compare different pieces of data**
- reveal the data at several **levels of detail,** from a broad overview to the fine structure
- serve a reasonably clear **purpose:** description, exploration, tabulation, or decoration
- be closely **integrated** with the statistical and verbal descriptions of a data set.

Reduce the clutter

Schwabish (2014, *JEP*)



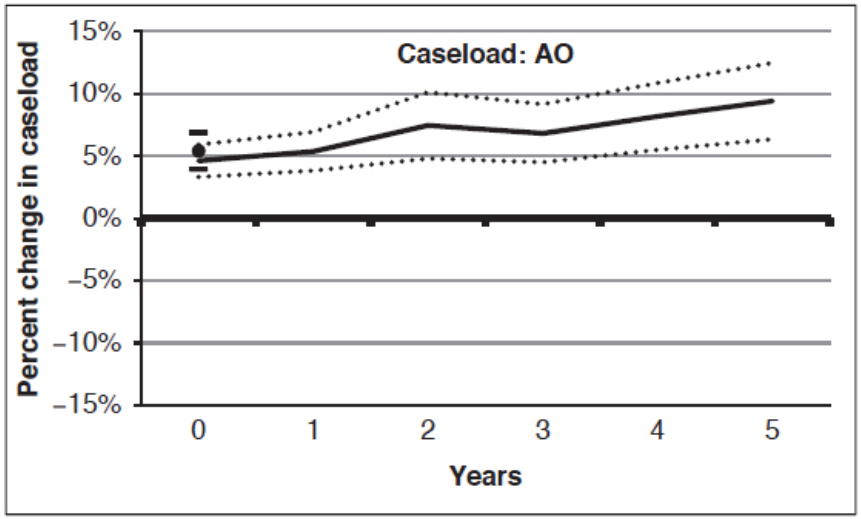
Option (a) vs. Option (b)

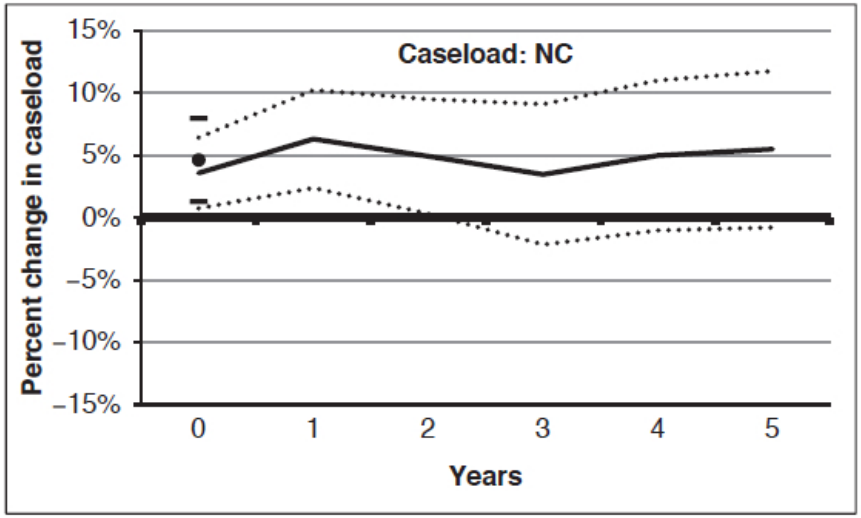
- Do not use the left option $\longrightarrow$ unnecessary clutter, only option (b) maximizes the information–ink ratio.

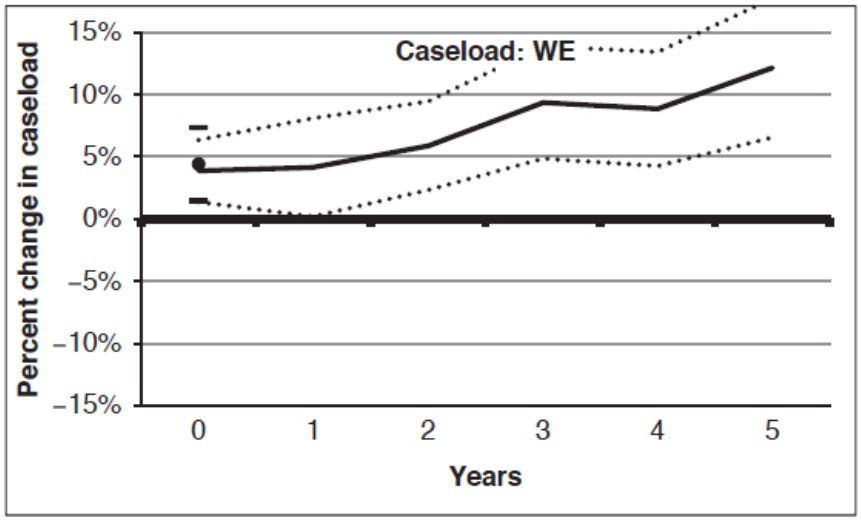
- Other examples of clutter:
  - dark or heavy gridlines
  - unnecessary tick marks, labels, or text
  - unnecessary icons or pictures
  - ornamental shading and gradients
  - unnecessary dimensions.

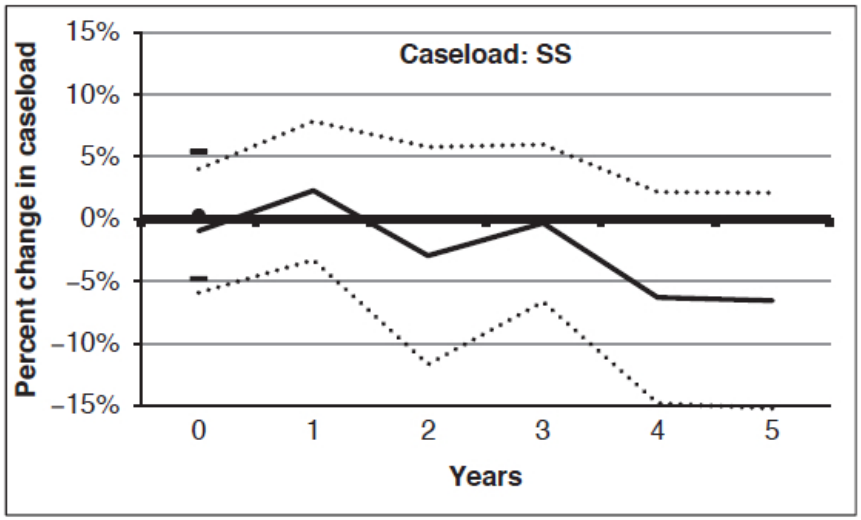
Some examples of good and bad graphs

Schwabish (2014, *JEP*)



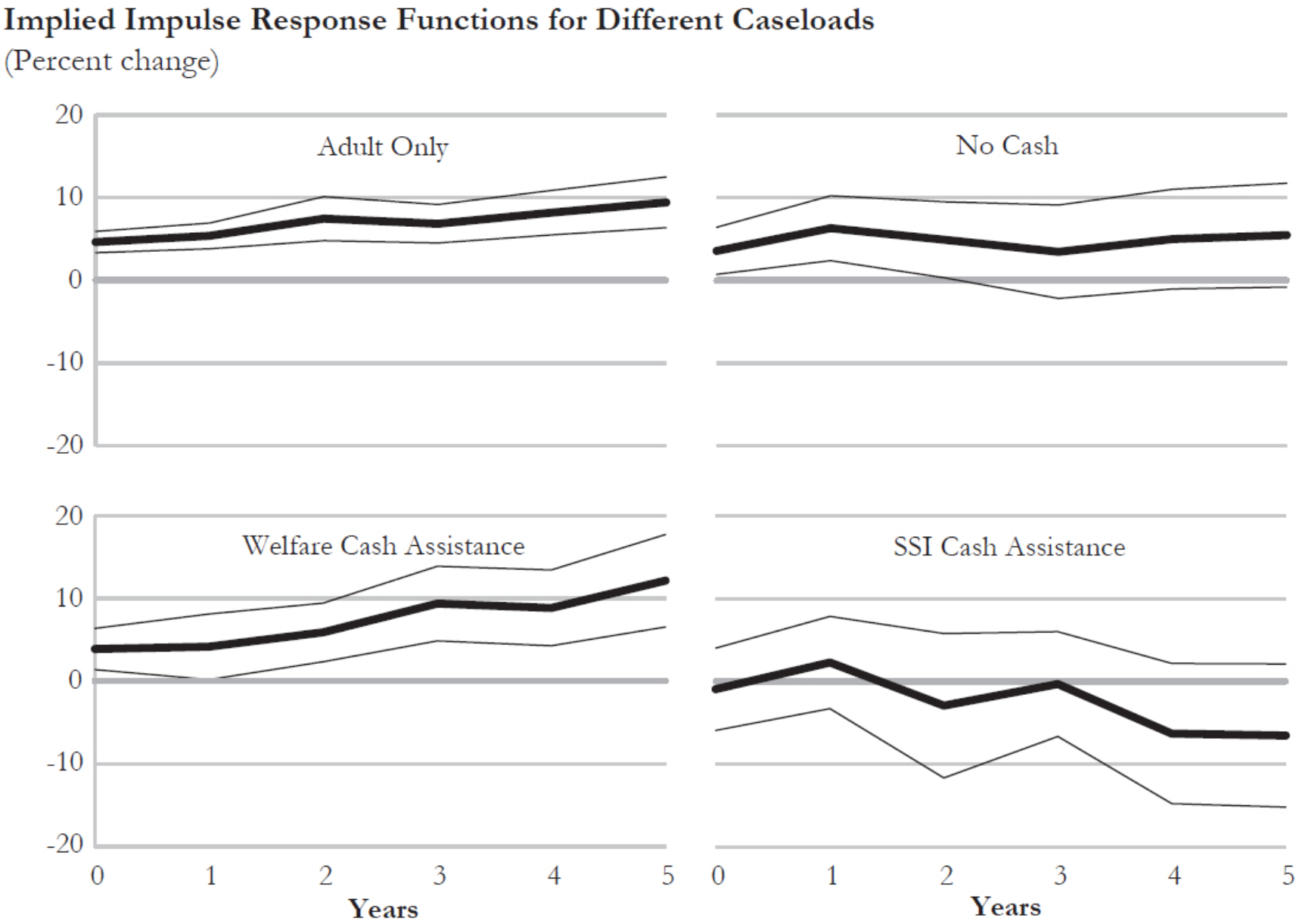






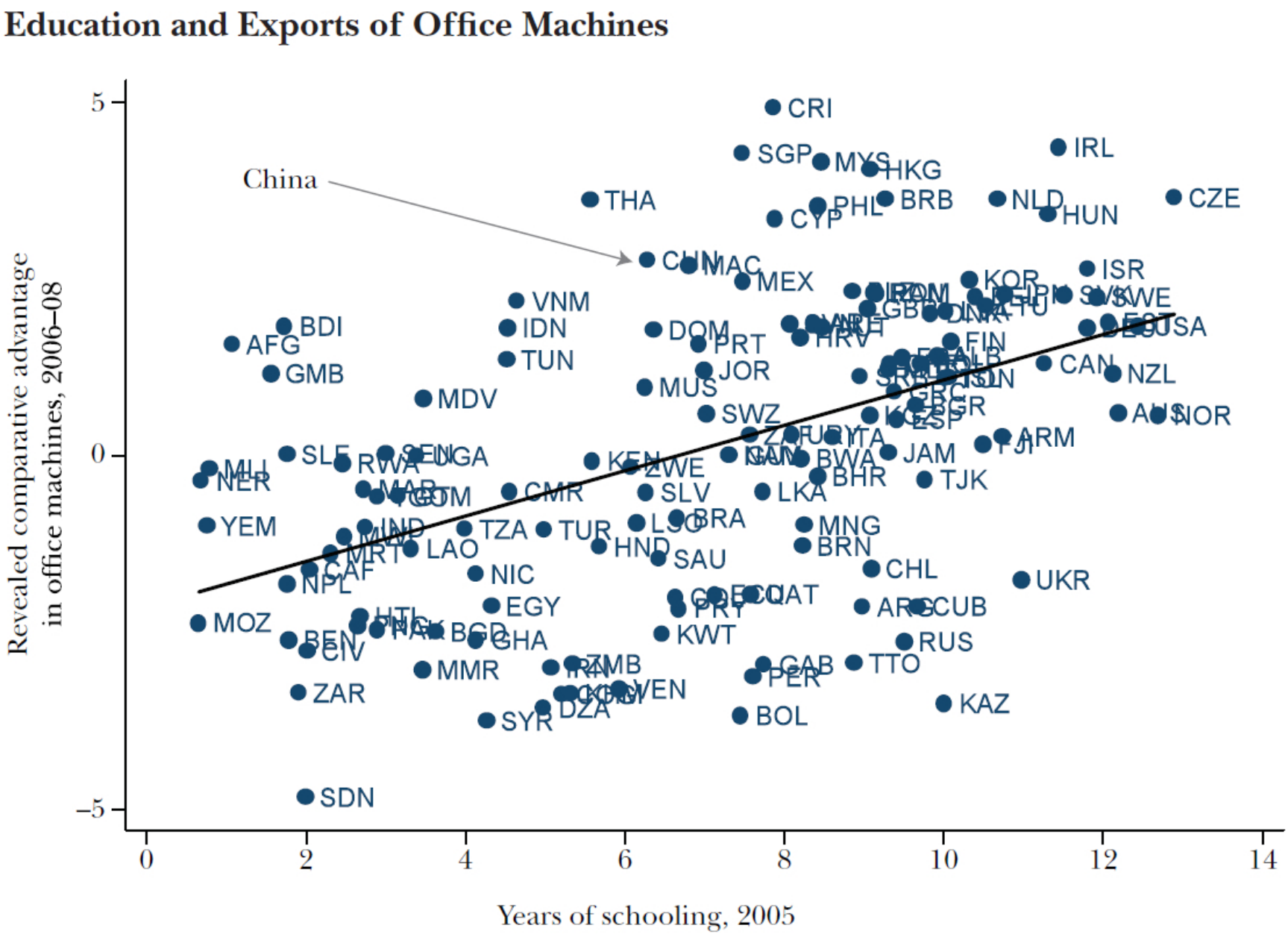
Some examples of good and bad graphs

Schwabish (2014, *JEP*)



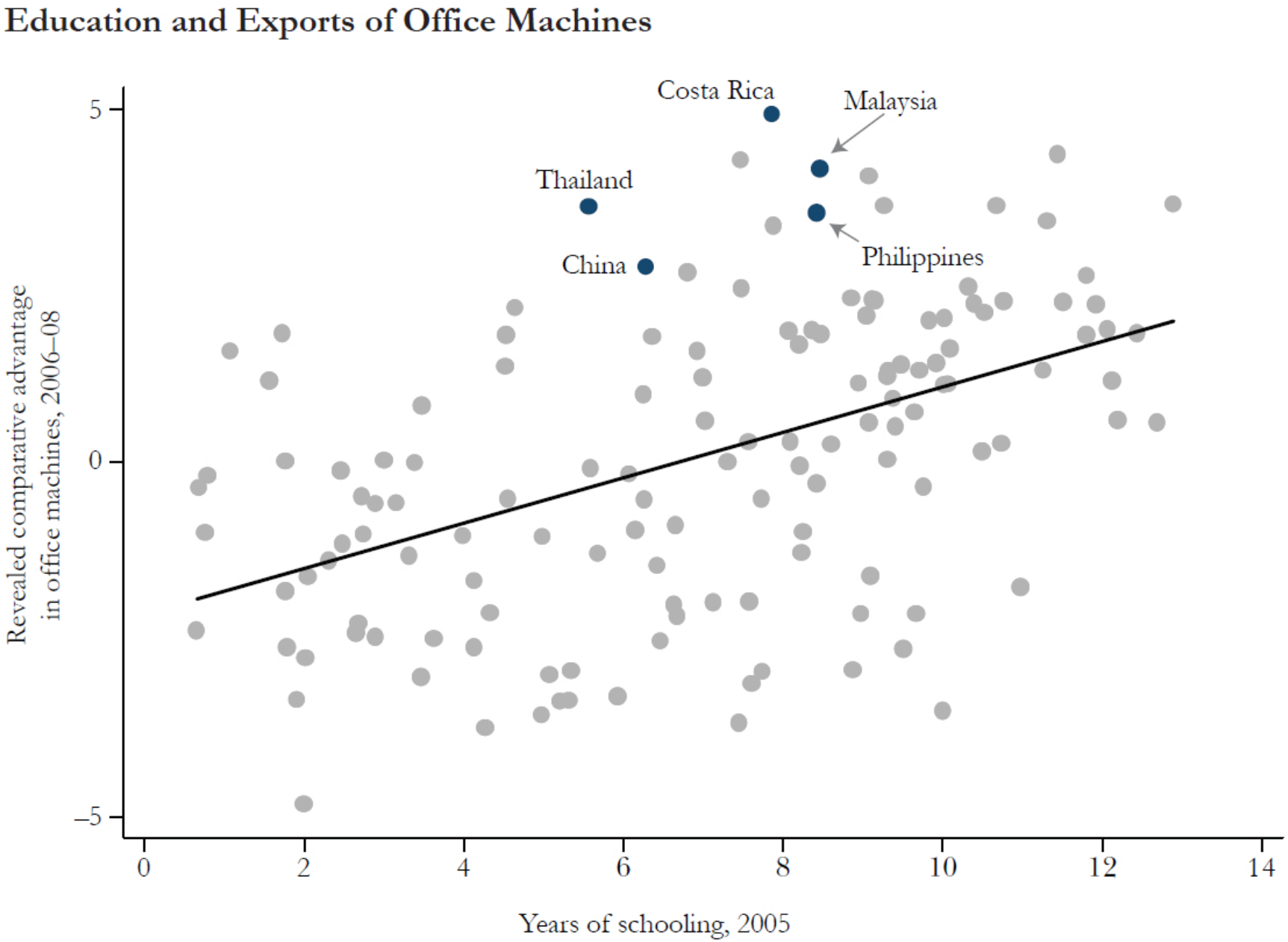
Some examples of good and bad graphs

Schwabish (2014, *JEP*)



Some examples of good and bad graphs

Schwabish (2014, *JEP*)

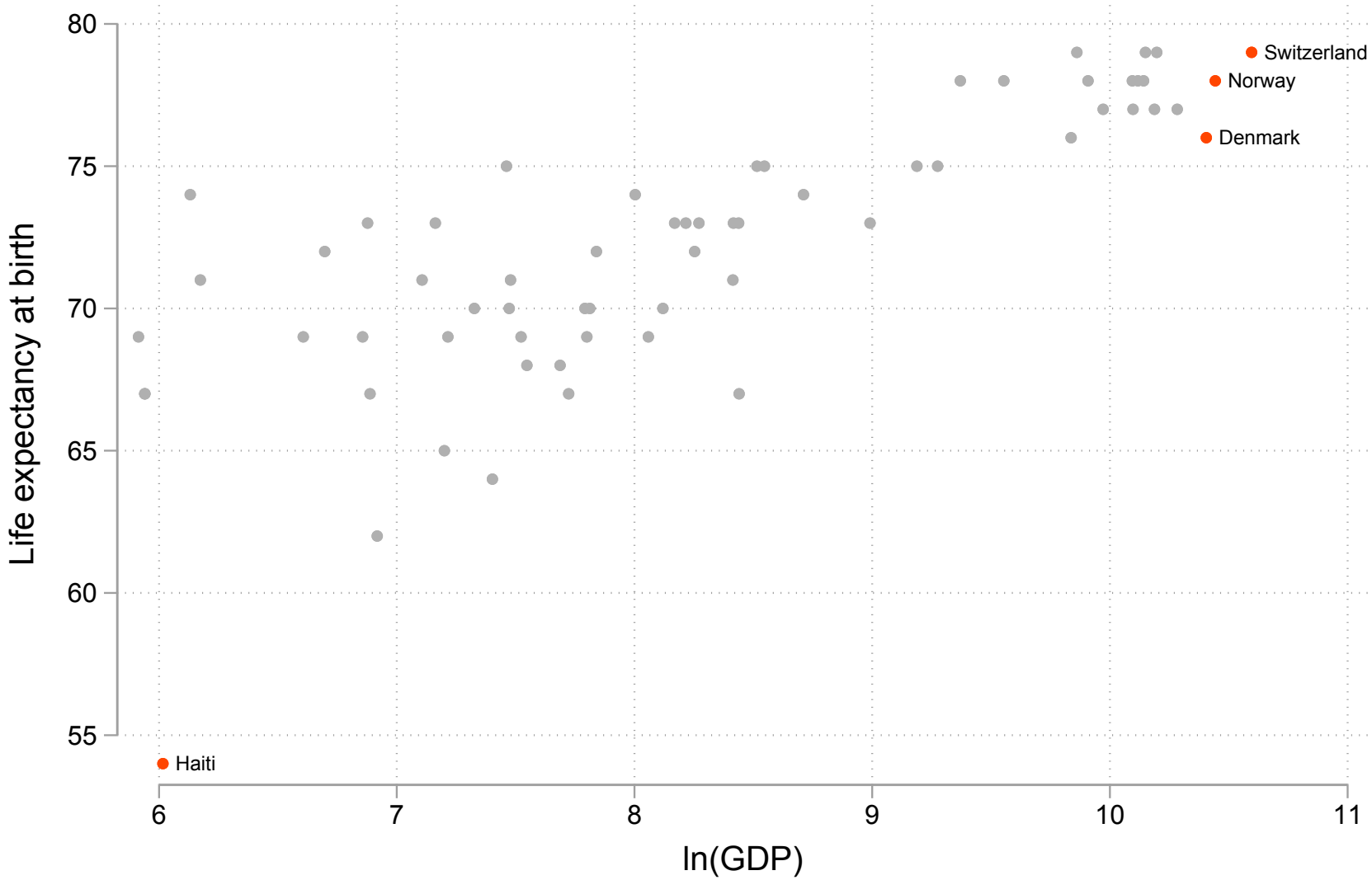


Intermezzo How can you draw such a graph?

```
. sysuse lifeexp.dta, clear
(Life expectancy, 1998)
. g lgnppc = ln(gnppc)
(5 missing values generated)
. g tag = inlist(country,"Haiti","Denmark","Norway","Switzerland")
. tw (scatter lexp lgnppc if tag == 0, msymbol(o) mcolor(gs11)) ///
>    (scatter lexp lgnppc if tag == 1, msymbol(o) mcolor("255 69 0") ///
>    mlab(country) mlabsize(vsmall) mlabpos(3)), xtitle("ln(GDP)") ///
>    legend(off)
. gr export "slides/graphs/tufte1.pdf", as(pdf) replace
(file slides/graphs/tufte1.pdf written in PDF format)
```

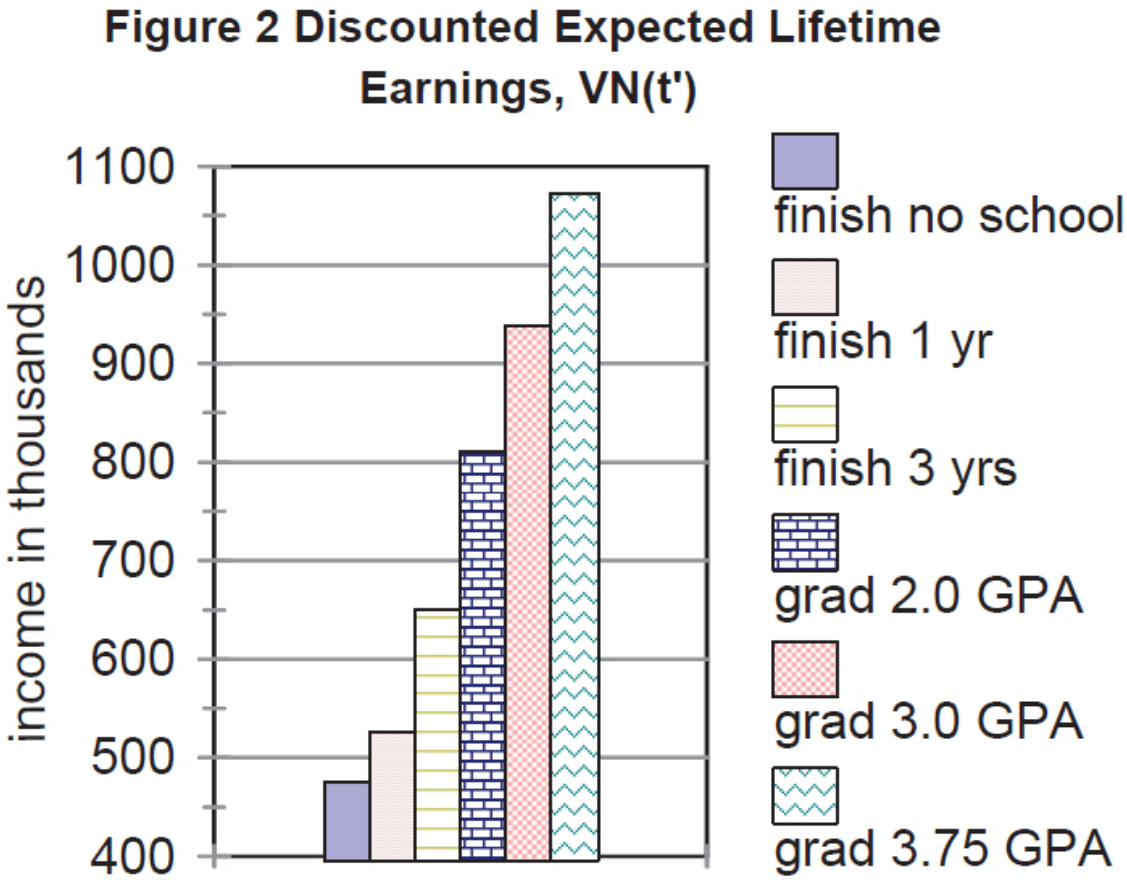
- It is essentially a set of overlaid scatterplots.
- Putting each label in a different position or using arrows to indicate labels is possible but tedious to code.
- *Exercise:* find a solution!

Intermezzo How can you draw such a graph?



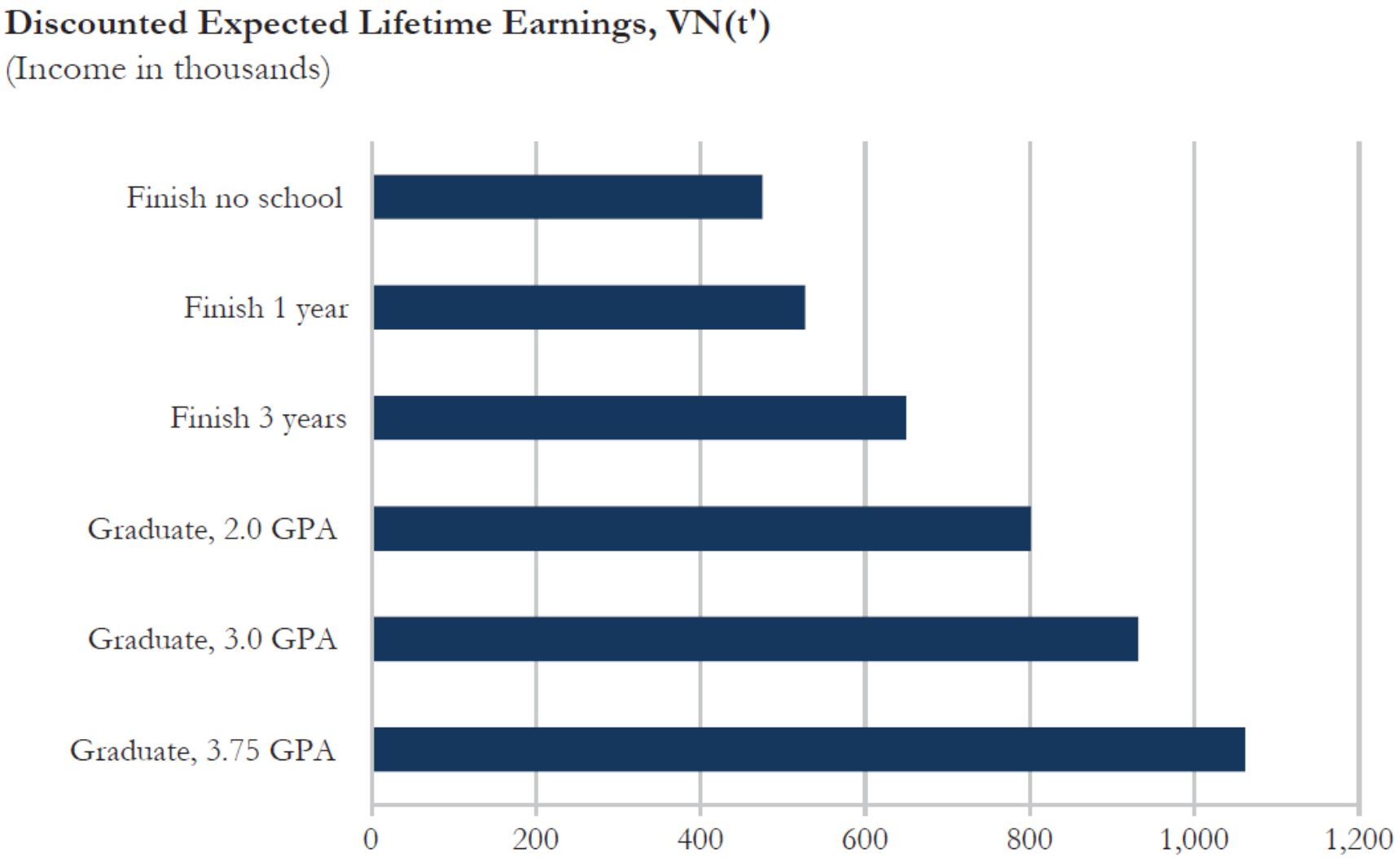
Some examples of good and bad graphs

Schwabish (2014, *JEP*)



Some examples of good and bad graphs

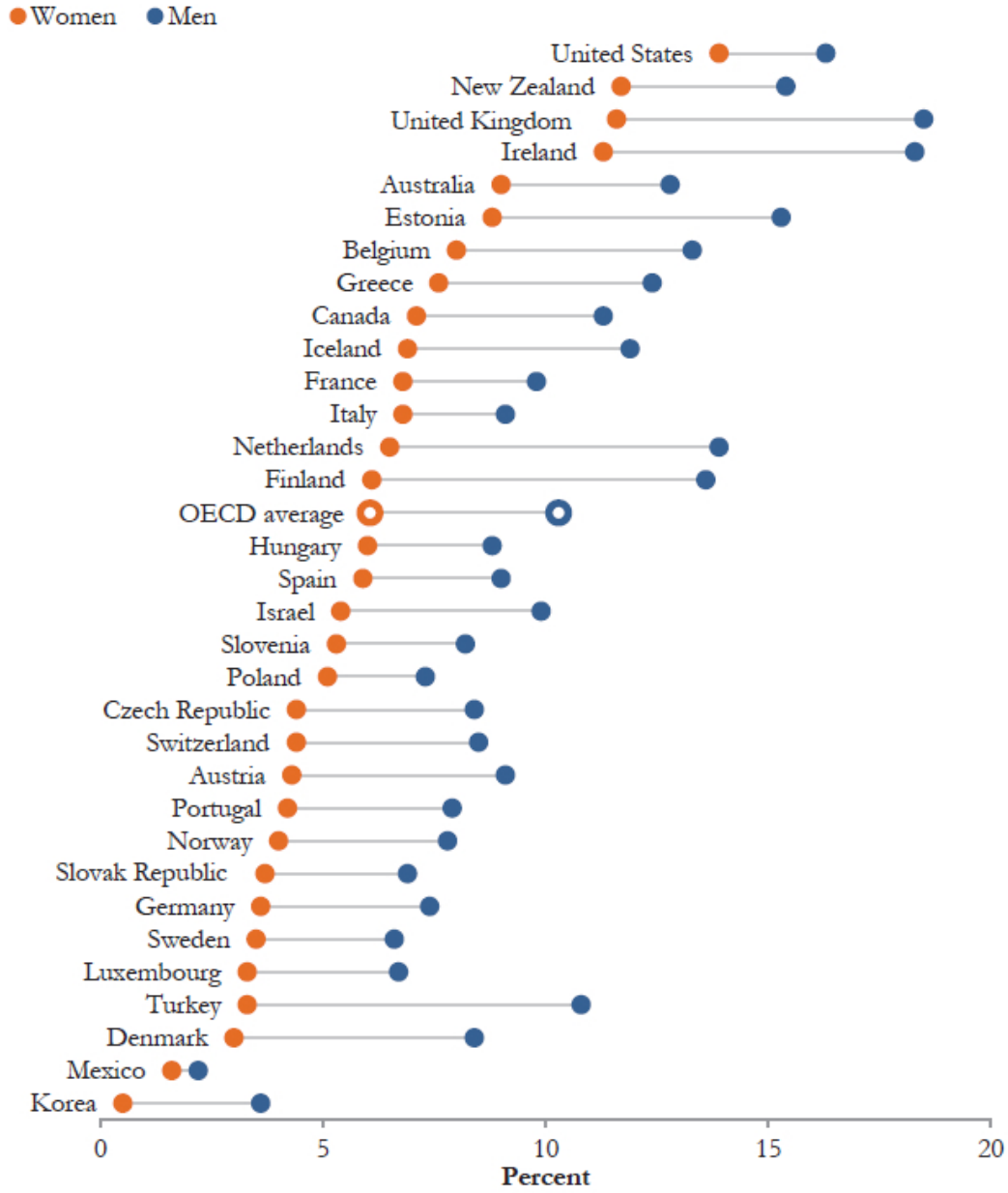
Schwabish (2014, *JEP*)



Some examples of good and bad graphs

Schwabish (2014, *JEP*)

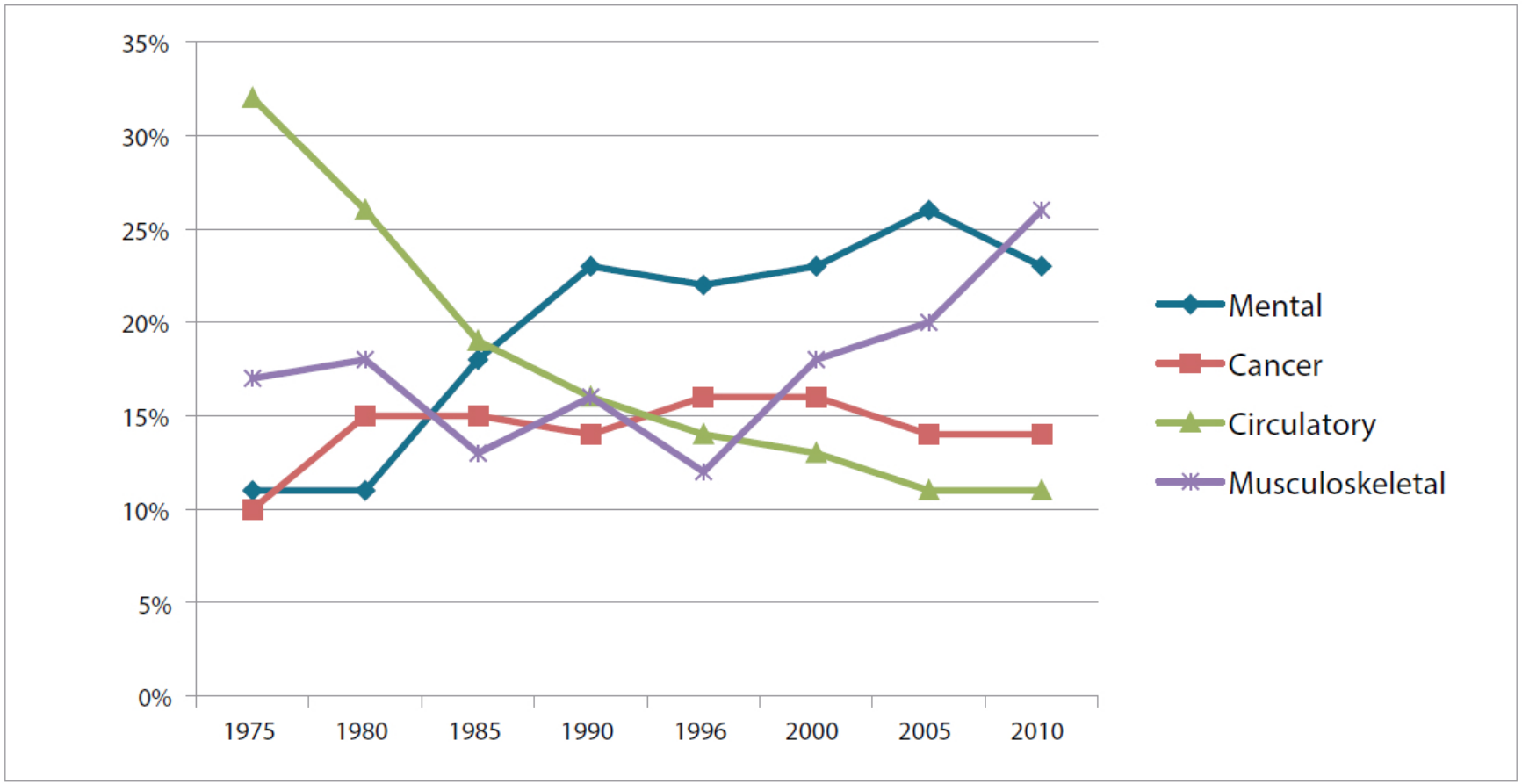
Percentage of Employed Who Are Senior Managers, by Gender, 2008
(percent)



The spaghetti chart

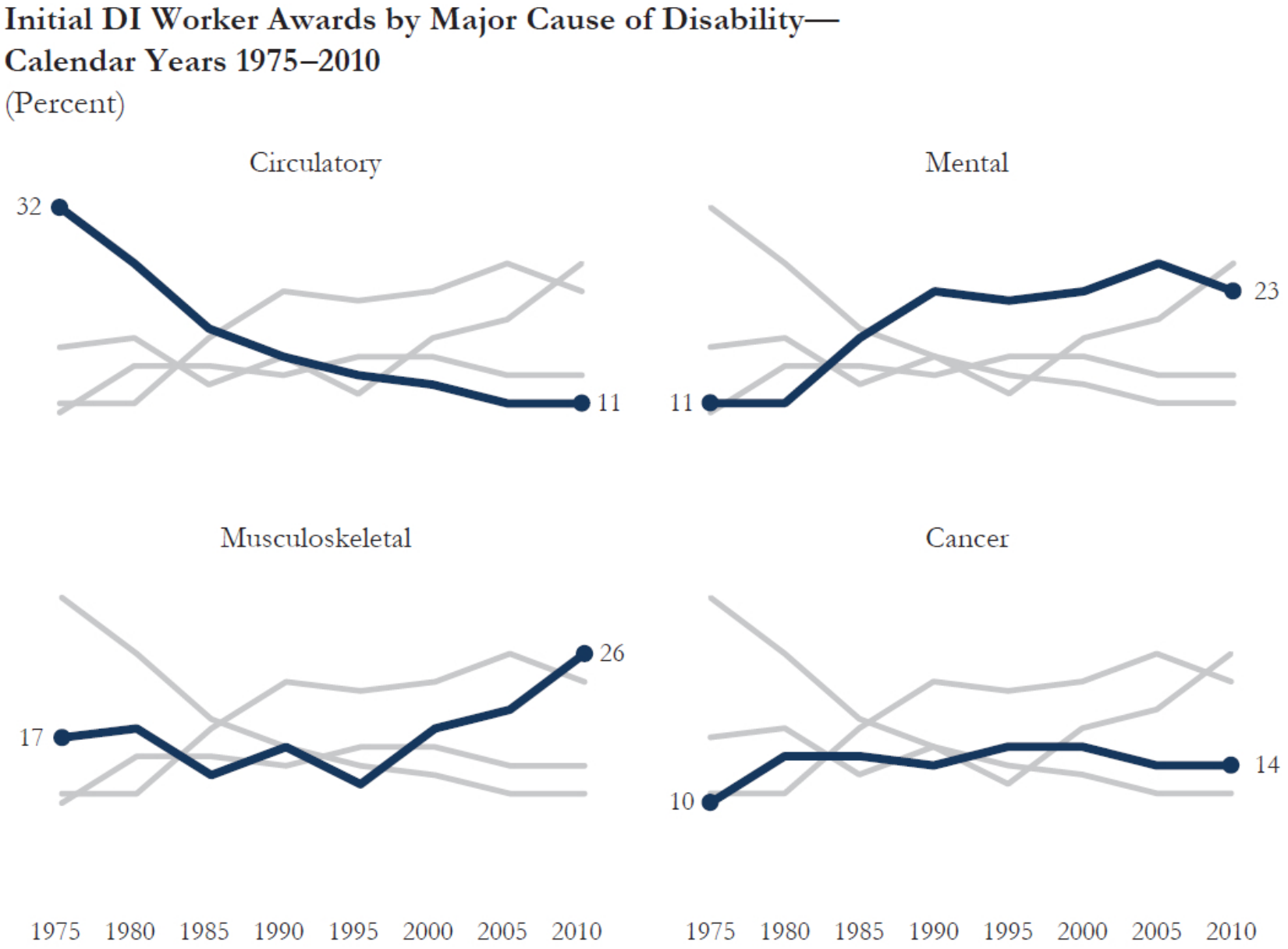
Schwabish (2014, *JEP*)

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

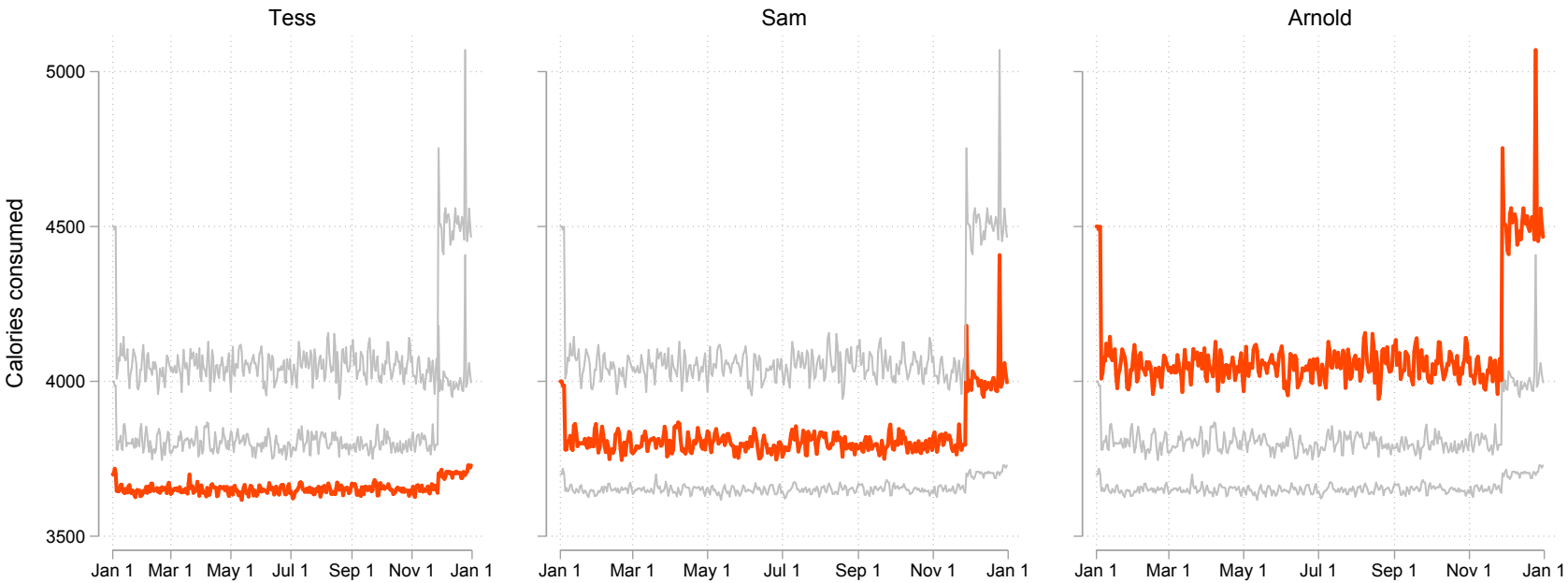


Use this instead of spaghetti charts

Schwabish (2014, *JEP*)



Intermezzo How can you draw such a graph?



- Not the best example, because the three time series are hardly overlapping anyways. Normally you would do that if you can't distinguish the series.
- I use three different graph commands with a `globaloptions` local, I think this makes more sense than looping with several `if` conditions.
- *Exercise* Instead of having the first of the respective month on the x-axis, try to keep the ticks but put the

```
. sysuse xtline1.dta, clear
. xtset person day
       panel variable:  person (strongly balanced)
        time variable:  day, 01jan2002 to 31dec2002
                delta:  1 day
.
. loc globaloptions "legend(off) xtitle("") xlab(#8, format(%tdMon_dd))"
.
. * graph 1
. #delimit ;
delimiter now ;
. tw      (line calories day if person == 1, lpattern(solid) lcolor("255 69 0") lwidth(*2))
>         (line calories day if person == 2, lpattern(solid) lcolor(gs12))
>         (line calories day if person == 3, lpattern(solid) lcolor(gs12)),
>         ylab(3500(500)5000) title("Tess") name(g1, replace) `globaloptions´
>   ;
. #delimit cr
delimiter now cr
.
. * graph 2
. #delimit ;
delimiter now ;
. tw      (line calories day if person == 1, lpattern(solid) lcolor(gs12))
>         (line calories day if person == 2, lpattern(solid) lcolor("255 69 0") lwidth(*2))
>         (line calories day if person == 3, lpattern(solid) lcolor(gs12)),
>         ylab(none) ytitle("") yticks(3500(500)5000, grid) title("Sam") name(g2, replace) `globaloptions´
>   ;
. #delimit cr
delimiter now cr
.
. * graph 3
. #delimit ;
delimiter now ;
. tw      (line calories day if person == 1, lpattern(solid) lcolor(gs12))
>         (line calories day if person == 2, lpattern(solid) lcolor(gs12))
>         (line calories day if person == 3, lpattern(solid) lcolor("255 69 0") lwidth(*2)),
>         ylab(none) ytitle("") yticks(3500(500)5000, grid) title("Arnold") name(g3, replace) `globaloptions´
>   ;
. #delimit cr
delimiter now cr
.
. gr combine g1 g2 g3, cols(3) scale(1.1) xsize(9)
```

Intermezzo **Two remarks**

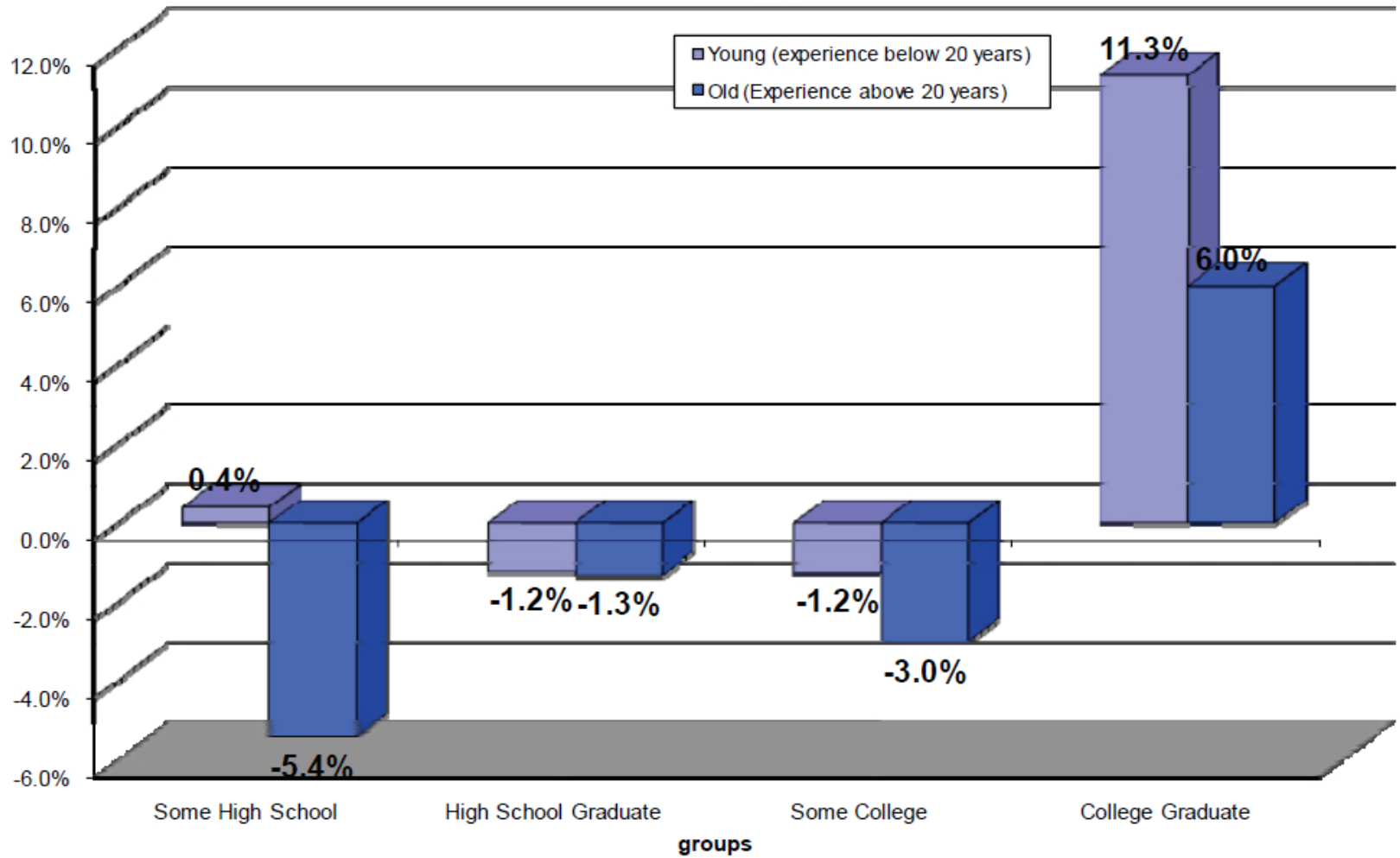
- **Don't use pie charts.**
  - Forces readers to make comparisons using the areas of the slices or the angles formed by the slices, something our visual perception does not accurately support. Donut charts are even worse.
- **Never use 3D charts.**
  - Why the 3rd dimension? Adds clutter but no information.
  - Distorts the information.

You will never see these graphs in scientific publications.

You know what's the worst? 3D pie charts.

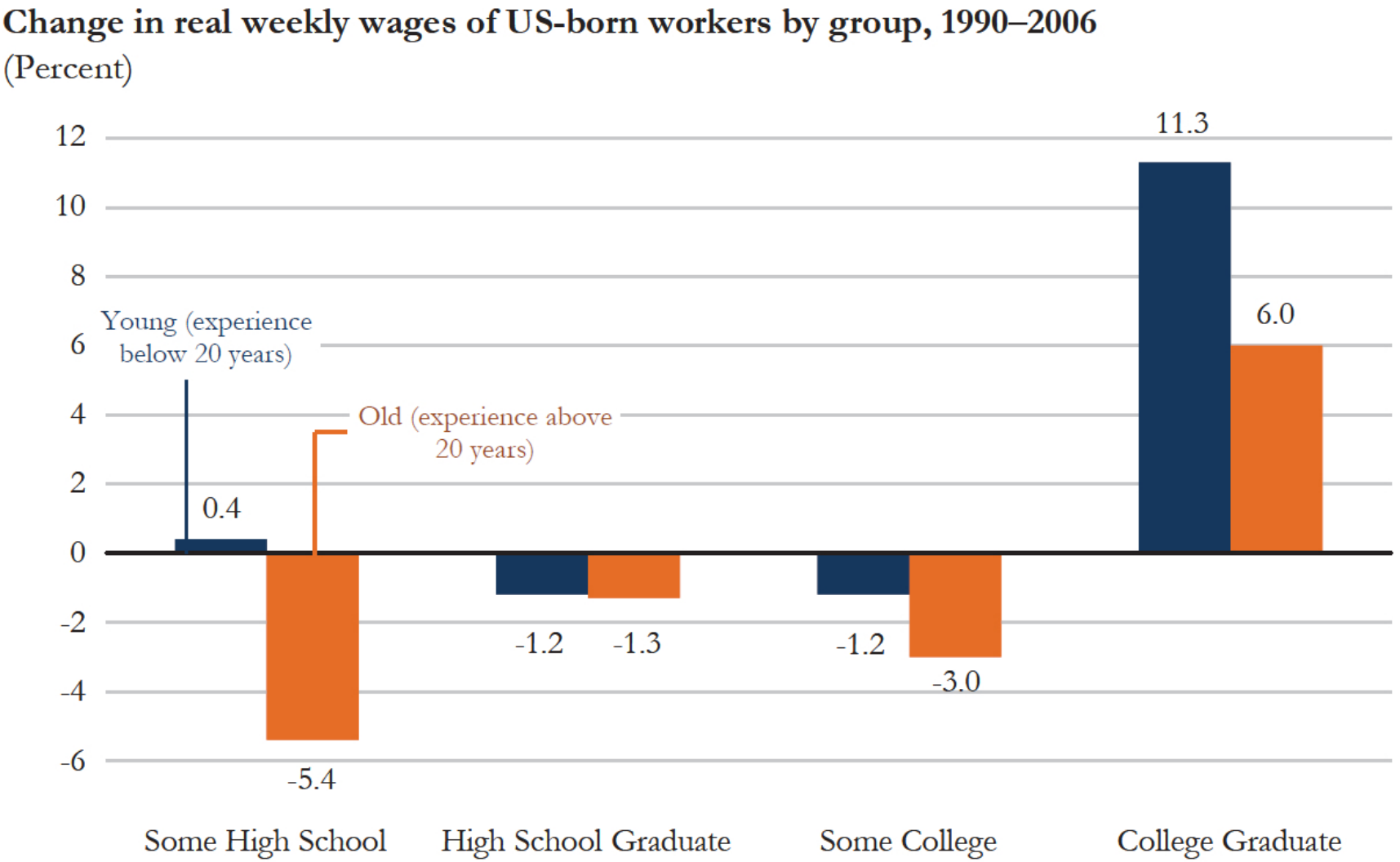
A horrible 3D chart

Schwabish (2014, *JEP*)



Use a bar chart instead

Schwabish (2014, *JEP*)



Pie charts

Schwabish (2014, *JEP*)

Figure 7A
A Pie Chart

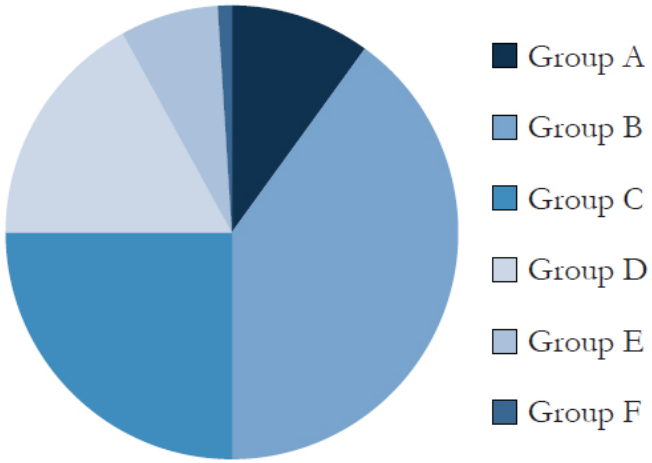
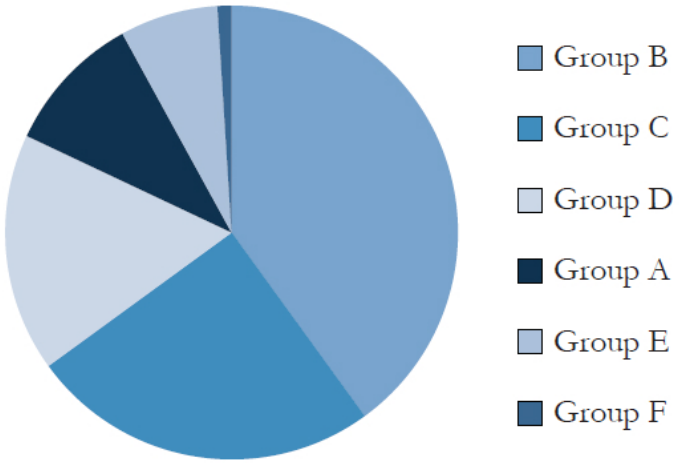


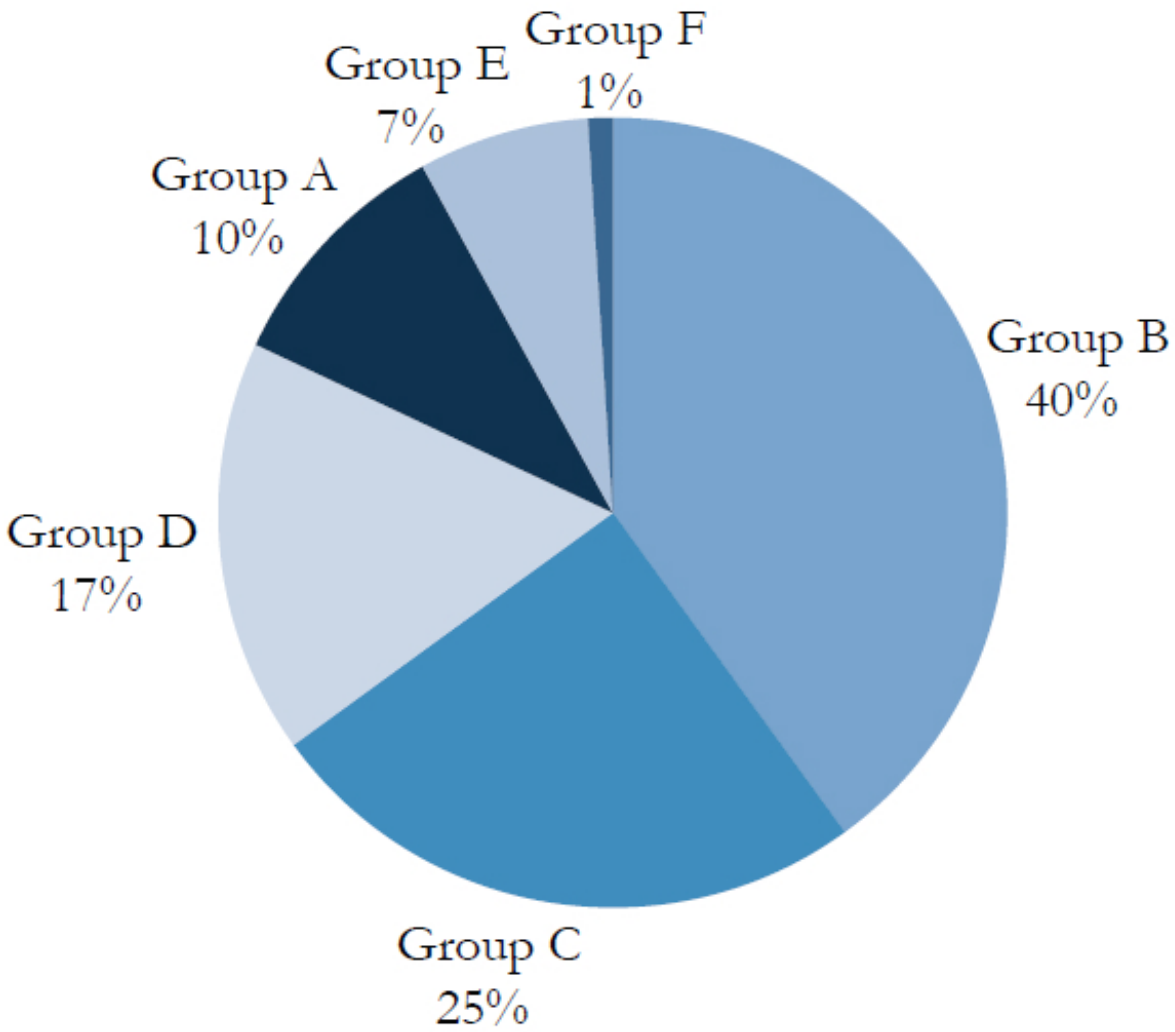
Figure 7B
B: A Pie Chart, Rotated



Let's try to guess the size of the slices.
Don't look at the next slide.

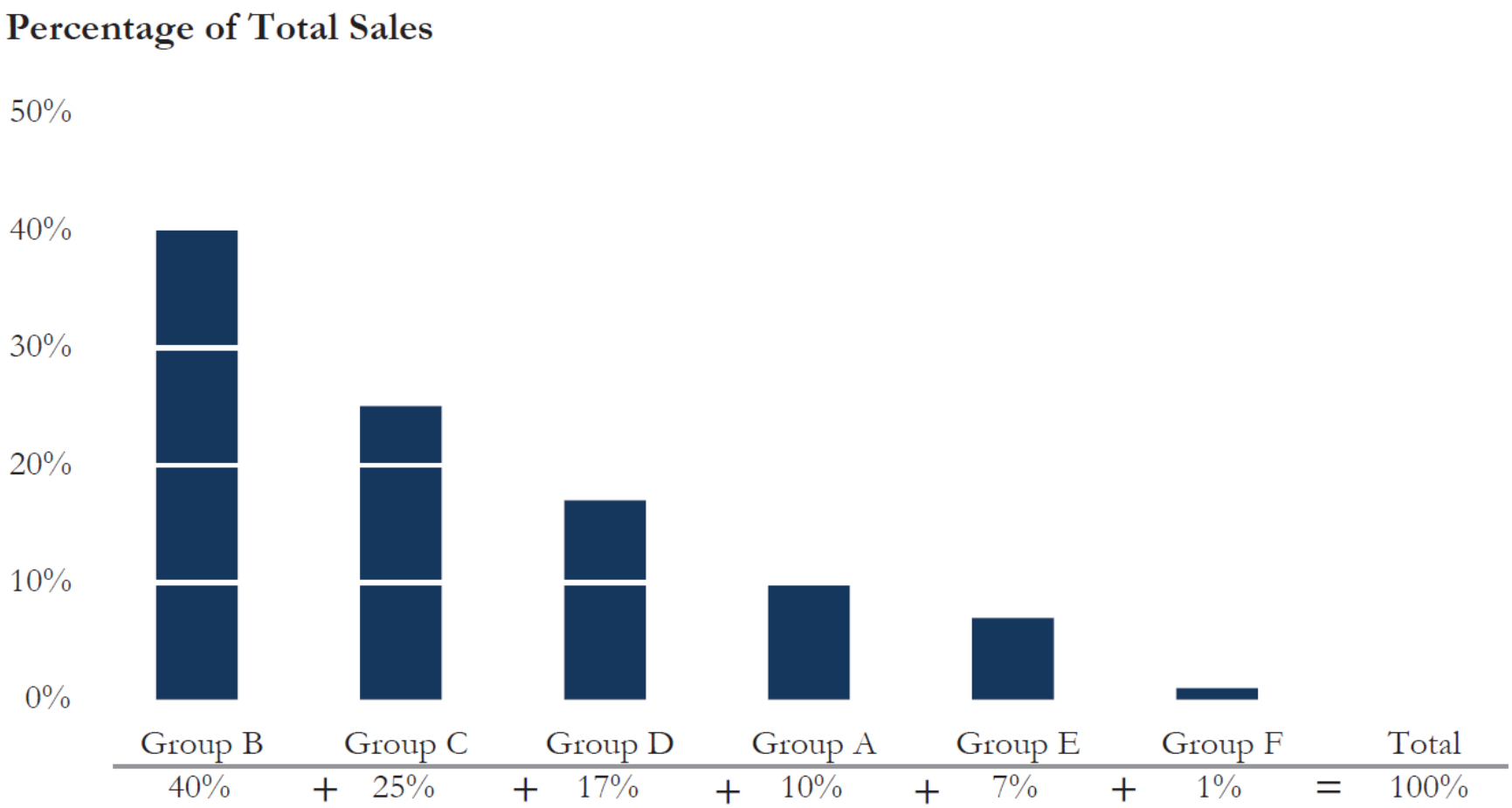
Pie charts

Schwabish (2014, *JEP*)



Try to guess the size here

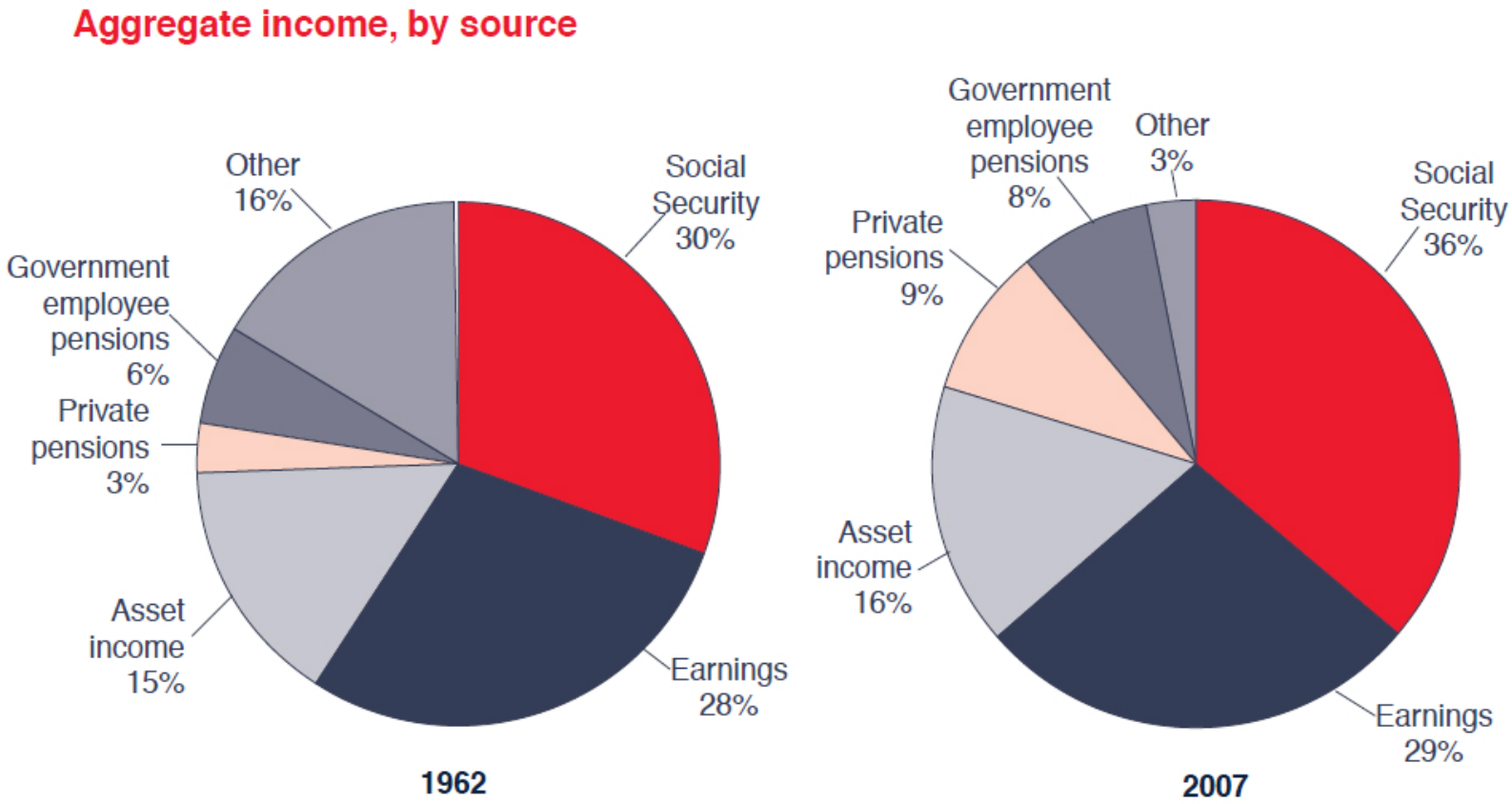
Schwabish (2014, *JEP*)



Comparing two pie charts

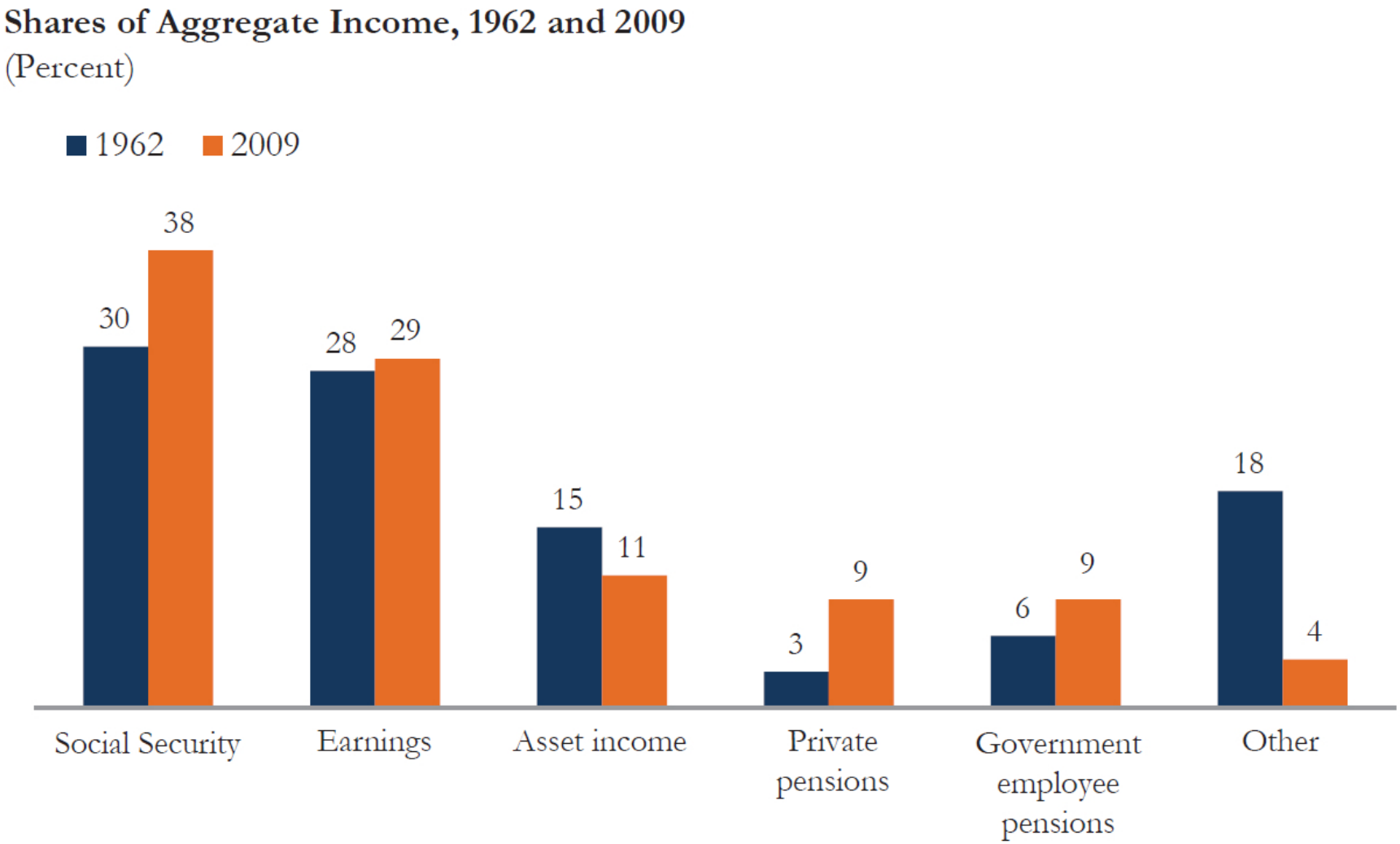
Schwabish (2014, *JEP*)

Shares of Aggregate Income, 1962 and 2007



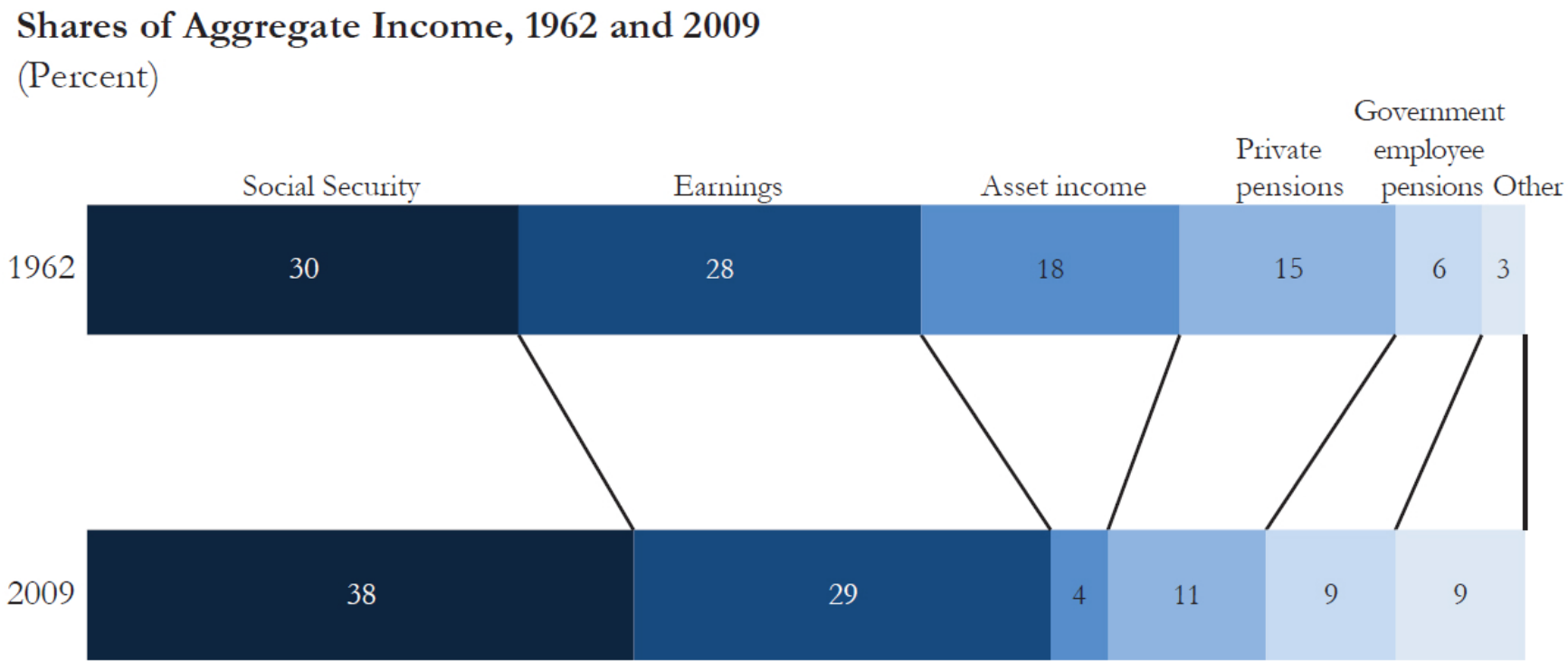
Use a bar chart

Schwabish (2014, *JEP*)



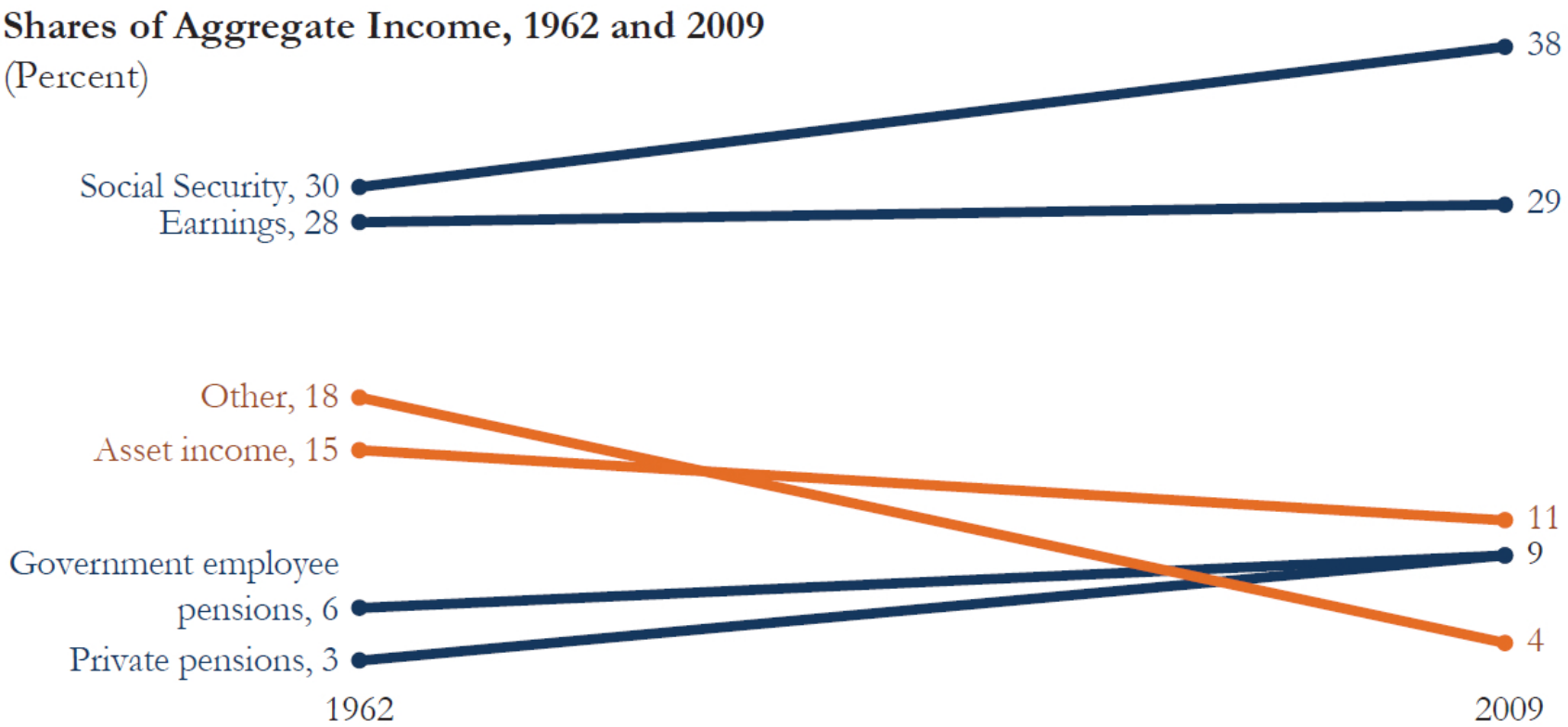
Use a stacked bar chart

Schwabish (2014, *JEP*)



Use a slope chart

Schwabish (2014, *JEP*)

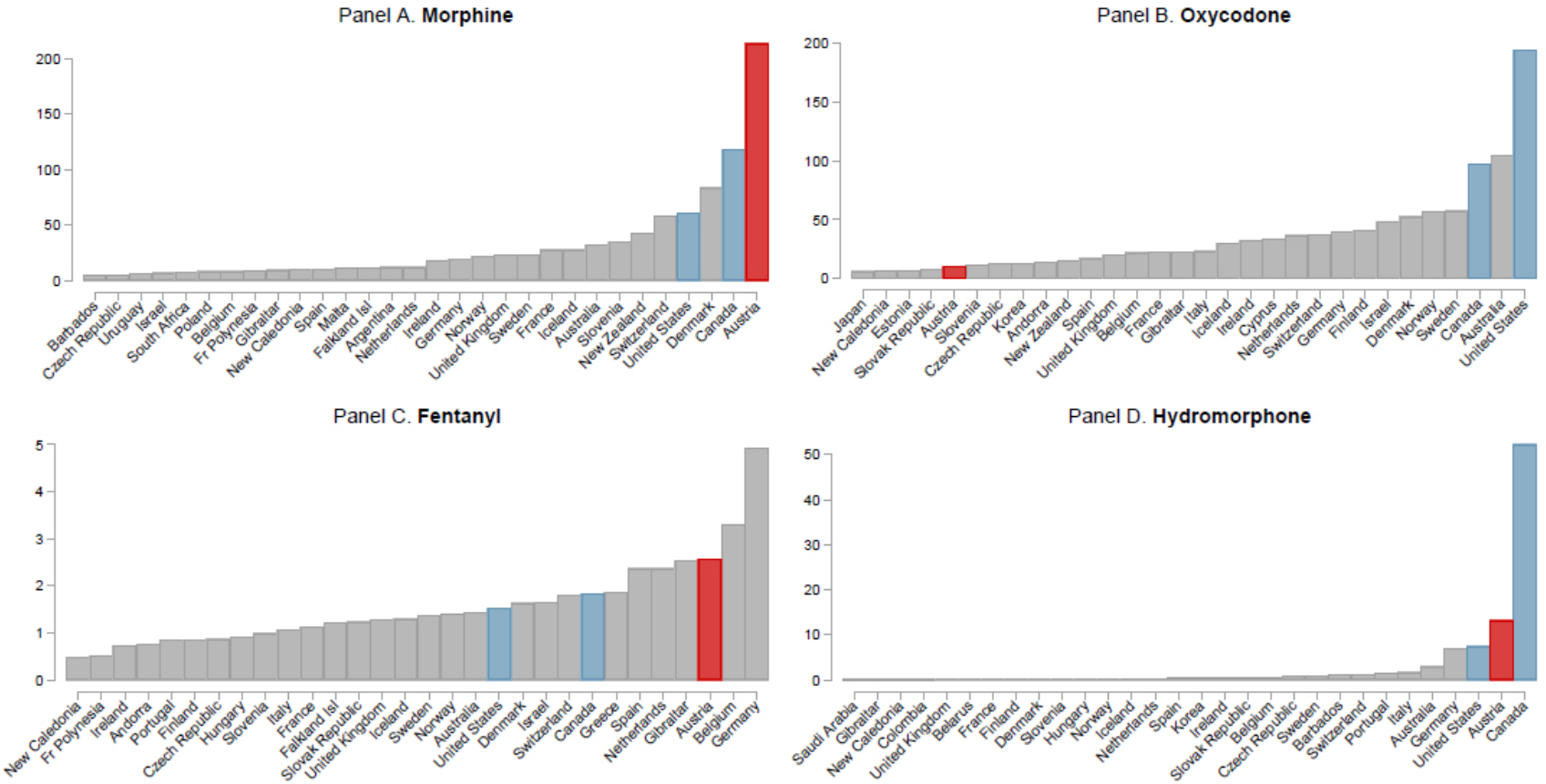


Graphs should be self explanatory

- Always compose a graph in a way that every reader immediately understands it without having to look in the text for explanation.
- Every graph should have a **caption** or a title (typically on top of the graph) and a **figure note,** explaining the graph.
  - The title should be short, e.g., *"The effect of x on y, 2000–2012"*
  - The figure note should be very detailed; it should allow the reader to understand the graph without having to refer to the main text.
- Use **integrated legends,** either right below the title, directly on the chart, or at the end of a line.
- Typically graphs are placed **after** the main text, but this is a question of preference. If you put them in the main text, make sure that they float on top of the page. Use landscape graphs if necessary.
- If you use colored graphs, make sure that no information is lost when it is printed in gray scales.

Graphs should be self explanatory

FIGURE 2 — Per capita prescriptions of popular opioid medications 2015, top 30 countries worldwide.

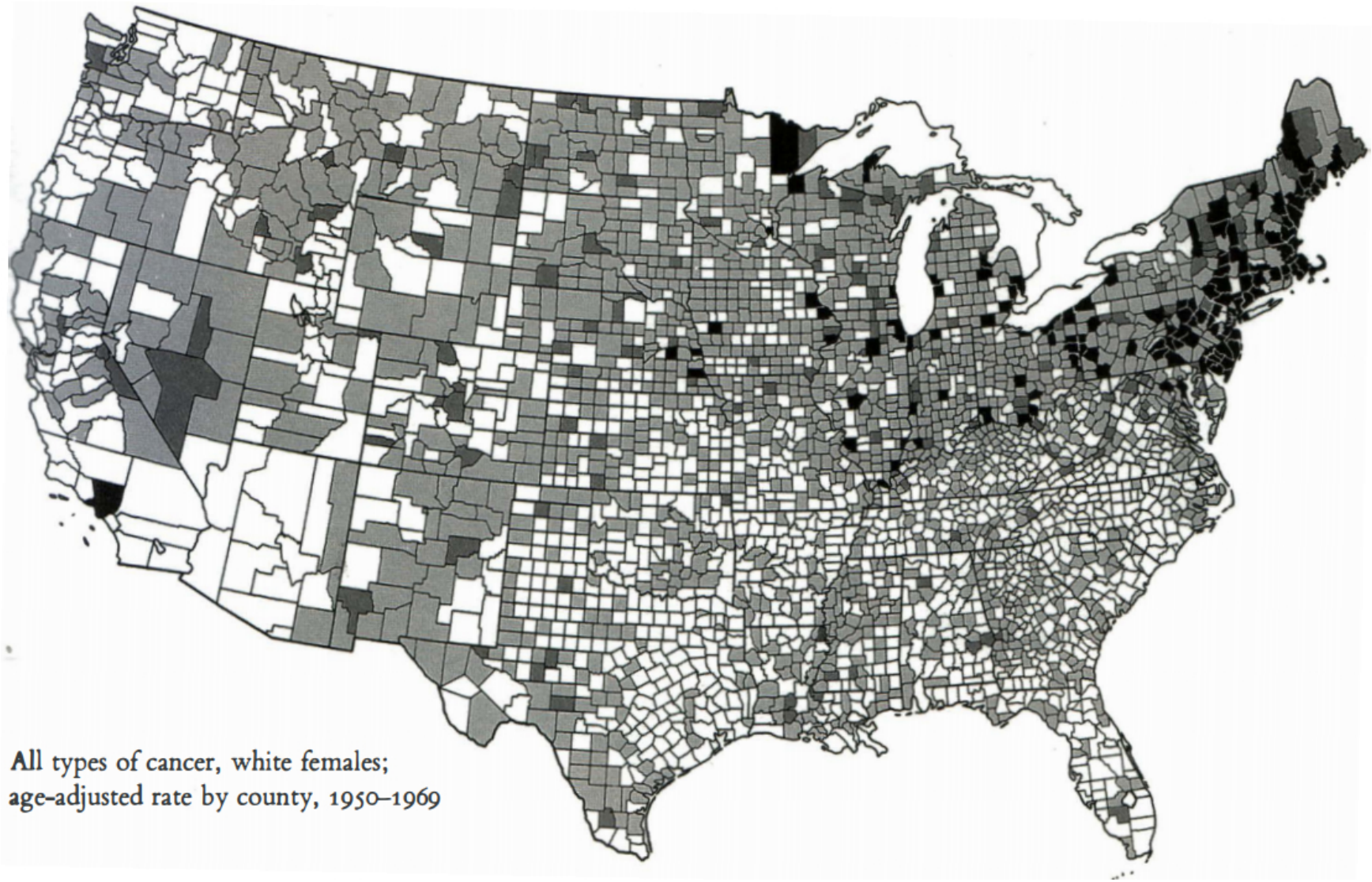


Notes: These graphs show the per capita consumption in mg of the respective medication in 2015 for the 30 countries worldwide with the highest consumption. The red bars ■ represent Austria and the blue bars ■ Canada and the United States. The data are compiled by the *Pain & Policy Studies Group* at the University of Wisconsin (URL: `http://www.painpolicy.wisc.edu/global`, accessed December 16, 2018). Originally, consumption data were taken from the *International Narcotics Control Board* (INCB) 2015 *Estimated World Requirements* report, while population data are from the *World Health Organization.*

Use maps whenever possible

- Tufte emphasizes the value of geographical maps, because they allow to show an incredible amount of detail that would not be possible to show in text or tables.
  - We devote an entire section to making maps in Stata.
- Immediately shows general overall patterns, but also makes it possible to detect very fine area-by-area detail.
- Attention is directed toward exploring the substantive content of the data rather than toward questions of methodology and technique.
- Maps also have flaws $\longrightarrow$ for example, they wrongly equate the visual importance of each area with its size rather than with the number of people living in the county (this can be circumvented).

Use maps whenever possible



All types of cancer, white females;
age-adjusted rate by county, 1950–1969

Good tables

Good tables have the following structure:

- Title
- Header
  - A matrix of column headings (and their subheadings) and side (row) headings
- Field/cells
  - Rows and columns containing the data
- Explanatory notes
  - Complements the info to fully understand the numbers presented and give additional information

Good tables

- Each column needs a heading
- Do not separate columns with vertical lines
- Do not use horizontal lines excessively
  - Top and bottom of the table
  - Line separating the heading from the main body
  - Rather use vertical spacing
- Use a reasonable number of post-decimal digits (max. 3)
- Align decimal points vertically with each other
- Add notes $\longrightarrow$ table should be self-contained

Good regression tables

- Indicate clearly the dependent variable, the treatment or main explanatory variable of interest, and the estimation method.
- Main field should contain ...
  - Coefficients (or marginal effects in non-linear models)
  - Standard errors (or confidence intervals or t-statistics)
  - Graphical indication of certain significance levels using asterisks ($*$)
- Additional information which may be useful
  - Number of observations
  - Mean and sd of dependent variable
  - Mean and sd of treatment var
  - Goodness of fit measure of fit
  - Diagnostics
- Show estimates graphically, especially if you, for example, compare coefficients between models

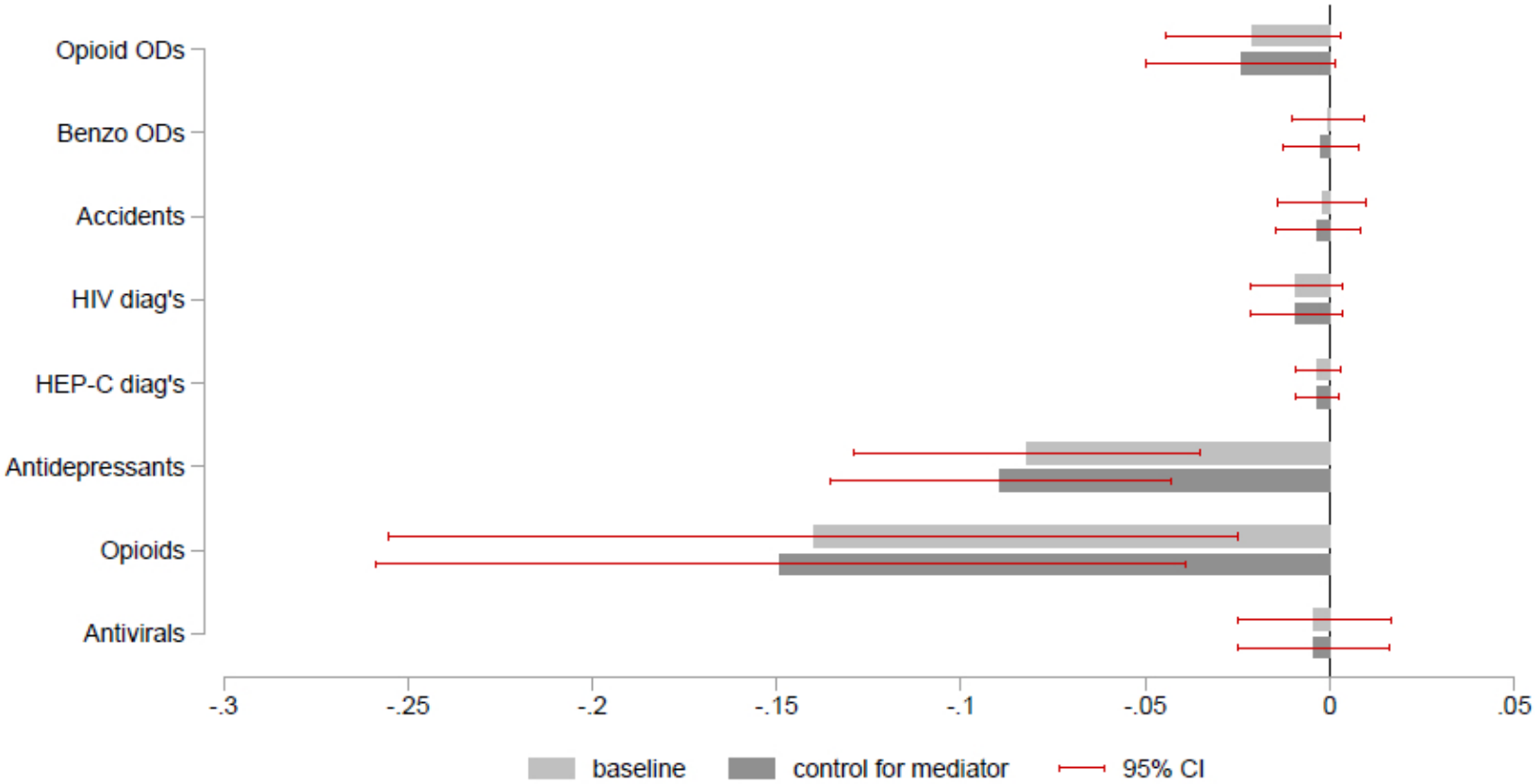
TABLE 5 — Estimated treatment effects on subsequent maternal outcomes

	(1) No. of next births	(2) Further birth	(3) Time to next birth†	(4) 20 year survival	(5) 40 year survival
Panel A. RDD					
Prenatal maternity leave	0.016	0.007	−0.002	−0.001	−0.007*
	(0.012)	(0.007)	(0.016)	(0.001)	(0.004)
No. of observations	7,350	7,350	3,619	7,350	7,350
Mean of outcome	0.70	0.49	7.10	0.99	0.92
Std. dev. of outcome	0.88	0.50	0.73	0.09	0.27
Kleinbergen-Paap *rK* Wald *F*-statistic	756.45	756.45	366.15	756.45	756.45
Panel B. OLS (only pre-treatment period)					
Prenatal maternity leave	0.003	0.003	−0.010	−0.000	−0.001
	(0.005)	(0.003)	(0.008)	(0.000)	(0.002)
No. of observations	3,721	3,721	1,816	3,721	3,721
Mean of outcome	0.69	0.49	7.11	0.99	0.93
Std. dev. of outcome	0.87	0.50	0.74	0.08	0.26

Notes: This table presents estimated treatment effects of extending compulsory ML duration by two weeks on different subsequent maternal fertility outcomes. Each cell represents a separate regression. The sample in Panel A consists of working mothers giving birth in April and June 1974, in panel B the sample is restricted to women giving birth in April 1974. 'No. of next births' (column 1) is a count variable measuring the number of children the mother has given birth to subsequently, 'further birth' (column 2) is a binary variable indicating whether the mother gave birth at least one more time, and 'time to next birth' (column 3) is the number of days passed until the mother gave birth again in logs, conditional on having another child. The outcomes '20 year survival' and '40 year survival' (columns 4 and 5) are binary variables indicating whether the mother was still alive 20 and 40 years after birth, respectively. In each specification we control for a binary variable indicating whether the child was born in wedlock, the mother's religion, whether the mother is an Austrian citizen, the province a mother lives in, and flexibly for age of the mother (separate dummies for every value of age between 20 and 34, and two additional categories indicating whether age is lower than 20 or higher than 34). Panel A presents fuzzy RDD estimates obtained via 2SLS where duration of ML is instrumented by assignment to the reform, panel B are simple OLS estimates where ML duration is used as an explanatory variable. Compulsory ML was extended by two weeks due to the reform, hence coefficients in panel A have to be multiplied by the same factor as well. Robust standard errors are in parentheses, stars indicate statistical significance: * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$.

† Time to next birth is conditional on giving birth again, thus the samples includes only mothers who had another child.

FIGURE A.4 — Change in LATEs when mediation through the usage of other benzodiazepines is controlled for, addiction-related outcomes.



Notes: This graph resembles Figure 8 in the main paper, but shows the change in LATEs for addiction-related outcomes. The gray line represents the LATE when a variable is controlled for that indicates whether other short-acting benzodiazepines (clonazepam, diazepam, nitrazepam, and oxazepam) were prescribed to the addict. The LATEs where mediators are controlled for are given as dark gray bars. For comparison, the baseline LATEs from Table 2, Panel A, are given as light gray bars. The graph also displays 95% confidence intervals.

E.2

Describing and comparing distributions

Different types of distributions

- Depending on the **types and distribution** of a variable, there are different possible visual representations.
  - Binary variables
  - Categorical variables
  - Count variables (discrete vars with a possibly large number of realizations)
  - Continuous variables
- The same goes if you are interested in **bivariate relationships,** e.g.,
  - Continuous vs. continuous $\longrightarrow$ scatter plot
  - Continuous vs. binary $\longrightarrow$ table with means and sd's by the binary var
  - Categorical vs. categorical $\longrightarrow$ two-way frequency table of bar chart
- If you don't know how a variable x is coded in your data, try one of the following commands
  - `codebook` x
  - `inspect` x
- There are two great books which show you examples of graphs and tables for every possible type of variable and combinations of variables:
  - Kohler and Kreuter (2012), *Data Analysis Using Stata*, 3rd edition, Stata Press.
  - Mitchell (2012), *A Visual Guide to Stata Graphics*, 3rd edition, Stata Press.

Different types of distributions

Find out how vars are coded

```
. codebook p_educ e_wage
```

---

p_educ

```
                 type:  numeric (byte)
                label:  educ

                range:  [0,5]                        units:  1
        unique values:  6                    missing .:  33,286/322,375

           tabulation:  Freq.   Numeric  Label
                         1,231        0  keine Pflichtschule
                        37,108        1  Pflichtschule
                       122,264        2  Lehre
                        50,449        3  mittlere Schule (o. Matura)
                        59,636        4  hoehere Schule (m. Matura)
                        18,401        5  Hochschule od. Universitaet
                        33,286        .
```

---

e_wage

```
                 type:  numeric (double)

                range:  [.00333333,1144276.2]        units:  1.000e-10
        unique values:  176,888              missing .:  0/322,375

                 mean:     25995
            std. dev:    16373.2

          percentiles:        10%       25%       50%       75%       90%
                           7863.87   15408.4   24618.2     33516   44050.6
```

Different types of distributions

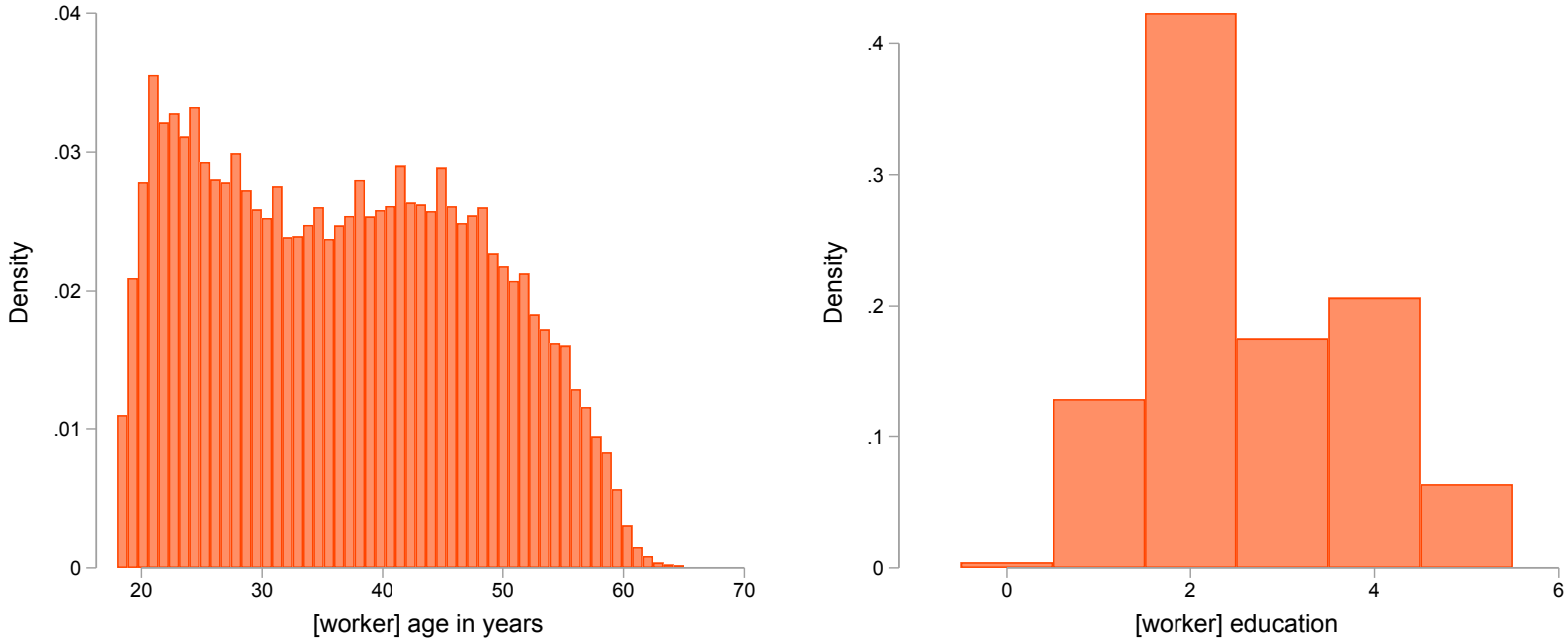
Find out how vars are coded

```
. inspect p_educ e_wage
p_educ:  [worker] education                    Number of Observations
-------------------------------------------------------------------------------
                                                    Total      Integers   Nonintegers
|          #                    Negative              -             -             -
|          #                    Zero              1,231         1,231             -
|          #                    Positive        287,858       287,858             -
|          #       #                          ----------    ----------    ----------
|          #   #   #            Total           289,089       289,089             -
|  .   #   #   #   #            Missing          33,286
+-------------------------
0                       5                      322,375
   (6 unique values)

      p_educ is labeled and all values are documented in the label.
e_wage:  [emp] annual wage                     Number of Observations
-------------------------------------------------------------------------------
                                                    Total      Integers   Nonintegers
|  #                            Negative              -             -             -
|  #                            Zero                  -             -             -
|  #                            Positive        322,375        28,999       293,376
|  #                                            ----------    ----------    ----------
|  #                            Total           322,375        28,999       293,376
|  #   .   .   .   .            Missing               -
+-------------------------
.0033333         1144276                       322,375
(More than 99 unique values)
```

Histograms

A general way to represent univariate distributions are **histograms.** In Stata, you can draw them using `hist`. Use the `discrete` option if you want to plot the distribution of a discrete variable.



The histogram is nothing else than an estimate of the probability distribution (pdf) of a variable. You have to select the **bin size** (i.e., you divide the range of values into a series of intervals). Smooth estimates of the pdf can be plotted, e.g., with `kdensity`.

E.3

Geographical maps

Geo maps

- As Tufte notes, maps are great.
  - Show both an overall pattern as well as incredible detail
- Creating a map in Stata requires multiple steps we have learned this semester, it is an adequate last exercise.
- *Exercise* Let's draw the average sick leave rate per patient in the data in 2010 for every Austrian municipality.

Shape files

- First, we need a shape file
- Popular geospatial vector data format to store geographic information
- The shape file format stores the data as primitive geometric shapes like points, lines, and polygons. These shapes, together with data attributes that are linked to each shape, create the representation of the geographic data.
- Official statistics agencies often offer shape files. Let's find one for Upper Austria.
  - Google *shape file + oberösterreich + gemeinden*
  - `www.data.gv.at` has *Gemeindegrenzen* and *Bezirksgrenzen* for (Upper) Austria
  - Make sure that the geo coding is consistent with your data
- Shape files always contain
  - `.shp` — shape format; the feature geometry itself
  - `.shx` — shape index format; positional index of the feature geometry
  - `.dbf` — attribute format; columnar attributes for each shape, in dBase IV format

Shape files

- I downloaded this: `http://e-gov.ooe.gv.at/at.gv.ooe.dorisdaten/DORIS_Basisdaten/GEMEINDEGRENZEN_GEN.zip`

- In order to get the coordinates into Stata, we use the user-written command

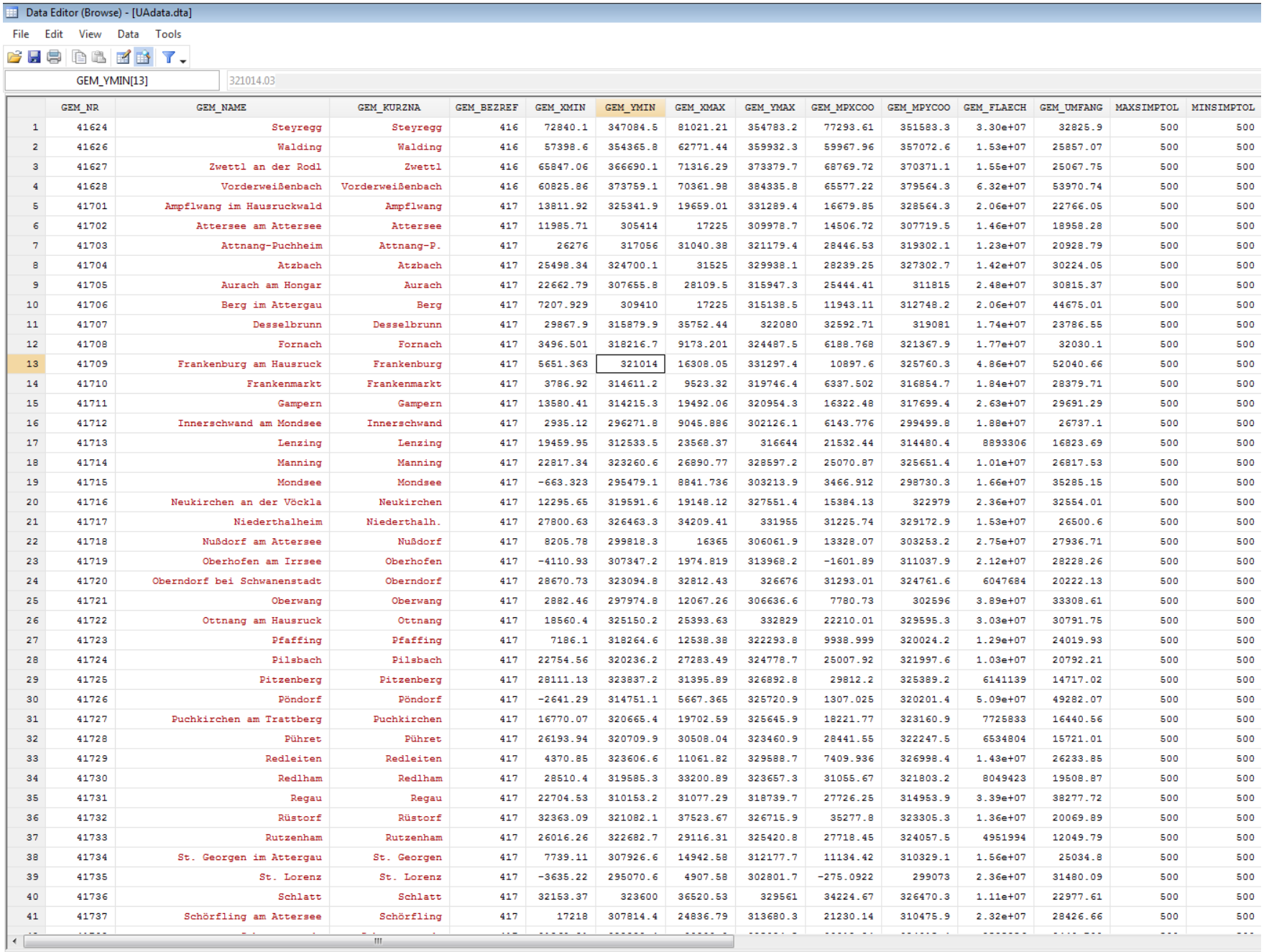
  `shp2dta`

  - Converts shape boundary files to Stata datasets
  - `database(`*filename*`)` makes new dataset containing the `.dbf` file data
  - `coordinates(`*filename*`)` makes new dataset containing the `.shp` file data

- This is the command I use:

```
shp2dta using "do/map/shapefile/GEMEINDEGRENZEN_GEN",
  database(UAdata) coordinates(UA) genid(id) replace
```

  $\Longrightarrow$ All files should have the same name (`GEMEINDEGRENZEN_GEN`), Stata automatically looks for the ones with the right ending.

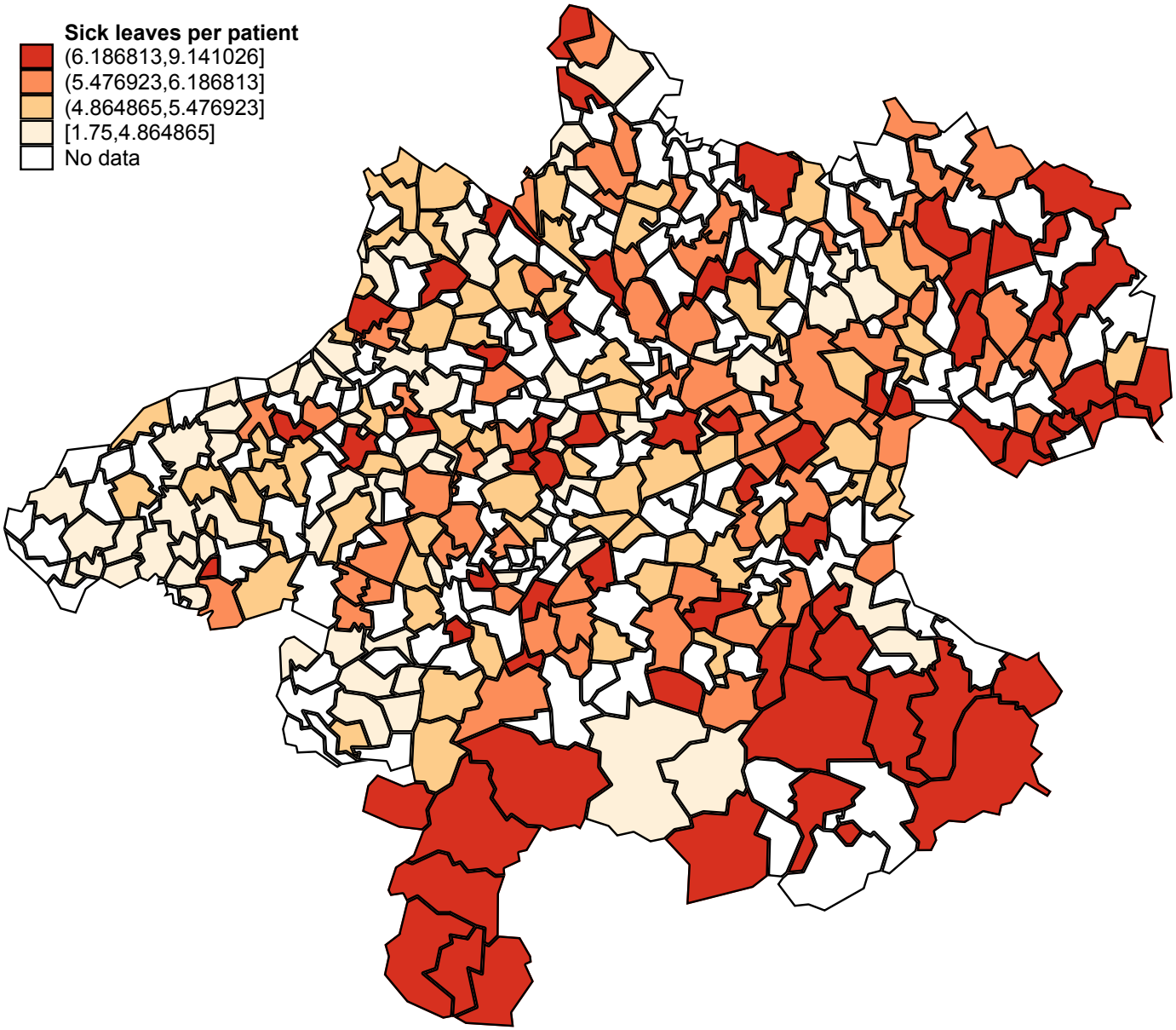


	GEM_NR	GEM_NAME	GEM_KURZNA	GEM_BEZREF	GEM_XMIN	GEM_YMIN	GEM_XMAX	GEM_YMAX	GEM_MPXCOO	GEM_MPYCOO	GEM_FLAECH	GEM_UMFANG	MAXSIMPTOL	MINSIMPTOL
1	41624	Steyregg	Steyregg	416	72840.1	347084.5	81021.21	354783.2	77293.61	351583.3	3.30e+07	32825.9	500	500
2	41626	Walding	Walding	416	57398.6	354365.8	62771.44	359932.3	59967.96	357072.6	1.53e+07	25857.07	500	500
3	41627	Zwettl an der Rodl	Zwettl	416	65847.06	366690.1	71316.29	373379.7	68769.72	370371.1	1.55e+07	25067.75	500	500
4	41628	Vorderweißenbach	Vorderweißenbach	416	60825.86	373759.1	70361.98	384335.8	65577.22	379564.3	6.32e+07	53970.74	500	500
5	41701	Ampflwang im Hausruckwald	Ampflwang	417	13811.92	325341.9	19659.01	331289.4	16679.85	328564.3	2.06e+07	22766.05	500	500
6	41702	Attersee am Attersee	Attersee	417	11985.71	305414	17225	309978.7	14506.72	307719.5	1.46e+07	18958.28	500	500
7	41703	Attnang-Puchheim	Attnang-P.	417	26276	317056	31040.38	321179.4	28446.53	319302.1	1.23e+07	20928.79	500	500
8	41704	Atzbach	Atzbach	417	25498.34	324700.1	31525	329938.1	28239.25	327302.7	1.42e+07	30224.05	500	500
9	41705	Aurach am Hongar	Aurach	417	22662.79	307655.8	28109.5	315947.3	25444.41	311815	2.48e+07	30815.37	500	500
10	41706	Berg im Attergau	Berg	417	7207.929	309410	17225	315138.5	11943.11	312748.2	2.06e+07	44675.01	500	500
11	41707	Desselbrunn	Desselbrunn	417	29867.9	315879.9	35752.44	322080	32592.71	319081	1.74e+07	23786.55	500	500
12	41708	Fornach	Fornach	417	3496.501	318216.7	9173.201	324487.5	6188.768	321367.9	1.77e+07	32030.1	500	500
13	41709	Frankenburg am Hausruck	Frankenburg	417	5651.363	321014	16308.05	331297.4	10897.6	325760.3	4.86e+07	52040.66	500	500
14	41710	Frankenmarkt	Frankenmarkt	417	3786.92	314611.2	9523.32	319746.4	6337.502	316854.7	1.84e+07	28379.71	500	500
15	41711	Gampern	Gampern	417	13580.41	314215.3	19492.06	320954.3	16322.48	317699.4	2.63e+07	29691.29	500	500
16	41712	Innerschwand am Mondsee	Innerschwand	417	2935.12	296271.8	9045.886	302126.1	6143.776	299499.8	1.88e+07	26737.1	500	500
17	41713	Lenzing	Lenzing	417	19459.95	312533.5	23568.37	316644	21532.44	314480.4	8893306	16823.69	500	500
18	41714	Manning	Manning	417	22817.34	323260.6	26890.77	328597.2	25070.87	325651.4	1.01e+07	26817.53	500	500
19	41715	Mondsee	Mondsee	417	-663.323	295479.1	8841.736	303213.9	3466.912	298730.3	1.66e+07	35285.15	500	500
20	41716	Neukirchen an der Vöckla	Neukirchen	417	12295.65	319591.6	19148.12	327551.4	15384.13	322979	2.36e+07	32554.01	500	500
21	41717	Niederthalheim	Niederthalh.	417	27800.63	326463.3	34209.41	331955	31225.74	329172.9	1.53e+07	26500.6	500	500
22	41718	Nußdorf am Attersee	Nußdorf	417	8205.78	299818.3	16365	306061.9	13328.07	303253.2	2.75e+07	27936.71	500	500
23	41719	Oberhofen am Irrsee	Oberhofen	417	-4110.93	307347.2	1974.819	313968.2	-1601.89	311037.9	2.12e+07	28228.26	500	500
24	41720	Oberndorf bei Schwanenstadt	Oberndorf	417	28670.73	323094.8	32812.43	326676	31293.01	324761.6	6047684	20222.13	500	500
25	41721	Oberwang	Oberwang	417	2882.46	297974.8	12067.26	306636.6	7780.73	302596	3.89e+07	33308.61	500	500
26	41722	Ottnang am Hausruck	Ottnang	417	18560.4	325150.2	25393.63	332829	22210.01	329595.3	3.03e+07	30791.75	500	500
27	41723	Pfaffing	Pfaffing	417	7186.1	318264.6	12538.38	322293.8	9938.999	320024.2	1.29e+07	24019.93	500	500
28	41724	Pilsbach	Pilsbach	417	22754.56	320236.2	27283.49	324778.7	25007.92	321997.6	1.03e+07	20792.21	500	500
29	41725	Pitzenberg	Pitzenberg	417	28111.13	323837.2	31395.89	326892.8	29812.2	325389.2	6141139	14717.02	500	500
30	41726	Pöndorf	Pöndorf	417	-2641.29	314751.1	5667.365	325720.9	1307.025	320201.4	5.09e+07	49282.07	500	500
31	41727	Puchkirchen am Trattberg	Puchkirchen	417	16770.07	320665.4	19702.59	325645.9	18221.77	323160.9	7725833	16440.56	500	500
32	41728	Pühret	Pühret	417	26193.94	320709.9	30508.04	323460.9	28441.55	322247.5	6534804	15721.01	500	500
33	41729	Redleiten	Redleiten	417	4370.85	323606.6	11061.82	329588.7	7409.936	326998.4	1.43e+07	26233.85	500	500
34	41730	Redlham	Redlham	417	28510.4	319585.3	33200.89	323657.3	31055.67	321803.2	8049423	19508.87	500	500
35	41731	Regau	Regau	417	22704.53	310153.2	31077.29	318739.7	27726.25	314953.9	3.39e+07	38277.72	500	500
36	41732	Rüstorf	Rüstorf	417	32363.09	321082.1	37523.67	326715.9	35277.8	323305.3	1.36e+07	20069.89	500	500
37	41733	Rutzenham	Rutzenham	417	26016.26	322682.7	29116.31	325420.8	27718.45	324057.5	4951994	12049.79	500	500
38	41734	St. Georgen im Attergau	St. Georgen	417	7739.11	307926.6	14942.58	312177.7	11134.42	310329.1	1.56e+07	25034.8	500	500
39	41735	St. Lorenz	St. Lorenz	417	-3635.22	295070.6	4907.58	302801.7	-275.0922	299073	2.36e+07	31480.09	500	500
40	41736	Schlatt	Schlatt	417	32153.37	323600	36520.53	329561	34224.67	326470.3	1.11e+07	22977.61	500	500
41	41737	Schörfling am Attersee	Schörfling	417	17218	307814.4	24836.79	313680.3	21230.14	310475.9	2.32e+07	28426.66	500	500

Shape files

- Sometimes the geo coding is **not** consistent with your data. Also in our case, the municipalities are coded using the so-called *Gemeindekennziffer*, but we have *Postleitzahlen* (zip codes) in our data.
- This means we have to convert between the two. We have to look for a file that contains 1:1 matches between *Gemeindekennziffern* and *Postleitzahlen*.
  - This may not be a perfect 1:1 match, but let's see what we can get.
- `https://www.statistik.at/web_de/klassifikationen/regionale_gliederungen/gemeinden/index.html`
  - This is a horrible database, but it will do the trick for now.
- Let's convert this to Stata format and match it to our sick leave data.
  - Let's do that in Stata, I will upload the code after the lesson.

The end result



Thank you!

It was a great first course. Feedback and criticism is highly welcome:
alexander.ahammer@jku.at

If you want to keep up with my work:
https://sites.google.com/view/alexanderahammer

Going forward ...

There is an infinite number of applications where you can use Stata, and with that come an infinite number of problems you will encounter. Here are some links that may help you:

- Blog Series on **programming an estimation command** [Link]
  - You may be required to program your own estimation command (especially if you strive for a career in academia, but not only then). This series of blog posts is a great and intuitive way to start, and additionally you may consider the book *'The Mata Book: A Book for Serious Programmers and Those Who Want to Be'* (William Gould, Stata Press).

- Read the **Stata journal** [Link]
  - Check the Stata journal for new user-written commands and tips related to workflow. Many theoretical econometricians program Stata suits and write corresponding articles in the Stata journal.

- There may exist **Stata books** for your field [Link]
  - Stata press publishes many great books. For many fields (e.g., *'Health Econometrics Using Stata,' 'Financial Econometrics Using Stata'*) and specific statistical and econometric problems (e.g., *'Bayesian Analysis with Stata'*) there are great books available. They not only focus on the practical implementation but also cover the necessary theoretical background.