Lecture 3a Overview of Plot Types

<u>Data Visualization · 1-DAV-105</u> Lecture by Broňa Brejová More details in the <u>notebook version</u>

Plan for today

- Types of variables (columns)
- Gallery of different plot types, some discussion of their properties
- Some notes on how to draw them in Python (more in the <u>notebook</u>)

Types of variables (columns)

Categorical / qualitative

- **Nominal:** values have no fixed ordering (for example, gender, country, color)
- **Ordinal:** values are ordered (for example education level primary / secondary / university; star rating 0-5)

Numerical / quantitative

- **Discrete (diskrétna):** typically counts
- **Continuous (spojitá):** typically measurements

Types of variables (columns)

Numerical / quantitative

- Discrete (diskrétna): typically counts
- **Continuous (spojitá):** typically measurements

Numerical variables also categorized as follows:

- Ratio (pomerová): if zero means "none", and it is meaningful to compute ratios / percentages (mass, length, duration, cost, ...)
- Interval (intervalová): does not have "true zero", we can subtract but not make ratios (temperature in degrees C, date)

Data for today

- Various country indicators downloaded from the World Bank for years 2000, 2010, 2020
- Population, area, GDP per capita, life expectancy, fertility (number of children per woman)
- Also classification into regions and income groups
- Which are categorical / numerical?

We will also use Gapminder/World bank life expectancy 1900-2021 from HW01, HW02

Scatter plot (bodový graf)

Good for two numerical variables (x and y).

In this plot, many points near left boundary, most space empty.

Solution 1: combine overall view and detail

Solution 2: log scale







Log-scale x-axis: draw at log(x) instead of x, but axis ticks show values of x

Adding a categorical variable with color





- Europe & Central Asia
- Middle East & North Africa
- East Asia & Pacific
- Sub-Saharan Africa
- Latin America & Caribbean
- North America

Adding a numerical variable with color scale



Adding numerical variable with marker size



Variable value should be proportional to circle area, not diameter!

Adding categorical variable with marker shape



Hard to read, particularly for many data points Showing population change between 2000 and 2020 If less than 1% change, marked as equal

Line graph (čiarový graf)



Emphasizing continuity between data points Data points can be also shown as markers

Adding categorical variable with color



Area graph (plošný graf)

Life expectancy in Slovakia and France



Y-axis must start at 0 (*Dis*)advantages compared to line graph?

Area graph (plošný graf)

Life expectancy in Slovakia and France



Y-axis must start at 0

Emphasizes differences more than line graph, but also more cluttered

Line graph with many lines



Countries with names Si..Sw, and having population at least 3 millions. Note that colors start to repeat. *What can/cannot we see here?*

Line graph with many lines



Countries with names Si..Sw, and having population at least 3 millions. Note that colors start to repeat.

Hard to follow individual lines, but shows general trends and comparisons.

Small multiples



A small plot for each value of a categorical variable Must have the same axes!

What can/cannot we see here?

Small multiples



A small plot for each value of a categorical variable

Must have the same axes!

Exact comparison difficult, but it is possible to notice different trends

Bar graph (stĺpcový/pruhový graf)

X-axis is categorical

Y-axis must start at 0



Bar graph with sorted columns



(Dis)advantages of sorting columns by size?

Bar graph with colored columns



What can we see from this graph?

Bar graphs can be horizontal



Bar labels are easier to read when horizontal

Dot plot

As bar graph but only dots shown at the top of the bar

Less clutter

X-axis does not need to start at 0 - better use of space

Can use multiple colors



Heatmap

Both axes categorical

Numerical value shown in a color scale

Compact display, but color scales harder to read



Pie chart (koláčový graf)



Obvious that percentages displayed Very large values are easy to see (here high income) Hard to compare similar values to each other Space use not good

Pie chart with values labeled



Easier to compare but still not ideal Labeling values also useful in other types of graphs

Stacked (skladaný) bar graph instead of pie chart



High income

Rectangles easier to compare than wedges Benefits from labeled values Middle colors hard to compare across bars Similar idea: stacked area plot (change in percentages over time)

Stacked area plot (skladaný plošný graf)



Life expectancy rounded down to a decade (to get a categorical variable) The number of countries for each decade group across years

Colored bar graph instead of pie chart



Easy to compare East Asia vs whole world. Not obvious that we show parts of a whole.



Colored bar graph instead of pie chart



Easy to compare income groups within region



	South Asia	1940 - A. M. C.
Strip plot	Europe & Central Asia	
One axis categorical	Middle East & North Africa	the second second
Other axis shows individual data points	East Asia & Pacific	tet tegener Kurk
Jitter added in categorical axis to avoid	Sub-Saharan Africa	1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.
point overlap	Latin America & Caribbean	eane.
	North America	2000 - C
		2.5 5.0

J

Fertility in 2020

Histogram

For 1D numerical data

Split values into bins, show bin sizes as bar graph

We could use colors to display 2 or more histograms

More in a lecture 7



Parallel coordinates

Good for multidimensional numerical data

Each column one dimension

Here scaled as % of maximum value

Hard to see individual lines, but can show trends, compare groups shown in color or selected data point vs others



Parallel categories

Good for multidimensional categorical data

Each column one dimension

The widths of ribbons correspond to the number of countries



Radar chart (radarový graf)

Hard-to-read version of parallel coordinates

Perhaps some justification in cyclical domains, such as average temperature in months of a year



Now some Python

Overview of libraries

- Matplotlib
- Seaborn: an extension of Matplotlib, convenient for many types of plots
- Plotly: basic usage similar to Seaborn, plots interactive by default

Part of the main table countries

	IS03	Region	Income Group	Population2000	Population2010	Population2020	Area	GDP2000	GDP2010
Country									
Afghanistan	AFG	South Asia	Low income	19542983.0	28189672.0	38972231.0	652860.0	NaN	562.499219
Albania	ALB	Europe & Central Asia	Upper middle income	3089026.0	2913021.0	2837849.0	28750.0	1126.683340	4094.349686
Algeria	DZA	Middle East & North Africa	Lower middle income	30774621.0	35856344.0	43451666.0	2381741.0	1780.376063	4495.921476
American Samoa	ASM	East Asia & Pacific	High income	58229.0	54849.0	46189.0	200.0	NaN	10446.863206
Andorra	AND	Europe & Central Asia	High income	66097.0	71519.0	77699.0	470.0	21620.465102	48237.890541

```
# create plot using Seaborn
axes = sns.scatterplot(data=countries, x='GDP2020', y='Expectancy2020',
                         hue='Region')
# set plot properties using methods from Matplotlib
axes.set xlabel('GDP per capita (US dollars), logscale')
axes.set ylabel('Life expectancy (years)')
axes.set title('Country indicators 2020')
axes.semilogx()
# place legend outside the plot:
axes.legend(bbox to anchor=(1.05, 1), loc=2)
                                                         Country indicators 2020
pass
                                                                        South Asia
                                                   85
```







Region

- South Asia
- Europe & Central Asia
- Middle East & North Africa
- East Asia & Pacific
- Sub-Saharan Africa
- Latin America & Caribbear
- North America





```
x='GDP2020', y='Expectancy2020', hue='Region',
style=diff_class, s=100,
markers={'increase':'^', 'decrease':'v', 'same':'.'})
```



display(life_exp_years)

1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 ... 2

Country

Afghanistan	29.4	29.5	29.5	29.6	29.7	29.7	29.8	29.9	29.9	30.0	
Albania	35.4	35.4	35.4	35.4	35.4	35.4	35.4	35.4	35.4	35.4	
Algeria	30.2	30.3	30.4	31.4	25.4	28.1	29.6	29.5	29.5	31.0	
Angola	29.0	29.1	29.2	29.3	29.3	29.4	29.4	29.5	29.6	29.7	
Antigua and Barbuda	33.8	33.8	33.8	33.8	33.8	33.8	33.8	33.8	33.8	33.8	

```
# list of numerical years from column names
years = [int(x) for x in life exp years.columns]
figure, axes = plt.subplots()
# plot two lines
axes.plot(years, life exp years.loc['Slovak Republic'], label='Slovakia')
axes.plot(years, life exp years.loc['France'], label='France')
# plot settings
axes.set xlabel('Year')
axes.set ylabel('Life expectancy (years)')
axes.set title('Life expectancy in Slovakia and France')
axes.legend()
                                                                    Life expectancy in Slovakia and France
                                                                  Slovakia
                                                                  France
pass
                                                               70
                                                              (years)
                                                              expectancy
09
09
```

1900

1920

1940

1960 Year 1980

2000

2020

```
figure, axes = plt.subplots()
# two filled areas, the second is semi-transparent
axes.fill_between(years, 0, life_exp_years.loc['Slovak Republic'], label='Slovakia')
axes.fill_between(years, 0, life_exp_years.loc['France'], label='France', alpha=0.5)
# plot settings as before...
```



```
# Lines for many countries easy in Seaborn
# ... works better with long table
life_exp_sel_long = (
    life_exp_sel.reset_index()
    .melt(id_vars=['Country'])
    .rename(columns={'variable':'Year', 'value':'Expectancy'})
    .astype({'Year':'int32'})
    Country Year Expectancy
```

display(life_exp_sel_long)

	Country	rear	Expectancy
0	Sierra Leone	1900	27.400000
1	Singapore	1900	34.200000
2	Slovak Republic	1900	37.900000
3	Somalia	1900	31.200000
4	South Africa	1900	34.500000
L 459	Sri Lanka	2021	76.399000
L 460	Sudan	2021	65.267000
461	Sweden	2021	83.156098
462	Switzerland	2021	83.851220
L 463	Syria	2021	72.063000

1464 rows × 3 columns



grid.set_axis_labels('Year', 'Life expectancy (years)')
grid.set_titles("{col_name}") # title of each plot will be country name



```
def rotate_bar_labels(axes, angle=45):
    """Auxiliary function for rotating bar plot labels by 45 degrees"""
    axes.tick_params(axis='x', labelrotation=angle, pad=-5)
    plt.setp(axes.get_xticklabels(), ha='right')
# sorting
```

```
life_exp_sel_2020_sorted = life_exp_sel_2020.sort_values('Expectancy')
# plotting
axes = sns.barplot(data=life_exp_sel_2020_sorted,
```

x='Country', y='Expectancy', color="C0")

axes.set_ylabel("Life expectancy in 2020")
axes.set_xlabel(None)
rotate bar labels(axes)





axes.set ylabel(None)

