Data Mining

Visualizations + EDA

Santiago Alonso-Díaz

Tecnólogico de Monterrey
EGADE, Business School

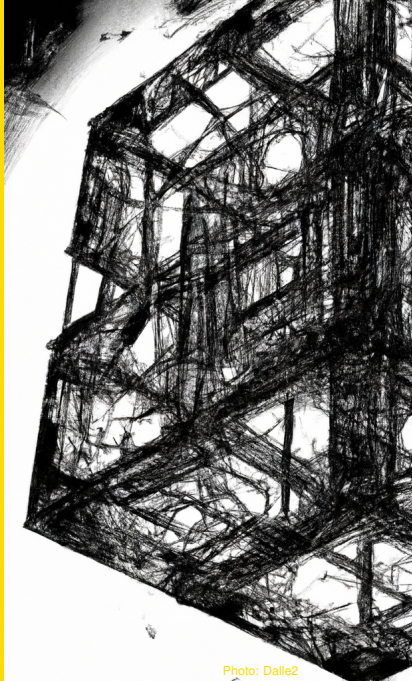


Photo: Dalle2

What to visualize/explore?

Table 2.1: Types of variables encountered in typical data visualization scenarios.

Type of variable	Examples	Appropriate scale	Description
quantitative/numerical continuous	1.3, 5.7, 83, $1.5x10^{-2}$	continuous	Arbitrary numerical values. These can be integers, rational numbers, or real numbers.
quantitative/numerical discrete	1, 2, 3, 4	discrete	Numbers in discrete units. These are most commonly but not necessarily integers. For example, the numbers 0.5, 1.0, 1.5 could also be treated as discrete if intermediate values cannot exist in the given dataset.
qualitative/categorical unordered	dog, cat, fish	discrete	Categories without order. These are discrete and unique categories that have no inherent order. These variables are also called *factors*.
qualitative/categorical ordered	good, fair, poor	discrete	Categories with order. These are discrete and unique categories with an order. For example, “fair” always lies between “good” and “poor”. These variables are also called *ordered factors*.
date or time	Jan. 5 2018, 8:03am	continuous or discrete	Specific days and/or times. Also generic dates, such as July 4 or Dec. 25 (without year).
text	The quick brown fox jumps over the lazy dog.	none, or discrete	Free-form text. Can be treated as categorical if needed.

Figure: Wilke, 2019

What to visualize/explore?

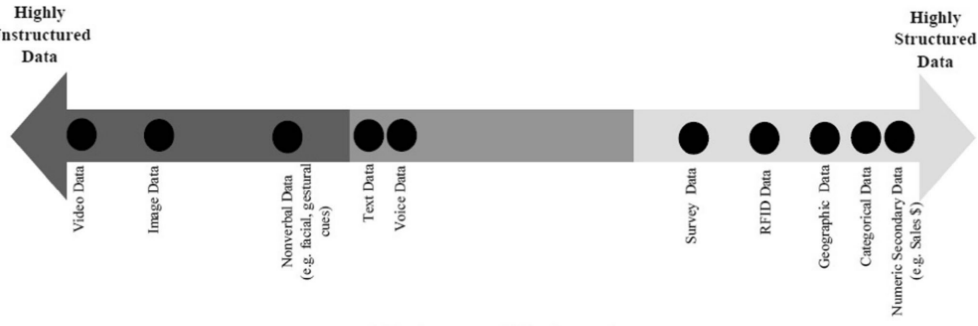


Figure: Balducci and Marinova, 2018

How to visualize?

Many types of charts.

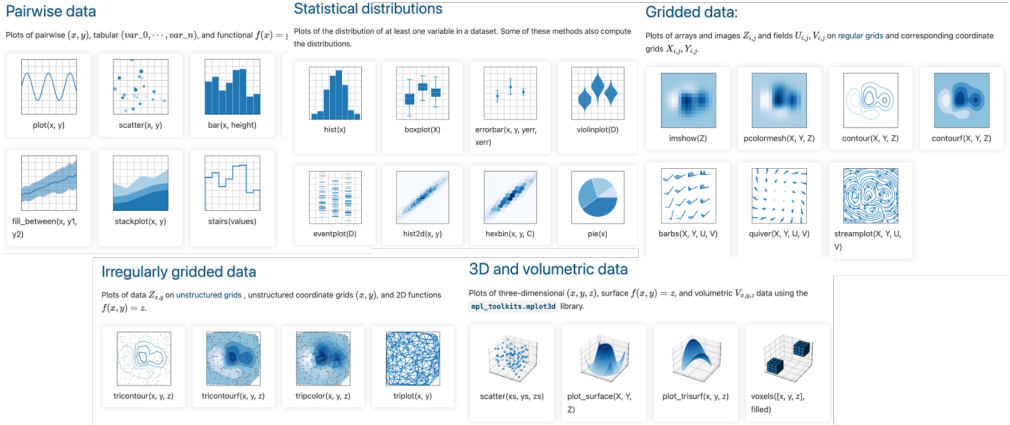


Figure: Matplotlib.org

Principles of visualization Wilke, 2019

Visualization needs to be proportional to the data values they represent.

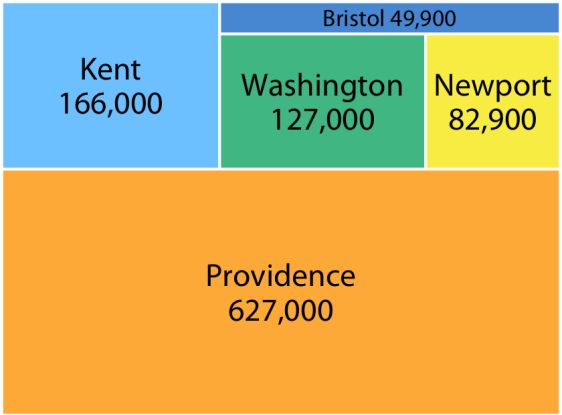


Figure: Wilke, 2019

Principles of visualization

Avoid overlaps e.g. with transparency and jittering

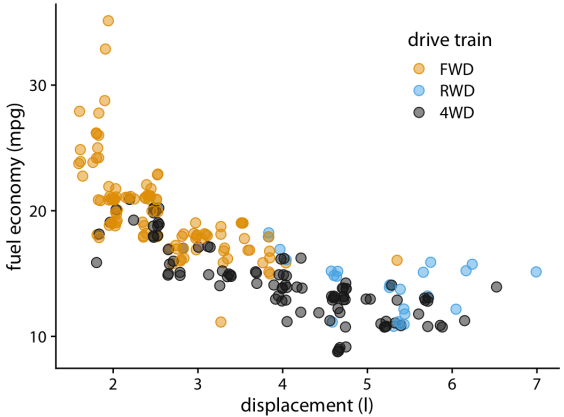


Figure: Wilke, 2019

Principles of visualization

Simple color scales. The rainbow scale looks bad and it is hard to read in this map.

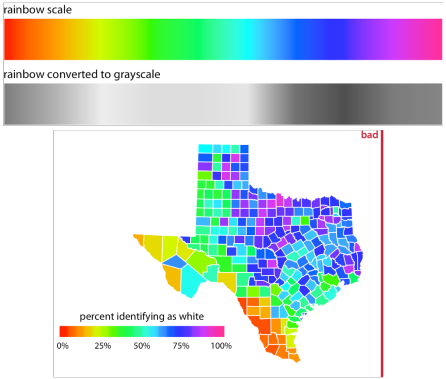


Figure: Wilke, 2019

Principles of visualization

Colorblind friendly schemes

original

deuteranomaly

protanomaly

tritanomaly

Figure: Wilke, 2019

Principles of visualization

Redundancy can be good. Here, color and shape enhance the difference in categories

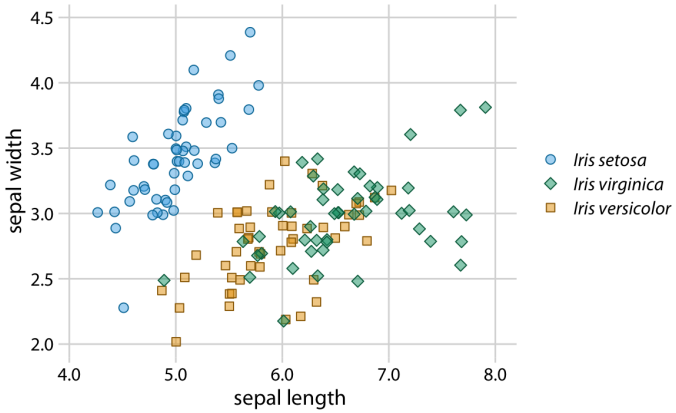


Figure: Wilke, 2019

Principles of visualization

Use consistent codes for the same categories across plots. Here the same color for each gender.

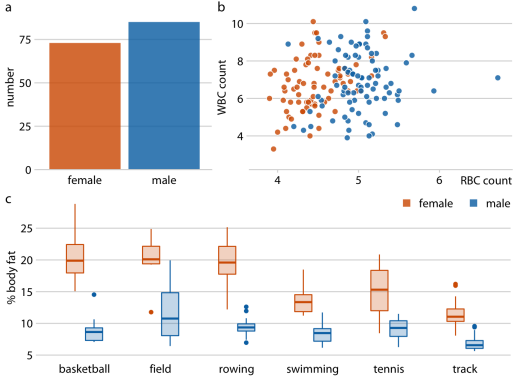


Figure: Wilke, 2019

Principles of visualization

Legends are not always necessary. Labels on the graphs are a good alternative.

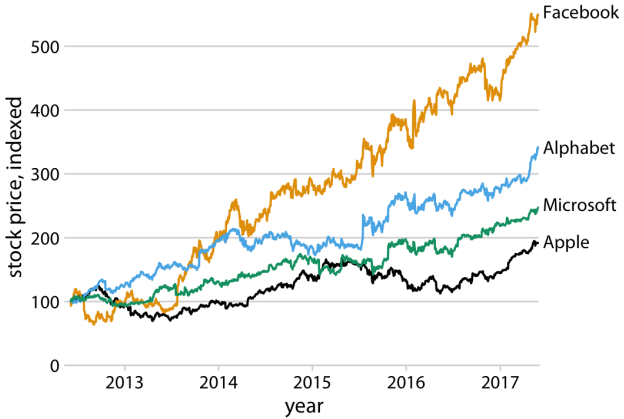


Figure: Wilke, 2019

Principles of visualization

Present heterogeneous results with multiple plots, with the same y-scale

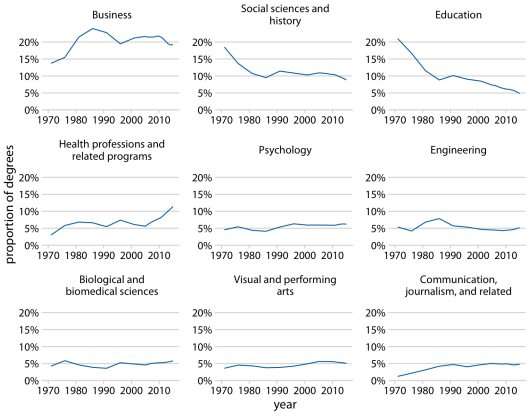


Figure: Wilke, 2019

Principles of visualization

Put title and axes labels

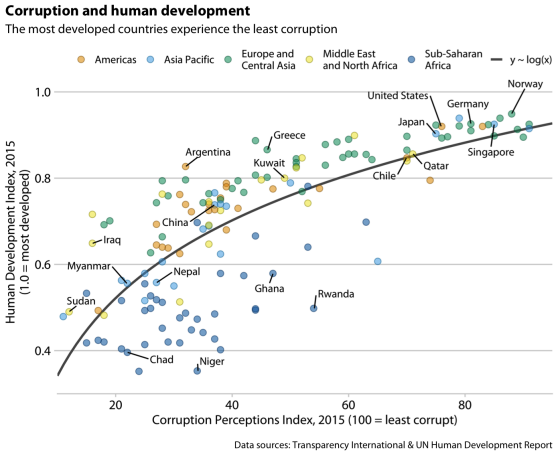


Figure: Wilke, 2019

Principles of visualization

Be generous with axes labels font size.

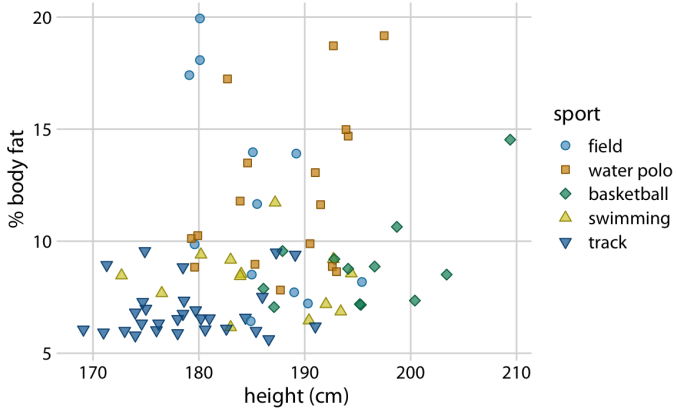


Figure: Wilke, 2019

Principles of visualization

Minimize ink devoted to non-data. The bottom plot respects more this principle than the top one. Drop grid lines?

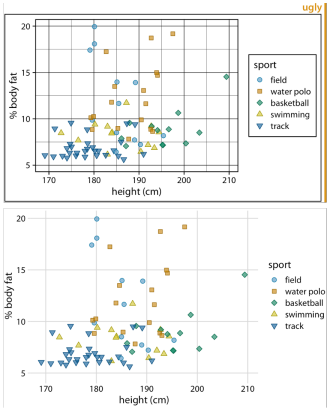
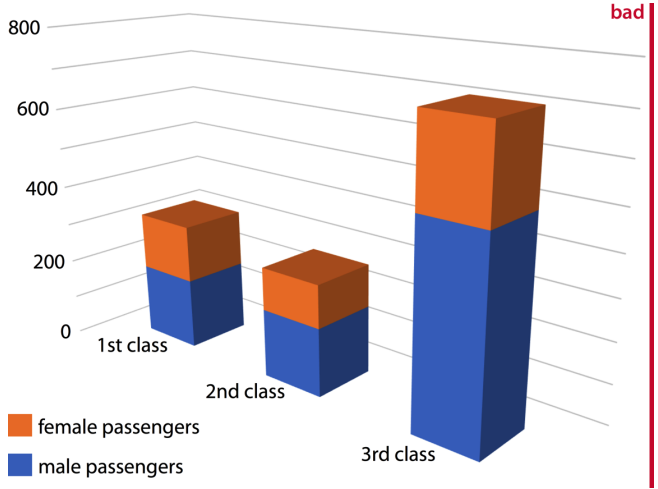


Figure: Wilke, 2019

Principles of visualization

Avoid 3D if unnecessary



Principles of visualization

If possible, tell a story. The left shows us that preprints stopped. The right why: another repository appeared.

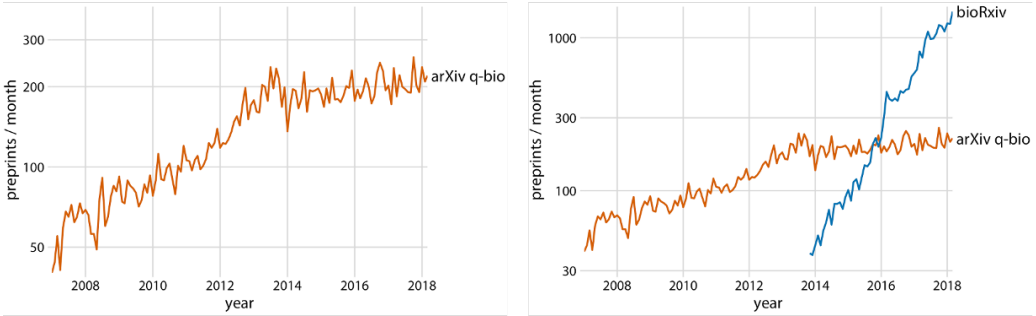


Figure: Wilke, 2019

Principles of visualization

When telling a story, prepare people for complex plots. This is simple (next slide the complex chart)

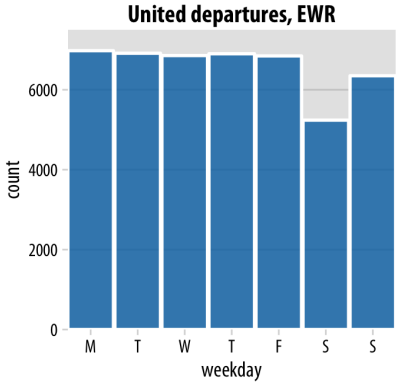


Figure: Wilke, 2019

Principles of visualization

This is more elaborated but similar to the simple one.

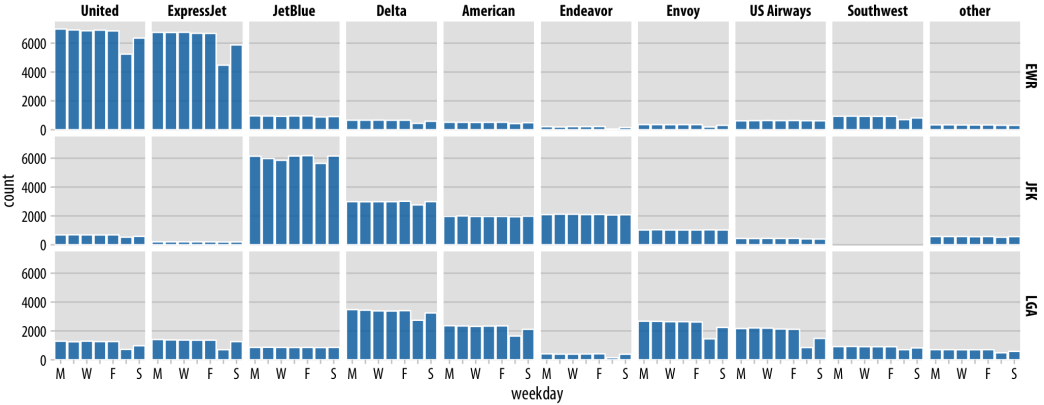


Figure: Wilke, 2019

Table of contents

Visualizations

Visualizations

Check all the available packages

Book: Wilke, 2019
Github repository: Wilke, 2019

Matplotlib + Seaborn

Go to $\sim$ \ Python \ Python_Viz_EDA \ DM_Viz_EDA.ipynb, sections Matplotlib and Seaborn

Grammar of graphics

Ideally, a good plotting syntax should be as close as possible to natural language.

The grammar of graphics approach thrives for that ideal

Grammar of graphics

In the ggplot approach, the grammar is layered. Each layer is a charting primitive.

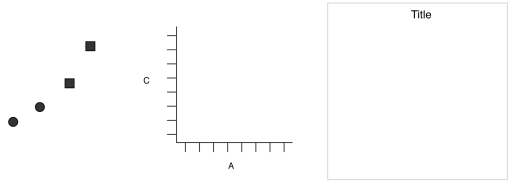


Figure 1. Graphics objects produced by (from left to right): geometric objects, scales and coordinate system, plot annotations.

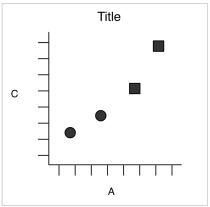


Figure: Wickham, 2010

Grammar of graphics

This is the generic syntax of a ggplot call:

```
ggplot(data = <DATA>) +
   <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
    ) +
    <COORDINATE_FUNCTION> +
   <FACET_FUNCTION>
```

... and concatenate similar as needed

Grammar of graphics

GEOM_FUNCTION: e.g. geom_boxplot()
MAPPINGS: e.g. x = years of ed, y = income (must be in <DATA>
STAT: e.g. stat_identity()
POSITION: e.g. position_dodge
COORDINATE_FUNCTION: e.g. coord_cartesian
FACET_FUNCTION: e.g. facet_grid

Grammar of graphics

Let's go to Python to see ggplot and HoloViews.

Interactive charts

We will continue with Holoviews, but it is good to know that other ones exist, and are widely used (Bokeh, Plotly, Dash)

Exploratory Data Analysis

EDA involves knowing your data, cleaning it, statistics and plotting techniques i.e. this course

But let's see basic tips in EDA

References

- **Balducci, B., & Marinova, D. (2018).**Unstructured data in marketing. *Journal of the Academy of Marketing Science*, *46*, 557–590.
- **Wickham, H. (2010).**A layered grammar of graphics. *Journal of computational and graphical statistics*, 3–28.
- **Wilke, C. O. (2019).** *Fundamentals of data visualization: A primer on making informative and compelling figures*. O'Reilly Media.