1 Lecture 12: Interactivity (continued), other tools, specialized plots

Data Visualization · 1-DAV-105

Lecture by Broňa Brejová

Acknowledgement: some materials inspired by lectures from Martina Bátorová in 2021

1.0.1 Several examples of infographics

Several examples that are close to data visualization: * Income by religious group in US (image, website) * Deadliest pandemics (website) * War casualties (website) * Game of Thrones relationships (website) * Emergency medical services in Slovakia 2019 (website)

Some explain other types of information: * Sitting and standing is bad (website)

1.1 Data visualization (DV) vs infographics (IG)

- **Target audience:** IG general public, DV often experts
- **Storytelling:** often in IG, can be created from multiple DV
- **Design and aesthetics:** more elaborate in IG, includes graphics elements and clipart (considered chart junk in DV)
- **Process of creation:** many simple tools for DV, IG time consuming, often created by collaboration of data analysis, domain experts and graphic designers

See also https://www.statsilk.com/blog/real-difference-between-infographics-and-data-visualizations

1.2 Interactivity

1.2.1 Examples

- PhD gender gap (website)
- Making it big (website)
- US cities with the same name (website)

1.2.2 Techniques in interactivity visualization

Similar to decisions made in designing a static plot:

- Selecting variables (x, y, color, …)
- Filtering data (selecting table rows)
- Highlighting points or groups
- Aggregating (display countries or region summaries)
- Zooming / panning
- Rescaling (log-scale) / reexpressing (e.g. % instead of counts)
- Sorting (e.g. bars in bargraphs)
- Displaying details (tooltips)
- Annotating
- Bookmarking

(Stephen Few)

1.2.3 Dashboard

- A display consisting of mutiple plots, summarizing current state of important indicators (e.g. of a business, pandemics, …)
- Inspired by dashboards in cars and planes
- Often interactive, but main features in default view

Two SARS-CoV-2 examples:

- https://covid19.who.int/
- https://nextstrain.org/ncov/global
  - many options: selecting color, filtering, highlighting, aggregating, zooming and panning (maps and tree), rescaling (time vs divergence), tooltips, bookmarking

1.2.4 Interactivity in Plotly Express

All Plotly plots by default have some interactivity:

- Filtering groups
- Zooming / panning
- Details
- Spike lines

Example 1: Country indicators from World Bank, https://databank.worldbank.org/home under CC BY 4.0 license.

Regions can be switched on and off.

[1]:
```
import plotly.express as px
import pandas as pd
url = 'http://compbio.fmph.uniba.sk/vyuka/viz/images/9/9d/World_bank.csv'
countries = pd.read_csv(url)

px.scatter(
        countries, x="GDP2018", y="Expectancy2018", color="Region",
        hover_data=['Country'],
        title="Country indicators 2018", log_x=True,
        width=800, height=500
    )
```

Example 2: Life expectancy data provided free by the Gapminder foundation under the CC-BY license.

Compare data along the x coordinate.

[2]:
```
url="http://compbio.fmph.uniba.sk/vyuka/viz/images/3/33/
 ↪Gapminder_life_expectancy_years.csv"
orig_expectancy = pd.read_csv(url)
expectancy = pd.melt(orig_expectancy, id_vars=["country"], var_name="year")
expectancy['year'] = expectancy['year'].astype(int)
```

[10]:
```
selected = expectancy.query("country=='Slovak Republic' or country=='France'")
fig=px.line(
        selected, x="year", y="value", color="country",
        title="Life expectancy", width=800, height=500
    )
fig.update_layout(hovermode="x unified")
```

1.2.5 More interaction with Dash by Plotly

- Dash library by Plotly allows adding control elements (selectors, sliders, buttons, …)
- https://dash.plotly.com/installation

[4]:
```
# installation of libraries:
! pip install jupyter-dash
```

```
Collecting jupyter-dash
  Downloading jupyter_dash-0.4.2-py3-none-any.whl (23 kB)
Requirement already satisfied: nest-asyncio in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from jupyter-
dash) (1.5.6)
Requirement already satisfied: ipython in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from jupyter-
dash) (8.11.0)
Collecting ansi2html
  Downloading ansi2html-1.8.0-py3-none-any.whl (16 kB)
Collecting dash
  Downloading dash-2.9.3-py3-none-any.whl (10.2 MB)
     |                | 10.2 MB 19.4 MB/s eta 0:00:01
Collecting retrying
  Downloading retrying-1.3.4-py3-none-any.whl (11 kB)
Requirement already satisfied: ipykernel in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from jupyter-
dash) (6.22.0)
Collecting requests
  Downloading requests-2.29.0-py3-none-any.whl (62 kB)
     |                | 62 kB 5.0 MB/s s eta 0:00:01
Collecting flask
  Downloading Flask-2.3.2-py3-none-any.whl (96 kB)
     |                | 96 kB 28.6 MB/s  eta 0:00:01
Requirement already satisfied: pickleshare in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (0.7.5)
Requirement already satisfied: pexpect>4.3; sys_platform != "win32" in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (4.8.0)
Requirement already satisfied: pygments>=2.4.0 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (2.14.0)
```

```
Requirement already satisfied: jedi>=0.16 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (0.18.2)
Requirement already satisfied: traitlets>=5 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (5.9.0)
Requirement already satisfied: stack-data in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (0.6.2)
Requirement already satisfied: matplotlib-inline in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (0.1.6)
Requirement already satisfied: decorator in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (5.1.1)
Requirement already satisfied: backcall in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (0.2.0)
Requirement already satisfied: prompt-toolkit!=3.0.37,<3.1.0,>=3.0.30 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipython->jupyter-dash) (3.0.38)
Collecting dash-core-components==2.0.0
  Downloading dash_core_components-2.0.0-py3-none-any.whl (3.8 kB)
Collecting dash-table==5.0.0
  Downloading dash_table-5.0.0-py3-none-any.whl (3.9 kB)
Requirement already satisfied: plotly>=5.0.0 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
dash->jupyter-dash) (5.13.1)
Collecting dash-html-components==2.0.0
  Downloading dash_html_components-2.0.0-py3-none-any.whl (4.1 kB)
Requirement already satisfied: six>=1.7.0 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
retrying->jupyter-dash) (1.16.0)
Requirement already satisfied: debugpy>=1.6.5 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipykernel->jupyter-dash) (1.6.6)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipykernel->jupyter-dash) (5.3.0)
Requirement already satisfied: psutil in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipykernel->jupyter-dash) (5.9.4)
Requirement already satisfied: packaging in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipykernel->jupyter-dash) (23.0)
Requirement already satisfied: comm>=0.1.1 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipykernel->jupyter-dash) (0.1.3)
```

```
Requirement already satisfied: tornado>=6.1 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipykernel->jupyter-dash) (6.2)
Requirement already satisfied: jupyter-client>=6.1.12 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipykernel->jupyter-dash) (8.1.0)
Requirement already satisfied: pyzmq>=20 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
ipykernel->jupyter-dash) (25.0.2)
Collecting charset-normalizer<4,>=2
  Using cached charset_normalizer-3.1.0-cp38-cp38-manylinux_2_17_x86_64.manylinu
x2014_x86_64.whl (195 kB)
Collecting urllib3<1.27,>=1.21.1
  Using cached urllib3-1.26.15-py2.py3-none-any.whl (140 kB)
Requirement already satisfied: idna<4,>=2.5 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
requests->jupyter-dash) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
requests->jupyter-dash) (2022.12.7)
Collecting blinker>=1.6.2
  Downloading blinker-1.6.2-py3-none-any.whl (13 kB)
Collecting itsdangerous>=2.1.2
  Using cached itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Requirement already satisfied: importlib-metadata>=3.6.0; python_version <
"3.10" in /home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
flask->jupyter-dash) (6.1.0)
Collecting Werkzeug>=2.3.3
  Downloading Werkzeug-2.3.3-py3-none-any.whl (242 kB)
     |                  | 242 kB 88.9 MB/s eta 0:00:01
Requirement already satisfied: click>=8.1.3 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
flask->jupyter-dash) (8.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
flask->jupyter-dash) (3.1.2)
Requirement already satisfied: ptyprocess>=0.5 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from pexpect>4.3;
sys_platform != "win32"->ipython->jupyter-dash) (0.7.0)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
jedi>=0.16->ipython->jupyter-dash) (0.8.3)
Requirement already satisfied: pure-eval in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from stack-
data->ipython->jupyter-dash) (0.2.2)
Requirement already satisfied: asttokens>=2.1.0 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from stack-
data->ipython->jupyter-dash) (2.2.1)
```

```
Requirement already satisfied: executing>=1.2.0 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from stack-
data->ipython->jupyter-dash) (1.2.0)
Requirement already satisfied: wcwidth in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from prompt-
toolkit!=3.0.37,<3.1.0,>=3.0.30->ipython->jupyter-dash) (0.2.6)
Requirement already satisfied: tenacity>=6.2.0 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
plotly>=5.0.0->dash->jupyter-dash) (8.2.2)
Requirement already satisfied: platformdirs>=2.5 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from jupyter-
core!=5.0.*,>=4.12->ipykernel->jupyter-dash) (3.2.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from jupyter-
client>=6.1.12->ipykernel->jupyter-dash) (2.8.2)
Requirement already satisfied: zipp>=0.5 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from importlib-
metadata>=3.6.0; python_version < "3.10"->flask->jupyter-dash) (3.15.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/home/bbrejova/viz/notebooks/venv/lib/python3.8/site-packages (from
Werkzeug>=2.3.3->flask->jupyter-dash) (2.1.2)
Installing collected packages: ansi2html, blinker, itsdangerous, Werkzeug,
flask, dash-core-components, dash-table, dash-html-components, dash, retrying,
charset-normalizer, urllib3, requests, jupyter-dash
Successfully installed Werkzeug-2.3.3 ansi2html-1.8.0 blinker-1.6.2 charset-
normalizer-3.1.0 dash-2.9.3 dash-core-components-2.0.0 dash-html-
components-2.0.0 dash-table-5.0.0 flask-2.3.2 itsdangerous-2.1.2 jupyter-
dash-0.4.2 requests-2.29.0 retrying-1.3.4 urllib3-1.26.15
```

[11]:
```python
from jupyter_dash import JupyterDash
from dash import dcc
from dash import html
from dash.dependencies import Input, Output
app = JupyterDash(__name__)
# create a list of all regions
regions = [{'label': region, 'value': region}
           for region in countries['Region'].unique()]
# create layout of items in application
app.layout = html.Div([
    dcc.Graph(id='graph'),
    dcc.Dropdown(
      id='region_list',
      options=regions,
      value=['Europe & Central Asia'],
      multi=True
    )
])
```

```
# define callback to update graph
@app.callback(
    Output('graph', 'figure'),
    [Input("region_list", "value")]
)
def update_figure(region_list):
    countries_subset = countries.query('Region in @region_list')
    return px.scatter(
        countries_subset, x="GDP2018", y="Expectancy2018",
        color="Fertility2018", hover_data=['Country'],
        title="Country indicators 2018", log_x=True, width=800, height=500
    )
# run application
app.run_server(mode='inline')
```

```
Dash is running on http://127.0.0.1:8050/


<IPython.lib.display.IFrame at 0x7f76910adbe0>
```

1.3 Other visualization tools

Non-programmers typically create plots in spreadsheets: * Excel (examples) * Google sheets (examples)

System R: programming language for statistical computing * Together with Python, very popular in data science * Built-in plots * Also other libraries, notably ggplot2 based on system called Grammar of Graphics (cheatsheet)

Javascript * Programming language popular in web programming * Google charts for Javascript (examples) * D3.js library (Data-Driven Documents)

Tableau * Advanced visualization tools, commercial * Gallery

Microsoft Power BI * Interactive data visualization software with a focus on business intelligence * An example

1.4 Several specialized visualization types

1.4.1 UML diagrams in computer science

- Display relationships between different classes or other components and aspects of software

https://commons.wikimedia.org/wiki/File:UML_diagrams_overview.svg Derfel73; Pmerson

1.4.2 Waterfall chart

- Used in bussiness analysis: financial, inventory, human resources etc.
- Displays effects decreasing or increasing a given value
- The first and last columns are bars displaying starting and final value
- Intermediate columns float, displaying changes from previous total
- Description

https://commons.wikimedia.org/wiki/File:Waterfallchart_ex2.jpg FusionCharts Blog, CC BY-SA 4.0

1.4.3 Funnel charts

- Display losses within a business process, e.g from website visit to actual purchase
- Horizontal bar chart with centered bars
- Beware: different from funnel plot in medical meta-analyses of multiple publications

[6]:
```
# example from https://plotly.com/python/funnel-charts/
data = dict(
    number=[39, 27.4, 20.6, 11, 2],
    stage=["Website visit", "Downloads", "Potential customers", "Requested␣
 ↪price", "invoice sent"])
fig = px.funnel(data, x='number', y='stage')
fig.show()
```

1.4.4 Gantt chart

- Used in management to display project schedule with different tasks and their planned duration
- Can also display current status of tasks and their dependencies

https://commons.wikimedia.org/wiki/File:GanttChartAnatomy.svg

1.4.5 Candlestick chart

- Similar to boxplot, used in financial data, e.g. stocks, currency exchange rates
- Line: minimum and maximum, box: opening and close, color: increase or decrease

https://commons.wikimedia.org/wiki/File:Candlestick_Chart_in_MetaTrader_5.png