Boosting your data analysis skills with data visualisation and ggplot2

Direction Access to and Reuse of Public Information

Unit EU Open Data and CORDIS

Sector EU Open Data
The context

This training course is organized in the scope of OP project within the ISA2 programme.

ISA2 supports the development of digital solutions enabling public administrations, businesses and citizens in Europe to benefit from interoperable cross-border and cross-sector public services.

How OP is involved in ISA2?

OP is aiming at developing open data related activities in the areas of:
- Data visualisation
- Linked open data
- Persistent identification
## Upcoming training & workshop sessions

<table>
<thead>
<tr>
<th>Topic</th>
<th>Type of session</th>
<th>Lux + webex</th>
<th>Bxl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telling your story through data visualisation</td>
<td>Training</td>
<td>25/06</td>
<td>28/06</td>
</tr>
<tr>
<td>Making great online data visualisations without coding</td>
<td>workshop</td>
<td>26/06</td>
<td>-</td>
</tr>
<tr>
<td>Going beyond bars and lines: practising non-standard data visualisation</td>
<td>Training</td>
<td>24/09</td>
<td>Sep-Oct</td>
</tr>
<tr>
<td>Making data visualisations like a pro: D3.js</td>
<td>Workshop</td>
<td>25/09</td>
<td>-</td>
</tr>
<tr>
<td>Applying data visualisation best practices in real use cases</td>
<td>workshop</td>
<td>24/10</td>
<td>-</td>
</tr>
</tbody>
</table>

and **webinars** (topic like for the trainings) ... stay tuned!

Data visualization events in 2019

**EU Datathon 2019**

- **Date:** 13 June 2019
- **Venue:** Residence Palace - Brussels
- **Website:** https://publications.europa.eu/eudatathon
- **e-mail:** op-datathon@publications.europa.eu

**EU DataViz 2019 - Data Visualisation for the Public Sector**

- **Date:** 12 November 2019
- **Venue:** European Convention Center - Luxembourg
- **Website:** https://publications.europa.eu/eudataviz
- **e-mail:** op-eu-dataviz@publications.europa.eu
Where to find the information about our training workshop and webinar sessions?

Visit our data visualisation community to find all the information about previous and upcoming sessions at:

Agenda

09:00  Introduction
      Visualisation in data analysis
      ggplot, R and RStudio

10:30  Coffee break

12:00 - 13:00  Lunch
      Visualising covariation
      Visualising multiple dimensions

14:30  Coffee break
      Visualising time series
      Saving and sharing visualisations
      Other tools

16:30  Q&A
1. INTRODUCTION
Participants

Institution/DG and role?
Experience with coding, R and ggplot2?
Expectations for today?
2. VISUALISATION IN DATA ANALYSIS?
EDA
What is it?
EDA
What is it?
John Tukey

"The simple graph has brought more information to the data analyst’s mind than any other device"

"The greatest value of a picture is when it forces us to notice what we never expected to see"
"EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task"
Your goal during EDA is to develop an understanding of your data.

EDA is an iterative cycle. You
- generate questions about your data
- search for answers by visualising, transforming, and modelling your data
- use what you learn to refine your questions and/or generate new questions
EDA
Tool requirements

Fast
Slice & dice data
Iterate over visualisations
Good defaults

Flexible
Many visualisation types
Combine visualisations

Integrates with data manipulation
The Tidyverse
EDA
R Tidyverse
tidyverse.org

Declaratively create graphics

Grammar for data manipulation

Get data in tidy format
3.
GGPLOT2
ggplot2
Grammar of graphics

R package for making visualisations

Based on the "Grammar of Graphics" by Leland Wilkinson

"A tool that enables us to concisely describe the components of a graphic"
ggplot2
Grammar of graphics

An example of how a visualisation is constructed from a data set

<table>
<thead>
<tr>
<th>Key</th>
<th>Response</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rarely</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Infrequently</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Occasionally</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>Frequently</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Not Sure</td>
<td>32</td>
</tr>
</tbody>
</table>
ggplot2
Grammar of graphics

Grammar of graphics specification

**Specification:**
FRAME: female
COORD: polar.theta()
GRAPH: bar(label(response), color(response), position.stack(), shape.rect())

**Graphic:**
\(\text{bar(<aesthetic attributes>) = perceivable graphic}\)

**Aesthetics:**
\(\text{label(response): "Rarely" } \mapsto \text{"Rarely"}, \text{"Infrequently" } \mapsto \text{"Infrequently"}, \text{"Occasionally" } \mapsto \text{"Occasionally"}, \text{"Frequently" } \mapsto \text{"Frequently"}, \text{"Not sure" } \mapsto \text{"Not sure"}\)
\(\text{color(response): "Rarely" } \mapsto \text{green, "Infrequently" } \mapsto \text{blue,} \text{"Occasionally" } \mapsto \text{yellow, "Frequently" } \mapsto \text{red,} \text{"Not Sure" } \mapsto \text{violet}\)
\(\text{position.stack(female): } [0,8) \mapsto [0,8), [0,11) \mapsto [8,19), [0,17) \mapsto [19,36), \text{[0,32) } \mapsto [36,68), [0,32) \mapsto 68,100]\)
\(\text{shape.rect(female): } [0,8) \mapsto [], [0,11) \mapsto \Box, [0,17) \mapsto \Box, \text{[0,32) } \mapsto \Box, [0,32) \mapsto \Box\)
ggplot2
Grammar of graphics

Grammar of graphics: result
ggplot2

Ggplot2 is an implementation of the grammar of graphics in R.

How do we go from data to a visualisation in ggplot2?

### Data

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>15</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>80</td>
<td>b</td>
<td></td>
</tr>
</tbody>
</table>

### Used aesthetics

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>80</td>
<td>b</td>
<td></td>
</tr>
</tbody>
</table>

---

**Plot**

**Title**

Data in aesthetic space

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>11</td>
<td>circle</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>circle</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>53</td>
<td>square</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>300</td>
<td>square</td>
<td></td>
</tr>
</tbody>
</table>
ggplot2

Map data to aesthetics of geometries

Data

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>a</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>a</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>15</td>
<td>b</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>80</td>
<td>b</td>
</tr>
</tbody>
</table>

ggplot(data=Data, aes(x=A, y=B, shape=D)) + geom_point()
library(ggplot2)

eexampledata <- read.csv("ggplot-example-data.csv")

View(exampledata)

ggplot(data = exampledata, aes(x = A, y = B, shape = D)) + geom_point()
Components of a ggplot2 plot

Data

Scales (map data to aesthetics)

Geometric objects

Coordinate system (a pie chart is just a bar chart in polar coords)

Statistics

Facets
4. **R & RSTUDIO**
Free, open source programming language

Initially for statistics and graphics
Today: maps, publishing documents, interactive dashboards, blogging, etc.

Big community: learning resources, packages, events
RStudio

Free, open source client for R

rstudio.com
rstudio.cloud

Rstudio online

Runs in the browser, no installation needed

Share R projects

rstudio.cloud
rstudio.cloud

Set up

Go to rstudio.cloud

Click "Sign Up" and create an account

Go to rstudio.cloud/project/351652

Click "Save a permanent copy"
```r
install.packages("ggplot2")
library(ggplot2)
ggplot(data = women, aes(x = weight, y = height)) +
geom_point()
```
RStudio interface

Source

Write scripts

Preview data

Uses tabs
RStudio interface

Console

Execute R code

Output of code

Errors and warnings
Source (aka Editor)

Console

Environment

Files
Plots
Packages
Help
RStudio interface

Environment

Loaded objects

History of commands

"Import dataset" wizard
### RStudio interface

<table>
<thead>
<tr>
<th>Files</th>
<th>File explorer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plots</td>
<td>ggplot output</td>
</tr>
<tr>
<td>Packages</td>
<td>Installed packages</td>
</tr>
<tr>
<td>Help</td>
<td>Documentation of functions and packages</td>
</tr>
</tbody>
</table>
R basics

Load data
"Import dataset" wizard
read.csv("mydatafile.csv")
eurostat::get_eurostat("ilc_hch10")

Assign
x <- 5
mydataframe <- read.csv("")

Inspecting data
View("")
head("")

Get help
?read.csv
?head
# R & Rstudio basics

## Packages

Install and load the ggplot2 package:

```r
install.packages("ggplot2")
library(ggplot2)
```

## Run code

- "Run" button in Source pane
- Using the console

## Saving script

- File => Save as
- .R files
Exercise
4-load-data.R

View(...) open data in the data viewer

read.csv(...) load csv files

library(...) load (installed) package

readxl::read_excel(...) load Excel files and sheets

?... open the documentation

eurostat package find and load eurostat data
5. DATA PROFILING
Data profiling

A first glimpse of the data

How big is the data?

What are the data types?

Missing data? Errors?

Starting to understand the data
Data profiling

Missing data
NA values

Inspect
Check the object under Environment

Run the data frame
Data frame will be outputted in the console
Exercise
5-1-data-profiling.R

ncol(...)  number of columns
nrow(...)  number of lines

typeof(...)  data type of a column

str(...)  compact display of R object

summary(...)  summary statistics

visdat::vis_dat(...)  visual profile of dataframe
Exercise
5-2-check-assumptions.R

geom_col()  bar and column charts

dplyr::filter()  from the dplyr package, filter
data on 1 or more conditions
6. VISUALISING DISTRIBUTIONS
Visualising distributions

1 numerical column
1 optional categorical column

Range of the distribution

Shape of the distribution (symmetrical, bimodal, long tails, ...)

Central values

Outliers
ggplot2 Cheat sheet

### Data Visualization with ggplot2

**Basics**

- ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same few components: a data set, a set of geoms—visual marks that represent data points, and a coordinate system.
- To display data values, map variables in the data set to aesthetic properties of the geom like size, color, and x and y locations.

**Geoms**

- Use a geom to represent data points, use the geom’s aesthetic properties to represent variables. Each function returns a layer.

**One Variable**

<table>
<thead>
<tr>
<th>Continuous</th>
<th>Two Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>a &lt;- ggplot(mpg, aes(hwy))</td>
<td>Continuous Bivariate Distribution</td>
</tr>
<tr>
<td>a + geom_area(stat = &quot;bin&quot;)</td>
<td>f &lt;- ggplot(movies, aes(year, rating))</td>
</tr>
<tr>
<td>x, y, alpha, color, fill, linetype, size, weight</td>
<td>f + geom_blank()</td>
</tr>
<tr>
<td>b + geom_area(aes = ...</td>
<td>f + geom_bin2d(binwidth = c(5, 0.5))</td>
</tr>
<tr>
<td>stat = &quot;bin&quot;)</td>
<td>x, y, alpha, min, max, ymin, ymax, alpha, color, fill, linetype, size, weight</td>
</tr>
<tr>
<td>a + geom_density(aes = ...</td>
<td>f + geom_density2d()</td>
</tr>
<tr>
<td>stat = &quot;density&quot;)</td>
<td>x, y, alpha, color, fill, lineend, size, weight</td>
</tr>
<tr>
<td>x, y, alpha, color, fill, linetype, size, weight</td>
<td>j &lt;- ggplot(cars, aes(speed, dist))</td>
</tr>
<tr>
<td>b + geom_density(aes = ...</td>
<td>j + geom_hex()</td>
</tr>
<tr>
<td>stat = &quot;density&quot;)</td>
<td>x, y, alpha, color, fill, lineend, size, weight</td>
</tr>
</tbody>
</table>

**Graphical Primitives**

<table>
<thead>
<tr>
<th>Continuous X, Continuous Y</th>
<th>Discrete X, Continuous Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>f &lt;- ggplot(mpg, aes(cty, hwy))</td>
<td>g &lt;- ggplot(mpg, aes(class, hwy))</td>
</tr>
<tr>
<td>f + geom_blank()</td>
<td>g + geom_bar(stat = &quot;identity&quot;)</td>
</tr>
<tr>
<td>f + geom_jitter()</td>
<td>x, y, alpha, color, fill, linetype, size, weight</td>
</tr>
<tr>
<td>f + geom_point()</td>
<td>x, y, alpha, color, fill, linetype, size, weight</td>
</tr>
<tr>
<td>f + geom_quantile()</td>
<td>x, y, alpha, color, fill, linetype, size, weight</td>
</tr>
<tr>
<td>f + geom_smooth(model = &quot;lm&quot;)</td>
<td>x, y, alpha, color, fill, linetype, size, weight</td>
</tr>
<tr>
<td>f + geom_text(aes(label = ...</td>
<td>geom_smooth()</td>
</tr>
<tr>
<td>x, y, label, alpha, angle, color, family, fontface, fontfamily, hjust, vjust, lineheight, size, vjust</td>
<td></td>
</tr>
</tbody>
</table>

**Discrete**

- a <- ggplot(mpg, aes(hwy))
- b + geom_bar() |

**Three Variables**

<table>
<thead>
<tr>
<th>Continuous</th>
<th>Discrete X, Discrete Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>f &lt;- ggplot(movies, aes(year, rating))</td>
<td>m + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)</td>
</tr>
<tr>
<td>f + geom_blank()</td>
<td>m + geom_tile(aes(fill = z))</td>
</tr>
<tr>
<td>f + geom_bin2d(binwidth = c(5, 0.5))</td>
<td>x, y, alpha, color, fill, linetype, size, weight</td>
</tr>
<tr>
<td>x, y, alpha, color, fill, lineend, size, weight</td>
<td>m + geom_point(jitter = TRUE)</td>
</tr>
</tbody>
</table>

**Maps**

- data <- data.frame(murder = USArrests$Murder, state = tolower(names(USArrests)))
- map <- map_data(state)
- i <- ggplot(data, aes(x = murder, y = USArrests))
- i + geom_map(aes(map_id = state, map = map) | expand_limits(xmin = map$xlim, xmax = map$xlim), map = map)
Distributions

Points

The easiest way to visualise a distribution: plotting each value

`geom_point()`

If you have overlap: use alpha or `geom_jitter()`
Distributions
Boxplots

`geom_boxplot()`

ggplot calculates the medians, quartiles and outliers for us
Exercise
6-1-distributions-points-boxplots.R

geom_point()  plot points (circles, symbols)

geom_boxplot()  plot boxplots

geom_jitter()  add jittering

geom_text()  plot text

geom_1() + geom_2()  combine geoms
Distributions
Boxplots alternatives
Distributions

Histograms

geom_histogram()

ggplot does the counting for us

binwidth/bins parameters is very important: try different bin sizes
Distributions

Frequency polygons

geom_freqpoly()

Same as histogram, but with lines instead of bars

Better suited to compare distributions between different categories
Distributions
Density plot

`geom_density()`

Smooth version of `geom_freqpoly()`

Use when underlying data is smooth
Distributions

Violin plot

geom_violin()  

Compact, mirrored version of geom_density()
Exercise
6-2-distributions-othergeoms.R

geom_histogram() plot histograms

geom_freqpoly() plot frequency polygons

geom_density() plot density estimates

geom_violin() plot violin plots
7.

VISUALISING COVARIATION
Visualising covariation

Two numerical variables

Optionally one categorical variable

Identify correlations

Detect outliers
Covariation Scatterplots

`geom_point()`

`geom_text()`
`geom_label()`

Use `geom_jitter()` for discrete or categorical variables
Covariation Scatterplots

Overcome overplotting by binning with

```
geom_bin2d()
geom_hex()
```

Or use transparency: `alpha`
Exercise

7-1-scatterplots.R

g`"om_point()` plot points

g`"om_bin2d()` bin points in rectangles

g`"om_hex()` bin points in hexagons

scale_x_continuous() customise continuous scale

scale_fill_continuous() customise color scale
Exercise: Why visualise?

7-2-anscombe.R

Dataset "anscombe"

View(anscombe)

Check the means with summary(anscombe)

Calculate some stats:
sd(anscombe$x1)
cor(anscombe$x1, anscombe$y1)

Now make scatterplots:

x1 ~ y1
x2 ~ y2
x3 ~ y3
x4 ~ y4

What do you see?
EDA
Why visualise?

Datasaurus
Covariation Heatmaps

2 categorical or discrete variables + 1 numerical variable

gemm_tile()
Exercise
7-3-heatmaps.R

`geom_tile()` plot rectangles
8. MULTIDIMENSIONAL DATA
Multidimensional Add aesthetics

Geoms have many aesthetics. For example, `geom_point()` understands the following:

- x*
- y*
- color
- fill
- group
- shape
- size
- stroke
Exercise: Why visualize?

8-1-gapminder.R

+ aes()  
  Add aesthetic to a plot

scale_x_log10()  
  Logarithmic scale

scale_size_area()  
  Size scale
Multidimensional Facetting

Make small multiple charts by facetting plots with `facet_wrap()` and `facet_grid()`

Free vs fixed scales
Exercise

8-2-facetting.R

**facet_wrap()**  plot split into small multiples for 1 variable

**facet_grid()**  plot split into small multiples for 2 variables
Multidimensional Parallel coordinates

Plot many dimensions and see correlation between them

**GGally::ggparcoord()**

Add boxplots
Exercise: Why visualize?

8-3-parallel-coordinates.R

GGally::parcoord() plot parallel coordinate plot
Multidimensional
Multiple views &
interactivity

Filters & interaction
Multiple visualisations
Example

Coordinated views
Example
9.

VISUALISING TIME SERIES
Time series

`geom_line()`

`geom_line()` with group or color aesthetic

`geom_area()`

`scale_x_time()`

`scale_x_date()`
Exercise

9-time-series.R

geom_line()  plot line charts

geom_area()  plot area charts

scale_x_date()
10. SAVING AND SHARING VISUALISATIONS
Saving

`ggsave()`

Plots panel => Export  

save a plot to your project/hard drive

`ggsave()`  

file formats  
dpi  
units
Sharing Rmarkdown

Mix text, R-code and R-output

Output to html, Word, Powerpoint, etc.

File => New file => R Markdown...

Example
(preparation for Why Budapest, Lithuania and Warsaw split themselves in two)
11.
OVERVIEW OF OTHER TOOLS
Visual EDA

Other tools

Python:
- pandas, seaborn, matplotlib
- altair

Without coding:
- Qlik
- Tableau
Q&A
Resources

Grammar of Graphics
A layered grammar of graphics

R & EDA
R for Data Science

ggplot2
Tidyverse: ggplot2
Data visualization: A practical introduction
Fundamentals of Data Visualisation
ggplot2 Cheat Sheet
**Upcoming training & workshop sessions**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Type of session</th>
<th>Lux + webex</th>
<th>Bxl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telling your story through data visualisation</td>
<td>Training</td>
<td>25/06</td>
<td>28/06</td>
</tr>
<tr>
<td>Making great online data visualisations without coding</td>
<td>workshop</td>
<td>26/06</td>
<td>-</td>
</tr>
<tr>
<td>Going beyond bars and lines: practising non-standard data visualisation</td>
<td>Training</td>
<td>24/09</td>
<td>Sep-Oct</td>
</tr>
<tr>
<td>Making data visualisations like a pro: D3.js</td>
<td>Workshop</td>
<td>25/09</td>
<td>-</td>
</tr>
<tr>
<td>Applying data visualisation best practices in real use cases</td>
<td>workshop</td>
<td>24/10</td>
<td>-</td>
</tr>
</tbody>
</table>

and **webinars (topic like for the trainings)** ... stay tuned!

Materials will be published on https://data.europa.eu/euodp/en/knowledge-center