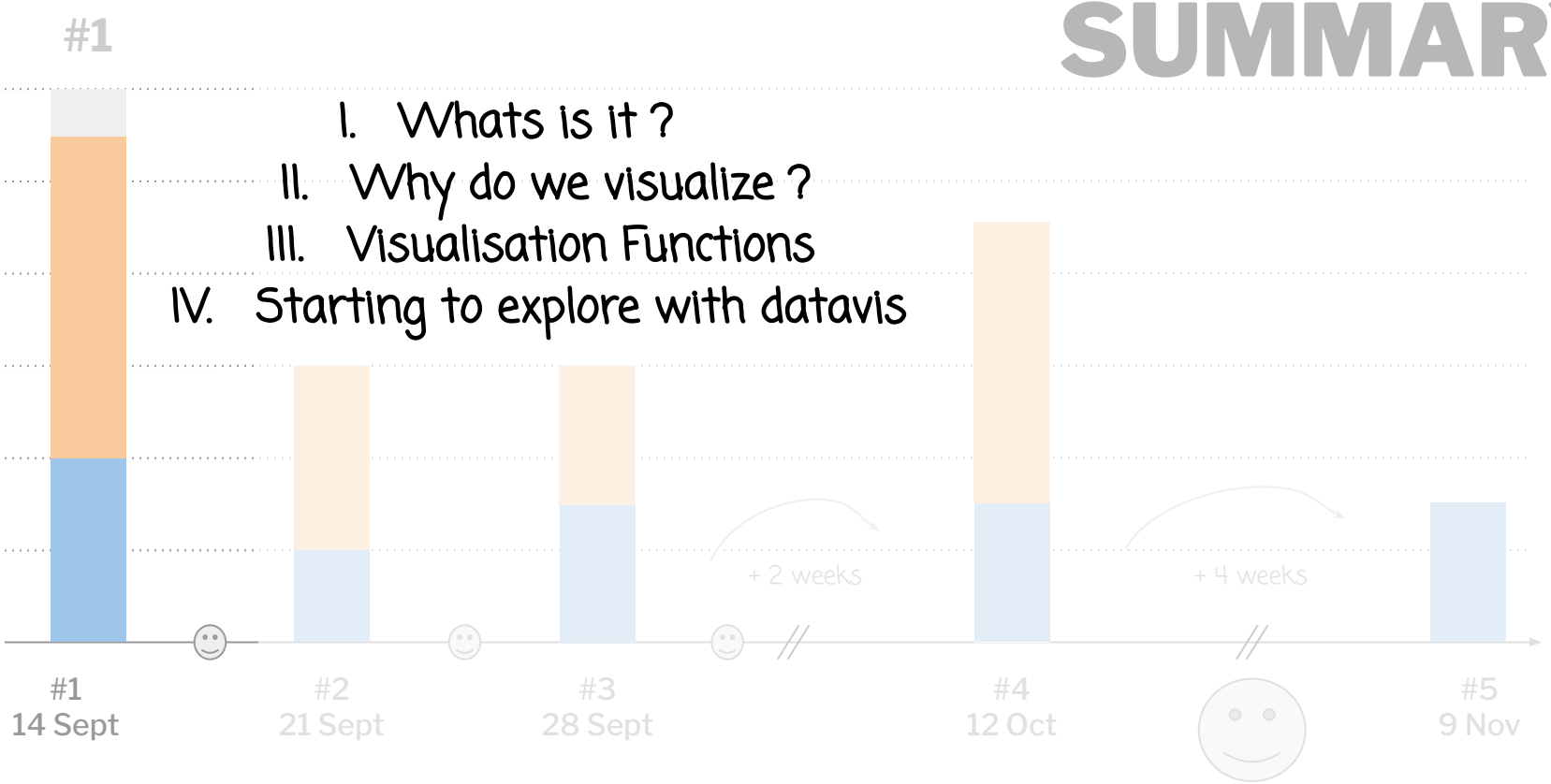
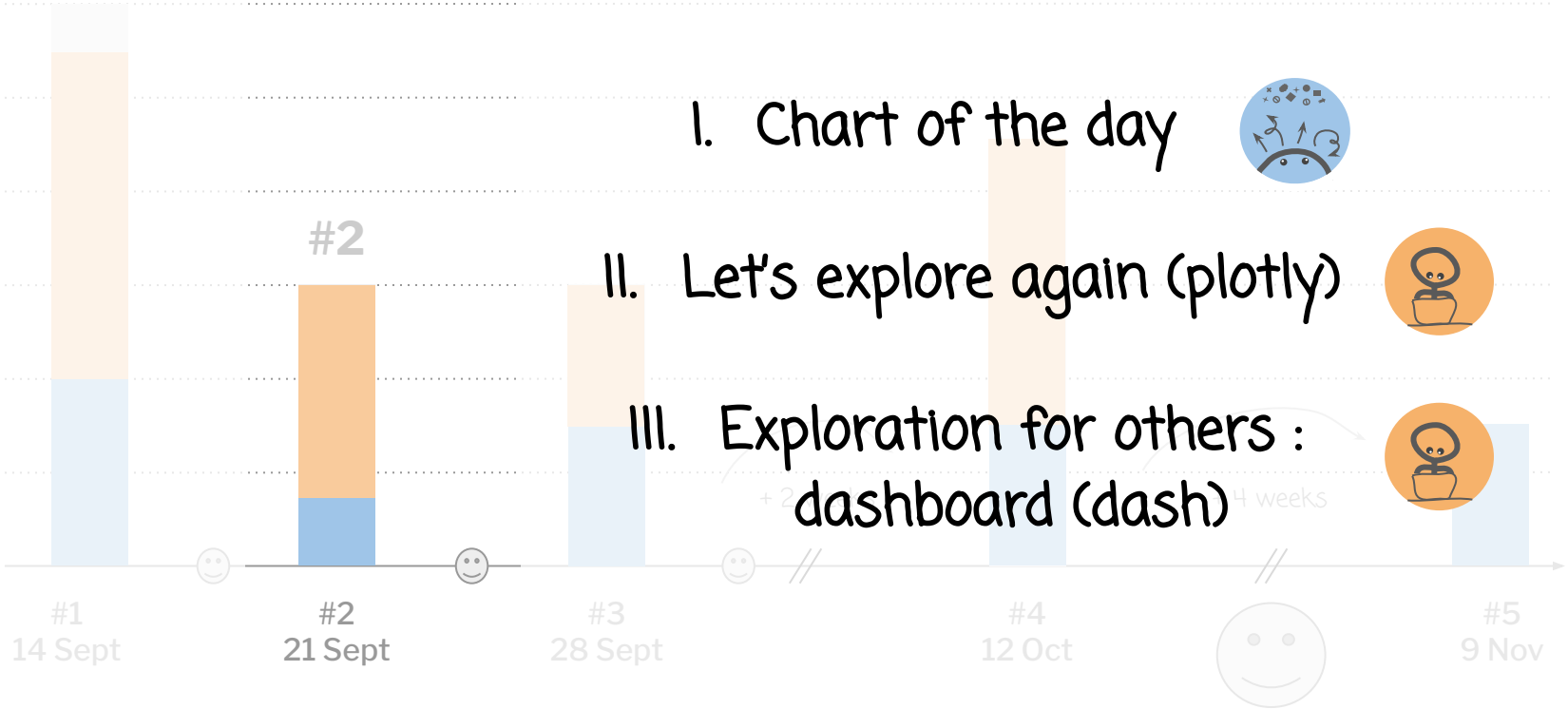
introduction to

DATA VISUALISATION

LAST DAY SUMMARY



TODAY



I. Chart of the day

II. Let's explore again (plotly)

III. Exploration for others : dashboard (dash)

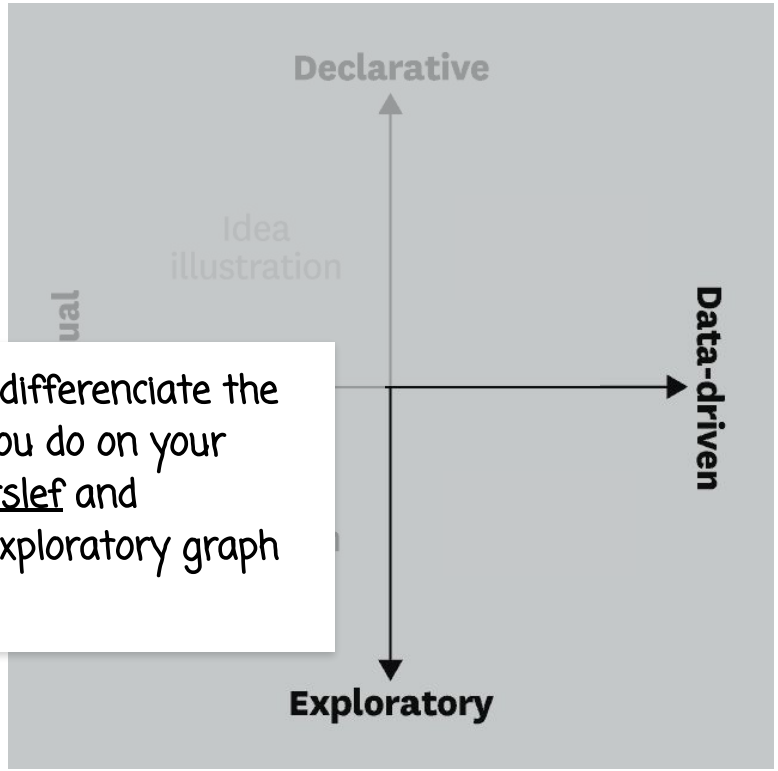
#2

I. Chart of the day

II. Let's explore again (plotly)

III. Exploration for others : dashboard

communicating information



We can also differenciate the exploration you do on your own, for yourslef and creating an exploratory graph for users

trying to figure something out → confirmation
→ discovery

Source : Harvard Business Review

exploratory graph/tools <u>for users</u>

dashboard ?

Tableau Multichannel Attribution ...
windsor.ai

Hyper – Admin & Dashboard Template ...
themes.getbootstrap.com

Sales Performance Dashboard | Sales ...
boldbi.com

dashboard examples from real companies ...
geckoboard.com

Sales Dashboards Template & KPI ...
clicdata.com

Zoho Flow dashboard
zoho.com

Dashboards
docs.datalust.co

Customer Service Insights dashboards ...
docs.microsoft.com

Dashboards For Project Manag...
clicdata.com

How to Build a Stunning Interactive ...
towardsdatascience.com

Dashboard Templates For Admins ...
colorlib.com

data visualisation et storytelling ...
fanvoice.com

Pin on Data visualization

Financial Performance Dashboard ...

Investor Dashboard - Personal ...

Business Dashboards ...

SEO dashboard using Google Analytics ...

How to design a sleek dashboard UI | by ...

exploratory graph/tools <u>for users</u>

dashboard ?

Tableau Multichannel Attribution ...
windsor.ai

Hyper – Admin & Dashboard Template ...
themes.getbootstrap.com

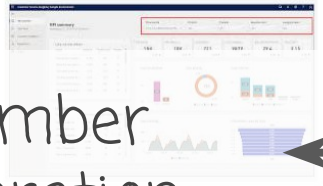
Sales Performance Dashboard | Sales ...
boldbi.com

dashboard examples from real companies ...
geckoboard.com

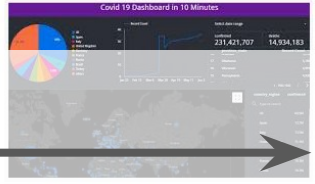
Sales Dashboards Template & KPI ...
clicdata.com

Zoho Flow dashboard
zoho.com

Dashboards
docs.datalust.co



Dashboards For Project Manag...
clicdata.com



How to Build a Stunning Interactive ...
towardsdatascience.com

data visualisation et storytelling ...
fanvoice.com

number decoration

data explorer

Pin on Data visualization

Financial Performance Dashboard ...

Investor Dashboard - Personal ...

Business Dashboards ...

SEO dashboard using Google Analytics ...

How to design a sleek dashboard UI | by ...

Source : [Moritz Stefaner Peak Spotting project] and [talk (from 23')]

problematic

German train system is open, can but don't have to book

Goal : maximize use without having too full trains

Indirect ways to affect travelers #

resources

Predictive models of passengers load but no tools for humans to work with that data

Many different tables in different places

Printing and drawing at peak time

Source : Moritz Stefaner Peak Spotting project and talk (from 23')

#2

problematic

German train system is open, can but don't have to book

Goal : maximize use without having too full trains

Indirect ways to affect travelers #

resources

Predictive models of passengers load but no tools for humans to work with that data

Many different tables in different places

Printing and drawing at peak time

process

Understand data and users

Immersion (work with real data)

Prototyping with accessible tools (notebooks, tableau ...) for real use

Source : [Moritz Stefaner Peak Spotting project](#) and [talk (from 23')](#)

#2

problematic

German train system is open, can but don't have to book

Goal : maximize use without having too full trains

Indirect ways to affect travelers #

resources

Predictive models of passengers load but no tools for humans to work with that data

Many different tables in different places

Printing and drawing at peak time

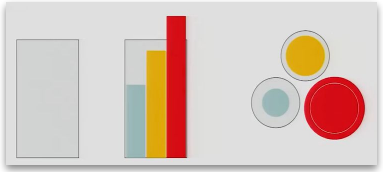
process

Understand data and users

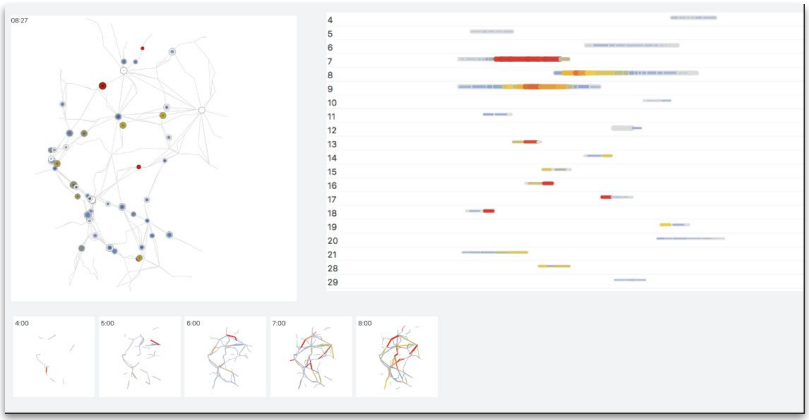
Immersion (work with real data)

Prototyping with accessible tools (notebooks, tableau ...) for real use

Define a specific visual langage to chose encodings



Favor quick explorations over perfectionism to find the right angle



Source : Moritz Stefaner Peak Spotting project and talk (from 23')

1. Chart of the day

#2

process

resources

Predictive models of passengers load but no tools for humans to work with that data

Many different tables in different places

Printing and drawing at peak time

Favor quick explorations over perfectionism to find the right angle

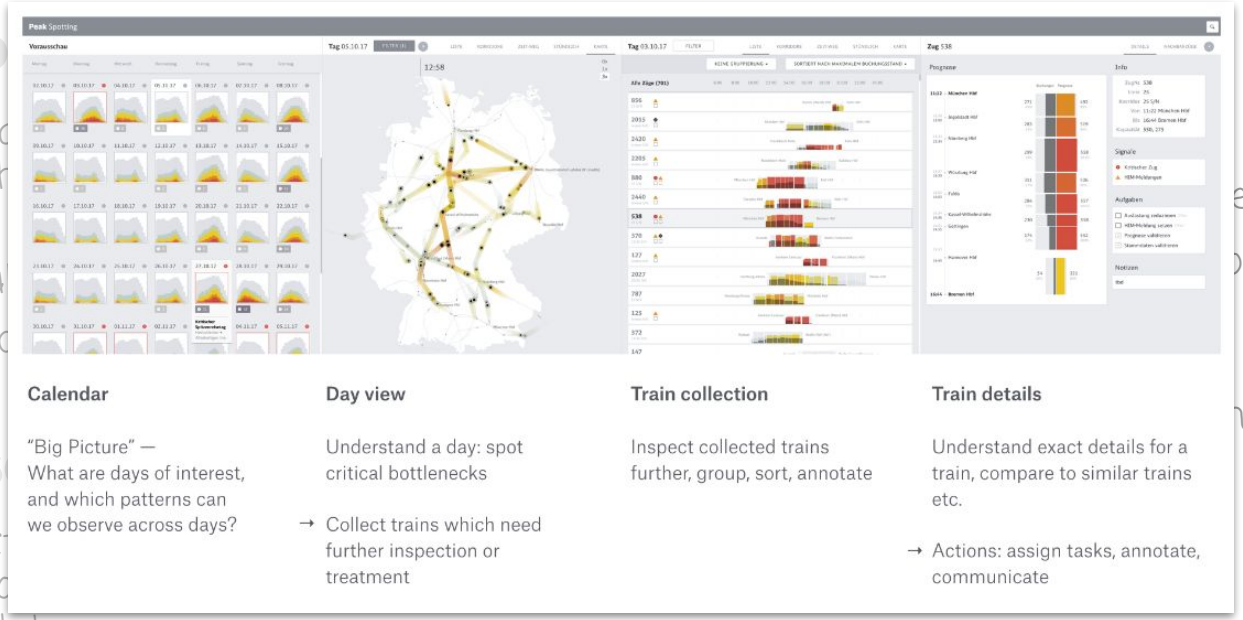
Doubt the data (missing ? do we measure the right thing ?)

Test alternatives and iterate

Source : Moritz Stefaner Peak Spotting project and talk (from 23')

#2

process



Refine and information architecture (combining views) + UI

Visualize + analyse user behavior : what's actual useful for users over time

Source : [Moritz Stefaner Peak Spotting project] and [talk (from 23')]

problematic

German train system is op
don't have to book

Goal : maximize use withou
too full trains

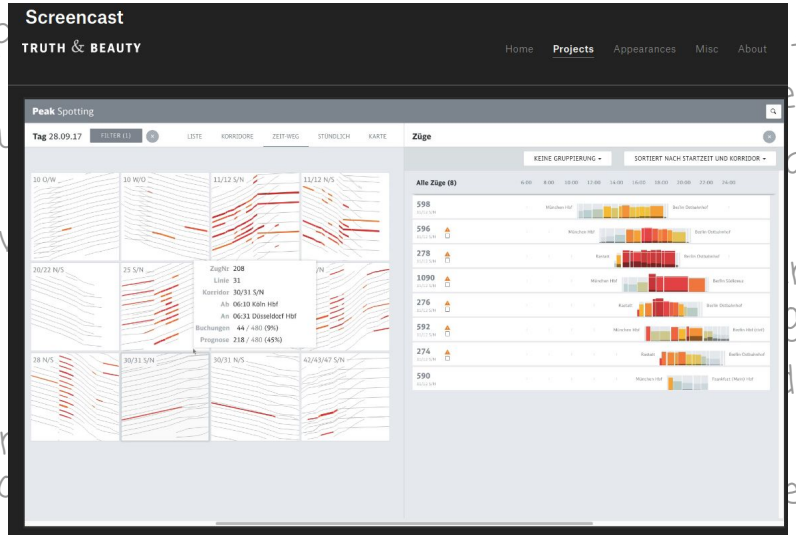
Indirect ways to affect trav

resources

Predictive models of passer
but no tools for humans to
that data

Many different tables in different places

Printing and drawing at peak time



process

Understand data and users

Immersion (work with real data)

tools
eal use

age to chose

r

ght angle

lo we

e

Refine and information architecture (combining views) + UI

Visualize + analyse user behavior : what's actual useful for users over time

Source : Moritz Stefaner Peak Spotting project and talk (from 23')

I. Chart of the day

II. Let's explore again (plotly)

III. Exploration for others : dashboard

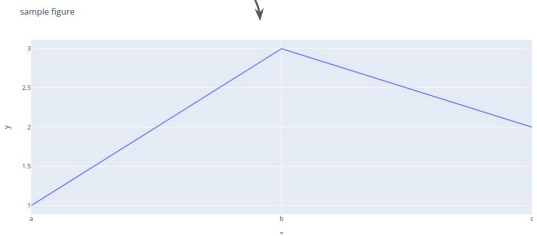
Let's explore again !

#1

Figures can be represented in Python either as dicts or as instances of the **plotly.graph_objects.Figure** class, and are serialized as text in JSON before being passed to Plotly.js

```
fig = px.line(
        x=["a","b","c"], y=[1,3,2],
        title="sample figure")

print(fig)
fig.show()
```




```
Figure({
  'data': [{'hovertemplate': 'x=%{x}<br>y=%{y}<extra></extra>',
            'legendgroup': '',
            'line': {'color': '#636efa', 'dash': 'solid'},
            'mode': 'lines',
            'name': '',
            'orientation': 'v',
            'showlegend': False,
            'type': 'scatter',
            'x': array(['a', 'b', 'c'], dtype=object),
            'xaxis': 'x',
            'y': array([1, 3, 2]),
            'yaxis': 'y'}],
  'layout': {'legend': {'tracegroupgap': 0},
             'template': '...',
             'title': {'text': 'sample figure'},
             'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'x'}},
             'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'y'}}}
})
```

data types

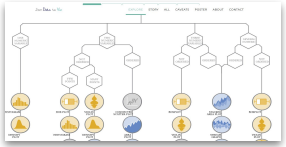


chart type

data relationship

GAPMINDER

jupyter plotly Dash

Source : From data to viz - Yan Holtz, Gapminder data

Let's explore again !

plotly

2 ways to create graphs

Plotly Express (included as the plotly.express module) is a high-level data visualization API that produces fully-populated graph object figures in single function-calls.

[See the gallery and examples]

The **plotly.graph_objects** module provides an automatically-generated hierarchy of classes called "graph objects" that may be used to represent figures, with a top-level class plotly.graph_objects.Figure

[When to tuse graph object]

recommandation

But adding more option, or creating specific graph, this one might be usefull as well

data types

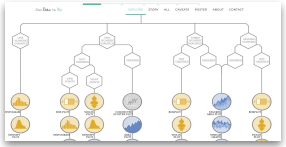


chart type

data relationship

GAPMINDER

jupyter plotly Dash

Source : [From data to viz - Yan Holtz], [Gapminder data]

Let's explore again !

plotly

Plotly Express (included as the plotly.express module) is a high-level data visualization API that produces fully-populated graph object figures in single function-calls.

See the gallery and examples

recommandation

```
fig = px.scatter(
        df,
        x="sepal_width", y="sepal_length",
        color="species")
fig.show()
```

Plotly Express currently includes the following functions:

- **Basics**: scatter, line, area, bar, funnel, timeline
- **Part-of-Whole**: pie, sunburst, treemap, funnel_area
- **1D Distributions**: histogram, box, violin, strip
- **2D Distributions**: density_heatmap, density_contour
- **Matrix Input**: imshow
- **3-Dimensional**: scatter_3d, line_3d
- **Multidimensional**: scatter_matrix, parallel_coordin
- **Tile Maps**: scatter_mapbox, line_mapbox, choropleth_
- **Outline Maps**: scatter_geo, line_geo, choropleth
- **Polar Charts**: scatter_polar, line_polar, bar_polar
- **Ternary Charts**: scatter_ternary, line_ternary

data types

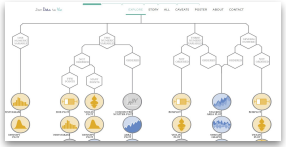


chart type

data relationship

GAPMINDER

jupyter plotly Dash

Source : From data to viz - Yan Holtz, Gapminder data

Let's explore again !

Data exploration

Step 1 : Context and data overview

Step 2 : Exploring each variable

Step 3 : Exploring variables relationships

1h

Dashboard making

1h

data types

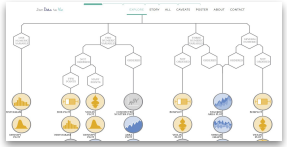


chart type

data relationship

GAPMINDER

jupyter plotly Dash

Source : From data to viz - Yan Holtz, Gapminder data

Process

visual

exploration

summary

Tools

Process

visual exploration summary

Tools

questions

data information

Context and data overview

list variables & types

Exploring each variable

overview

get to know a dataset

Exploring variables relationships

questions relationships : chart types

create variables

explore new charts

Reflect

answers ?

conclusion ?

next steps ?

Process

visual

exploration

summary

Tools

questions

data information

Context and data overview list variables & types

Exploring each variable overview get to know a dataset

Exploring variables relationships questions relationships : chart types

create variables explore new charts

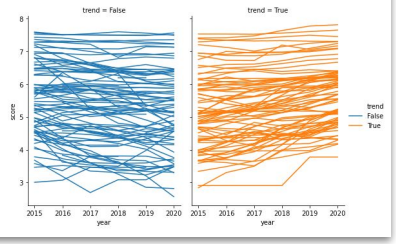
Reflect answers ? conclusion ? next steps ?

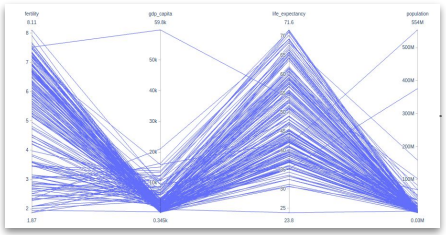
Out[87]:

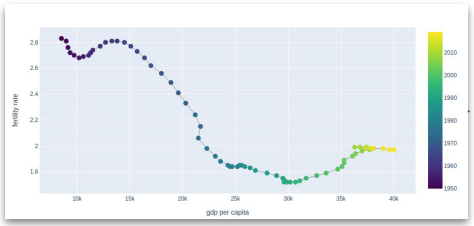
	country	trend	year	score
0	Afghanistan	False	2015	3.575
1	Albania	False	2015	4.959
2	Algeria	False	2015	5.605
3	Angola	False	2015	4.033
4	Argentina	False	2015	6.574

rée [115]:
```
# plotting the
sns.relplot(
    data=df_melted,
    x="year", y="score",
    hue="trend", col="trend",
    units='country', estimator=None,
    kind="line",
    height=5, aspect=.75, facet_kws=dict(sharex=False),
)
```

Out[115]: <seaborn.axisgrid.FacetGrid at 0x7f75d09675e0>







questions

data information

Context and data overview

list variables & types

Exploring each variable

overview

get to know dataset

Exploring variables relationships

questions relationships : chart types

create variables

explore new charts

Reflect

answers ?

conclusion ?

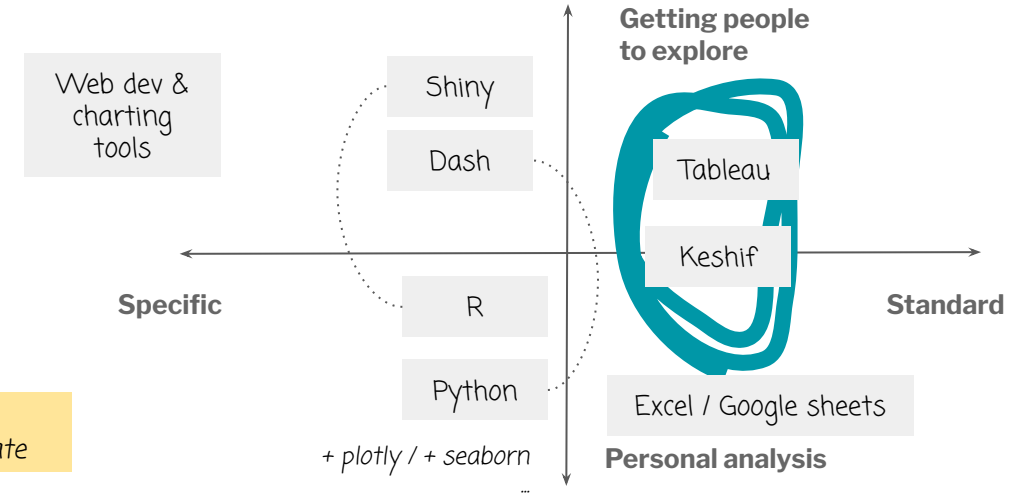
next steps ?

Process

visual exploration summary

Tools

/!\ Rough overview nor exaustive nor very accurate



I. Chart of the day

II. Let's explore again (plotly)

III. Exploration for others : dashboard

#1

Let's explore again !

Preliminary

Data exploration

Dashboard making

Step 1 : Dash discovery

Step 2 : Setting up a simple real example from our data

Step 3 : Getting to know callbacks

1h

data types

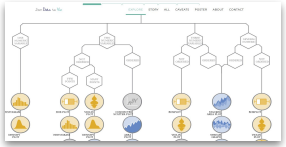


chart type

data relationship

GAPMINDER

Source : From data to viz - Yan Holtz, Gapminder data

Let's explore again !

#1

```
import XXX

external_stylesheets =
['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__,
external_stylesheets=external_stylesheets)

app.layout = html.Div([
    html.H6("Change the value in the text box to see callbacks in
action!"),
    html.Div(["Input: ",
              dcc.Input(id='my-input', value='initial value', type='text')]),
    html.Br(),
    html.Div(id='my-output'),
])

@app.callback(
    Output(component_id='my-output',
component_property='children'),
    [Input(component_id='my-input',
component_property='value')]
)
def update_output_div(input_value):
    return 'Output: {}'.format(input_value)

if __name__ == '__main__':
    app.run_server(debug=True)
```

Layout

Callback

data types

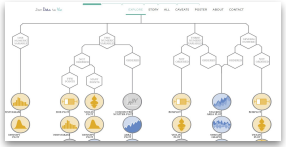


chart type

data relationship

GAPMINDER

jupyter plotly Dash

Source : From data to viz - Yan Holtz, Gapminder data

#1

Let's explore again !

```
import XXX

external_stylesheets =
['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__,
external_stylesheets=external_stylesheets)

app.layout = html.Div([
    html.H6("Change the value in the text box to see callbacks in
action!"),
    html.Div(["Input: ",
              dcc.Input(id='my-input', value='initial value', type='text')]),
    html.Br(),
    html.Div(id='my-output'),
])
```

```
@app.callback(
    Output(component_id='my-output',
component_property='children'),
    [Input(component_id='my-input',
component_property='value')]
)
def update_output_div(input_value):
    return 'Output: {}'.format(input_value)

if __name__ == '__main__':
    app.run_server(debug=True)
```

Layout

dash_html_components

html.H1(children='Hello Dash')

<h1>Hello Dash</h1> HTML element

Callback

data types

chart type

data relationship

GAPMINDER

jupyter plotly Dash

Source : From data to viz - Yan Holtz, Gapminder data

#1

Let's explore again !

```
import XXX

external_stylesheets =
['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__,
external_stylesheets=external_stylesheets)

app.layout = html.Div([
    html.H6("Change the value in the text box to see callbacks in
action!"),
    html.Div(["Input: ",
              dcc.Input(id='my-input', value='initial value', type='text')]),
    html.Br(),
    html.Div(id='my-output'),
])
```

```
@app.callback(
    Output(component_id='my-output',
component_property='children'),
    [Input(component_id='my-input',
component_property='value')]
)
def update_output_div(input_value):
    return 'Output: {}'.format(input_value)

if __name__ == '__main__':
    app.run_server(debug=True)
```

Layout

dash_html_components

dash_core_components

dropdowns, graphs, markdown blocks ...
See the gallery

Callback

data types

chart type

data relationship

GAPMINDER

jupyter plotly Dash

Source : From data to viz - Yan Holtz, Gapminder data

#1

Let's explore again !

```
import XXX

external_stylesheets =
['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__,
external_stylesheets=external_stylesheets)

app.layout = html.Div([
    html.H6("Change the value in the text box to see callbacks in
action!"),
    html.Div(["Input: ",
              dcc.Input(id='my-input', value='initial value', type='text')]),
    html.Br(),
    html.Div(id='my-output'),
])
```

```
@app.callback(
    Output(component_id='my-output', ...)),
    [ Input(component_id='my-input', ...) ]
)
def update_output_div(input_value):
    return 'Output: {}'.format(input_value)

if __name__ == '__main__':
    app.run_server(debug=True)
```

Layout

dash_html_components

dash_core_components

Callback

dcc.Input(id='my-input' ...

html.Div(id='my-output')

component_id='my-inpput'

Input

def update function

Output

component_id='my-output'

data types

chart type

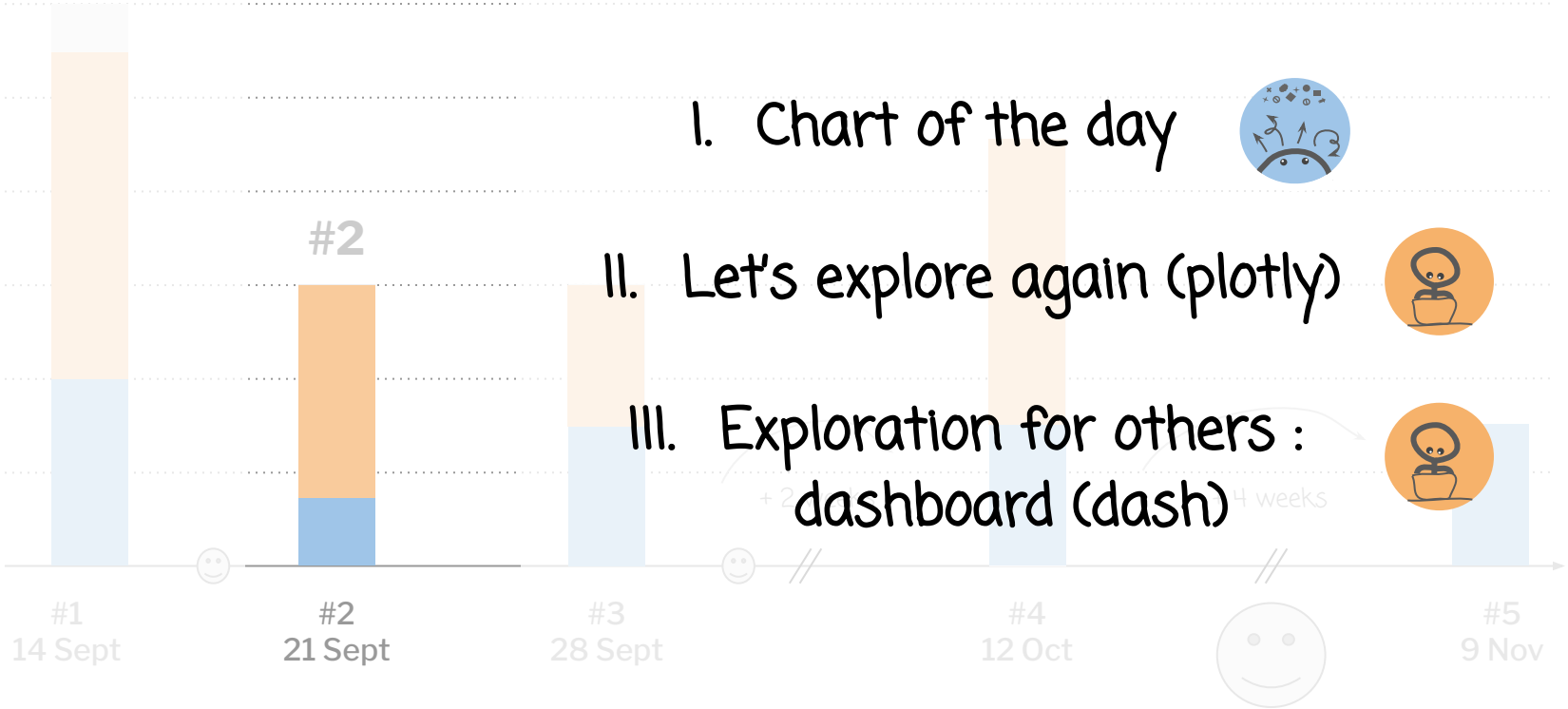
data relationship

GAPMINDER

jupyter plotly Dash

Source : From data to viz - Yan Holtz, Gapminder data

DAY SUMMARY



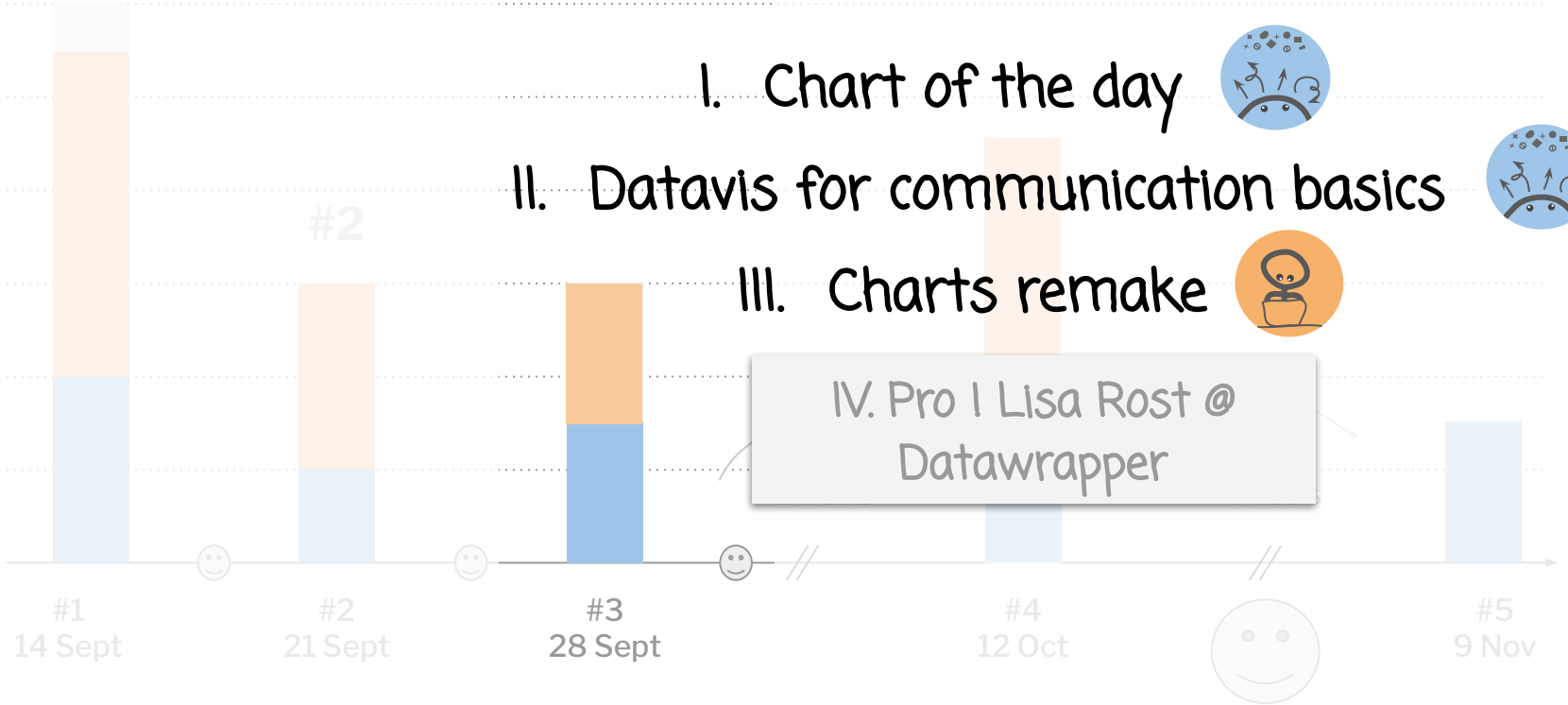
I. Chart of the day

II. Let's explore again (plotly)

III. Exploration for others : dashboard (dash)

WHAT'S NEXT

Show findings using visualisation



I. Chart of the day

II. Datavis for communication basics

III. Charts remake

IV. Pro I Lisa Rost @ Datawrapper