Saturdays.AI
Barcelona

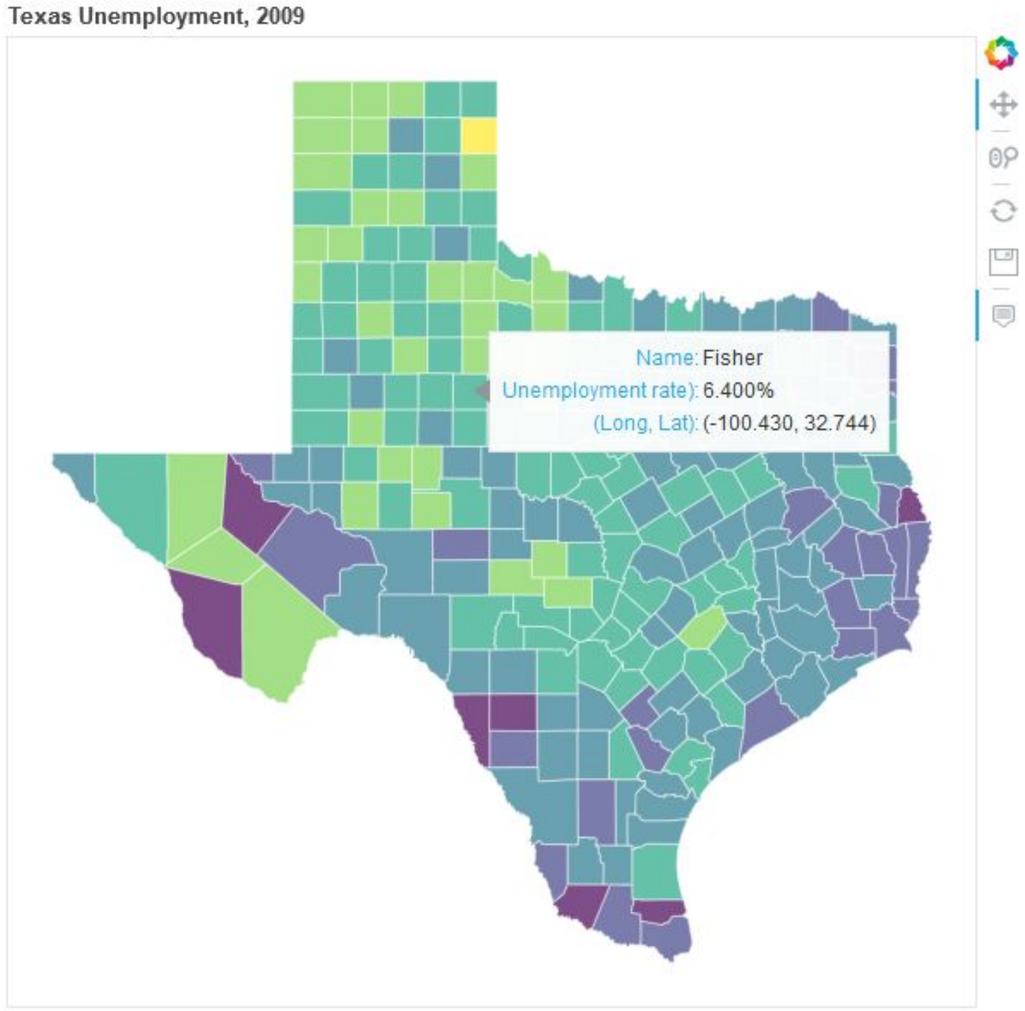
Data Visualization

by Albert Sanchez Lafuente

Get ready for the future AI!

Index

- Pandas
- Pandas profiling
- Matplotlib
- Seaborn
- Bokeh
- Altair
- Maps
- D-Tale



Datasets

Saturdays.AI Barcelona

	Mes	data science	machine learning	deep learning	categorical
2	2004-01-01	12	18	4	1
3	2004-02-01	12	21	2	1
4	2004-03-01	9	21	2	1
5	2004-04-01	10	16	4	1
6	2004-05-01	7	14	3	1
7	2004-06-01	9	17	3	1
8	2004-07-01	9	16	3	1
9	2004-08-01	7	14	3	1
10	2004-09-01	10	17	4	1

temporal.csv

	País	data science	machine learning	deep learning
2	Santa Elena	100.0	100.0	52.0
3	India	99.0	77.0	25.0
4	Ruanda			
5	Lesoto			
6	Singapur	91.0	79.0	52.0
7	Zimbabue			
8	Botsuana			
9	Nepal	77.0	49.0	
10	Nigeria	72.0	25.0	8.0
11	Etiopía			

mapa.csv

https://github.com/albertsl/datasets/tree/master/popularidad

Saturdays.AI
Barcelona

Pandas

Pandas

```
import pandas as pd
df = pd.read_csv('datos.csv')
df.head(10)
```

	Mes	data science	machine learning	deep learning	categorical
0	2004-01-01	12	18	4	1
1	2004-02-01	12	21	2	1
2	2004-03-01	9	21	2	1
3	2004-04-01	10	16	4	1
4	2004-05-01	7	14	3	1
5	2004-06-01	9	17	3	1
6	2004-07-01	9	16	3	1
7	2004-08-01	7	14	3	1
8	2004-09-01	10	17	4	1
9	2004-10-01	8	17	4	1

Pandas

df.describe()

	data science	machine learning	deep learning	categorical
count	194.000000	194.000000	194.000000	194.000000
mean	20.953608	27.396907	24.231959	0.257732
std	23.951006	28.091490	34.476887	0.438517
min	4.000000	7.000000	1.000000	0.000000
25%	6.000000	9.000000	2.000000	0.000000
50%	8.000000	13.000000	3.000000	0.000000
75%	26.750000	31.500000	34.000000	1.000000
max	100.000000	100.000000	100.000000	1.000000

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 194 entries, 0 to 193
Data columns (total 5 columns):
Mes                 194 non-null object
data science        194 non-null int64
machine learning    194 non-null int64
deep learning       194 non-null int64
categorical         194 non-null int64
dtypes: int64(4), object(1)
memory usage: 7.7+ KB
```

Pandas please show me all the data

When there is a lot of info, pandas doesn't show everything (which I don't like)

31	c_32	c_33	c_34	c_35	...	c_460	c_461	c_462	c_463	c
1	1	1	1	1	...	1	1	1	1	
1	1	1	1	1	...	1	1	1	1	
1	1	1	1	1	...	1	1	1	1	
1	1	1	1	1	...	1	1	1	1	
1	1	1	1	1	...	1	1	1	1	
...	...	...	...	...	...	...	...	...	...	
1	1	1	1	1	...	1	1	1	1	
1	1	1	1	1	...	1	1	1	1	
1	1	1	1	1	...	1	1	1	1	
1	1	1	1	1	...	1	1	1	1	
1	1	1	1	1	...	1	1	1	1	

Pandas please show me all the data

Let’s fix it!

```
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

Now we can see the whole dataset (careful with big datasets, they may take too much time)

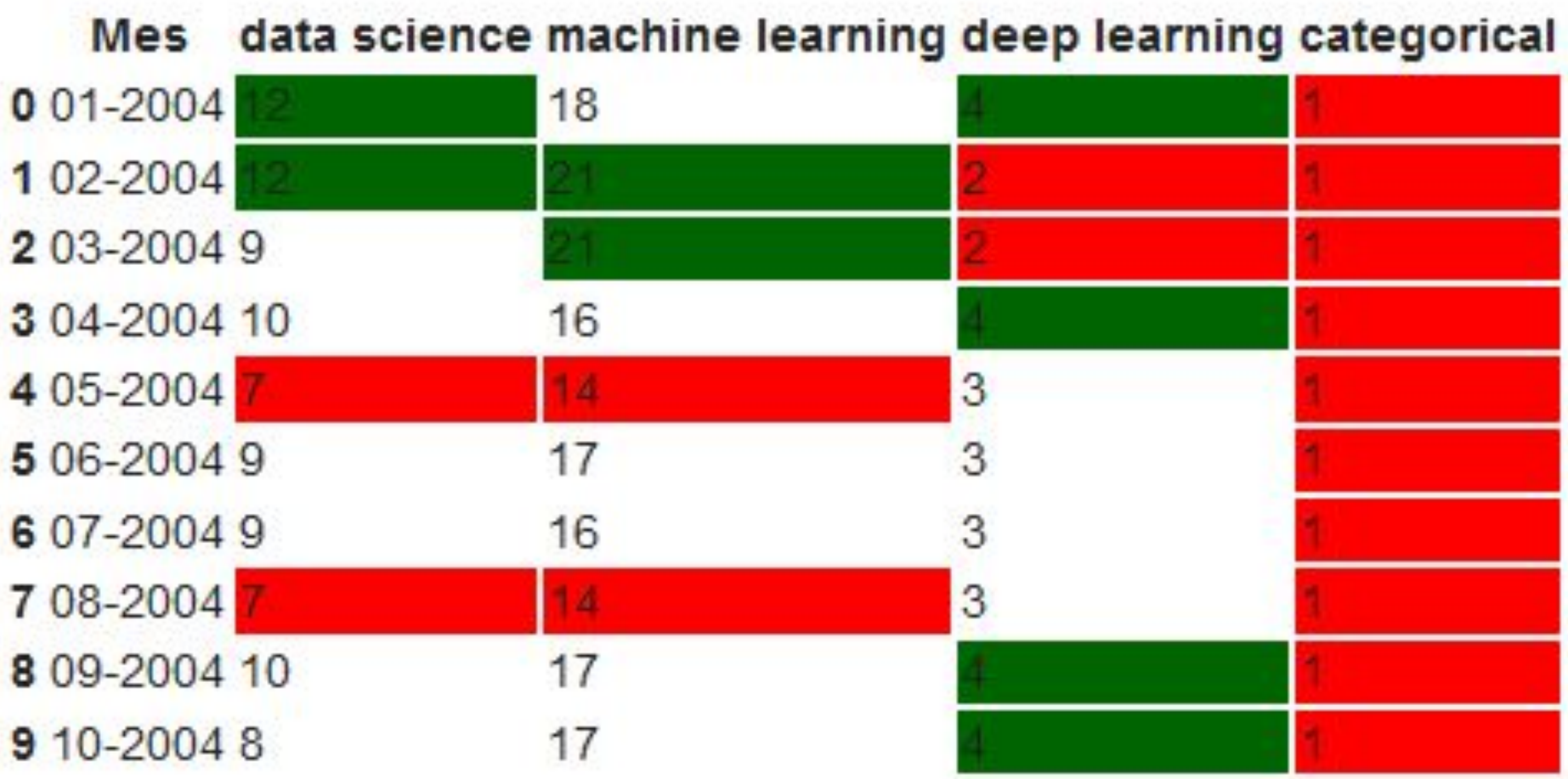
Improving Pandas

```
format_dict = {'data science':'${0:,.2f}', 'Mes':'{:%m-%Y}', 'machine learning':'{:.2%}'}
df.head().style.format(format_dict)
```

	Mes	data science	machine learning	deep learning
0	01-2004	$12.00	1800.00%	4
1	02-2004	$12.00	2100.00%	2
2	03-2004	$9.00	2100.00%	2
3	04-2004	$10.00	1600.00%	4
4	05-2004	$7.00	1400.00%	3

Pandas - Highlight min and max values

```
df.head(10).style.format(format_dict).highlight_max(color='darkgreen').highlight_min(color='#ff0000')
```



	Mes	data science	machine learning	deep learning	categorical
0	01-2004	12	18	4	1
1	02-2004	12	21	2	1
2	03-2004	9	21	2	1
3	04-2004	10	16	4	1
4	05-2004	7	14	3	1
5	06-2004	9	17	3	1
6	07-2004	9	16	3	1
7	08-2004	7	14	3	1
8	09-2004	10	17	4	1
9	10-2004	8	17	4	1

Pandas - Colour Gradient

```
df.head(10).style.format(format_dict).background_gradient(subset=['data science', 'machine learning'], cmap='BuGn')
```

	Mes	data science	machine learning	deep learning	categorical
0	01-2004	12	18	4	1
1	02-2004	12	21	2	1
2	03-2004	9	21	2	1
3	04-2004	10	16	4	1
4	05-2004	7	14	3	1
5	06-2004	9	17	3	1
6	07-2004	9	16	3	1
7	08-2004	7	14	3	1
8	09-2004	10	17	4	1
9	10-2004	8	17	4	1

Pandas - Bars

```
df.head(10).style.format(format_dict).bar(color='red', subset=['data science', 'deep learning'])
```

	Mes	data science	machine learning	deep learning	categorical
0	01-2004	12	18	4	1
1	02-2004	12	21	2	1
2	03-2004	9	21	2	1
3	04-2004	10	16	4	1
4	05-2004	7	14	3	1
5	06-2004	9	17	3	1
6	07-2004	9	16	3	1
7	08-2004	7	14	3	1
8	09-2004	10	17	4	1
9	10-2004	8	17	4	1

Saturdays.AI
Barcelona

Pandas Profiling

Pandas Profiling

```
from pandas_profiling import ProfileReport
prof = ProfileReport(df)
prof.to_file('output.html')
```

It's [interactive](#)!

Overview

Dataset info

Number of variables	4
Number of observations	194
Total Missing (%)	0.0%
Total size in memory	6.2 KiB
Average record size in memory	32.7 B

Variables types

Numeric	1
Categorical	0
Boolean	0
Date	1
Text (Unique)	0
Rejected	2
Unsupported	0

Warnings

`machine learning` is highly correlated with `data science` (ρ = 0.98573) Rejected
`deep learning` is highly correlated with `machine learning` (ρ = 0.9877) Rejected

Variables

Mes
Date

Distinct count	194	Minimum	2004-01-01 00:00:00
Unique (%)	100.0%	Maximum	2020-02-01 00:00:00
Missing (%)	0.0%		
Missing (n)	0		
Infinite (%)	0.0%		
Infinite (n)	0		

Toggle details

data science
Numeric

Distinct count	54	Mean	20.954
Unique (%)	27.8%	Minimum	4
Missing (%)	0.0%	Maximum	100
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

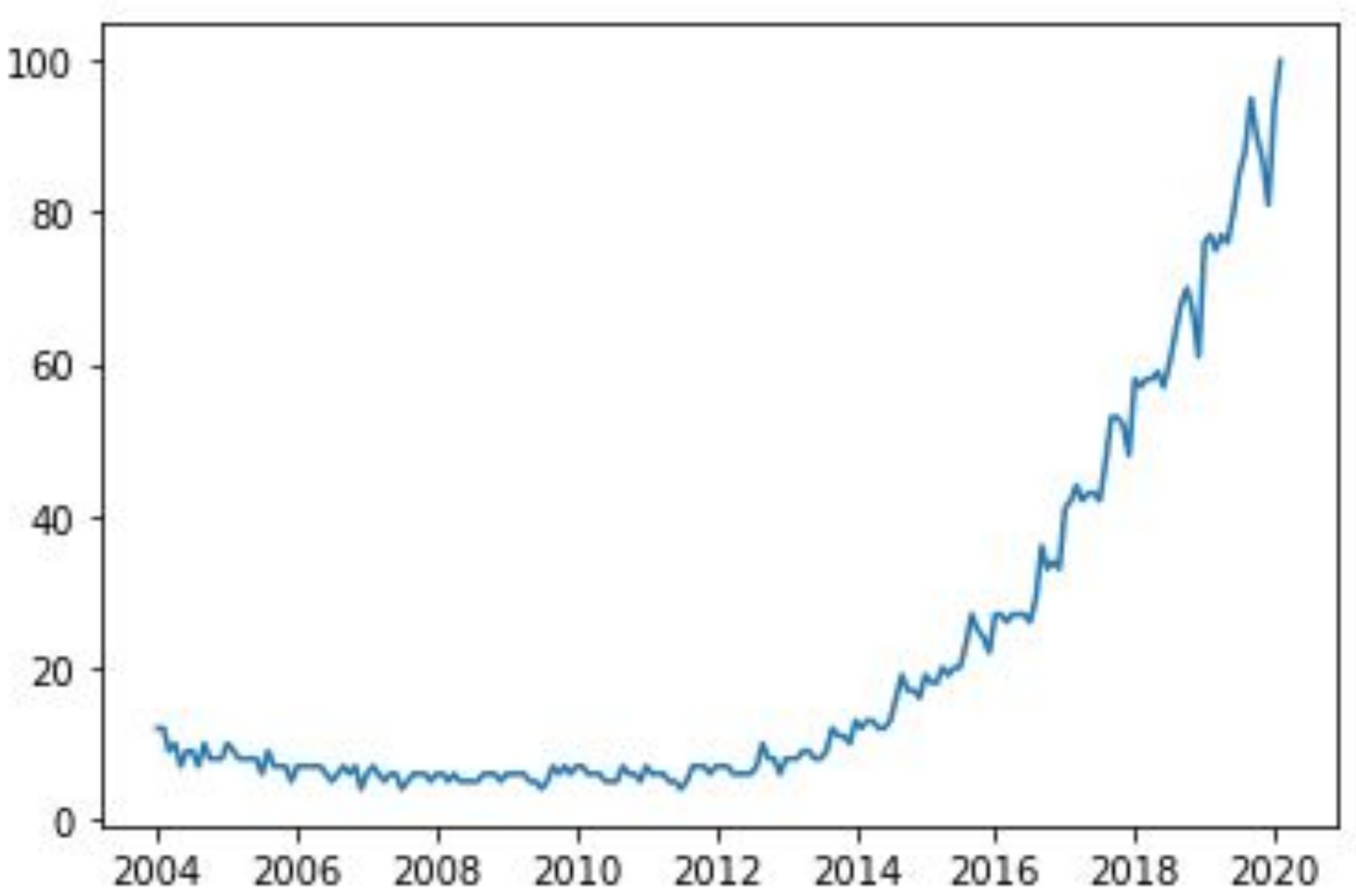
Toggle details

Saturdays.AI
Barcelona

Matplotlib

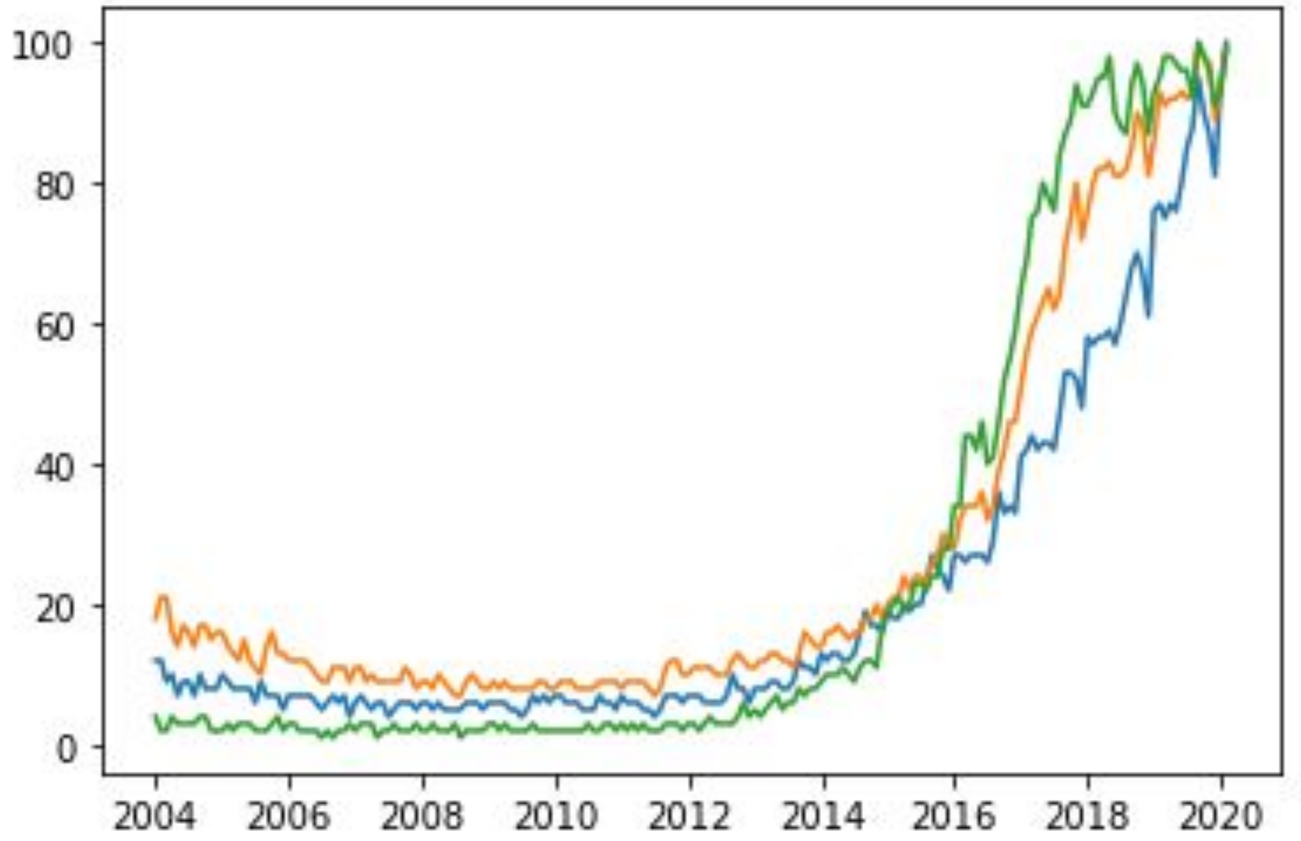
Matplotlib

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.plot(df['Mes'], df['data science'], label='data science')
```



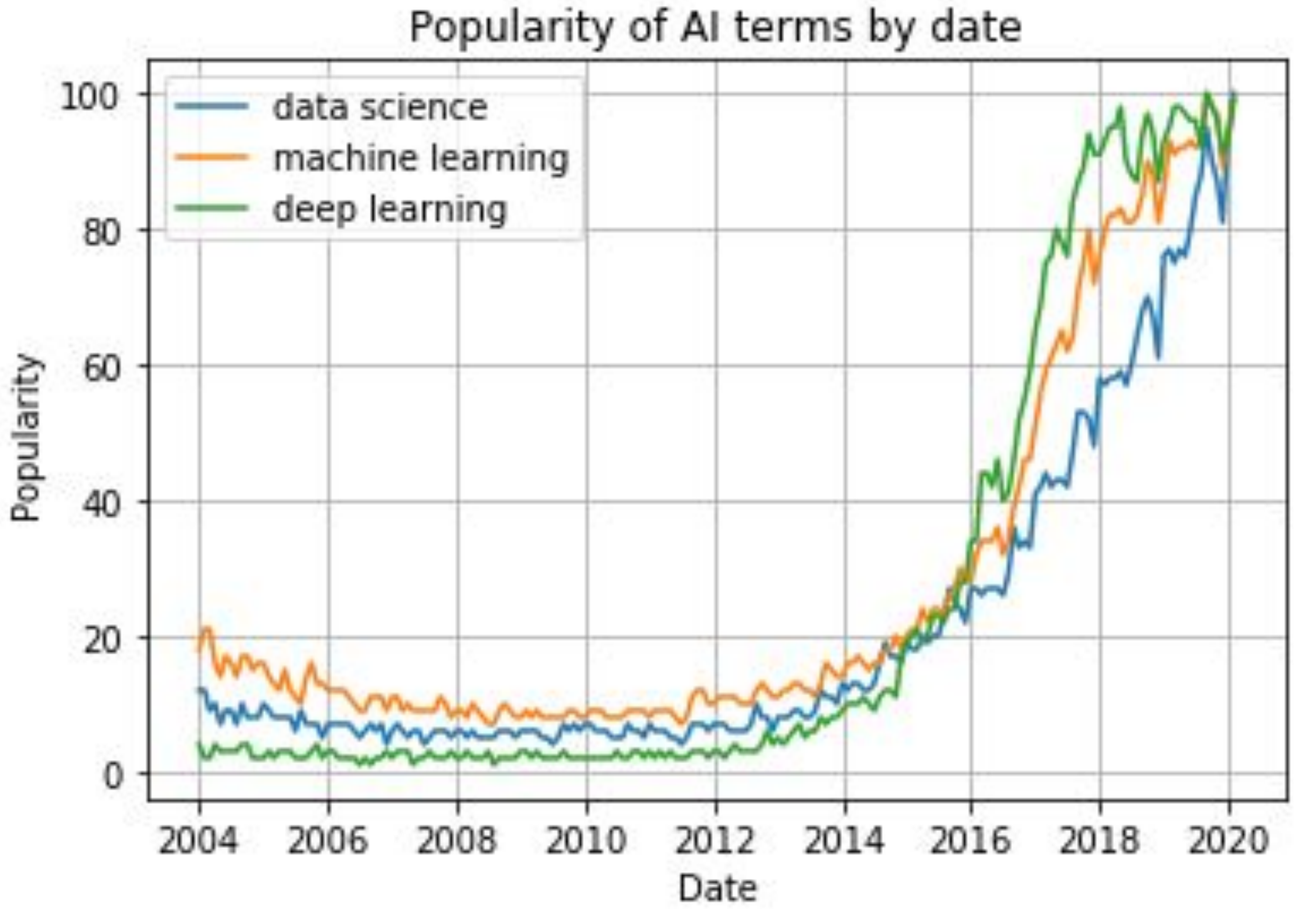
Matplotlib

```
plt.plot(df['Mes'], df['data science'], label='data science')
plt.plot(df['Mes'], df['machine learning'], label='machine learning')
plt.plot(df['Mes'], df['deep learning'], label='deep learning')
```



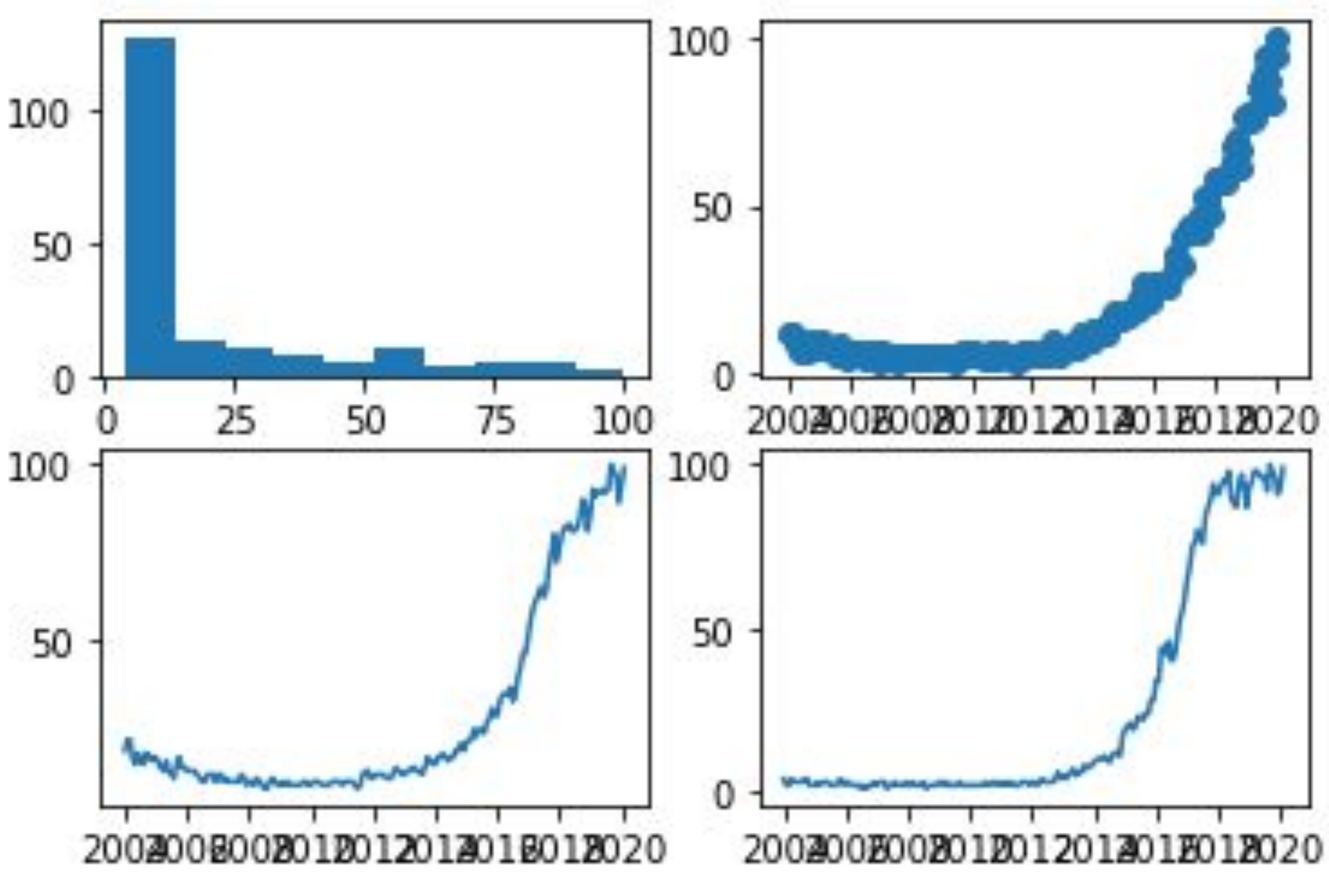
Matplotlib

```
plt.plot(df['Mes'], df['data science'], label='data science')
plt.plot(df['Mes'], df['machine learning'], label='machine learning')
plt.plot(df['Mes'], df['deep learning'], label='deep learning')
plt.xlabel('Date')
plt.ylabel('Popularity')
plt.title('Popularity of AI terms by date')
plt.grid(True)
plt.legend()
```



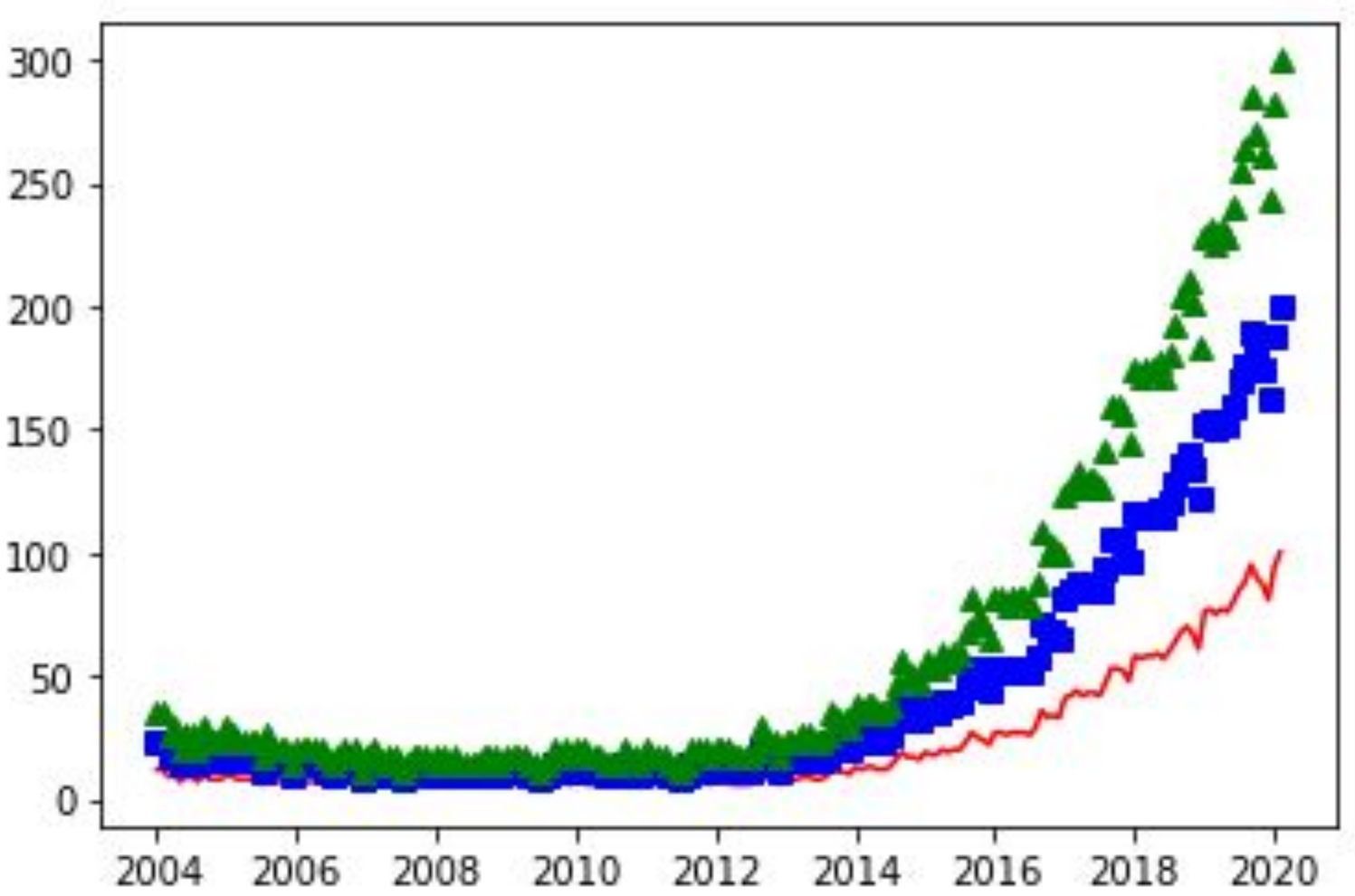
Matplotlib - Multiple plots

```
fig, axes = plt.subplots(2,2)
axes[0, 0].hist(df['data science'])
axes[0, 1].scatter(df['Mes'], df['data science'])
axes[1, 0].plot(df['Mes'], df['machine learning'])
axes[1, 1].plot(df['Mes'], df['deep learning'])
```



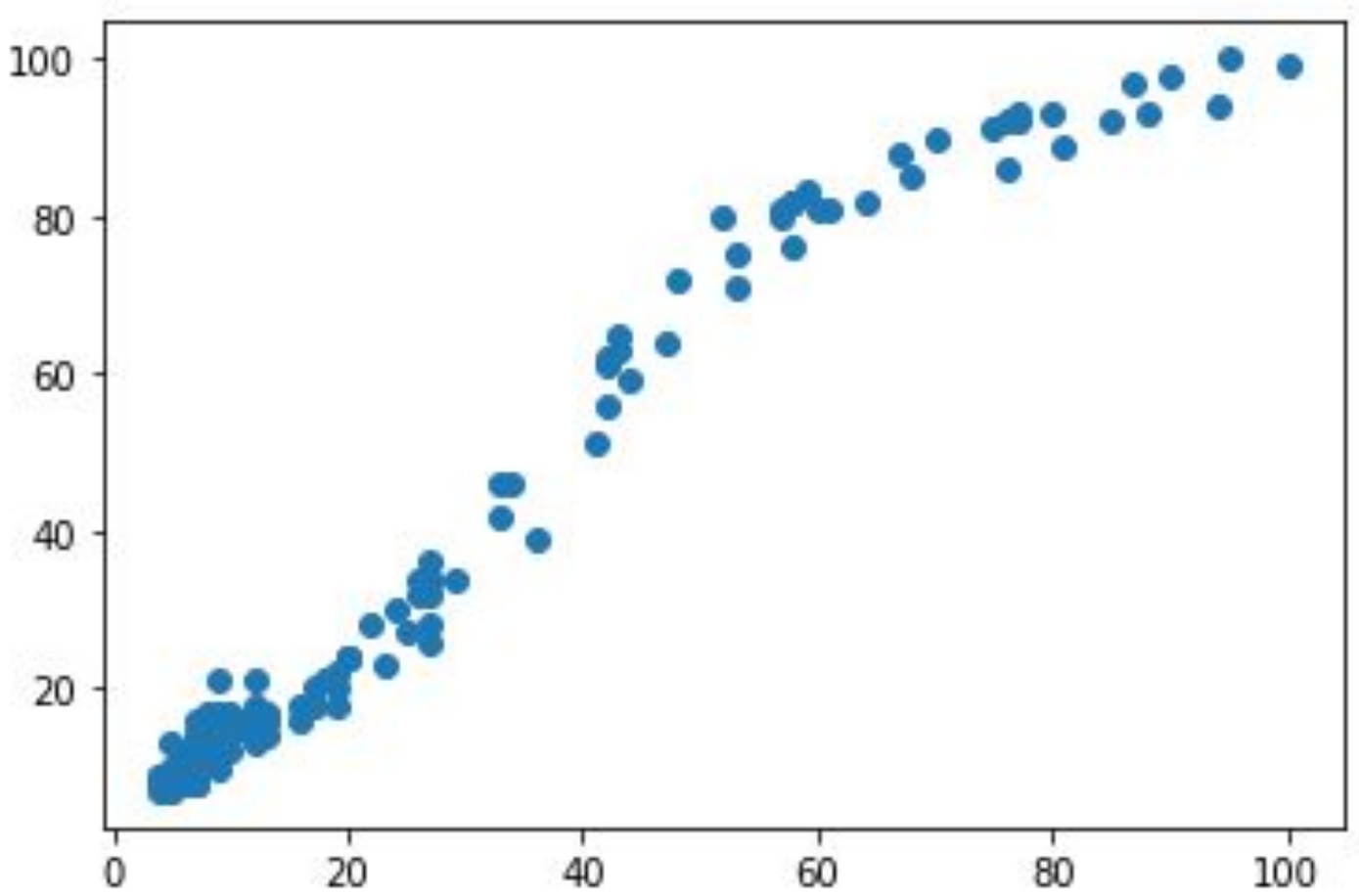
Matplotlib

```
plt.plot(df['Mes'], df['data science'], 'r-')
plt.plot(df['Mes'], df['data science']*2, 'bs')
plt.plot(df['Mes'], df['data science']*3, 'g^')
```



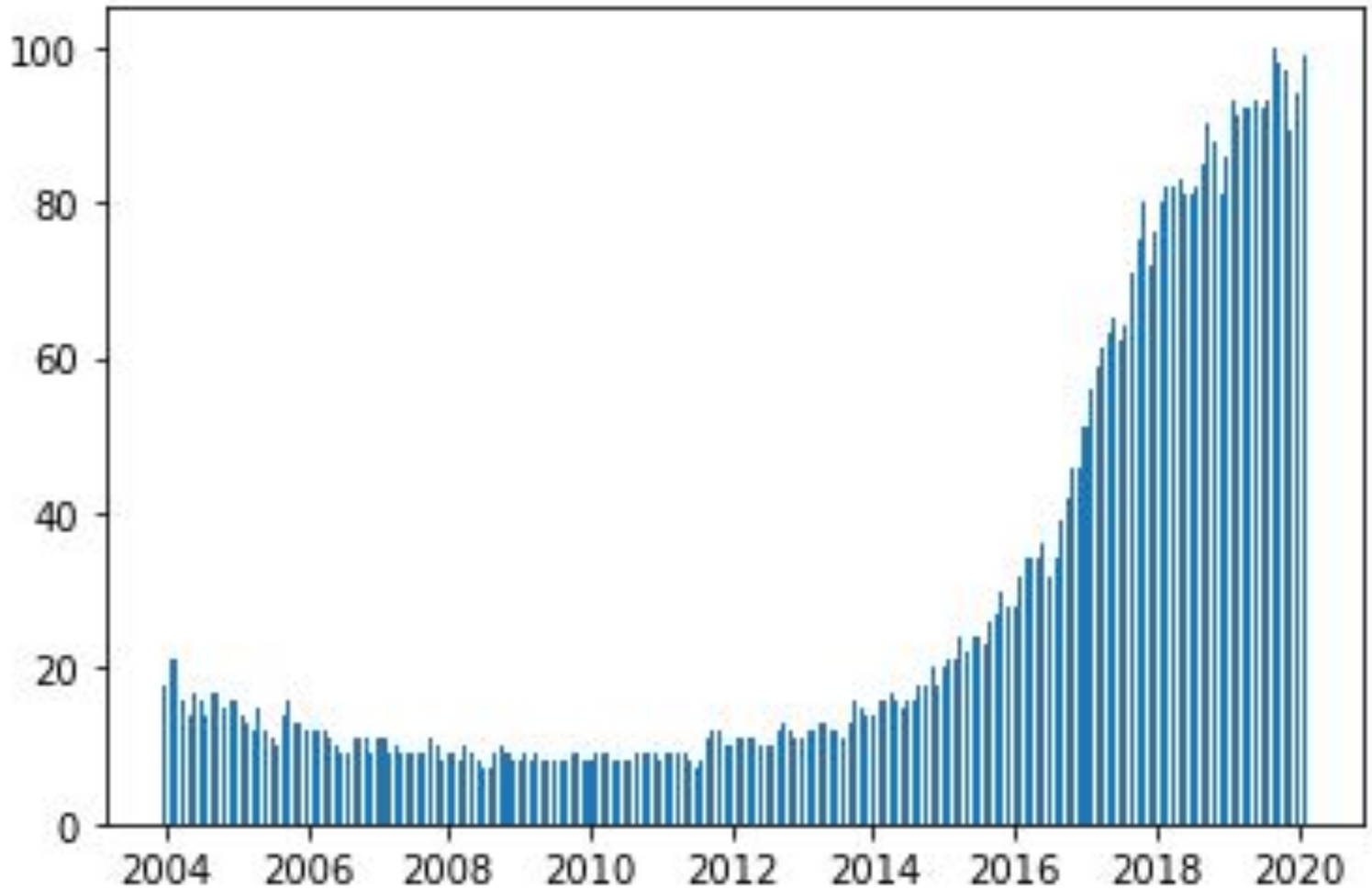
Matplotlib

```
plt.scatter(df['data science'], df['machine learning'])
```



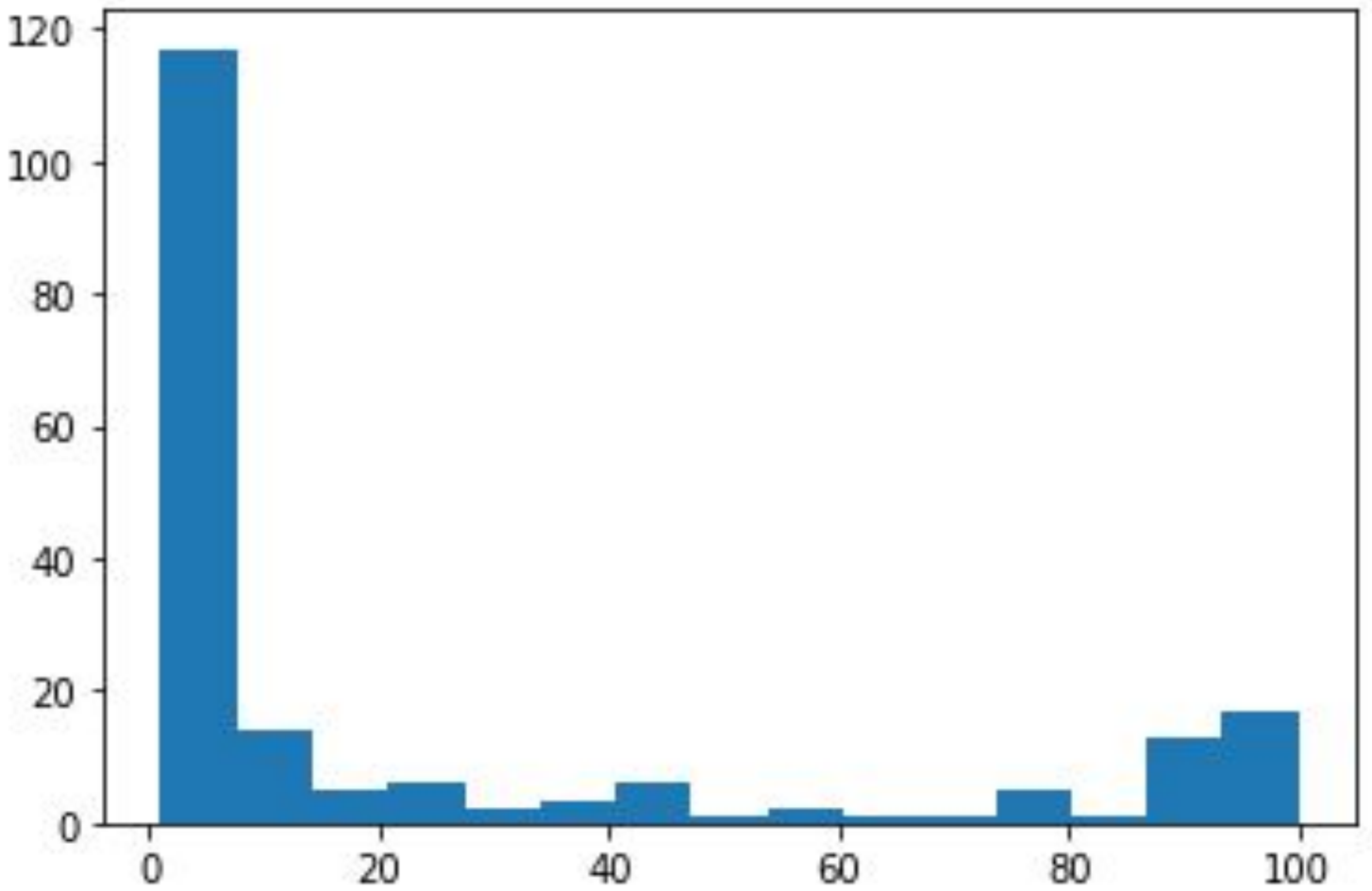
Matplotlib

```
plt.bar(df['Mes'], df['machine learning'], width=20)
```



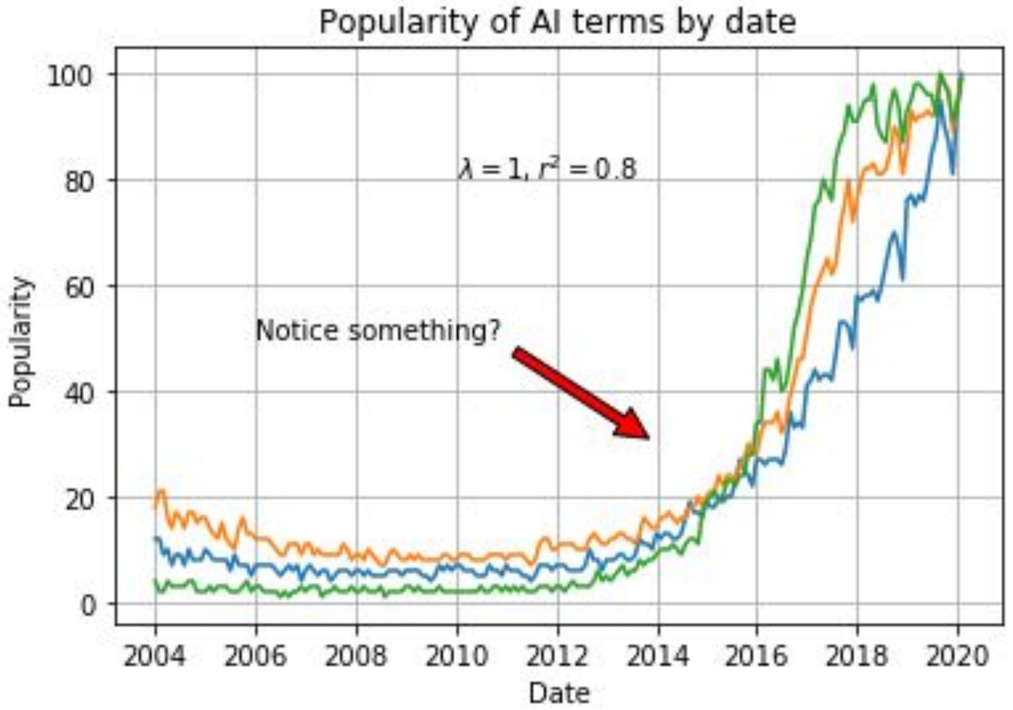
Matplotlib

```
plt.hist(df['deep learning'], bins=15)
```



Matplotlib - Text and Markers

```
plt.plot(df['Mes'], df['data science'], label='data science')
plt.plot(df['Mes'], df['machine learning'], label='machine learning')
plt.plot(df['Mes'], df['deep learning'], label='deep learning')
plt.xlabel('Date')
plt.ylabel('Popularity')
plt.title('Popularity of AI terms by date')
plt.grid(True)
plt.text(x='2010-01-01', y=80, s=r'$\lambda=1, r^2=0.8$') #Coordinates use the same units as the axis
plt.annotate('Notice something?', xy=('2014-01-01', 30), xytext=('2006-01-01', 50), arrowprops={'facecolor':'red', 'shrink':0.05})
```

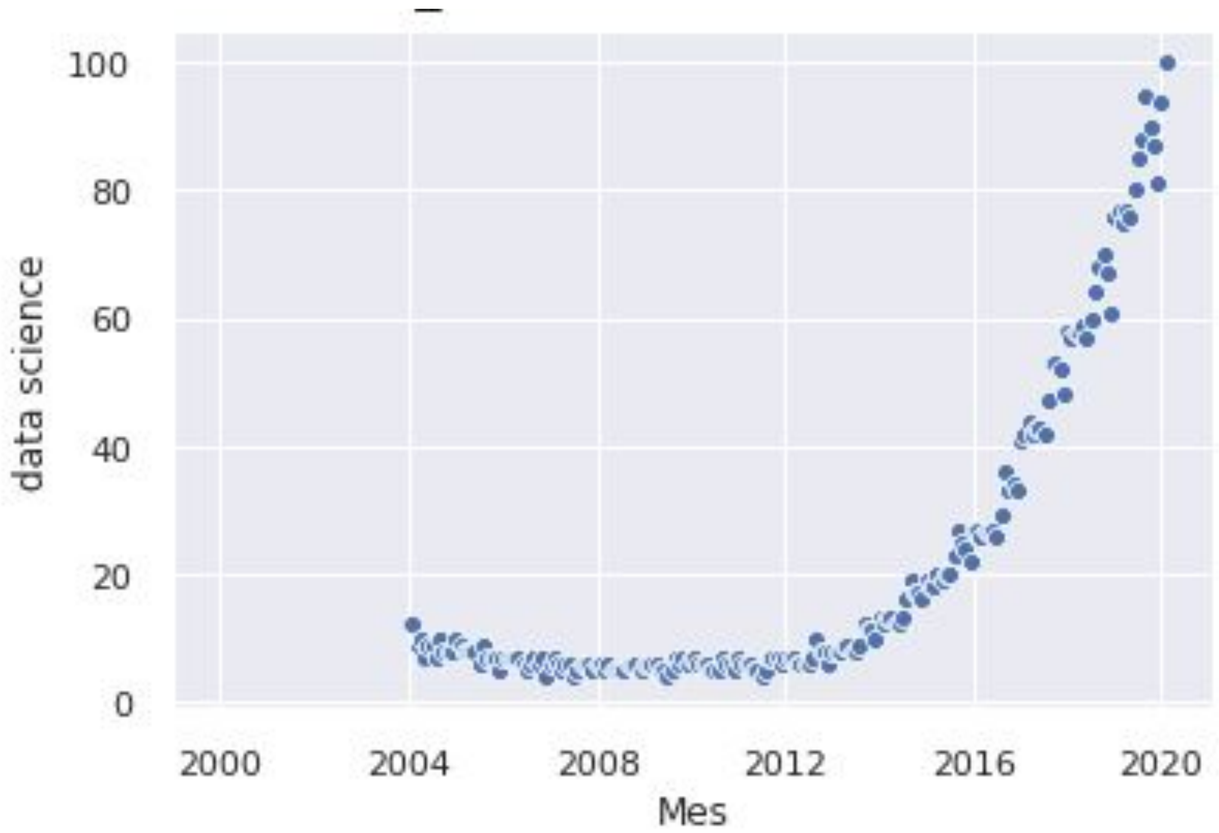


Saturdays.AI
Barcelona

Seaborn

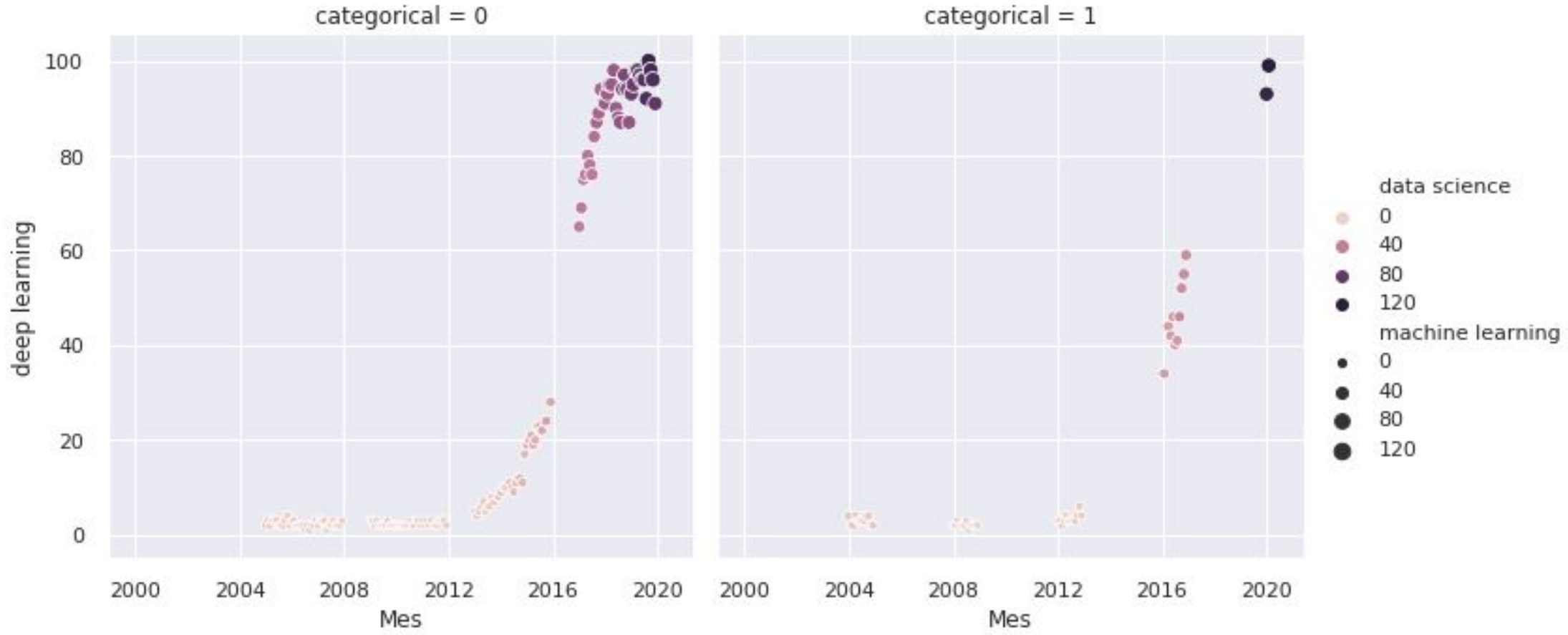
Seaborn

```
import seaborn as sns
sns.set()
sns.scatterplot(df['Mes'], df['data science'])
```



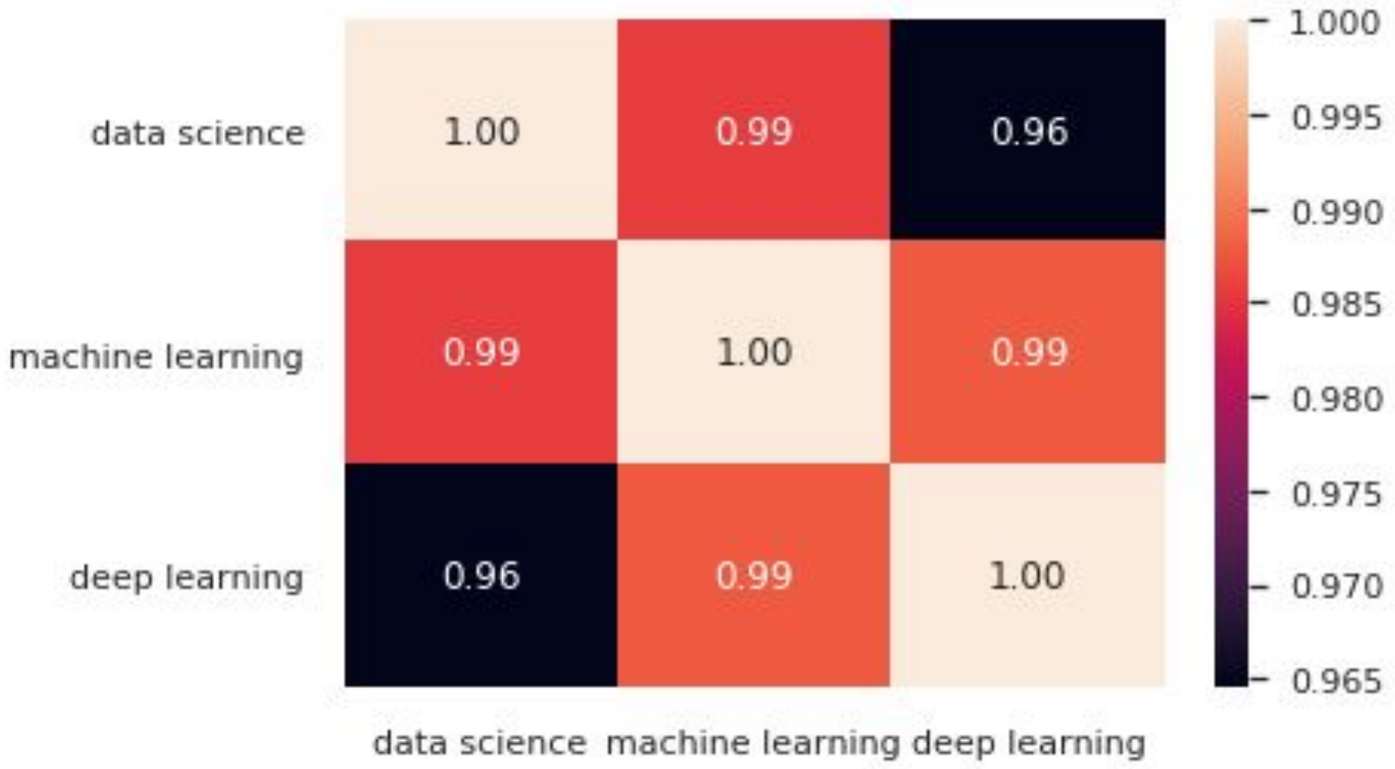
Seaborn - Plot multiple variables

```
sns.relplot(x='Mes', y='deep learning', hue='data science', size='machine learning', col='categorical', data=df)
```



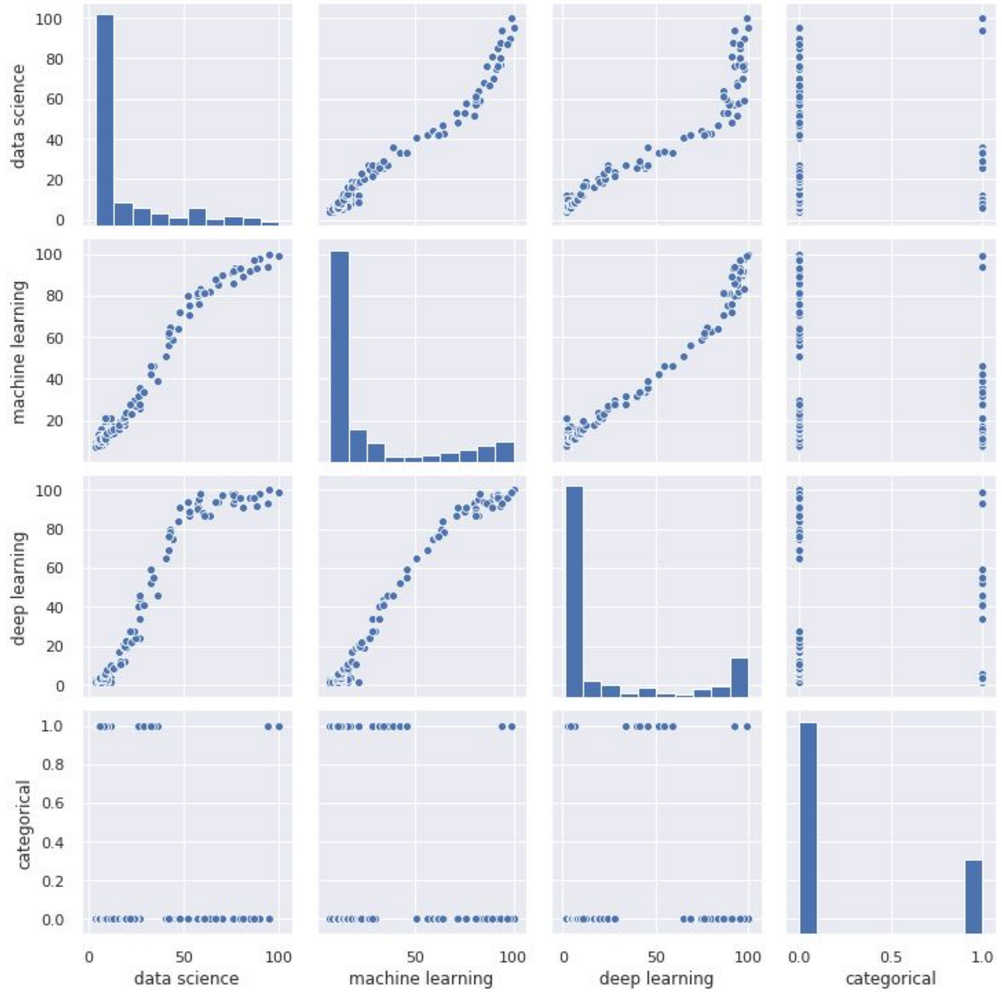
Seaborn - Heatmap of correlations

```
sns.heatmap(df.corr(), annot=True, fmt='.2f')
```

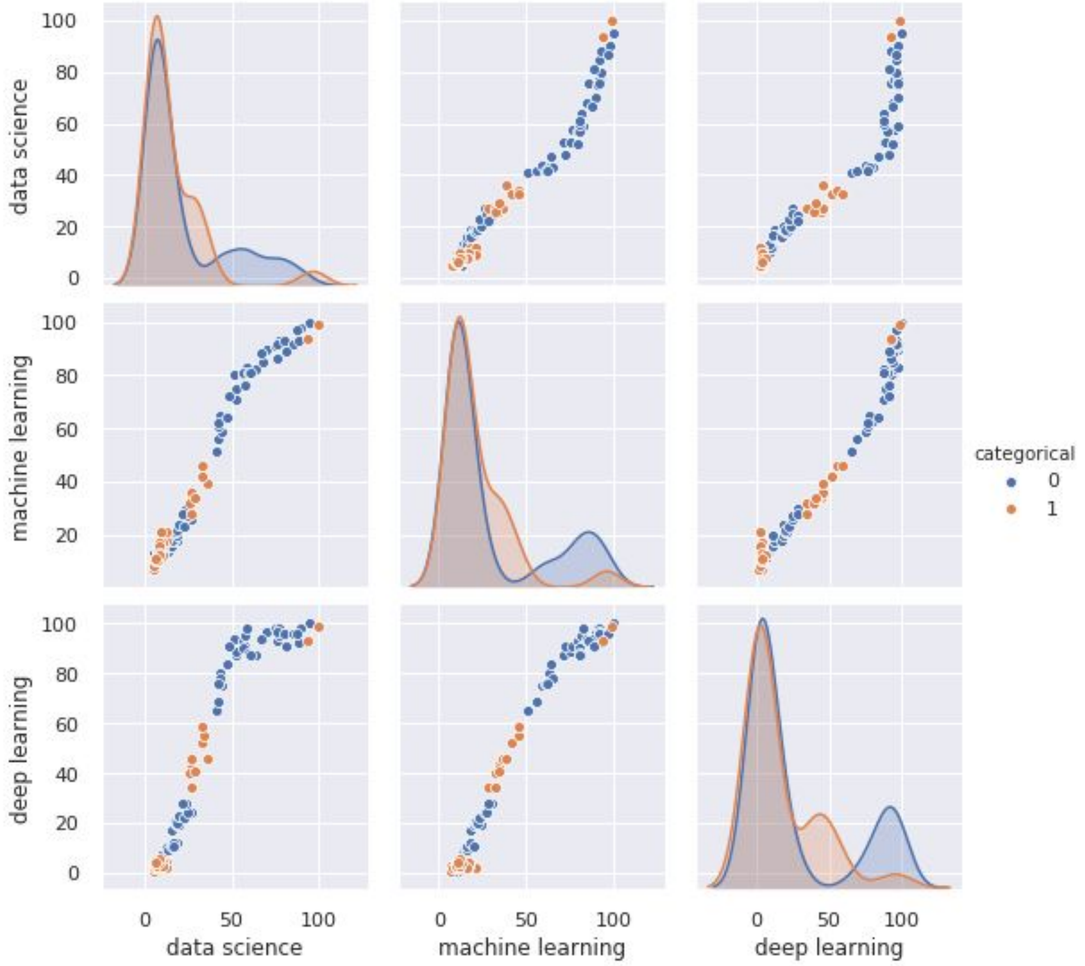


Seaborn - Pairplot

```
sns.pairplot(df)
```

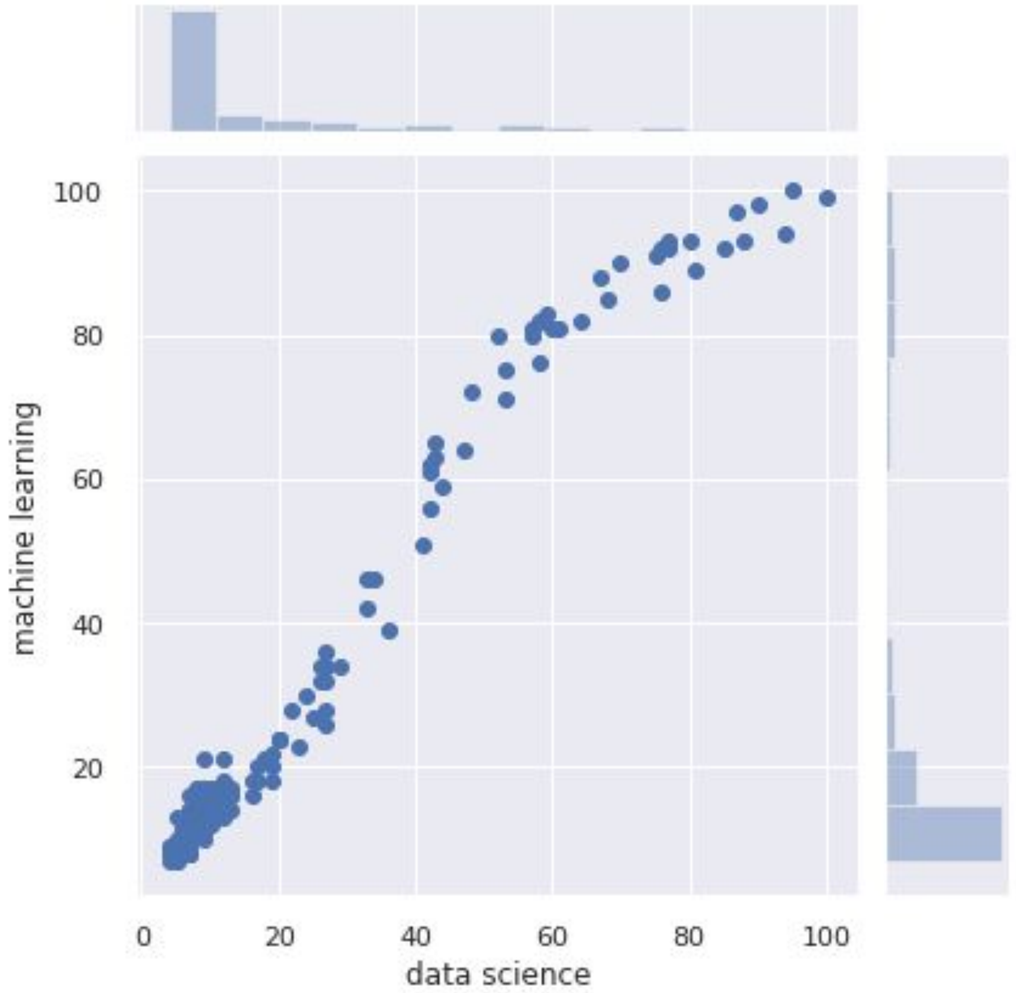



```
sns.pairplot(df, hue='categorical')
```



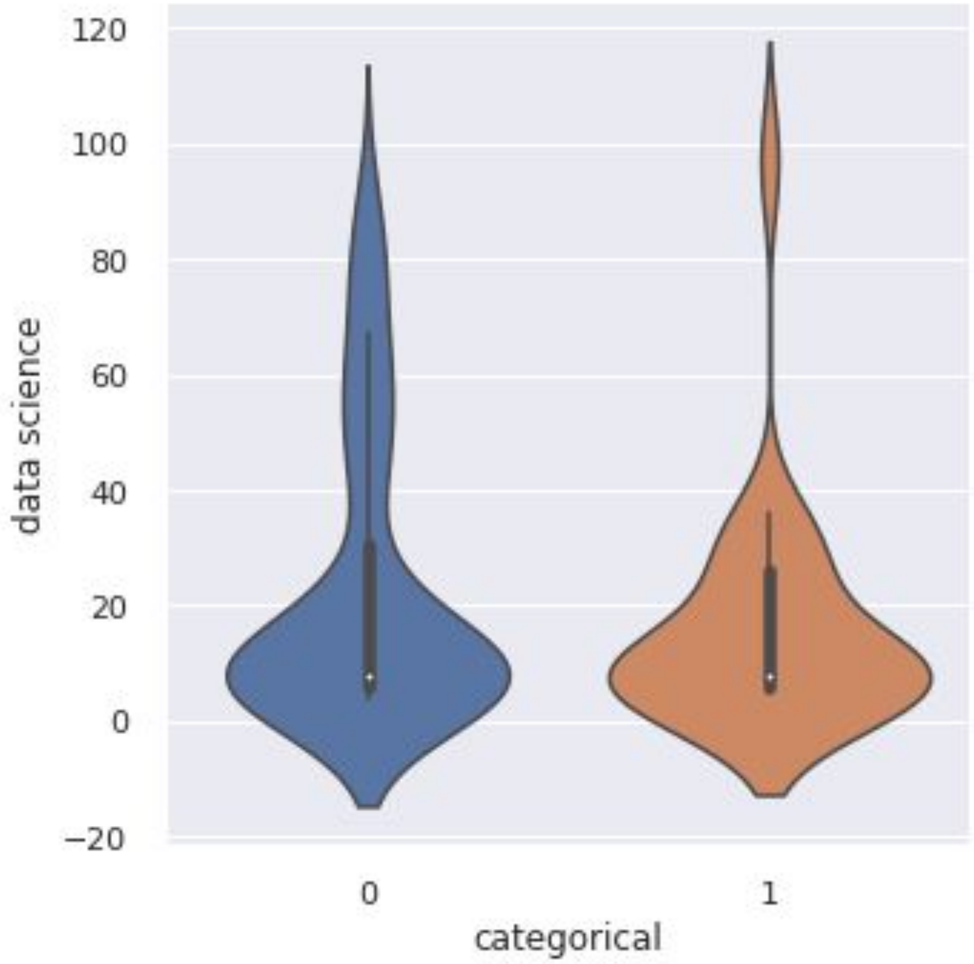
Seaborn - Jointplot

```
sns.jointplot(x='data science', y='machine learning', data=df)
```



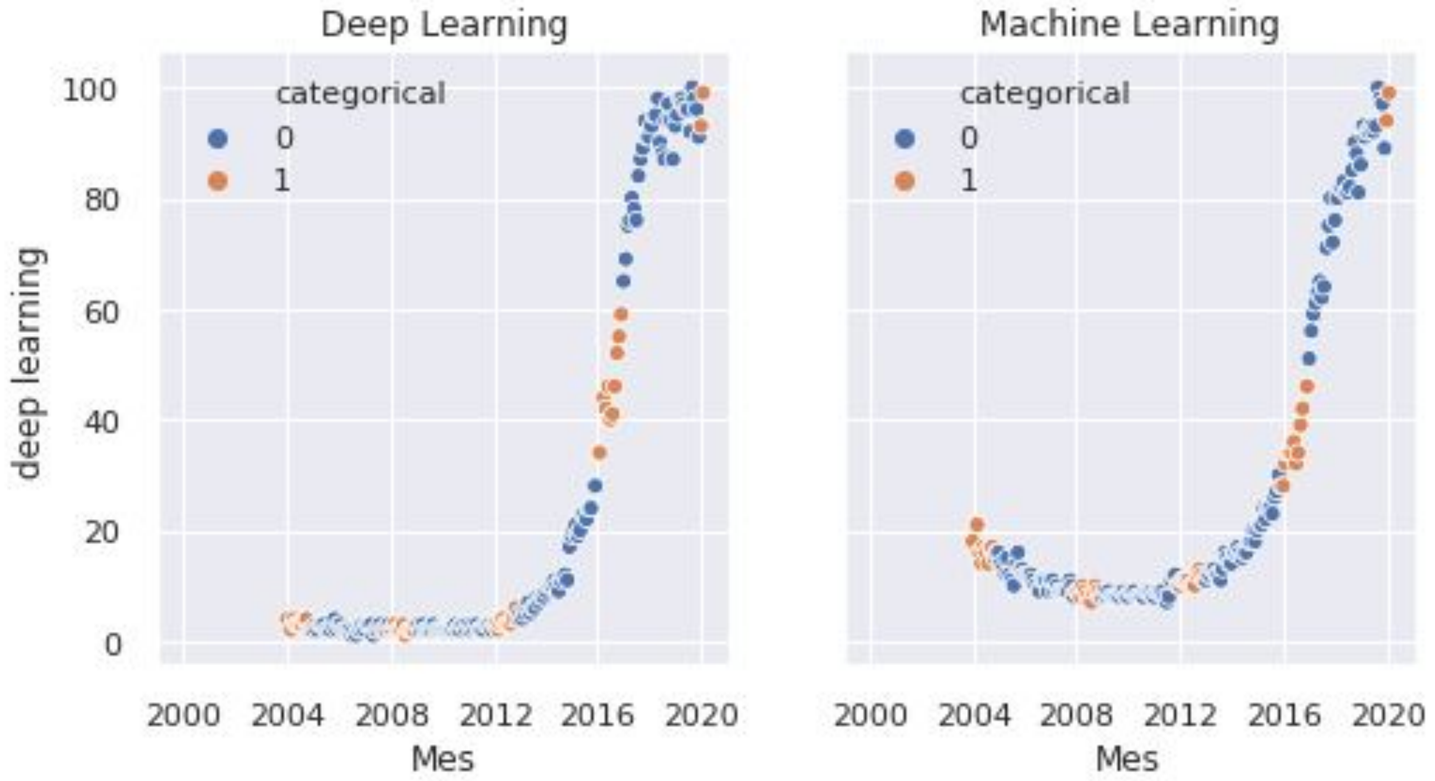
Seaborn - Violin Plot

```
sns.catplot(x='categorical', y='data science', kind='violin', data=df)
```



Seaborn - Multiple plots

```
fig, axes = plt.subplots(1, 2, sharey=True, figsize=(6, 4))
sns.scatterplot(x="Mes", y="deep learning", hue="categorical", data=df, ax=axes[0])
axes[0].set_title('Deep Learning')
sns.scatterplot(x="Mes", y="machine learning", hue="categorical", data=df, ax=axes[1])
axes[1].set_title('Machine Learning')
```



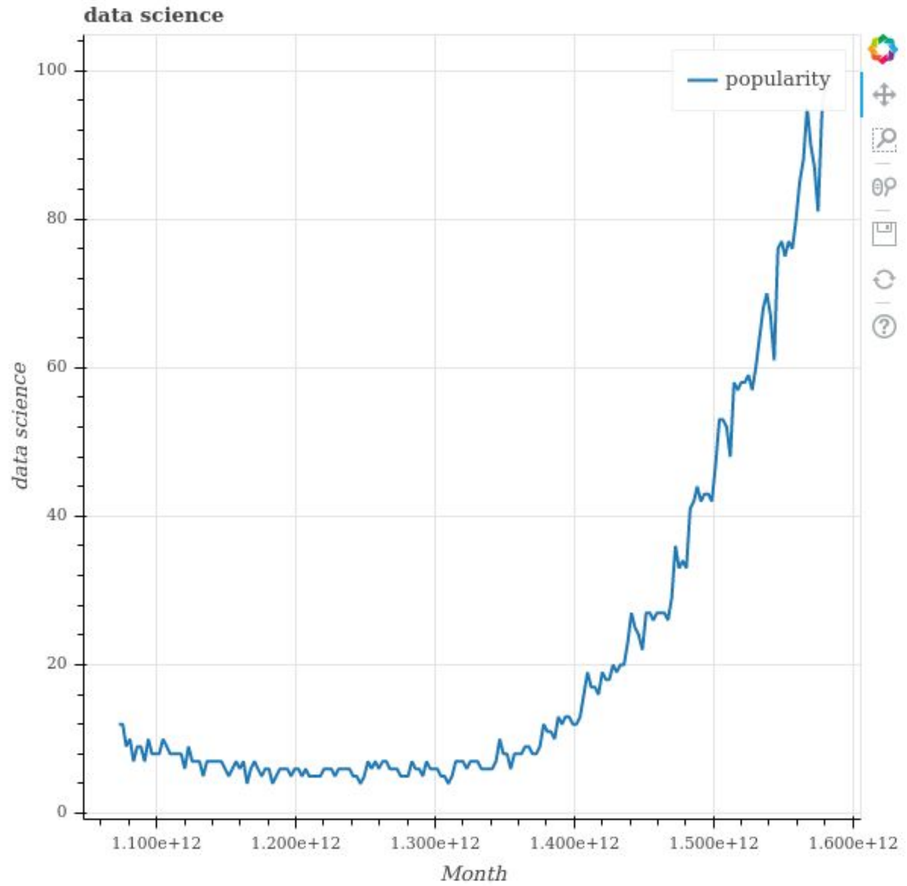
Saturdays.AI
Barcelona

Bokeh

Bokeh

```
p = figure(title='data science', x_axis_label='Month', y_axis_label='data science')
p.line(df['Mes'], df['data science'], legend='popularity', line_width=2)
save(p)
```

It’s [interactive](#)!



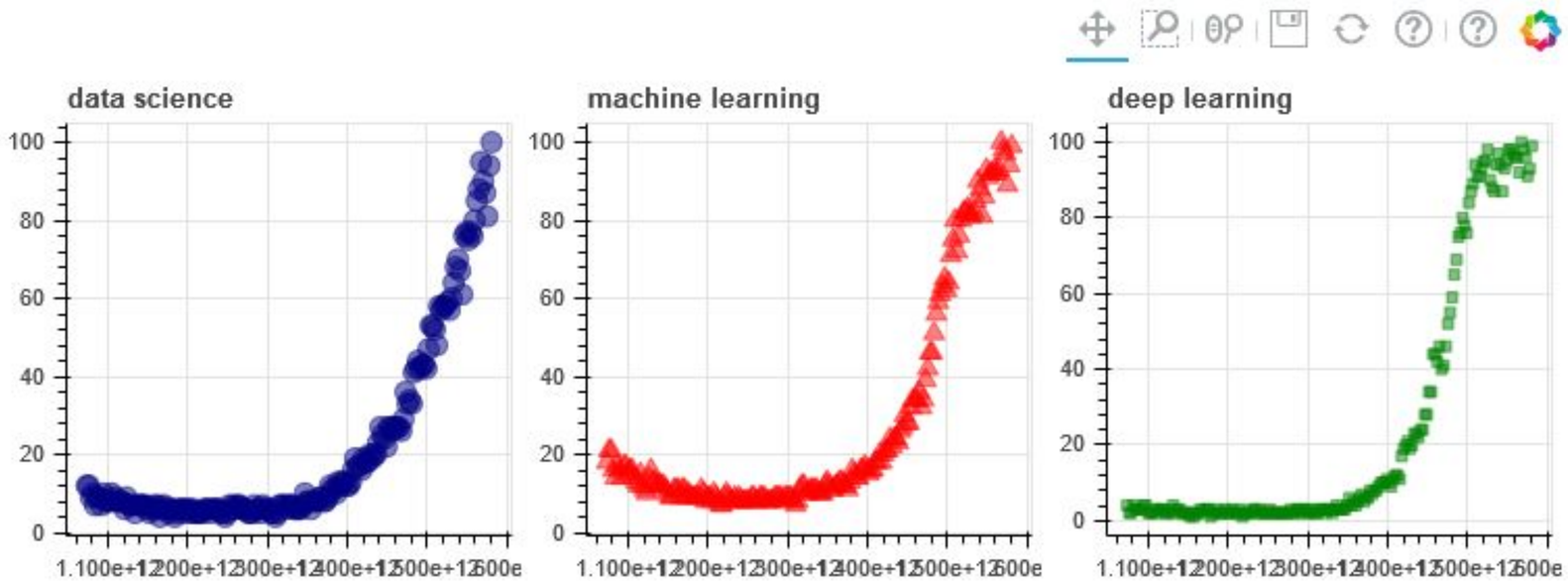
Bokeh - Multiple charts in the same file

```
output_file('multiple_graphs.html')

s1 = figure(width=250, plot_height=250, title='data science')
s1.circle(df['Mes'], df['data science'], size=10, color='navy', alpha=0.5)
s2 = figure(width=250, height=250, x_range=s1.x_range, y_range=s1.y_range, title='machine learning') #share both axis range
s2.triangle(df['Mes'], df['machine learning'], size=10, color='red', alpha=0.5)
s3 = figure(width=250, height=250, x_range=s1.x_range, title='deep learning') #share only one axis range
s3.square(df['Mes'], df['deep learning'], size=5, color='green', alpha=0.5)

p = gridplot([[s1, s2, s3]])
save(p)
```

It's [interactive](#)!

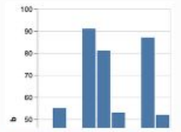


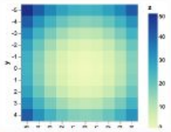
Saturdays.AI
Barcelona

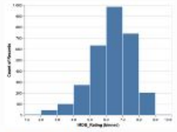
Altair

Altair

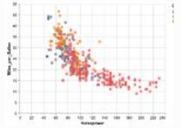
Simple Charts

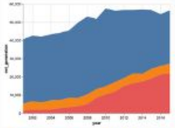

Simple Bar Chart

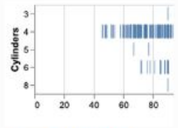

Simple Heatmap


Simple Histogram


Simple Line Chart

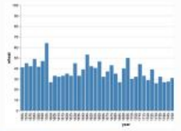

Simple Scatter Plot with Tooltips

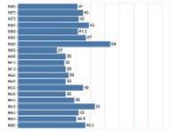

Simple Stacked Area Chart

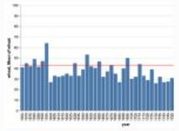


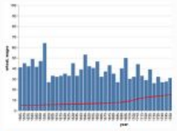
Simple Strip Plot

Bar Charts

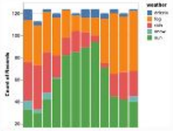

Bar Chart with Highlighted Bar

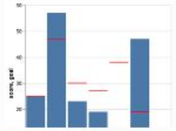

Bar Chart with Labels

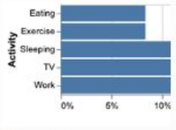

Bar Chart with Line at Mean


Bar Chart with Line on Dual Axis

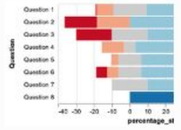

Bar Chart with Negative Values

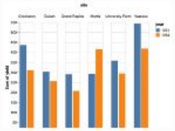

Bar Chart with rounded edges

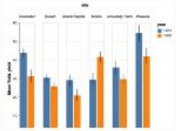

Bar and Tick Chart

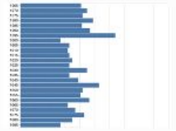


Calculating Percentage of Total


Diverging Stacked Bar Chart


Grouped Bar Chart


Grouped Bar Chart with Error Bars

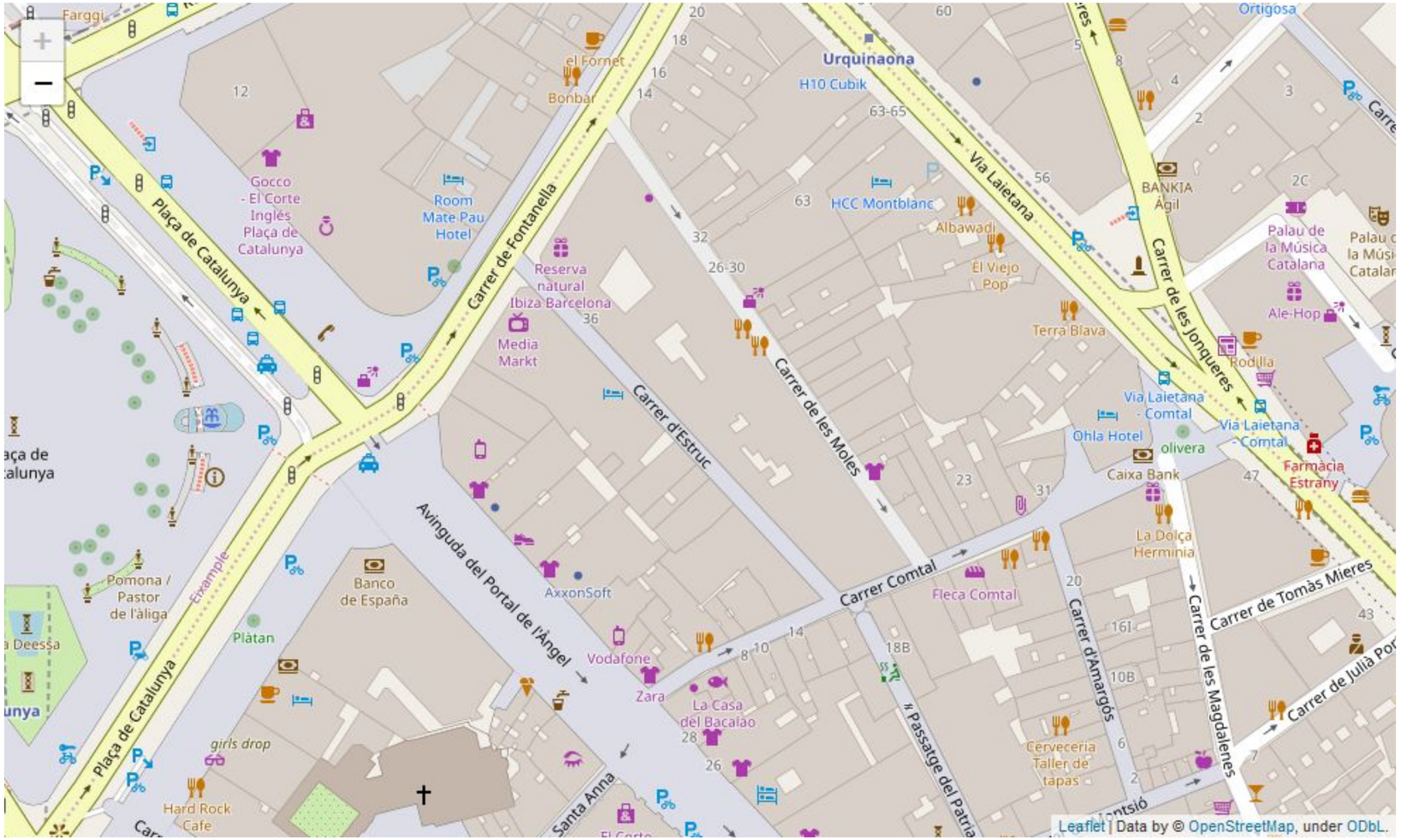

Horizontal Bar Chart

Saturdays.AI
Barcelona

Maps with Folium

Folium

```
import folium
m_1 = folium.Map(location=[41.387, 2.172659], tiles='openstreetmap', zoom_start=18)
```

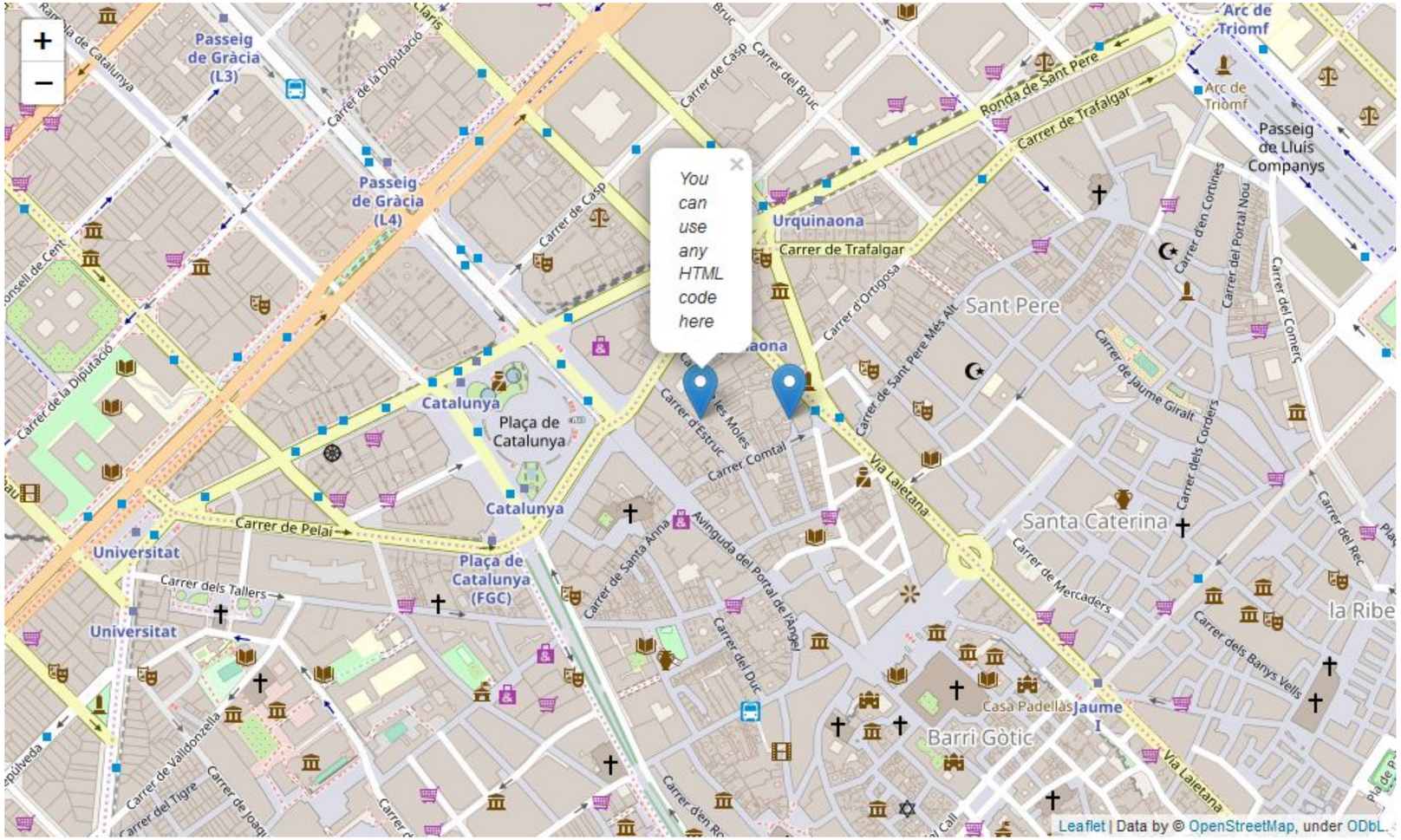


It's [interactive](#)!

Folium - Adding Markers

```
m2 = folium.Map(location=[41.387, 2.172659], tiles='openstreetmap', zoom_start=16)

folium.Marker([41.387, 2.172659], popup='<i>You can use any HTML code here</i>', tooltip='We are here').add_to(m2)
folium.Marker([41.387, 2.174], popup='<b>You can use any HTML code here</b>', tooltip='click me').add_to(m2)
```



It's interactive!

GeoCoding with Geopandas

(78.66774 22.35111)

Folium - BubbleMap

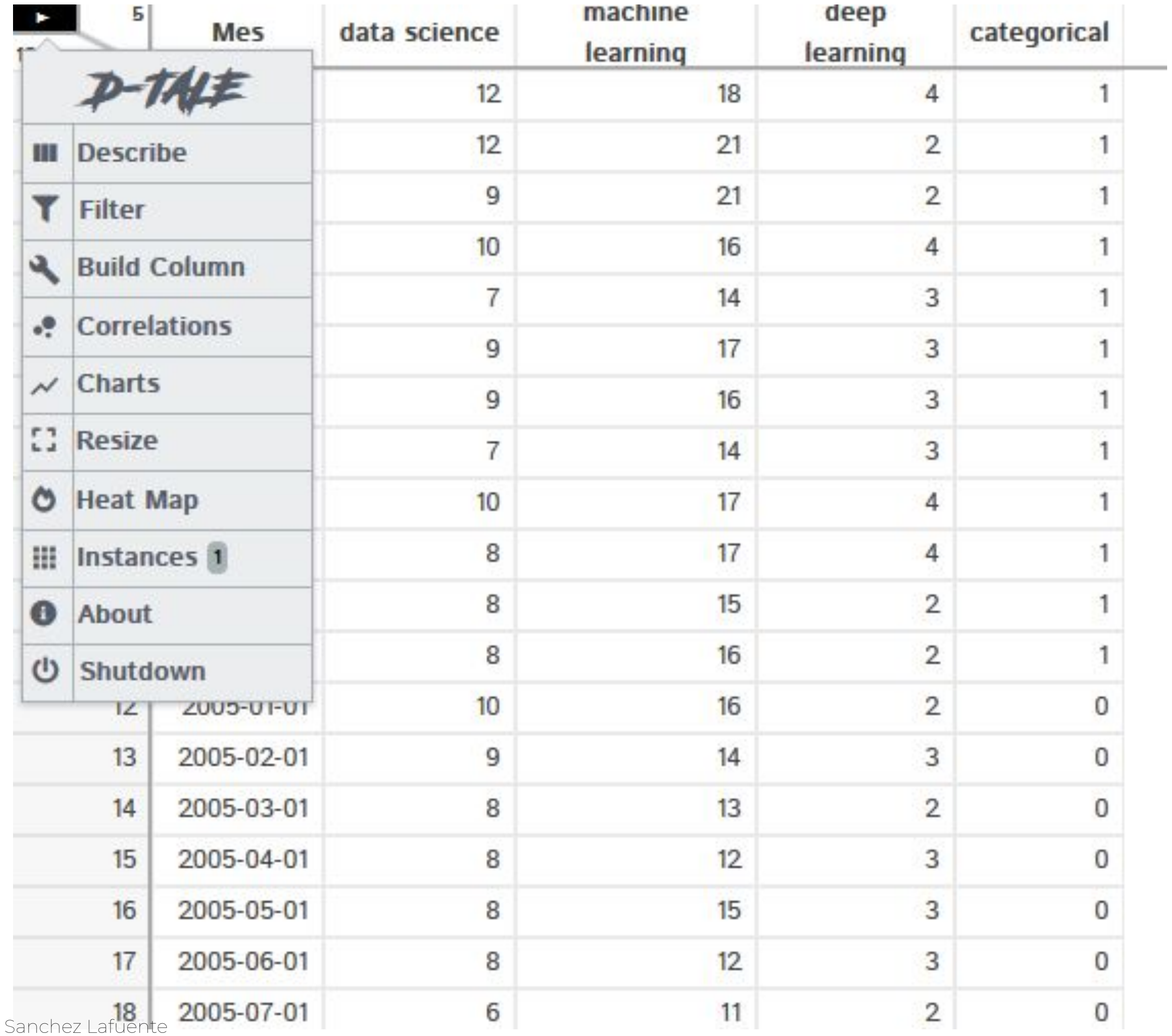
It's [interactive](#)!

Saturdays.AI
Barcelona

D-Tale

D-Tale

Super-new library, came out last week (February 20th, 2020)



	Mes	data science	machine learning	deep learning	categorical
		12	18	4	1
		12	21	2	1
		9	21	2	1
		10	16	4	1
		7	14	3	1
		9	17	3	1
		9	16	3	1
		7	14	3	1
		10	17	4	1
		8	17	4	1
		8	15	2	1
		8	16	2	1
12	2005-01-01	10	16	2	0
13	2005-02-01	9	14	3	0
14	2005-03-01	8	13	2	0
15	2005-04-01	8	12	3	0
16	2005-05-01	8	15	3	0
17	2005-06-01	8	12	3	0
18	2005-07-01	6	11	2	0

It's interactive!

...But I can't show you with our own data

Saturdays.AI
Barcelona

Wrap-up

What should I use?

- Start with pandas / pandas profiling to understand the data
- Follow with Matplotlib / Seaborn to make more plots to understand the data, see what works
- Once you understand the data you can use whatever library gives you the graph you want to make. Look at their galleries….

Gallery of examples

Matplotlib

- https://matplotlib.org/gallery/index.html

Seaborn

- https://seaborn.pydata.org/examples/index.html

Bokeh

- https://docs.bokeh.org/en/latest/docs/gallery.html

Altair

- https://altair-viz.github.io/gallery/index.html

Folium

- https://nbviewer.jupyter.org/github/python-visualization/folium/tree/master/examples/