Accelerating Visual Data Exploration with Lux

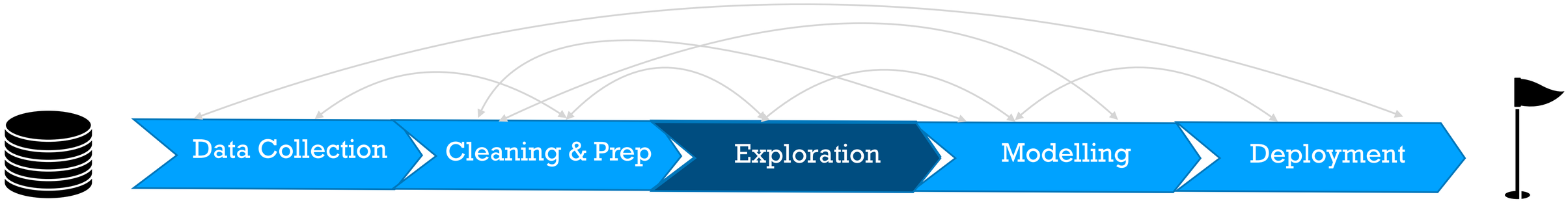
https://github.com/lux-org/lux

Doris Jung-Lin Lee (UC Berkeley)

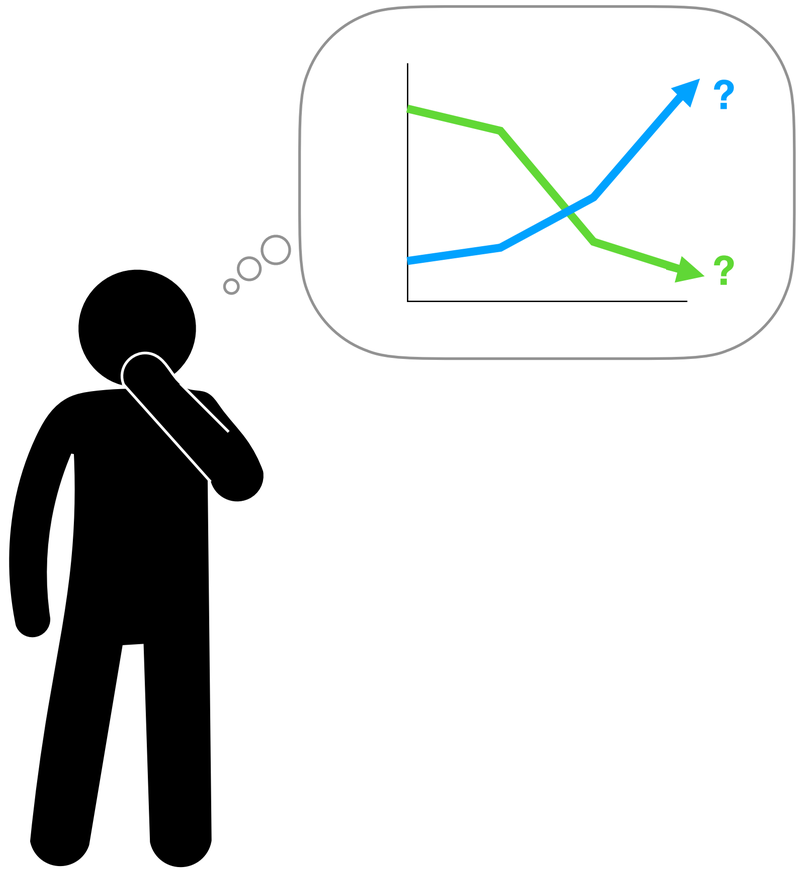
WiDS 2021

Berkeley SCHOOL OF INFORMATION

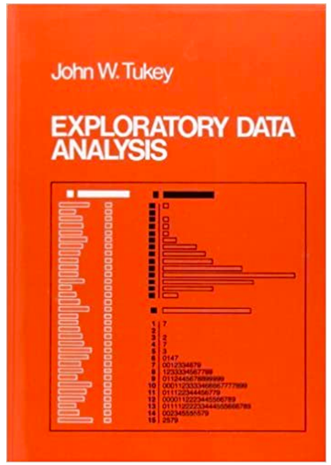
riselab UC Berkeley


Data Collection
Cleaning & Prep
Exploration
Modelling
Deployment

Exploratory Data Analysis (EDA)

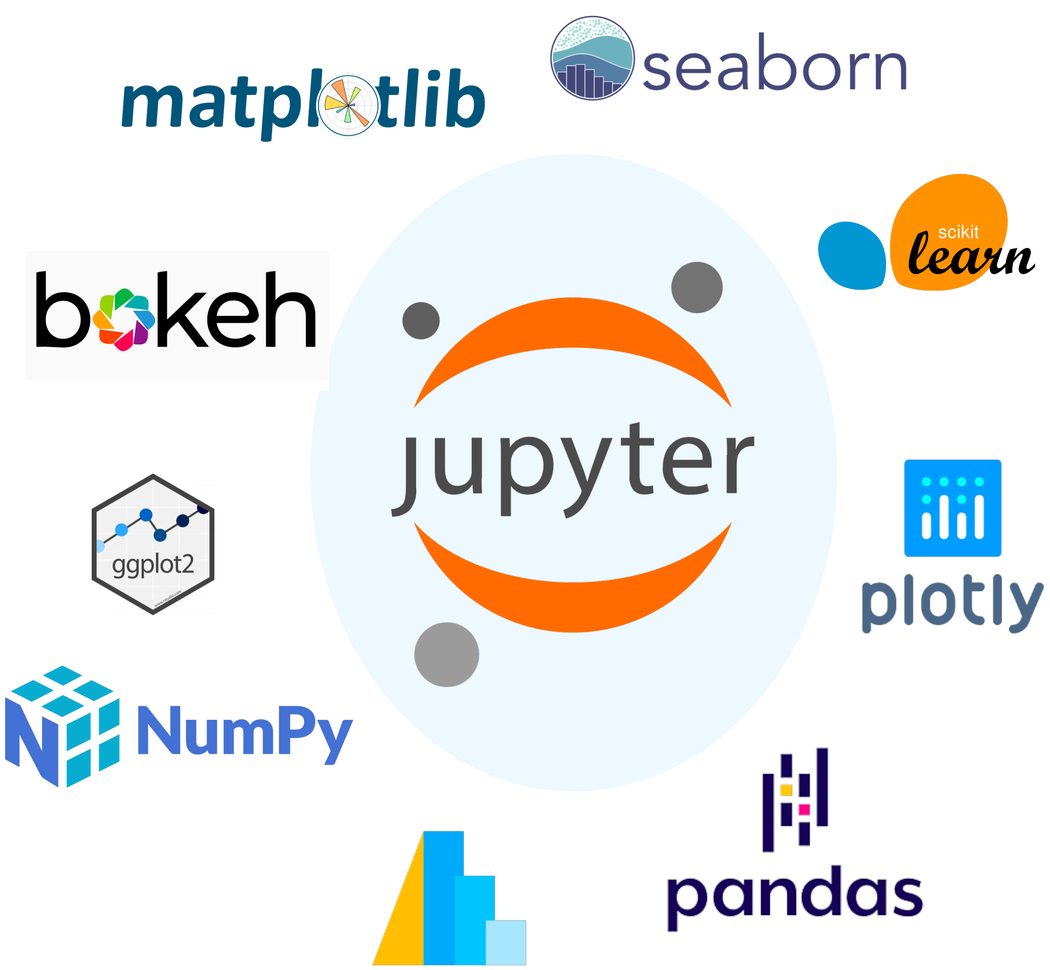


"Exploratory data analysis is an attitude, a state of flexibility, a willingness to look for those things that we believe are not there, as well as those that we believe to be there."



— John Tukey (1970)

State of EDA in 2021: Data Science in Notebooks



jupyter ChurnAnalysis Last Checkpoint: 2 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

```
plot_layout = go.Layout(
        xaxis={"type": "category"},
        title='Online Backup',
        plot_bgcolor  = "rgb(243,243,243)",
        paper_bgcolor  = "rgb(243,243,243)",
    )
fig = go.Figure(data=plot_data, layout=plot_layout)
pyoff.iplot(fig)
```

In [ ]:

```
df_plot = df_data.groupby('DeviceProtection').Churn.mean().reset_index()
plot_data = [
    go.Bar(
        x=df_plot['DeviceProtection'],
        y=df_plot['Churn'],
        width = [0.5, 0.5, 0.5],
        marker=dict(
        color=['green', 'blue', 'orange'])
    )
]

plot_layout = go.Layout(
        xaxis={"type": "category"},
        title='Device Protection',
        plot_bgcolor  = "rgb(243,243,243)",
        paper_bgcolor  = "rgb(243,243,243)",
    )
fig = go.Figure(data=plot_data, layout=plot_layout)
pyoff.iplot(fig)
```

In [ ]:

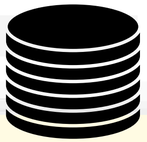
```
df_plot = df_data.groupby('TechSupport').Churn.mean().reset_index()
plot_data = [
    go.Bar(
        x=df_plot['TechSupport'],
        y=df_plot['Churn'],
        width = [0.5, 0.5, 0.5],
        marker=dict(
        color=['green', 'blue', 'orange'])
    )
]
```

State of EDA in 2021: Data Science in Notebooks

Dataframes with pandas

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechS
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	No	
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...		

7043 rows × 21 columns



jupyter ChurnAnalysis Last Checkpoint: 2 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code

```
plot_layout = go.Layout(
        xaxis={"type": "category"},
        title='Online Backup',
        plot_bgcolor  = "rgb(243,243,243)",
        paper_bgcolor  = "rgb(243,243,243)",
    )
fig = go.Figure(data=plot_data, layout=plot_layout)
pyoff.iplot(fig)
```

In [ ]:

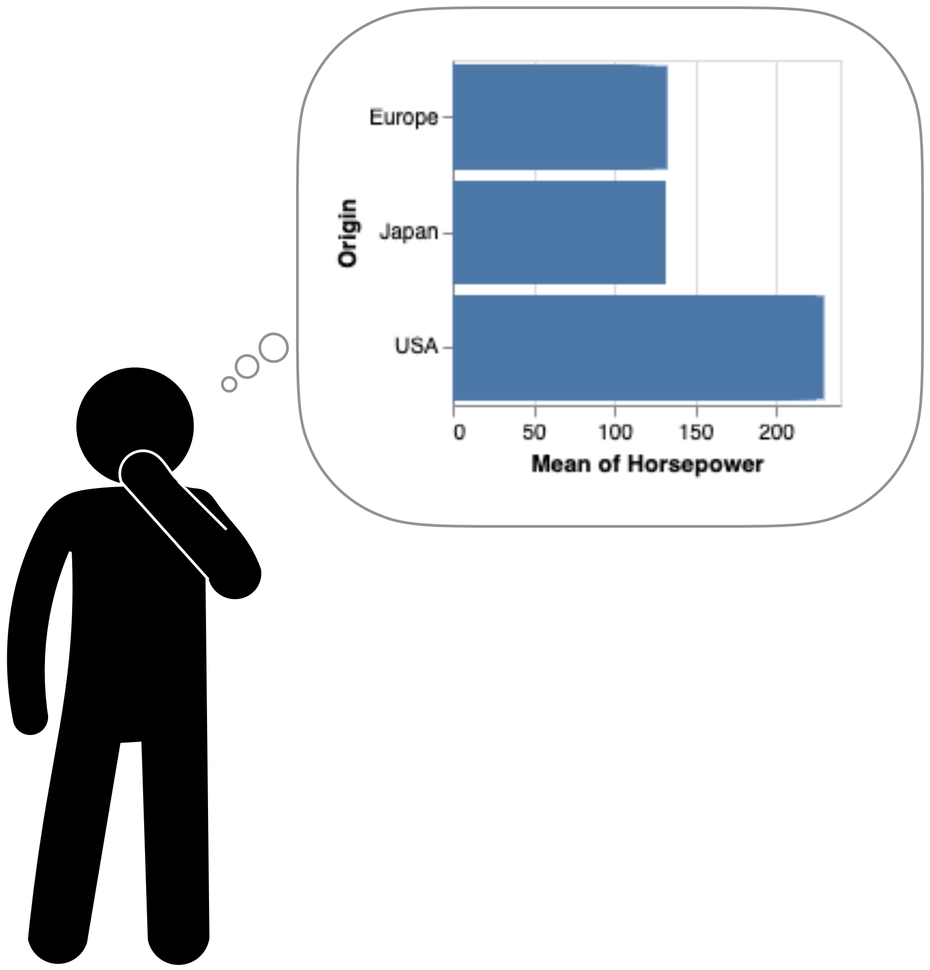
```
df_plot = df_data.groupby('DeviceProtection').Churn.mean().reset_index()
plot_data = [
    go.Bar(
        x=df_plot['DeviceProtection'],
        y=df_plot['Churn'],
        width = [0.5, 0.5, 0.5],
        marker=dict(
        color=['green', 'blue', 'orange'])
    )
]

plot_layout = go.Layout(
        xaxis={"type": "category"},
        title='Device Protection',
        plot_bgcolor  = "rgb(243,243,243)",
        paper_bgcolor  = "rgb(243,243,243)",
    )
fig = go.Figure(data=plot_data, layout=plot_layout)
pyoff.iplot(fig)
```

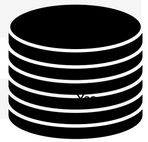
In [ ]:

```
df_plot = df_data.groupby('TechSupport').Churn.mean().reset_index()
plot_data = [
    go.Bar(
        x=df_plot['TechSupport'],
        y=df_plot['Churn'],
        width = [0.5, 0.5, 0.5],
        marker=dict(
        color=['green', 'blue', 'orange'])
    )
]
```

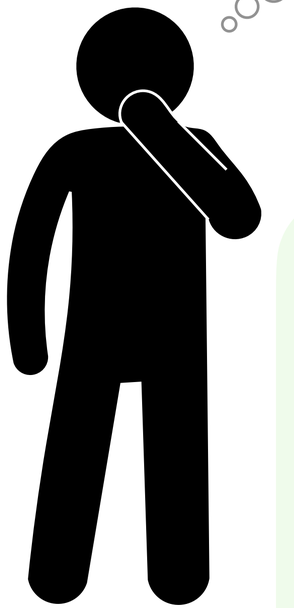
What’s inside my dataframe?

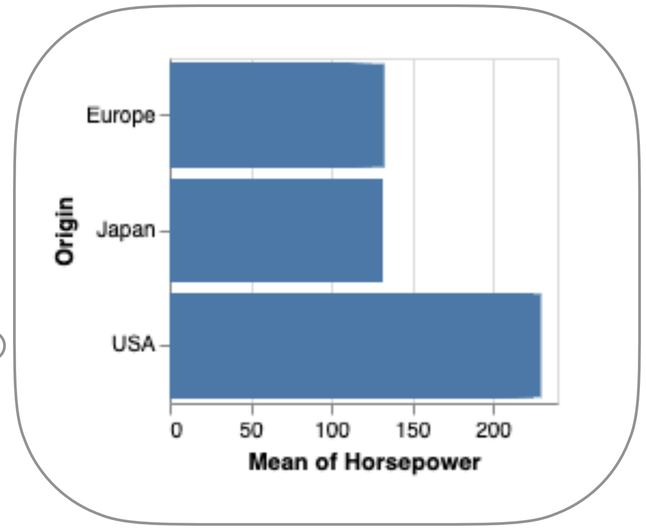


	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	Techs
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...		
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...		
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...		

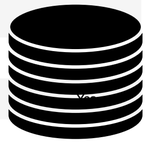


Visualizing DataFrames = LOTS of code ! 😫





	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	Techs
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...		
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...		
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...		



How should my visualization look like?

What portions of the data should I look at?

What marks and chart type should I use?

Should I apply binning or grouping on my data?

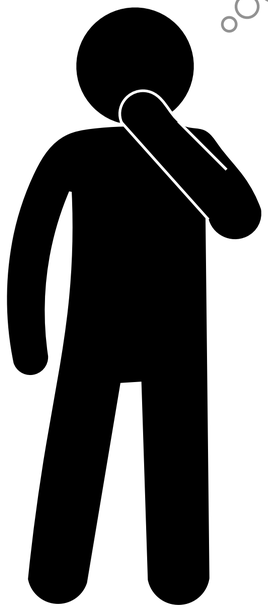
What chart annotations should I add?

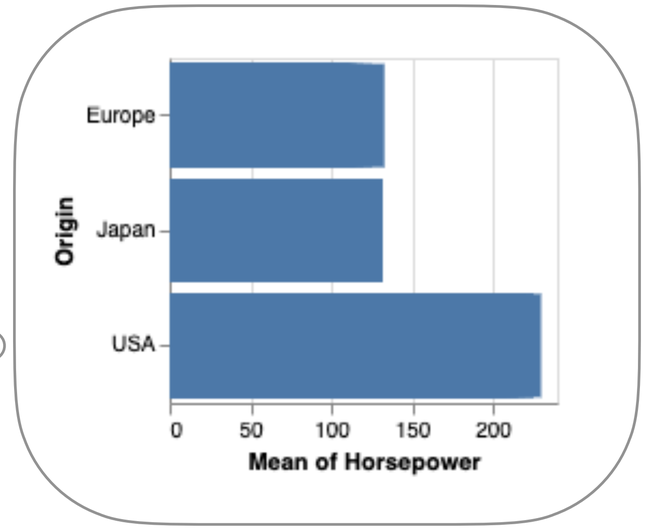
What data transformations should I apply on my data?

What do I put on the axes and channels?

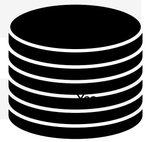
What attributes should I visualize?

Visualizing DataFrames = LOTS of code ! 😫





	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	Tech
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...		
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...		
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...		



```
import plotly.graph_objects as go
import pandas as pd
df = pd.read_csv("data/cars.csv")
barVal = df.groupby("Origin").mean()
["Horsepower"]
fig = go.Figure(go.Bar(
            x= barVal,
            y= barVal.index,
            orientation='h'))
fig.update_layout(
    xaxis_title="Mean of Horsepow
    yaxis_title="Origin",
)
```

plotly

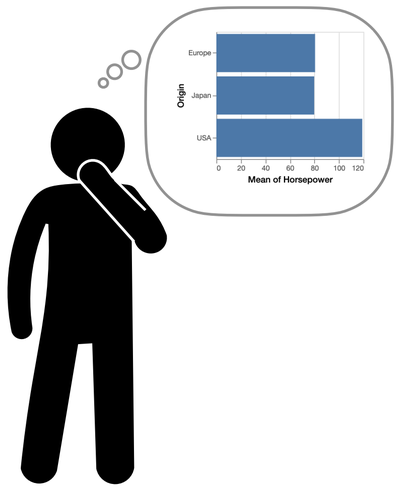
```
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv("data/cars.csv")
barVal = df.groupby("Origin").mean()["Horsepower"]
y_pos = range(len(barVal))
plt.barh(y_pos,barVal, align='center', alpha=0.5)
plt.yticks(y_pos,list(barVal.index))
plt.xlabel('Mean of Horsepower')
plt.ylabel('Origin')
plt.show()
```

matplotlib

LUX

github.com/lux-org/lux

Always-on visualizations of dataframes in-situ notebooks




```
df.intent=[“Origin”,“Horsepower”]
```

Toggle Pandas/Lux

	Name	MilesPerGal	Cylinders	Displacement	Horsepower	Weight	Acceleration	Year	Origin
0	chevrolet chevelle malibu	18.0	8	307.0	130	3504	12.0	1970-01-01	USA
1	buick skylark 320	15.0	8	350.0	165	3693	11.5	1970-01-01	USA
2	plymouth satellite	18.0	8	318.0	150	3436	11.0	1970-01-01	USA
3	amc rebel sst	16.0	8	304.0	150	3433	12.0	1970-01-01	USA
4	ford torino	17.0	8	302.0	140	3449	10.5	1970-01-01	USA
...	...	...	...	...	...	...	...	...	...
387	ford mustang gl	27.0	4	140.0	86	2790	15.6	1982-01-01	USA
388	vw pickup	44.0	4	97.0	52	2130	24.6	1982-01-01	Europe
389	dodge rampage	32.0	4	135.0	84	2295	11.6	1982-01-01	USA
390	ford ranger	28.0	4	120.0	79	2625	18.6	1982-01-01	USA
391	chevy s-10	31.0	4	119.0	82	2720	19.4	1982-01-01	USA

392 rows × 9 columns

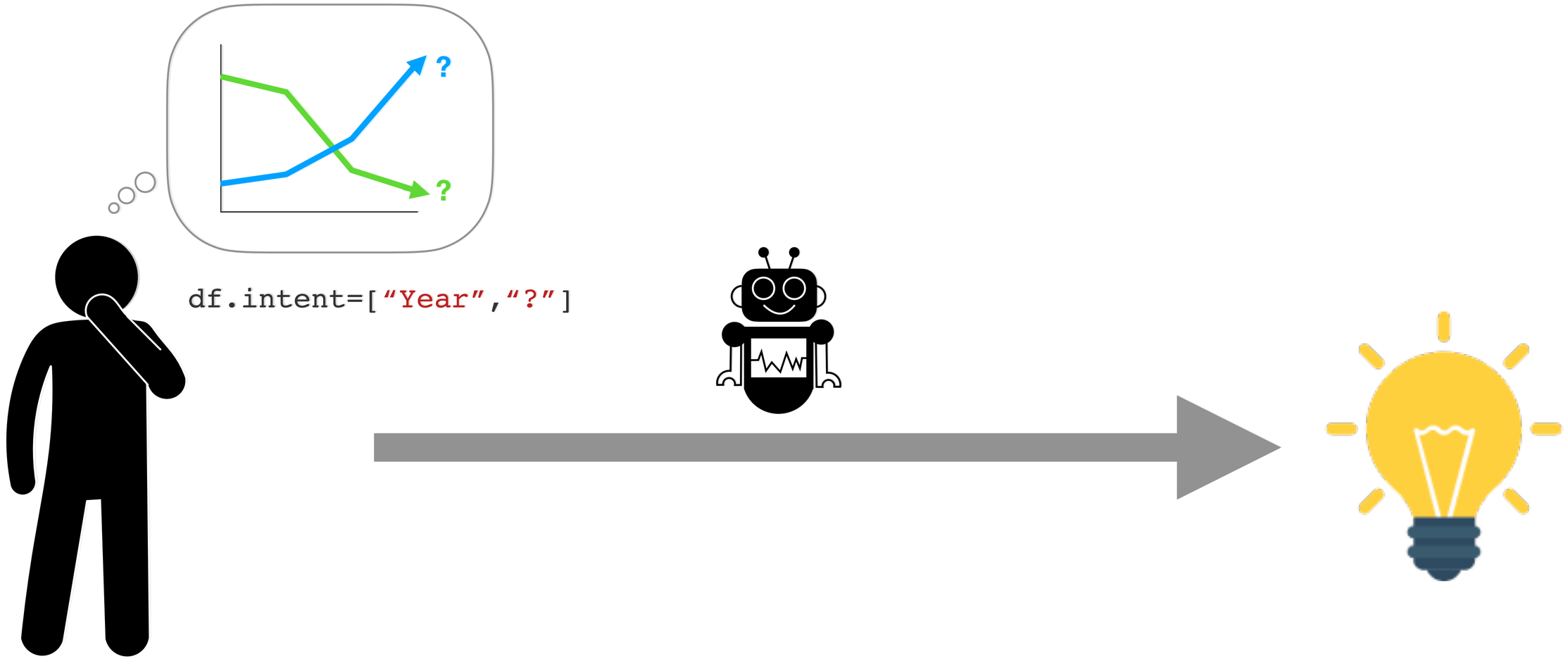
In-Situ Visualization of Dataframes

In [ ]:
```
import pandas as pd
import lux
```

In [ ]:
```
df = pd.read_csv("car.csv")
```

In [ ]:
```
df
```

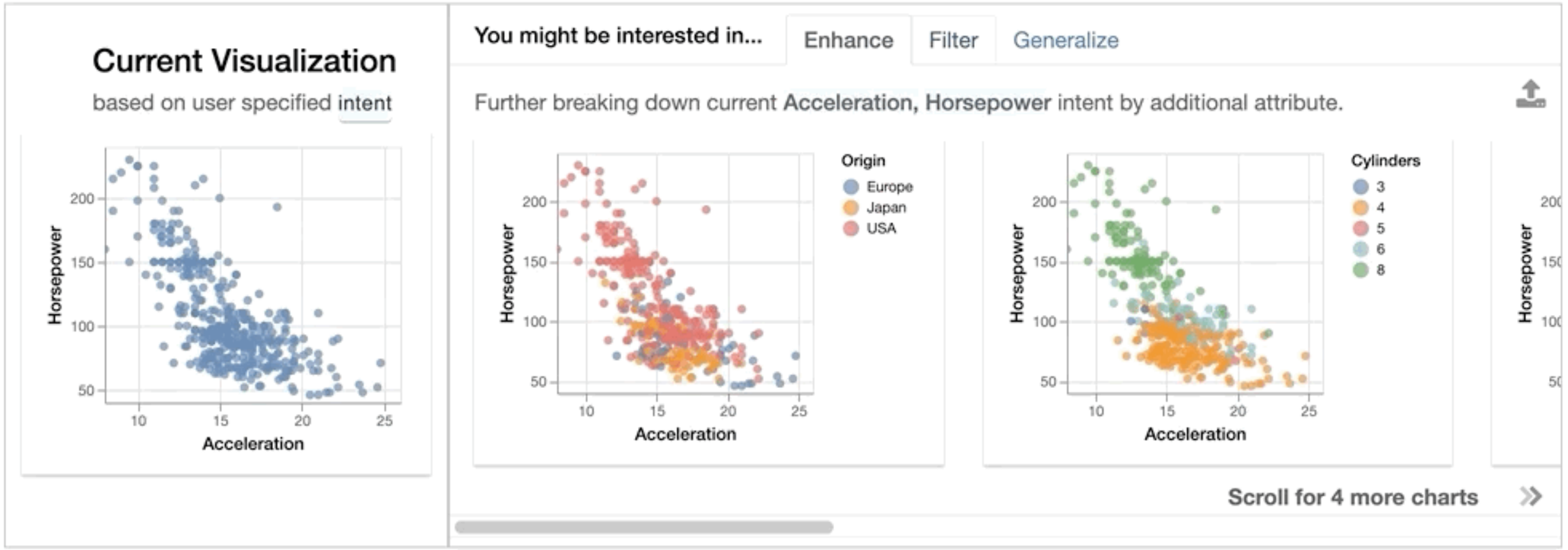
Powerful, Intuitive Intent Specification Language



Intent-Driven, Next-step Recommendations

```
In [8]: df.intent = ["Acceleration","Horsepower"]
        df
```

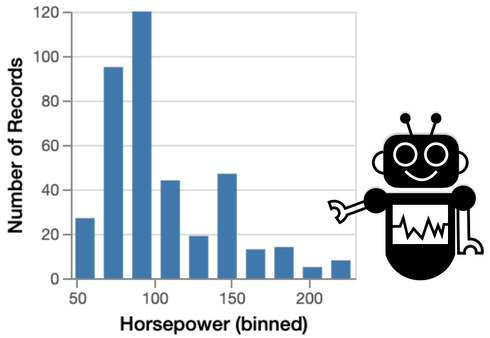
Toggle Pandas/Lux



Quick, on-demand visualizations based on intent

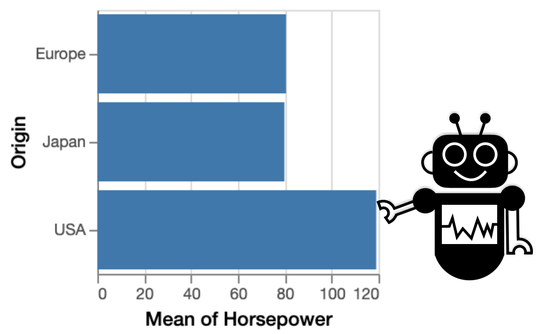
Let's start by looking at Horsepower!

```
In [1]: Vis(["Horsepower"],df)
```



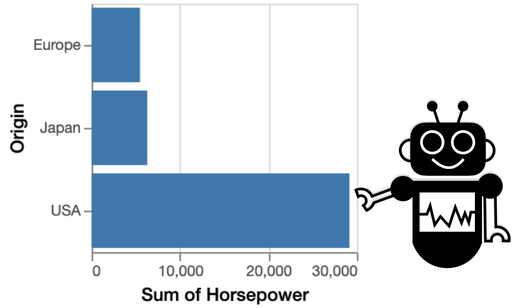
Ok, now add Origin to this.

```
In [2]: Vis(["Horsepower","Origin"],df)
```



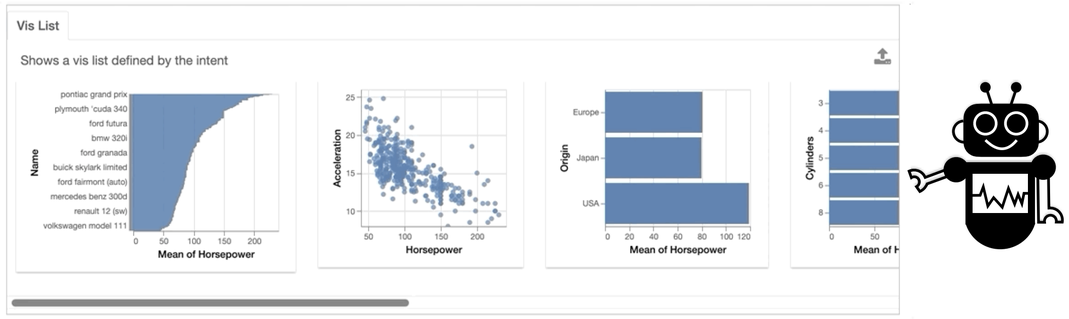
Hmm...maybe aggregate by sum instead of average

```
In [3]: Vis(["Origin",
        lux.Clause("Horsepower",aggregation="sum")],df)
```



What does other visualizations involving Horsepower look like?

```
In [4]: VisList(["?","Horsepower"],df)
```



lux-org / lux

Star 778 | Fork 102

Code | Issues 48 | Pull requests 3 | ZenHub | Actions | Projects | Wiki

master | 3 branches | 6 tags | Go to file | Add file | Code

dorisjlee avoid current_vis expire upon expire_recs (#278) dfa64eb 2 hours ago 598 commits

conda.recipe	Conda setup (#264)	4 days ago
doc	Update Documentation for Data Type (#252)	3 days ago
lux	avoid current_vis expire upon expire_recs (#278)	2 hours ago
tests	avoid current_vis expire upon expire_recs (#278)	2 hours ago
.gitignore	Global shared variable in test (#144) (#149)	3 months ago

README.md

LUX

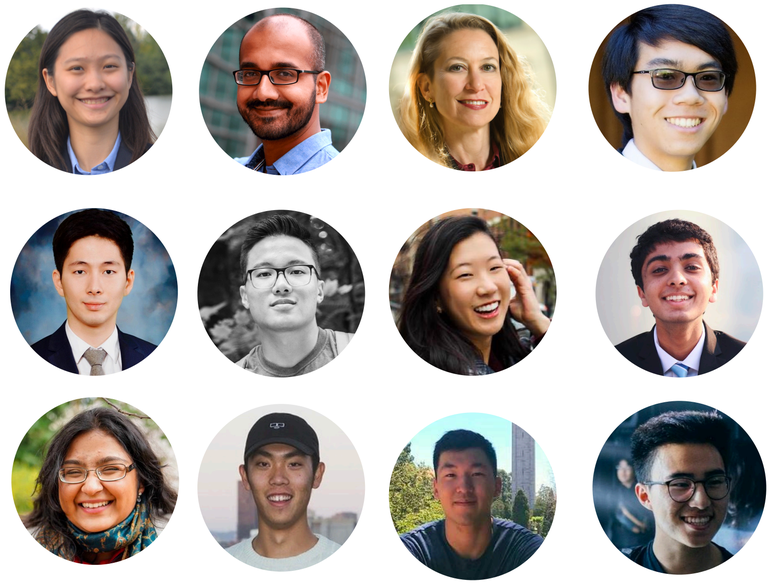
A Python API for Intelligent Visual Discovery

build passing | pypi package 0.2.3 | docs passing | chat Slack | email signup | launch binder | codecov 78% | @lux_api 18

Lux is a Python library that makes data science easier by automating aspects of the data exploration process. Lux facilitate faster experimentation with data, even when the user does not have a clear idea of what they are looking for. Visualizations are displayed via an interactive widget that allow users to quickly browse through large collections of visualizations directly within their Jupyter notebooks.

Berkeley SCHOOL OF INFORMATION

riselab UC Berkeley



@lux_api

github.com/lux-org/lux

tinyurl.com/try-lux