Lux: Towards Intelligent Visual Data Discovery

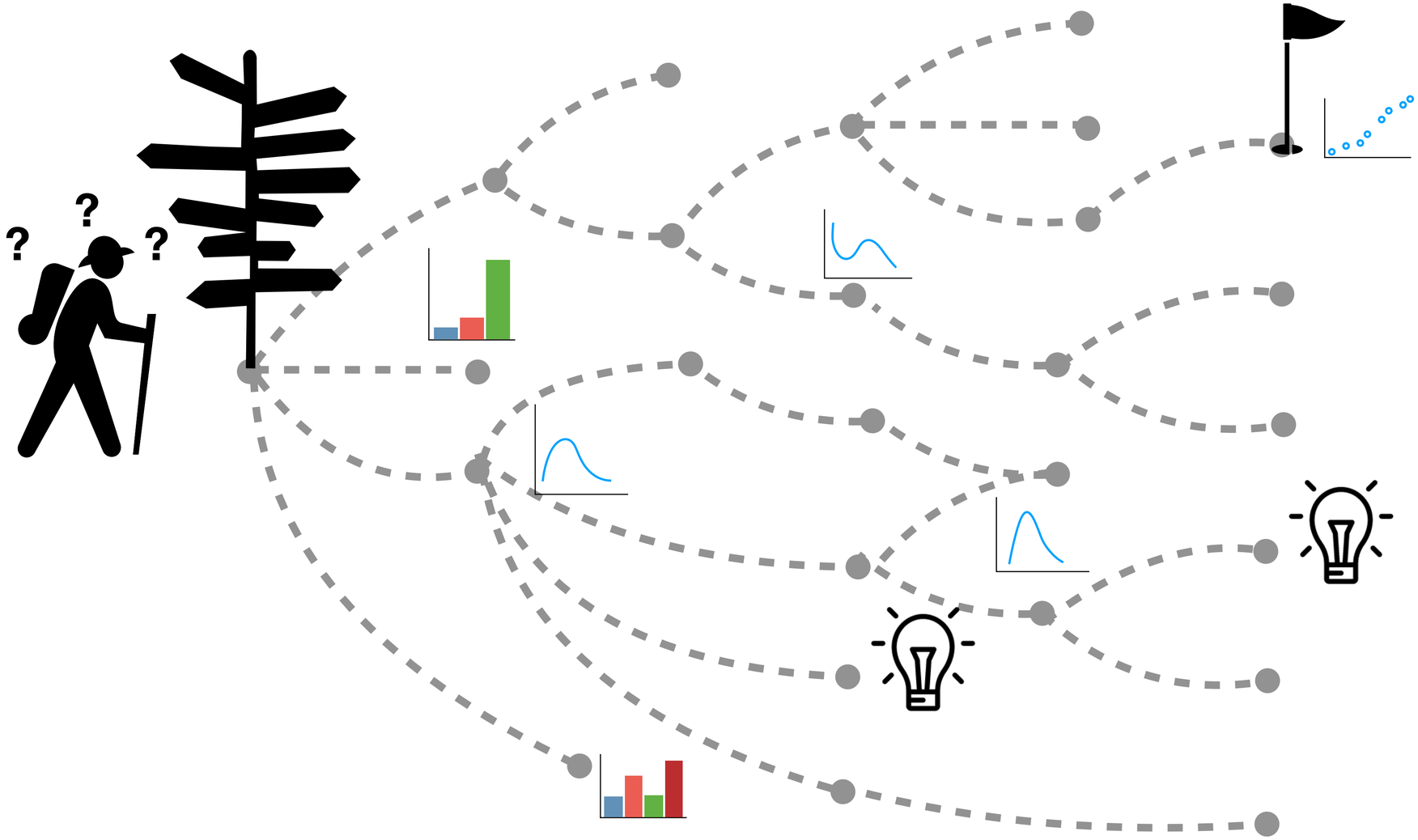
https://github.com/lux-org/lux

Doris Jung-Lin Lee, Thyne Boonmark, Aditya Parameswaran (Berkeley)
Vidya Setlur, Melanie Tory (Tableau Research)
Jaewoo Kim, Karrie Karahalios (UIUC)

“The purpose of computing is insight, not numbers”

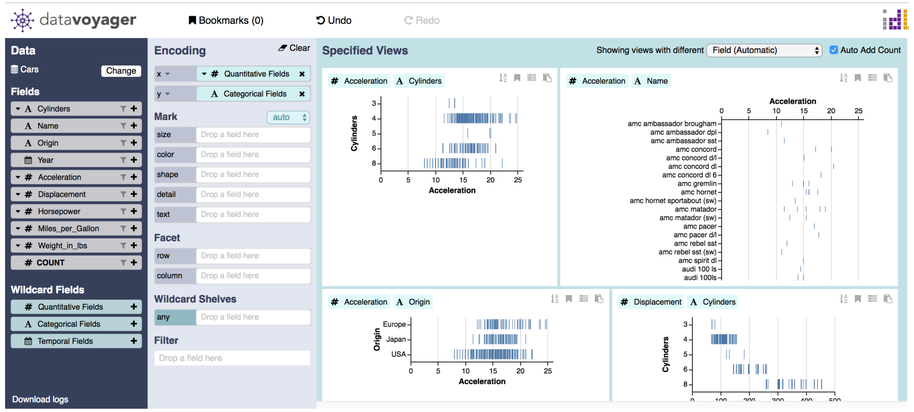
–Richard Hamming

Data exploration is tedious, iterative, and overwhelming

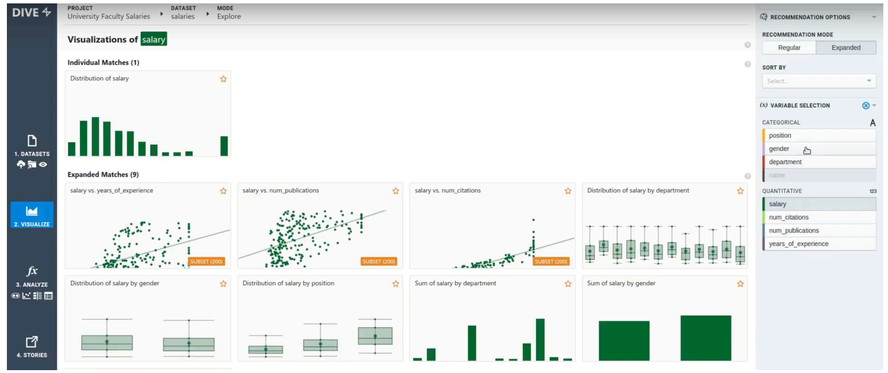


Visualization Recommendation Systems

Suggest interesting patterns and trends in data

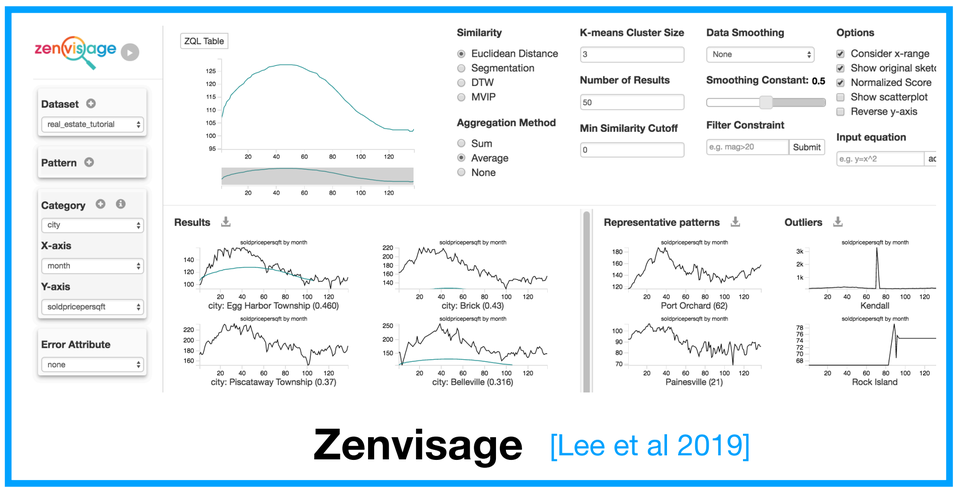


Voyager [Wongsuphasawat et al 2017]

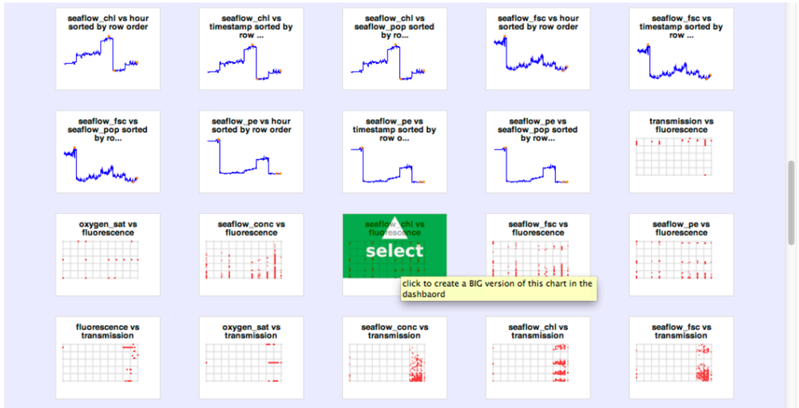


DIVE [Hu et al 2018]

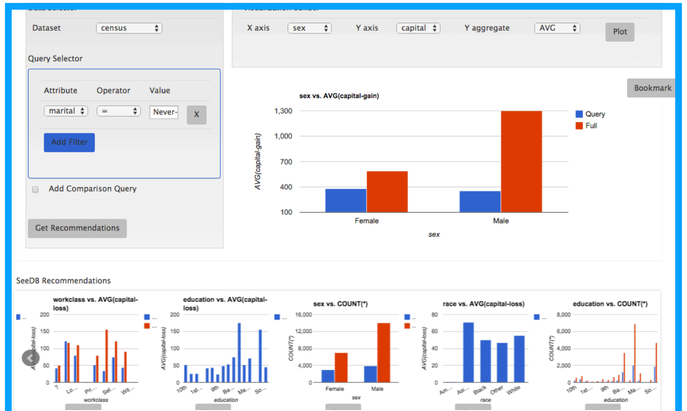
VisPilot [Lee et al 2019]



Zenvisage [Lee et al 2019]



VizDeck [Key et al 2012]



SEEDB [Vartak et al 2016]

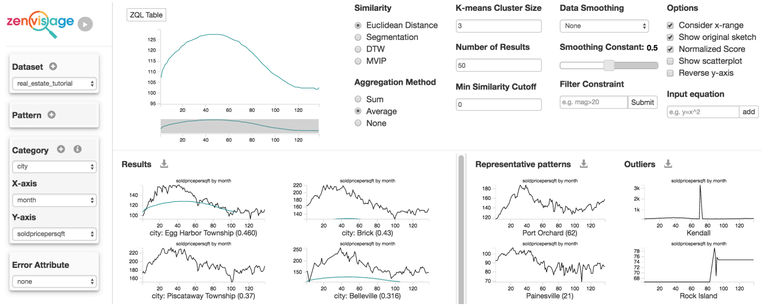
Challenge #1: Limited Task Support

Voyager

[Wongsuphasawat et al 2017]

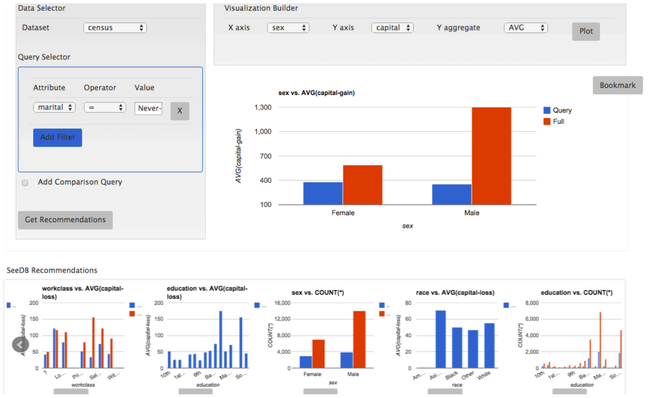
Zenvisage

[Lee et al 2019]

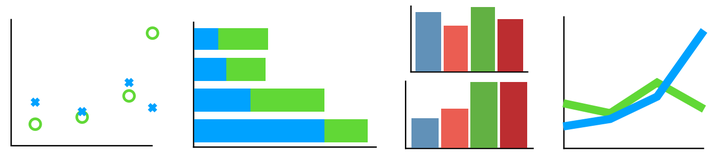


SEEDB

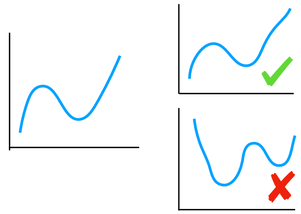
[Vartak et al 2016]



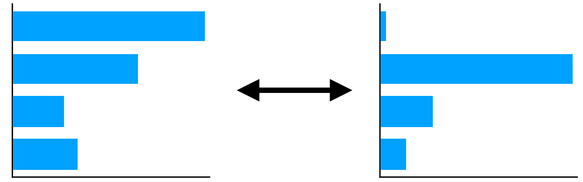
Perceptual effectiveness



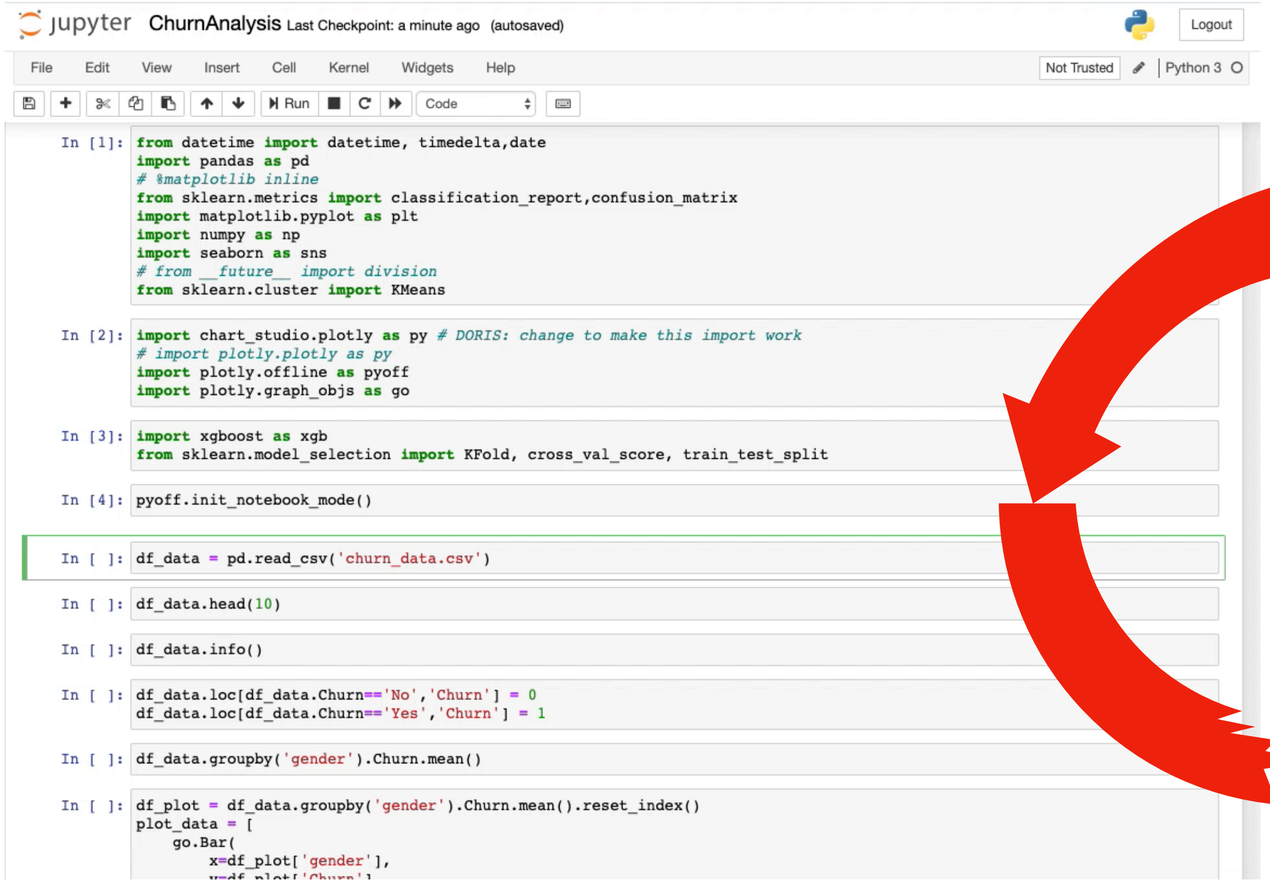
Similarity

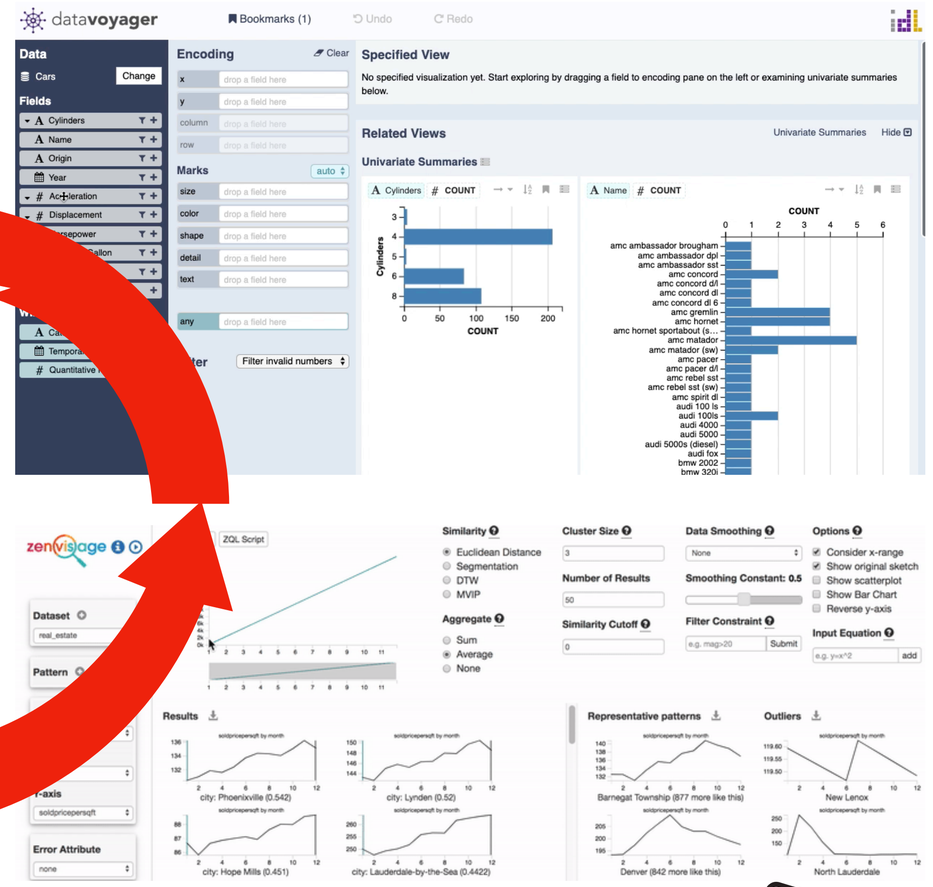


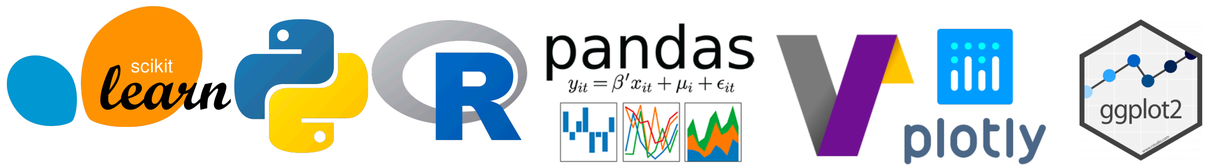
Deviation



Challenge #2: Disconnected Workflow







LUX

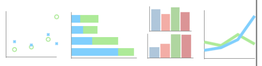
Challenge #1: Limited Task Support

Voyager
[Wongsuphasawat et al 2017]

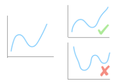
Zenvisage
[Lee et al 2019]

SEEDB
[Vartak et al 2016]

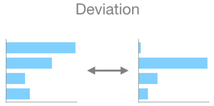
Perceptual effectiveness



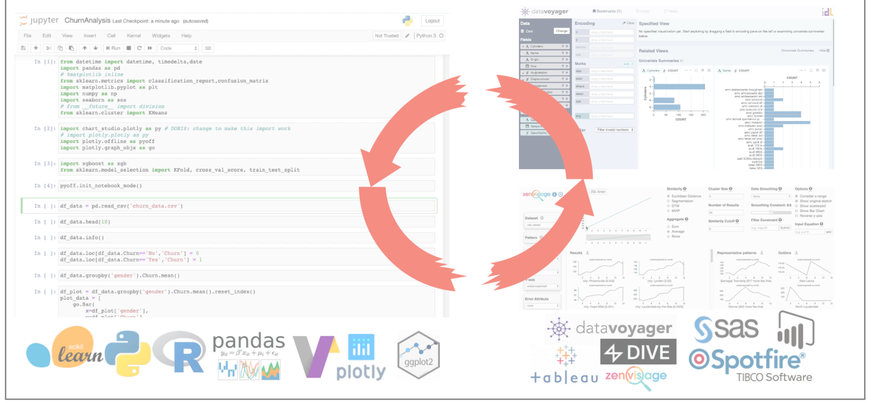
Similarity



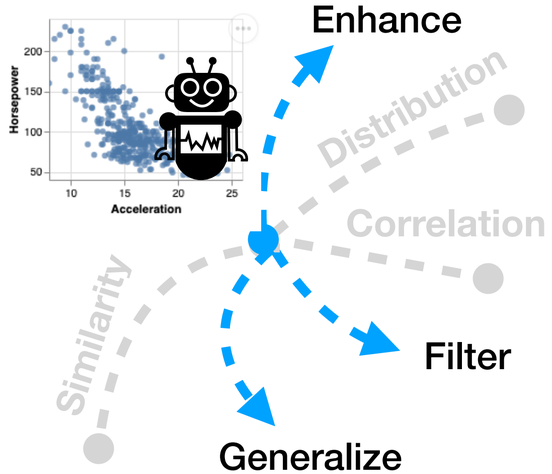
Deviation



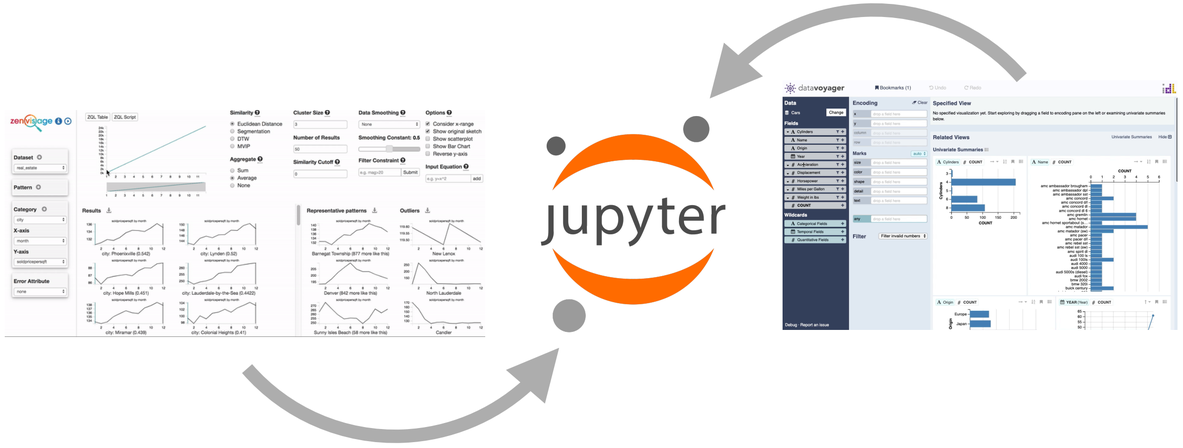
Challenge #2: Disconnected Workflow



General, Extensible Analytics Framework



In-situ Jupyter Widget



jupyter LuxDemo Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [1]:
```
import lux
```

Load in a dataset of 392 different cars from 1970-1982:

In [ ]:
```
dataset = lux.Dataset("lux/data/car.csv",schema=[{"Year":{"dataType":"date"}}])
```

In [ ]:
```
dataset.df.head()
```

In [ ]:
```
dobj = lux.DataObj(dataset)

result = dobj.showMore()

result.display()
```

Intuitively, we expect cars with more horsepower means higher acceleration, but we are actually seeing the opposite of that trend.

Let's learn more about whether there are additional factors that is affecting this relationship.

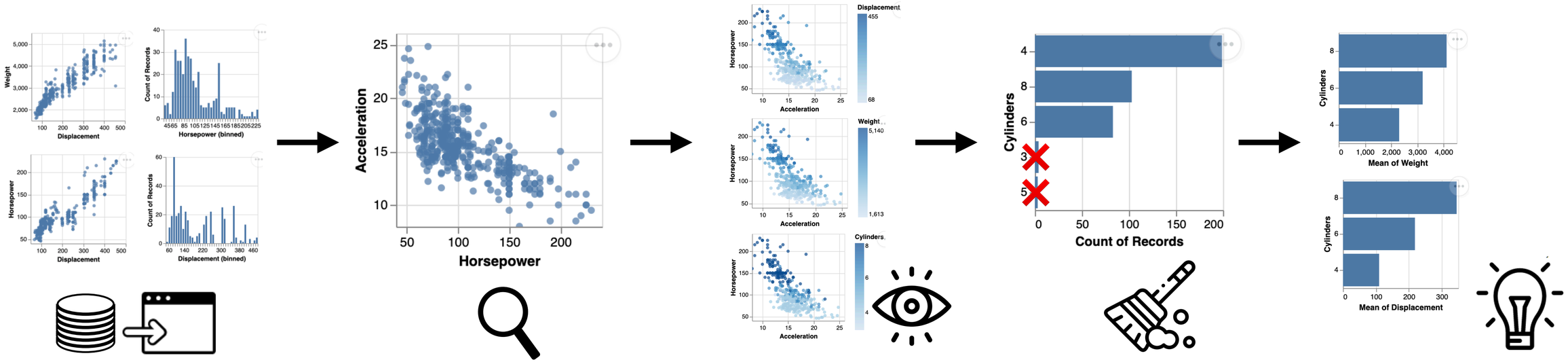
In [ ]:
```
dobj = lux.DataObj(dataset,[lux.Column("Acceleration",dataModel="measure"),
                            lux.Column("Horsepower",dataModel="measure")])
result = dobj.showMore()
result.display()
```

In Enhance, all the added variable (color), except `MilesPerGal`, shows a trend for the value being higher on the upper-left end, and value decreases towards the bottom-right.

Now given these three other variables, let's look at what the `Displacement` and `Weight` is like for different `Cylinder` cars.

Demo Summary

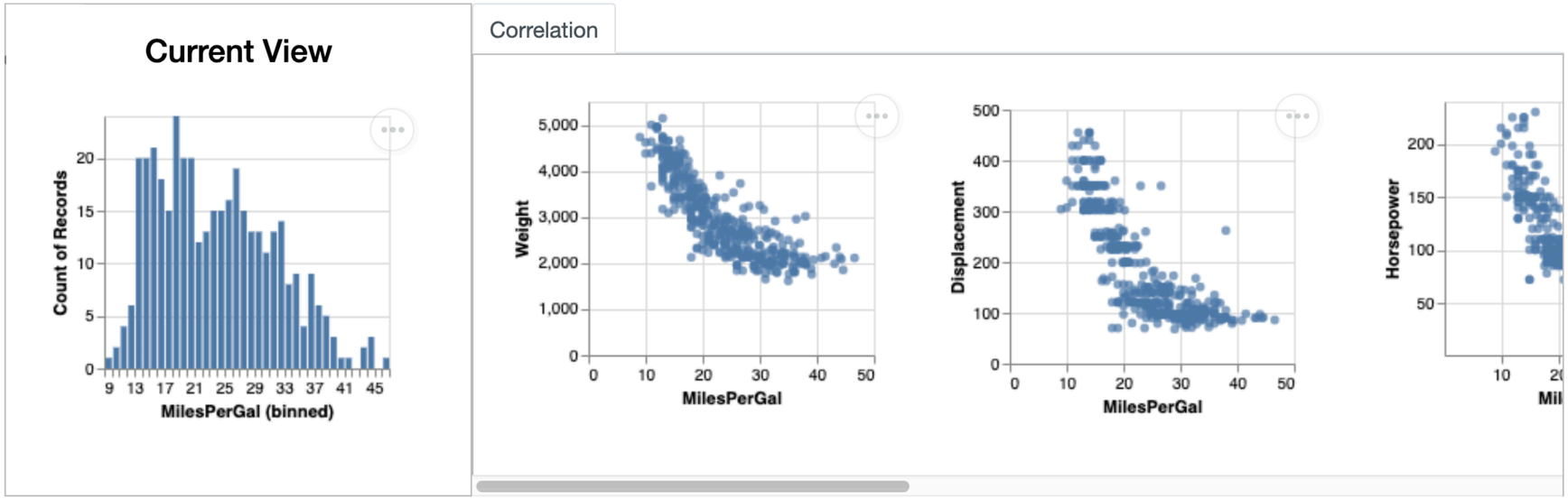
- Seamless integration: Load, clean data with pandas
- Overview: Correlation, Distribution
- ShowMore: Enhance, Filter, Generalize
- Generate vis collection on-the-fly and for rapid browsing



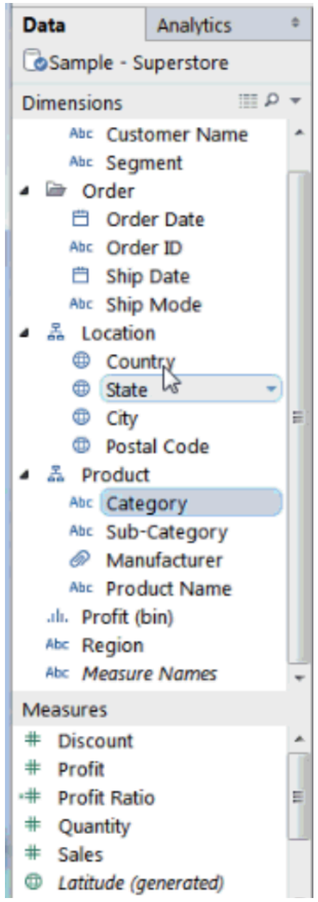
Lux language design

Target + **Action**

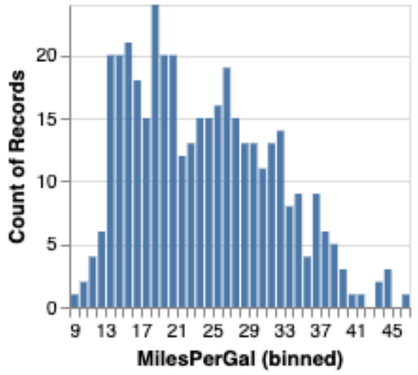
```
dobj = DataObj(dataset,[Column("MilesPerGal"),Column("?",dataModel="measure")])
dobj.correlation()
```



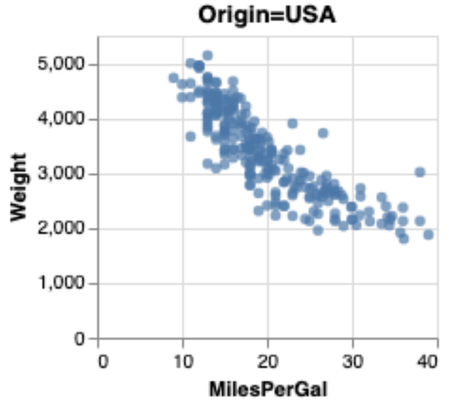
Target: Representing Data Aspects of Interest



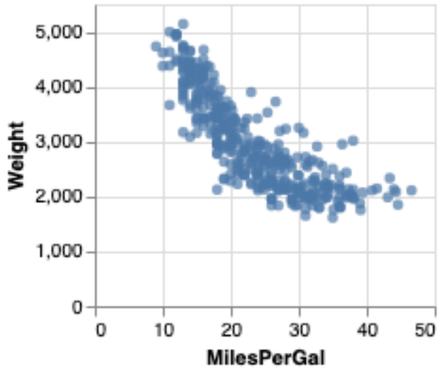
[Column(“MilesPerGal")]

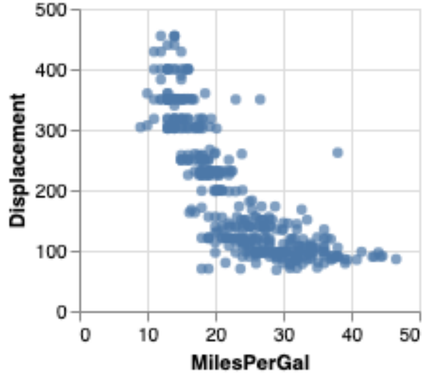


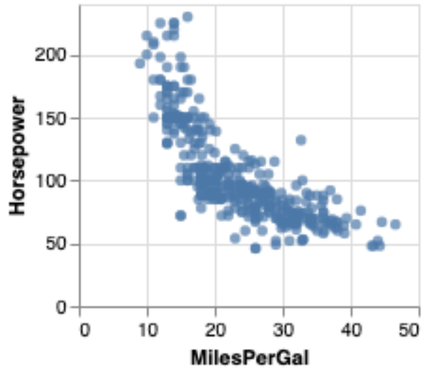
[Column("MilesPerGal"),Column("Weight"),
Row("Origin","USA")]



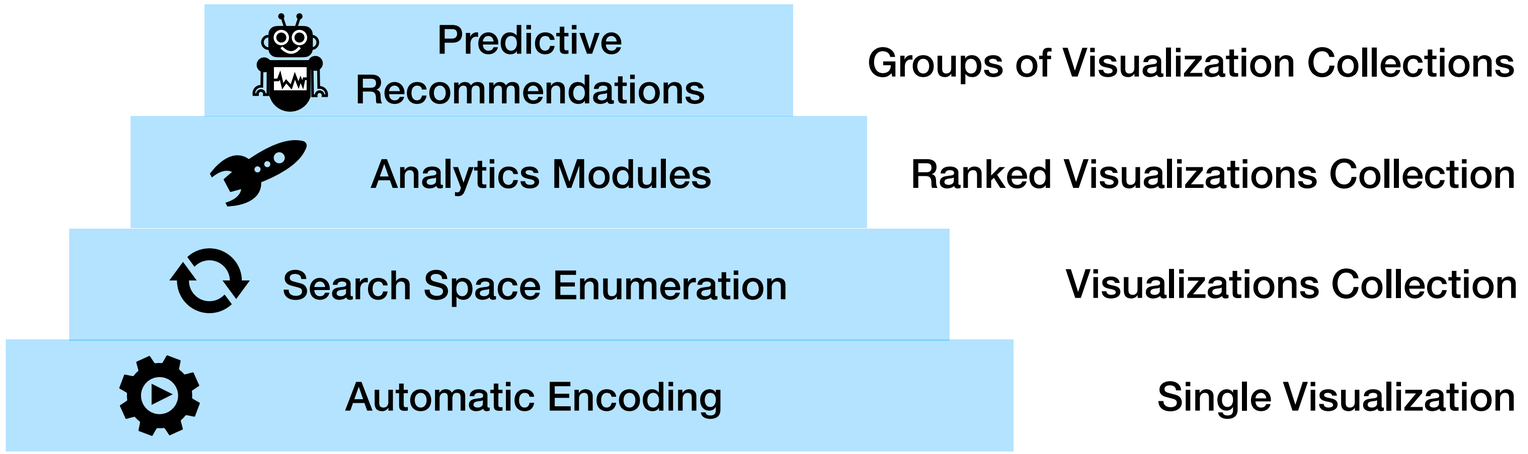
[Column("MilesPerGal"),Column("?",dataModel="measure")]



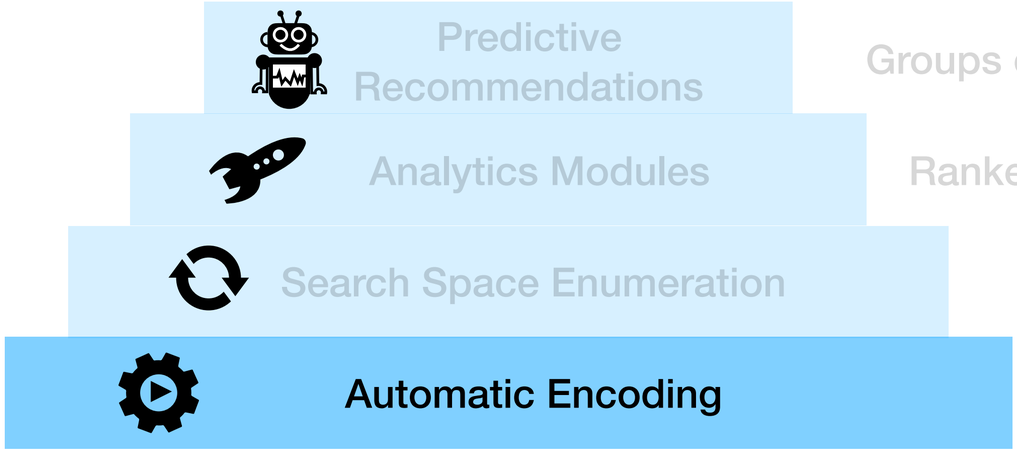




Actions: Hierarchy of Lux capabilities



Actions: Hierarchy of Lux capabilities



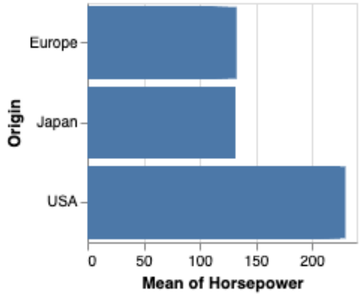
Groups of Visualization Collections

Ranked Visualizations Collection

Visualizations Collection

Single Visualization

```
import lux
dataset = lux.Dataset("data/cars.csv")
dobj = lux.DataObj(dataset,[lux.Column("Horsepower"),lux.Column("Origin")])
dobj.display()
```




```
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 Horsepower",
    yaxis_title="Origin",
)
fig.show()
```

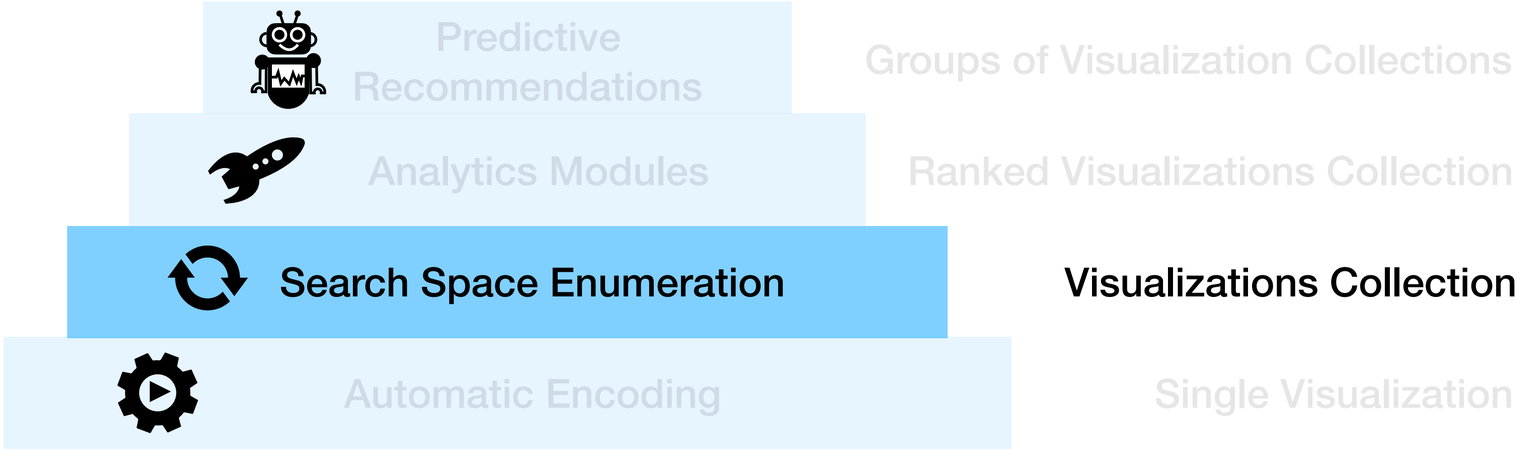
plotly

```
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

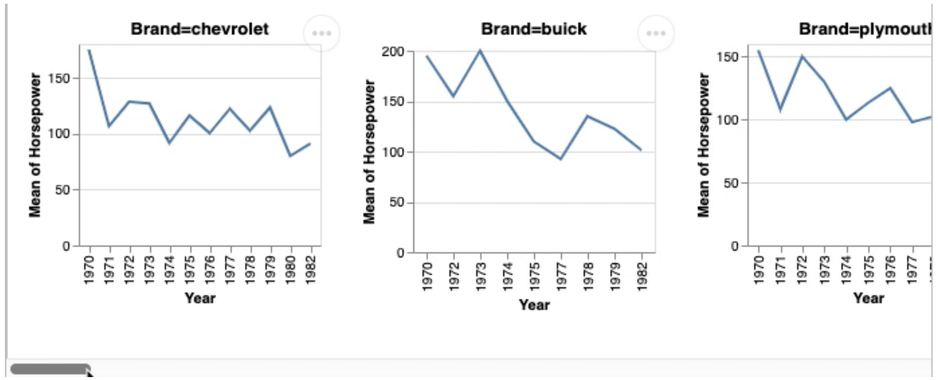
Inspired by Tableau's ShowMe

Actions: Hierarchy of Lux capabilities



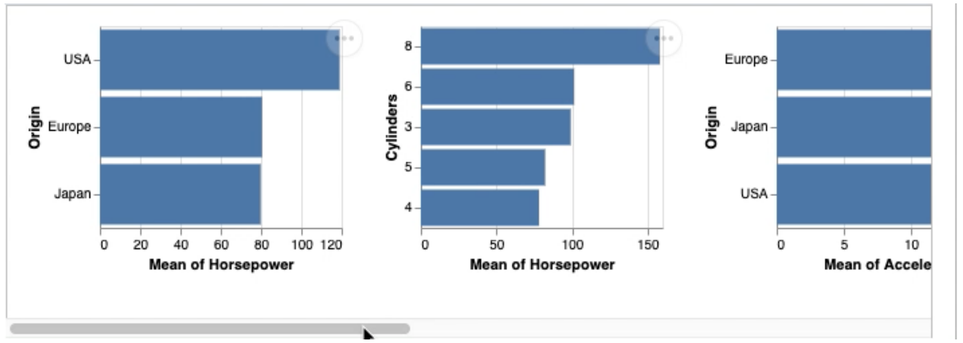
Iterate over filter values

```
dobj = DataObj(dataset,[Column("Horsepower",channel="y"),
                        Column("Year",channel="x"),
                        Row("Brand","?")])
```



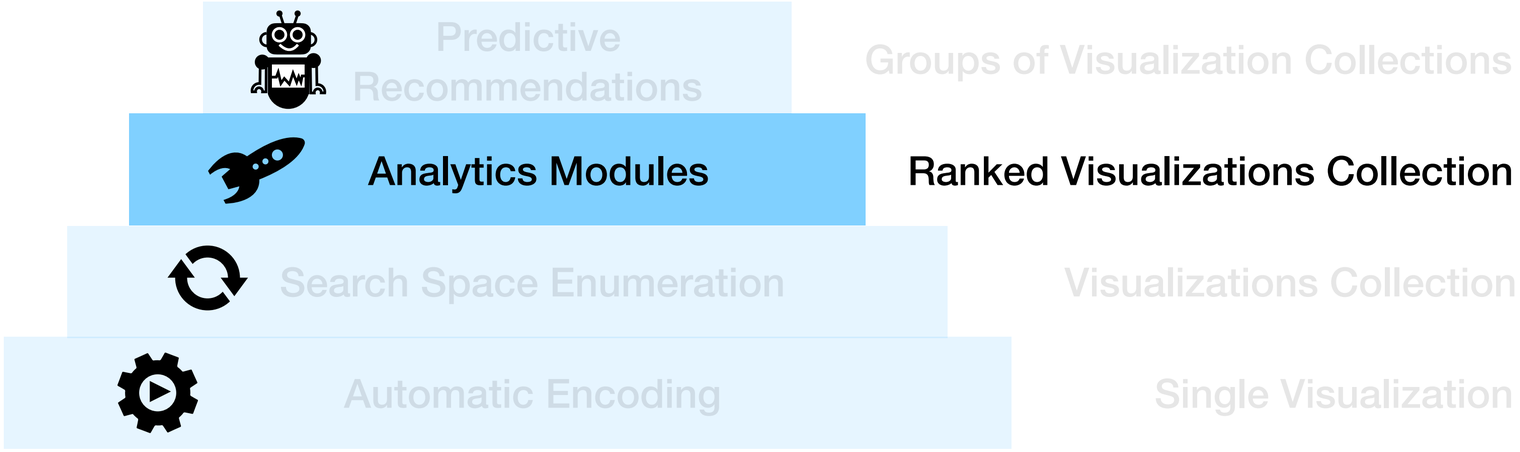
Iterate over attributes

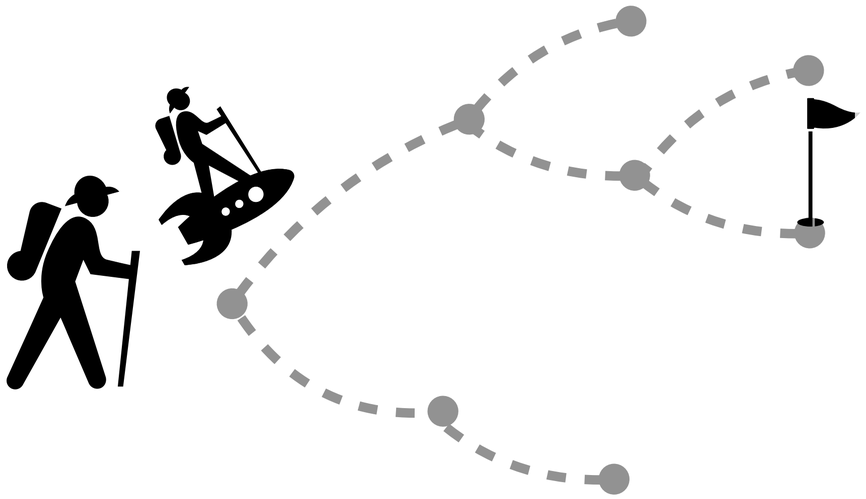
```
dobj = DataObj(dataset,[
Column(["Horsepower","Acceleration","Weight"]),
Column(["Origin","Cylinders"],channel="x")])
```

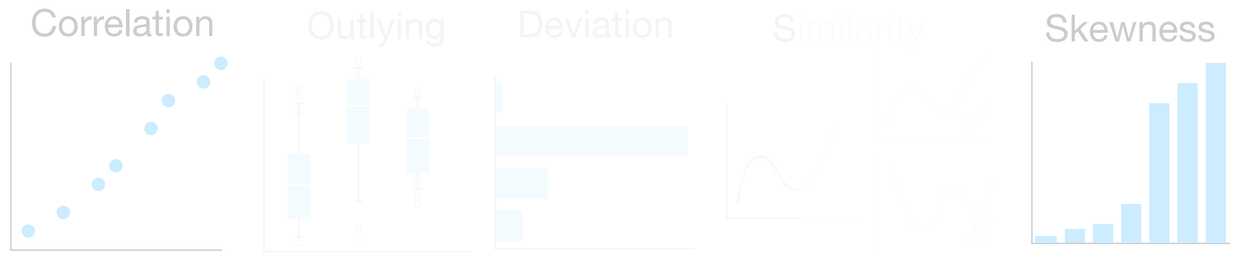


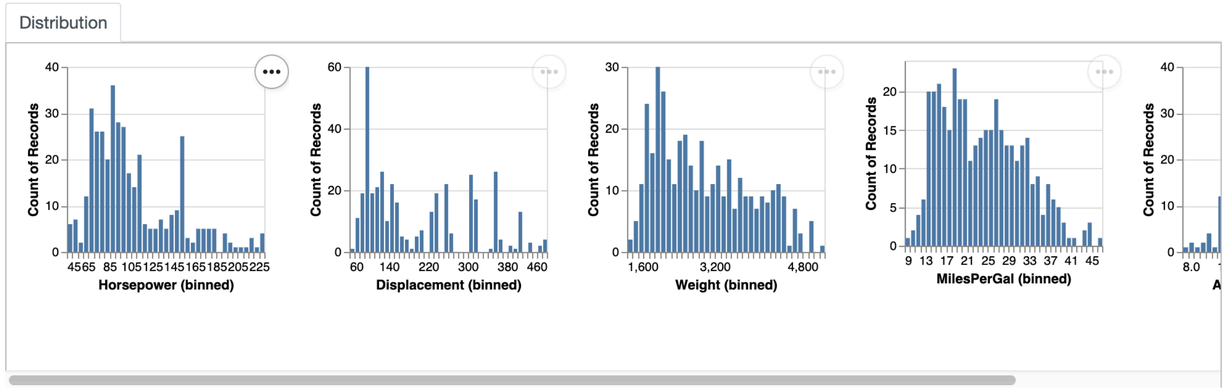
Inspired by Zenvisage's ZQL and Voyager's CompassQL

Actions: Hierarchy of Lux capabilities









Actions: Hierarchy of Lux capabilities

Predictive Recommendations

Groups of Visualization Collections

Analytics Modules

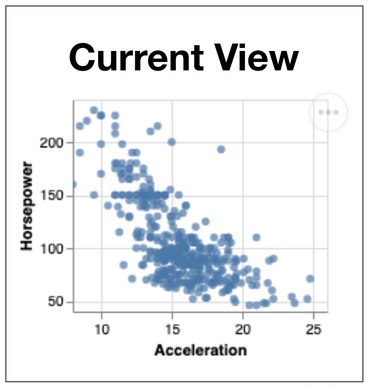
Ranked Visualizations Collection

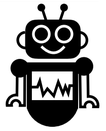
Search Space Enumeration

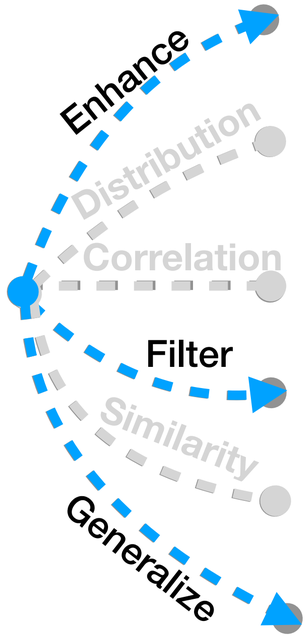
Visualizations Collection

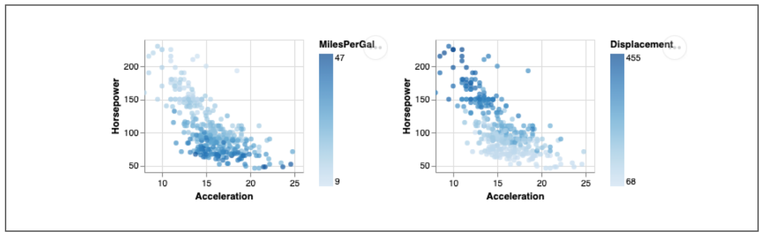
Automatic Encoding

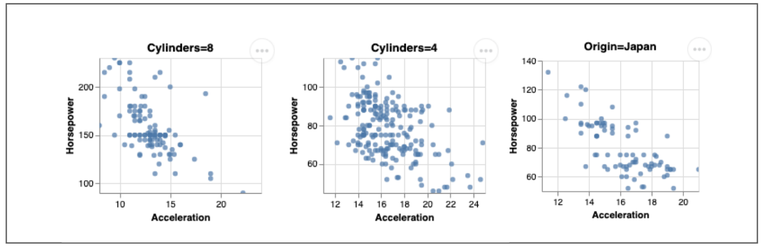
Single Visualization

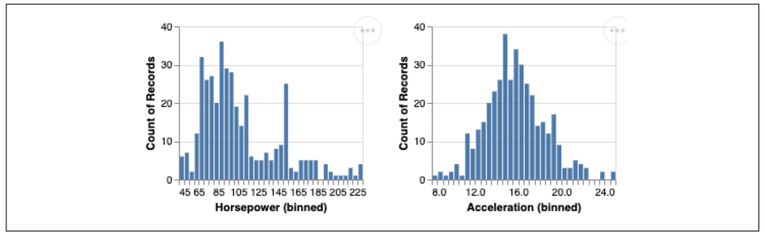




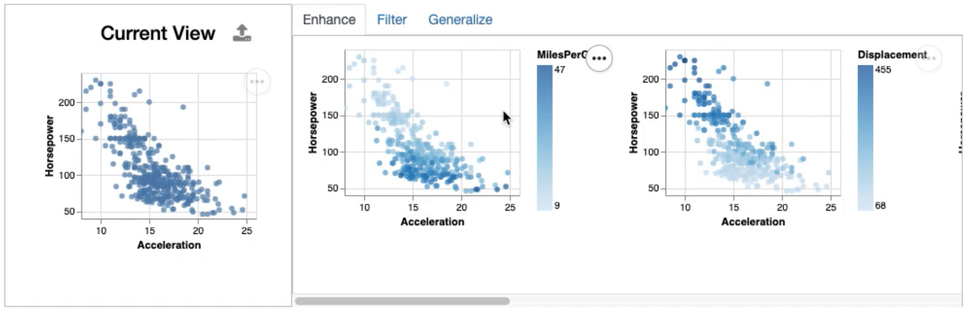









```
dobj = lux.DataObj(dataset,[lux.Column("Acceleration",dataModel="measure"),
                            lux.Column("Horsepower",dataModel="measure")])
result = dobj.showMore()
result
```



Lux: Towards Intelligent Visual Data Discovery

UC Berkeley



Tableau Research


University of Illinois


github.com/lux-org/lux

- Expressive partial specification of data aspects
- In-situ browsing of visualization collection
- Analytics modules accelerating discovery
- Predictive “next-steps” in user’s exploration

riselab UC Berkeley

LUX

Contact:

dorislee@berkeley.edu