Supporting Effective Data Exploration at the Speed of Thought

Doris Jung-Lin Lee (UC Berkeley)
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 2020

Programming Tools

Interactive Interfaces
State of EDA in 2020

Programming Tools

Interactive Interfaces

- Qlik
- Looker
- SAS
- Spotfire®
- Tableau
- SPSS
- TIBCO Software
#1: Disconnect b/w Code + Interactive Tools

**Programming Tools**

```python
# spring boot
import numpy as np
import sklearn.metrics

# ChurnAnalysis
from sklearn.metrics import classification_report, confusion_matrix

def classify(x):
    return np.where(x > 0.5, 1, 0)

# Interactive Interfaces
In [2]: import plotly

# Virtualization
In [3]: def df_data.head():

# Data Analysis
In [4]: df_data = pd.read_csv('churn_data.csv')

# Visualization
In [5]: df_data.head()
#2: Plotting requires LOTS of code + decisions

- How should my visualization look like?
- What type of chart should I use?
- How do I process my data to generate the visualization?

```python
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()
```

```python
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()
```
#3: Trial-and-error is tedious and overwhelming
df.set_intent(["Origin", "Horsepower"])
Basic Walkthrough

To use Lux, simply add the line `import lux`

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

We first load in the Cars dataset with 392 different cars from 1970-1982, which contains information about its Horsepower, MilesPerGal, Acceleration, etc.

```python
In [ ]:
df = pd.read_csv("../lux/data/car.csv")
df["Year"] = pd.to_datetime(df["Year"], format="%Y") # change pandas dtype for the column "Year" to datatype
```

We print out the dataframe, we see the default Pandas display and we can toggle to a set of recommendations generated by Lux. Lux returns four sets of visualizations to show an overview of the dataset.

```python
In [ ]:
df
```

Here we spot this scatterplot visualization Acceleration vs. Horsepower. Intuitively, we expect cars with higher horsepower to have 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. Using the set_intent function, we indicate to Lux that we are interested in the attributes Acceleration and Horsepower.

```python
In [ ]:
df.set_intent(['Acceleration', 'Horsepower'])
df
```

On the left, we see that the Current Visualization corresponds to our specified intent. On the right, we see different tabs of recommendations:

- Enhance shows what happens when we add an additional variable to the current selection
- Then, we have the Filter tab, which adds an additional filter, while fixing the selected variables on the X Y axes
- Finally, we have Generalize which removes one of the attributes, to display the more general trend.
Data Exploration at the Level of Thought
## Actions: Recommended Visualization of Dataframes

- **High Correlation**
- **Low Correlation**

<table>
<thead>
<tr>
<th>Name</th>
<th>PredominantDegree</th>
<th>HighestDegree</th>
<th>FundingModel</th>
<th>Region</th>
<th>Geography</th>
<th>AdmissionRate</th>
<th>ACTMedian</th>
<th>SATAverage</th>
<th>AverageCost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama A&amp;M University</td>
<td>Bachelor's</td>
<td>Graduate</td>
<td>Public</td>
<td>Southeast</td>
<td>Mid-size City</td>
<td>0.8869</td>
<td>17</td>
<td>823</td>
<td>18988</td>
</tr>
<tr>
<td>University of Alabama at Birmingham</td>
<td>Bachelor's</td>
<td>Graduate</td>
<td>Public</td>
<td>Southeast</td>
<td>Mid-size City</td>
<td>0.89673</td>
<td>25</td>
<td>1146</td>
<td>19900</td>
</tr>
<tr>
<td>University of Alabama in Huntsville</td>
<td>Bachelor's</td>
<td>Graduate</td>
<td>Public</td>
<td>Southeast</td>
<td>Mid-size City</td>
<td>0.8062</td>
<td>26</td>
<td>1180</td>
<td>20306</td>
</tr>
<tr>
<td>Alabama State University</td>
<td>Bachelor's</td>
<td>Graduate</td>
<td>Public</td>
<td>Southeast</td>
<td>Mid-size City</td>
<td>0.5135</td>
<td>17</td>
<td>830</td>
<td>17400</td>
</tr>
<tr>
<td>The University of Alabama</td>
<td>Bachelor's</td>
<td>Graduate</td>
<td>Public</td>
<td>Small</td>
<td>City</td>
<td>0.5655</td>
<td>26</td>
<td>1171</td>
<td>28717</td>
</tr>
</tbody>
</table>

**Correlation**

- Skewed Distribution
- Normal Distribution
- Uneven Distribution
- Even Distribution

**Visualization**

- **Correlation**
- **Distribution**
- **Occurrence**

**Show frequency of occurrence for categorical attributes.**

**Uneven Distribution**

**Even Distribution**

- **Number of Records**
- **Count of Record**
- **Region**
- **Name**

**Toggle Pandas/Lux**

Scroll for 3 more charts
Powerful, Intuitive Intent Specification Language
df.set_intent(["Acceleration","Horsepower"])
Quick, on-demand visualizations

Let’s start by looking at Horsepower!

Vis([“Horsepower”], df)

![Histogram showing binned Horsepower distribution.]

Ok, now add Origin to this.

Vis([“Horsepower”, “Origin”], df)

![Bar chart showing mean Horsepower by Origin.]

Hmm…maybe aggregate by sum instead of average

Vis([“Origin”, lux.Clause(“Horsepower”, aggregation=“sum”)], df)

![Bar chart showing sum of Horsepower by Origin.]

What does other visualizations involving Horsepower look like?

VisList([“?”,”Horsepower”], df)

![List of visualizations related to Horsepower.]

Shows a vis list defined by the client.
Supporting Effective Data Exploration at the Speed of Thought

Sign up as Early User:
tinyurl.com/lux-signup