Introduction to Data Science Programming in Python

J. Hathaway - Data Science Program Chair

Disclaimers

I am focusing on modern tools for data science in Python - Polars over Pandas, Plotly over Matplotlib, and Streamlit over Dash. These modern tools reflect the best of

- declarative programming (task focused programming)
- clean grammar (language abstraction that allows intuitive but complex actions)
- industry respect (all three tools are very popular for the quality and have rapid growth)

Agenda

1. Introduction and Set-up (30 minutes)
2. Polars for data munging (45 minutes)
3. Break (10 minutes)
4. Plotly for data visualization (45 minutes)
5. Streamlit for dashboards (45 minutes)

J. Hathaway

Data Scientists with ~20 years of industry experience and 8 years in Academia. Undergraduate degree in Economics (University of Utah) and a graduate degree in Statistics (BYU).


(https://raw.githubusercontent.com/hathawayj/ghana_datascience/master/im

Checking our installation

1. Python Installed
2. VS Code Installed
3. Python VS Code Extension Installed
4. Python packages installed.

```
pip install polars[all] plotly streamlit
```

Introduction to Polars and Data Munging (speed)

Polars is a lightning fast DataFrame library/in-memory query engine. Its embarrassingly parallel execution, cache efficient algorithms and expressive API makes it perfect for efficient data wrangling, data pipelines, snappy APIs and so much more. Polars is about as fast as it gets, see the results in the H2O.ai benchmark. Polars Website

Introduction to Polars and Data Munging (declarative API)

Polars functions (Contexts & Expressions) are human readable and they align with standard SQL methods as well as the very popular big data package PySpark.

Contexts

Selection: `df.select([..])`, `df.with_columns([..])` Filtering: `df.filter()` Groupby / Aggregation: `df.groupby(..).agg([..])`

Expressions

```
new_df = df.select(
  pl.col("names").n_unique().alias("unique"),
  pl.approx_unique("names").alias("unique_approx"),
  (pl.col("nrs").sum() + 5).alias("nrs + 5"),
  pl.col("integers").cast(pl.Float32).alias("integers_as_floats"),
  pl.col("animal").str.n_chars().alias("letter_count")
)
```

Polars programming

Now let's practice using Polars with our installation of Python

1. Read data, melt data, and save as `.parquet` (01_read.py)
2. Munge data for visualization (02_munge.py)

Introduction to Data Visualization

Our eyes are drawn to colors and patterns. We can quickly identify red from blue, and squares from circles. Our culture is visual, including everything from art and advertisements to TV and movies. Data visualization is another form of visual art that grabs our interest and keeps our eyes on the message.

Advantages of data visualization:

- Easily sharing information.
- Interactively explore opportunities.
- Visualize patterns and relationships.

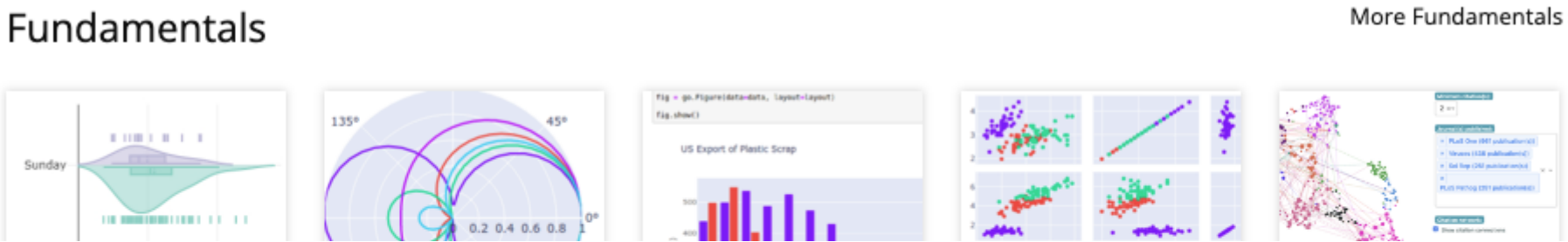
Disadvantages:

- Biased or inaccurate information.
- Correlation doesn't always mean causation.
- Core messages can get lost in

Introduction to **Plotly** for Data Visualization

Built on top of the Plotly JavaScript library (plotly.js), plotly enables Python users to create beautiful interactive web-based visualizations that can be displayed in Jupyter notebooks, saved to standalone HTML files, or served as part of pure Python-built web applications. The plotly Python library is sometimes referred to as "plotly.py" to differentiate it from the JavaScript library.

Plotly.js is built on top of d3.js and stack.gl, Plotly.js is a high-level, declarative charting library. plotly.js ships with over 40 chart types, including 3D charts, statistical graphs, and SVG maps.



Plotly programming

Now let's practice using Plotly with our installation of Python

1. Plotly practice (03_begin_plotly.py)
2. Data visualization with our munged data (03_explore.py)

Introduction to Dashboarding

Introduction to **Streamlit** dashboards

Streamlit turns data scripts into shareable web apps in minutes in pure Python. A faster way to build and share data apps with no front-end experience required.

```
1  import streamlit as st
2
3  st.write("""
4  # My first app
5  Hell|
```

Streamlit programming

Now let's practice using Streamlit with our installation of Python

1. Streamlit practice (04_dashboard.py)