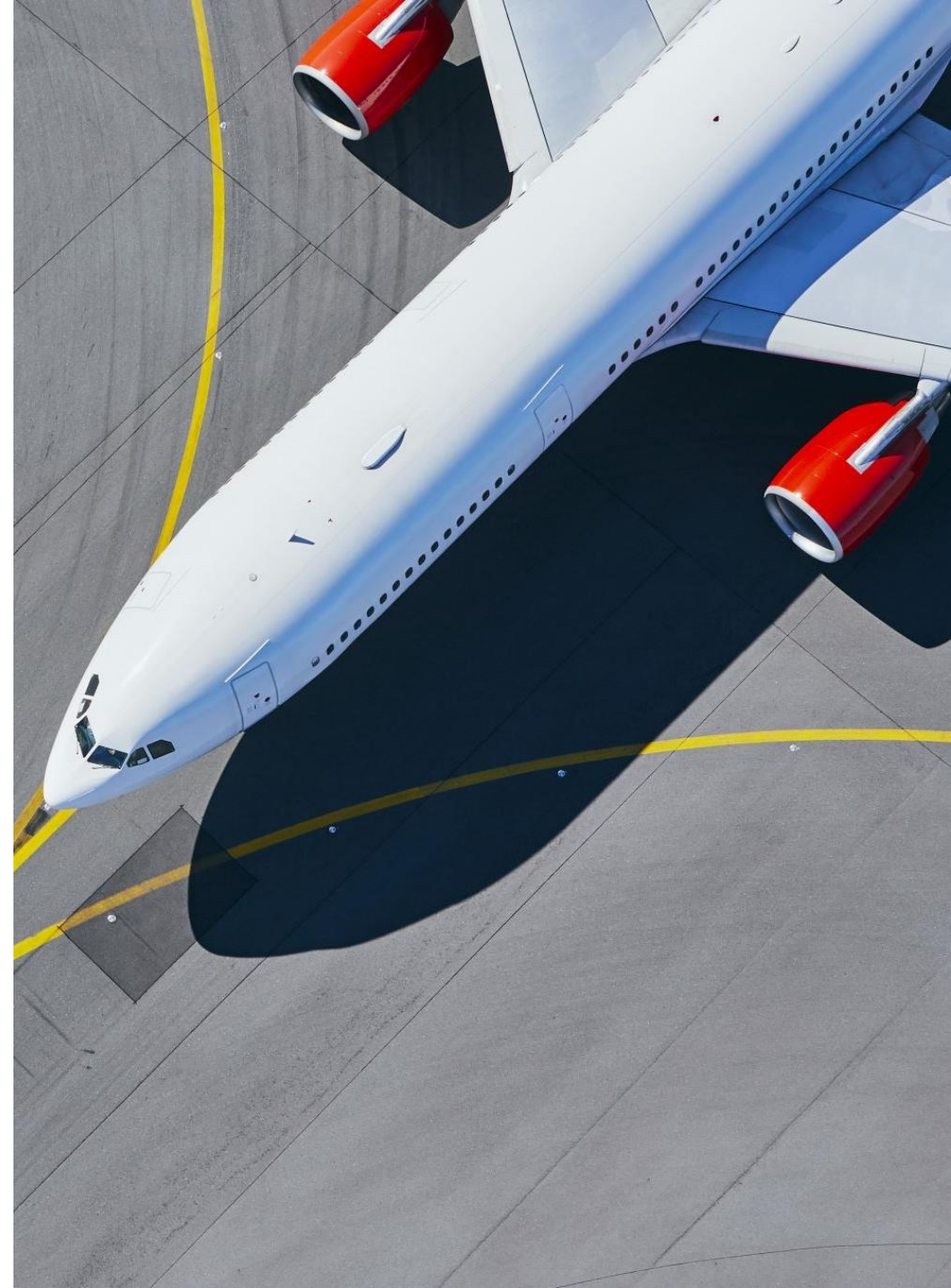


Beat The Crowd

Technology Review

Background and Use Cases

- Flight delays and cancellations can be stressful and uncertain, especially in the context of the Covid-19 pandemic. By analyzing data on airlines and routes, we can identify those with the highest likelihood of providing a smooth travel experience.
- Our approach focuses on two use cases:
 - Providing analysis of domestic US airlines based on arrival/departure delay, denied boarding, and approximate travel fare.
 - Building a data pipeline that extracts raw data from the Bureau of Transportation Statistics and provides it to users in different file formats, enabling them to perform their analysis.
- We will need Python packages such as Seaborn or Plotly to visualize the raw data so that users can draw conclusion easily and understand the underlying pattern from the data.



Python Package Choices

- Name : Seaborn
 - Author : Michael Waskom
 - Summary : Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
-
- Name : Plotly
 - Author : Plotly Technologies Inc.
 - Summary : Plotly's Python graphing library makes interactive, publication-quality graphs. We can make line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axes, polar charts, and bubble charts.

Package Comparison

Plotly

- Interactive visualizations: Plotly allows users to create interactive charts and dashboards that can be embedded in web pages or shared on social media. Users can zoom in and out, hover over data points for more information, and even select subsets of the data to view.
- Wide range of chart types: Plotly offers a wide variety of chart types, including line charts, scatter plots, bar charts, pie charts, and more. Users can also create 3D charts and surface plots.

Seaborn

- Beautiful default styles: Seaborn comes with several default themes that make it easy to create beautiful visualizations without much customization.
- High-level interface: Seaborn provides a high-level interface that allows users to create complex visualizations with just a few lines of code.

Package Comparison

Plotly

- Multi-platform support: Plotly supports multiple programming languages, including Python, R, MATLAB, and JavaScript. It also integrates with popular data analysis libraries like Pandas and NumPy.
- Community-driven development: Plotly has a large and active community of developers who contribute to the library and provide support to users.

Seaborn

- Support for statistical analysis: Seaborn is designed for statistical data visualization and provides several tools for visualizing relationships between variables and exploring distributions of data.
- Integration with Pandas: Seaborn integrates well with the Pandas library, which makes it easy to work with data in a tabular format.

Your Choice

We've chosen to use **Plotly** for this project.

Plotly has stronger capabilities for creating interactive visualizations, while Seaborn offers more robust statistical analysis functionality.

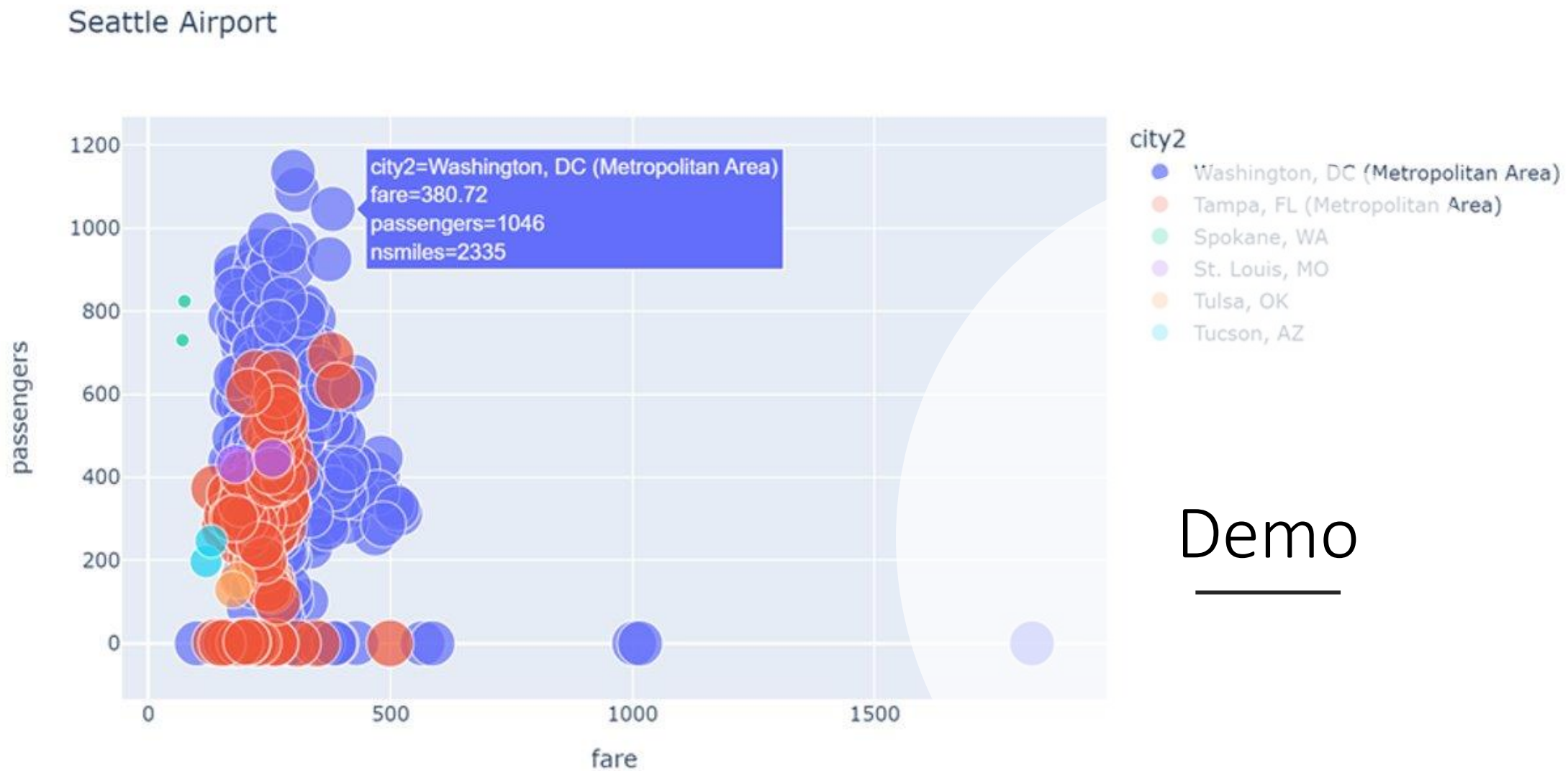
Our goal for this project is to create visualizations that are easy for a user to understand and interact with. We will not be focusing heavily on statistical analysis within these visualizations, so we will not need a high level of statistical functionality to create our graphics.

Because of this, Plotly will better suit our project.

Drawbacks and Remaining Constraints

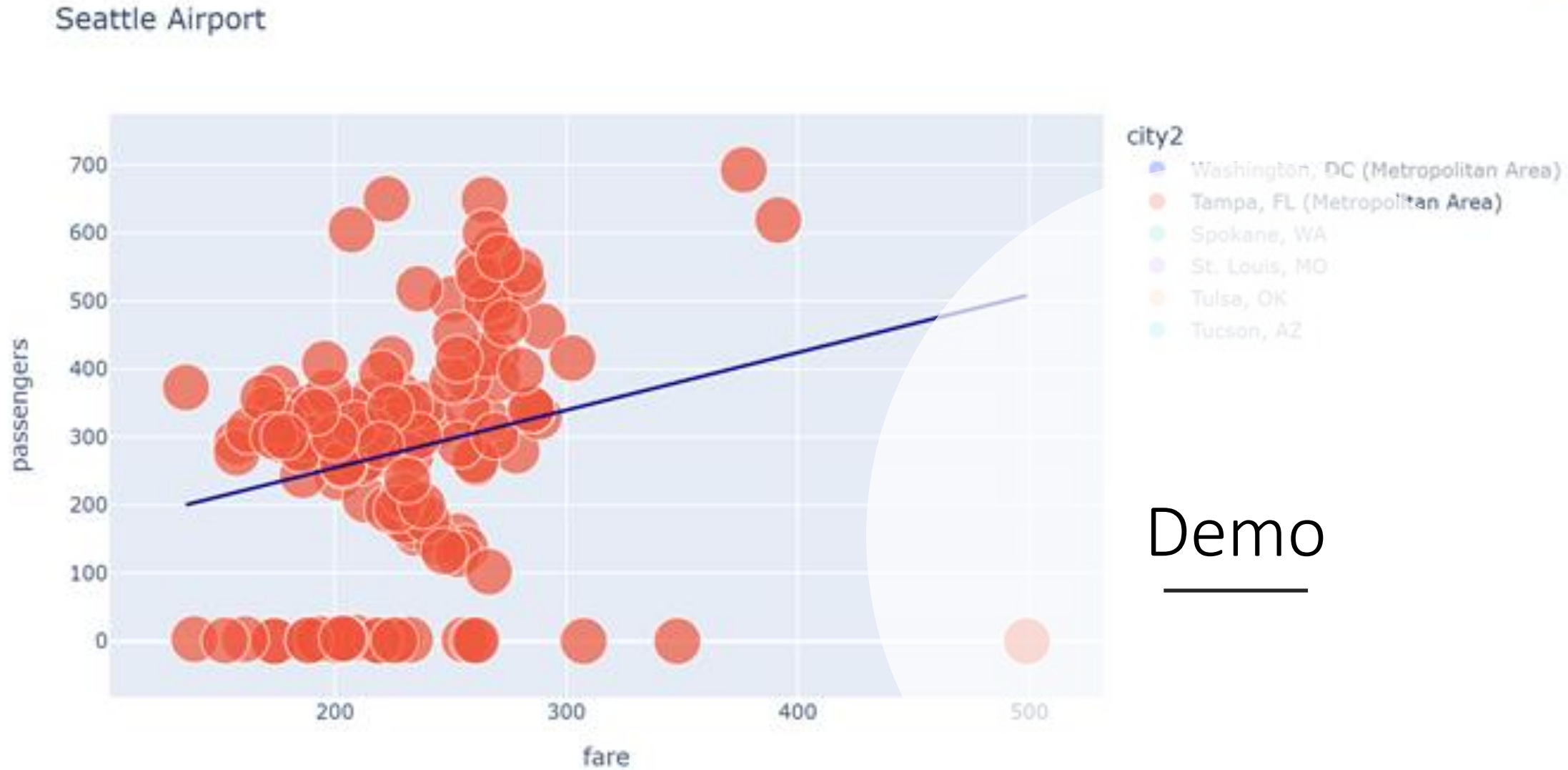
- Drawbacks
 - Small color palette
 - Slow with animations/color
 - Slow with large datasets
- Remaining Constraints:
 - Licensing limitations may impact use in certain applications
 - Integration with other data tools may be limited

```
In [24]: fig = px.scatter(df, x="fare", y="passengers", color="city2", size="nsmiles", hover_data=['nsmiles'], title="Seattle Airport")  
fig.show()
```



Demo


```
In [31]: fig = px.scatter(df, x="fare", y="passengers", color="city2", size="nsmiles", hover_data=['nsmiles'],  
                        trendline='ols', trendline_color_override='darkblue', title="Seattle Airport")  
fig.show()
```



Appendix

- [go.Sacttergl takes tremendously longer to generate the figure for with large dataset using datetime/timestamp as index · Issue #3357 · plotly/plotly.py \(github.com\)](#)
- [Plotly express bar: slow to setup with animations and/or color · Issue #4071 · plotly/plotly.py \(github.com\)](#)