The University of Arizona

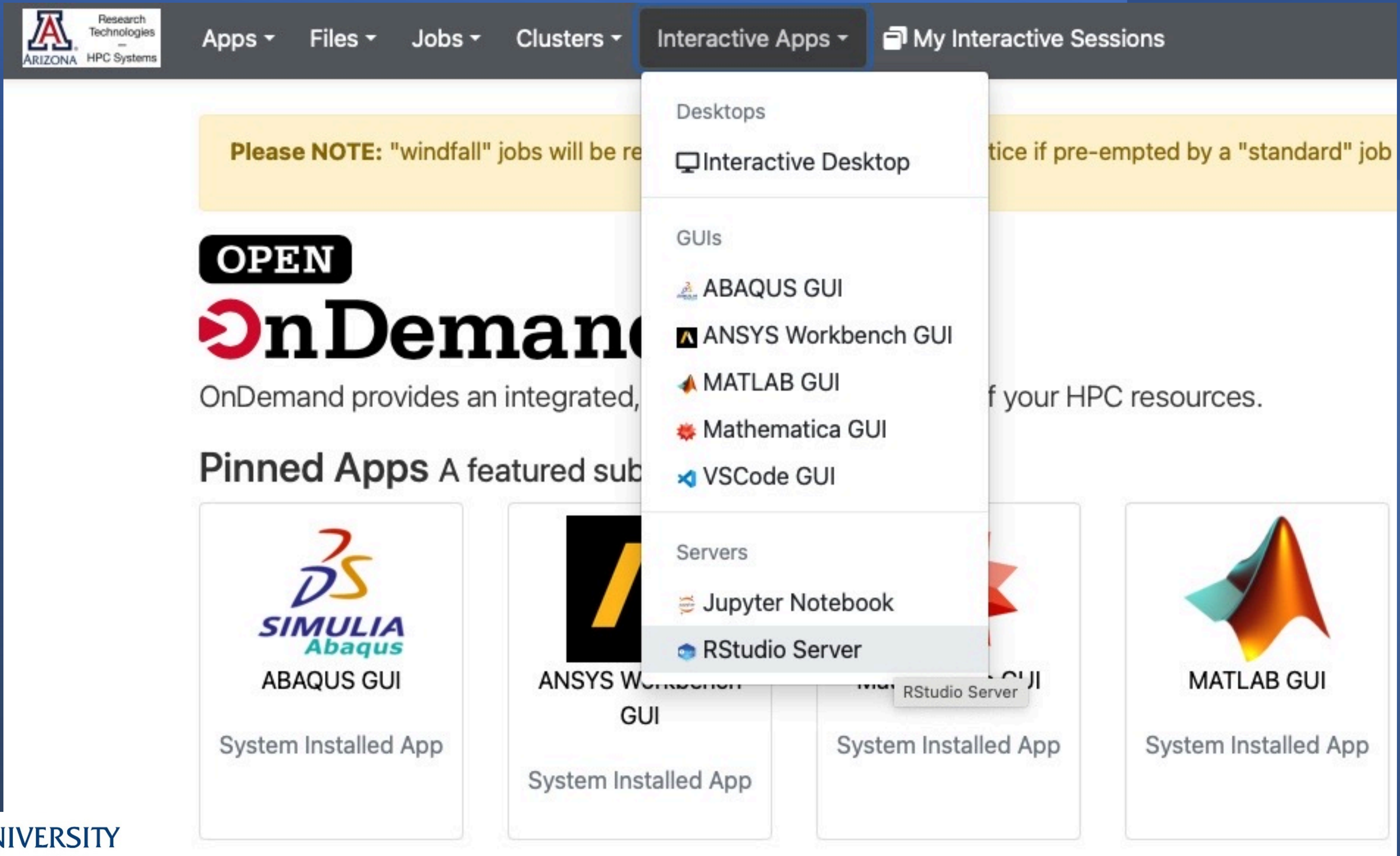
Intro to Visualization on HPC

2024

Using HPC

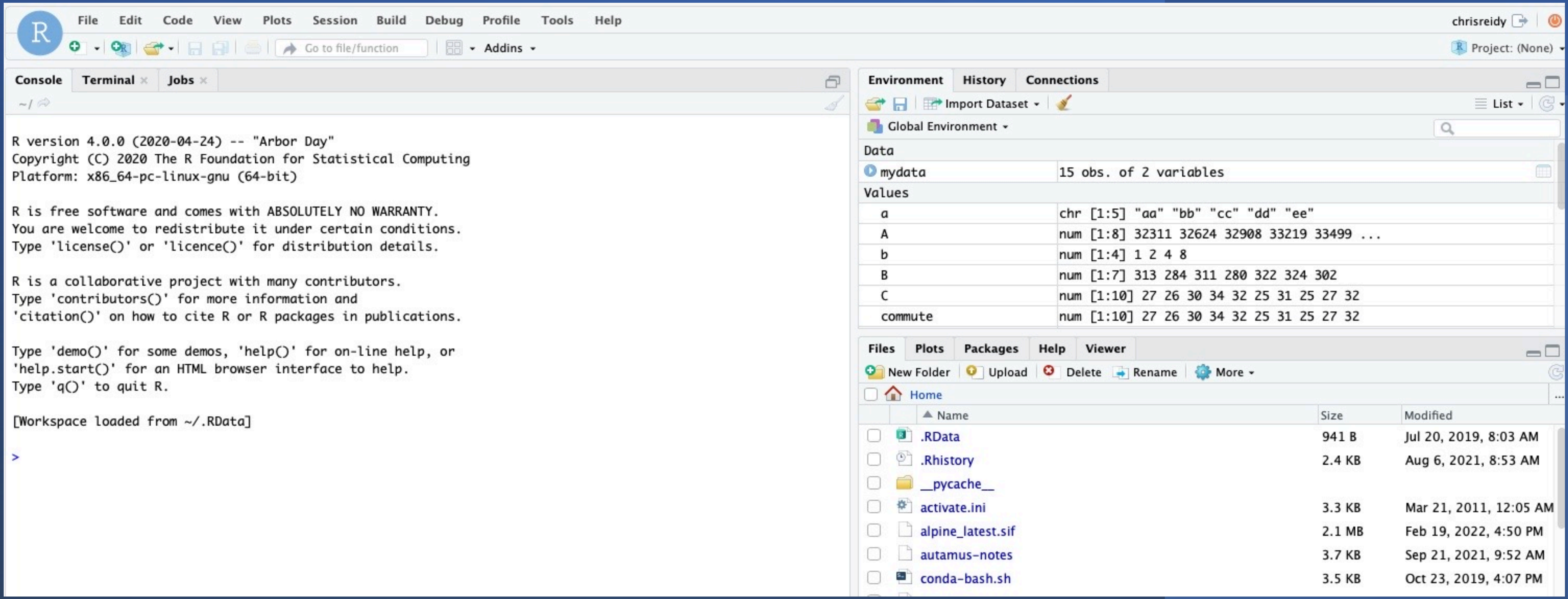
Need some introductory slides about visualization

Using RStudio on HPC



THE UNIVERSITY OF ARIZONA

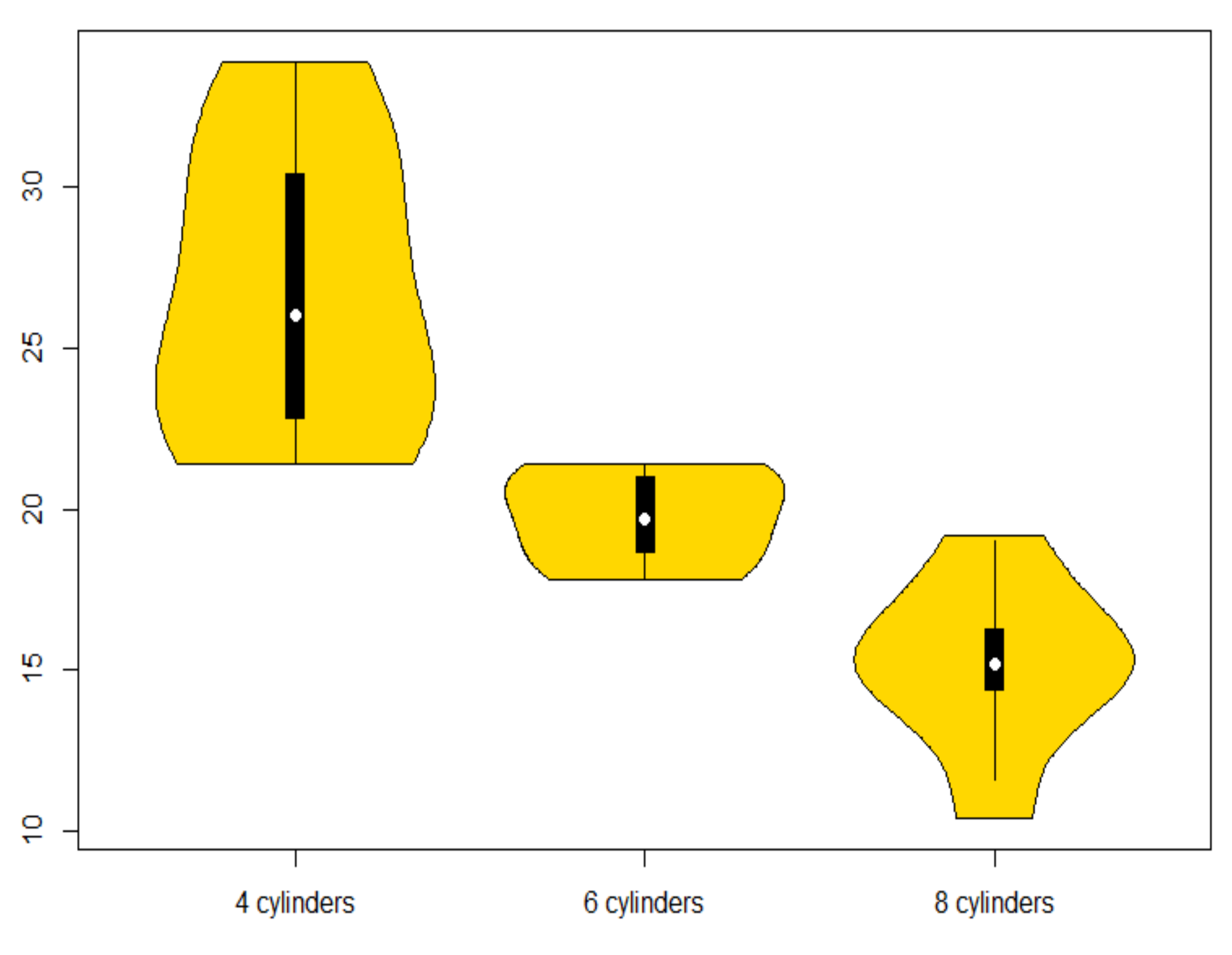
Using RStudio on HPC



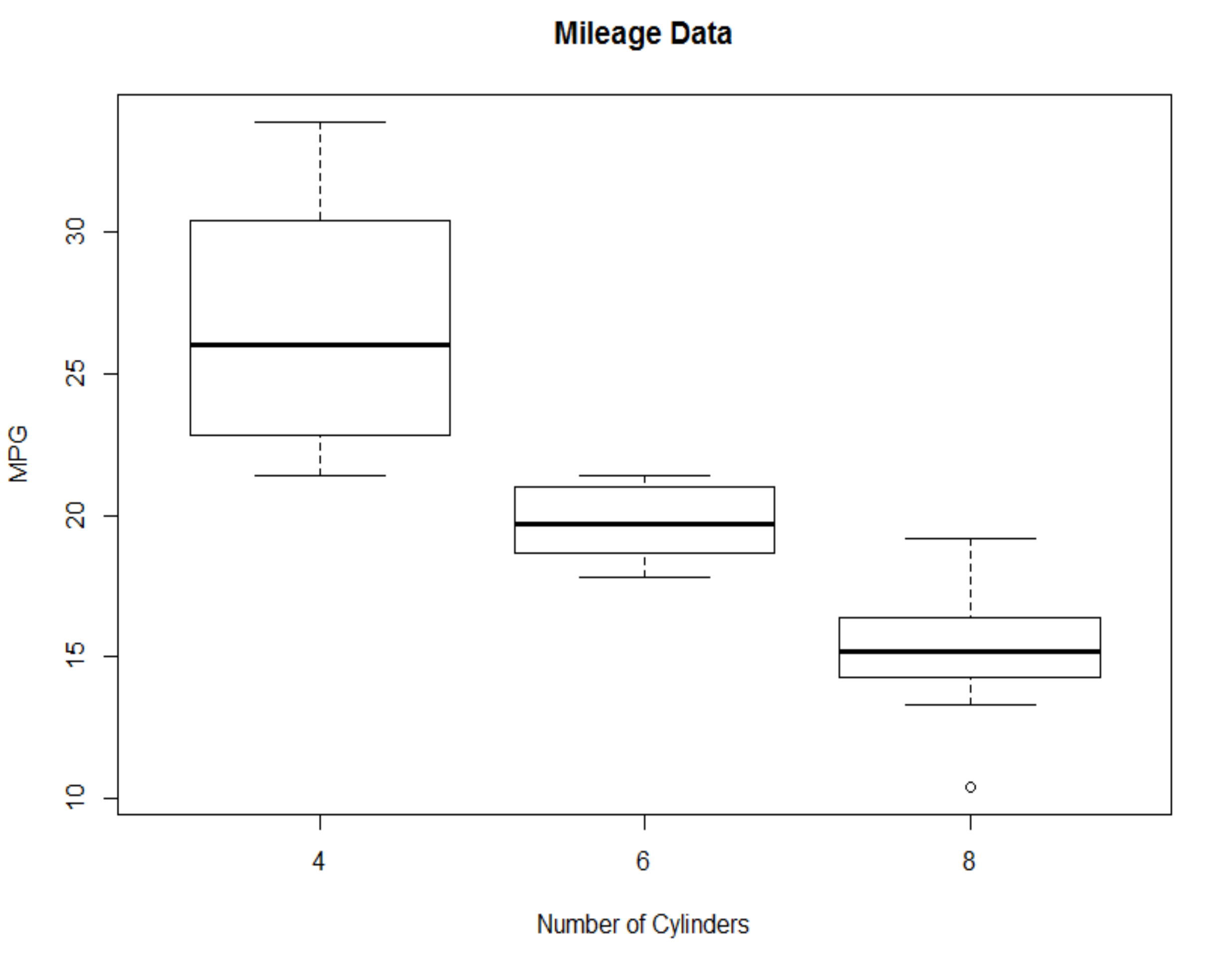
THE UNIVERSITY OF ARIZONA

Using RStudio on HPC

Violin Plot



Box Plot

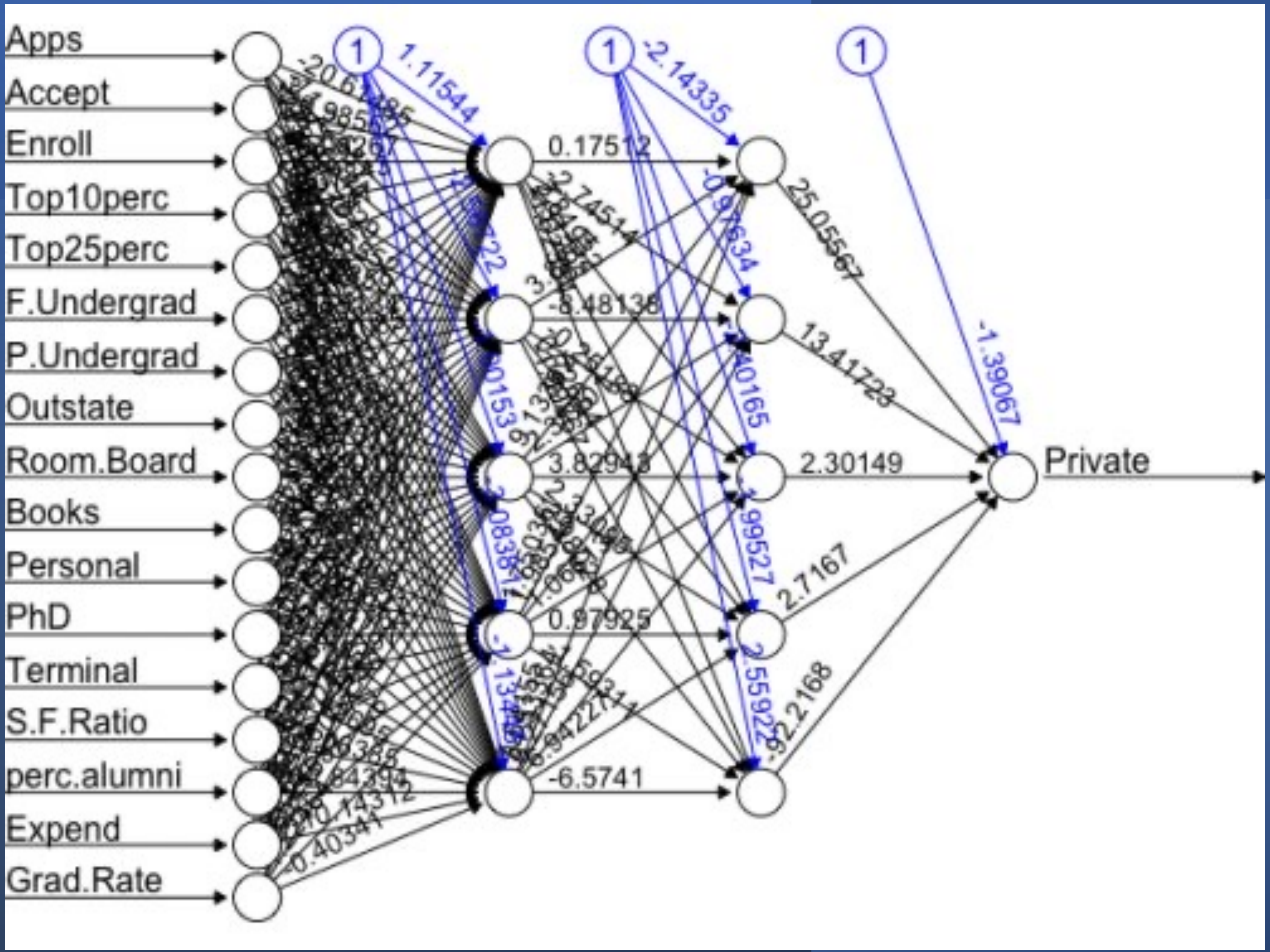


THE UNIVERSITY OF ARIZONA

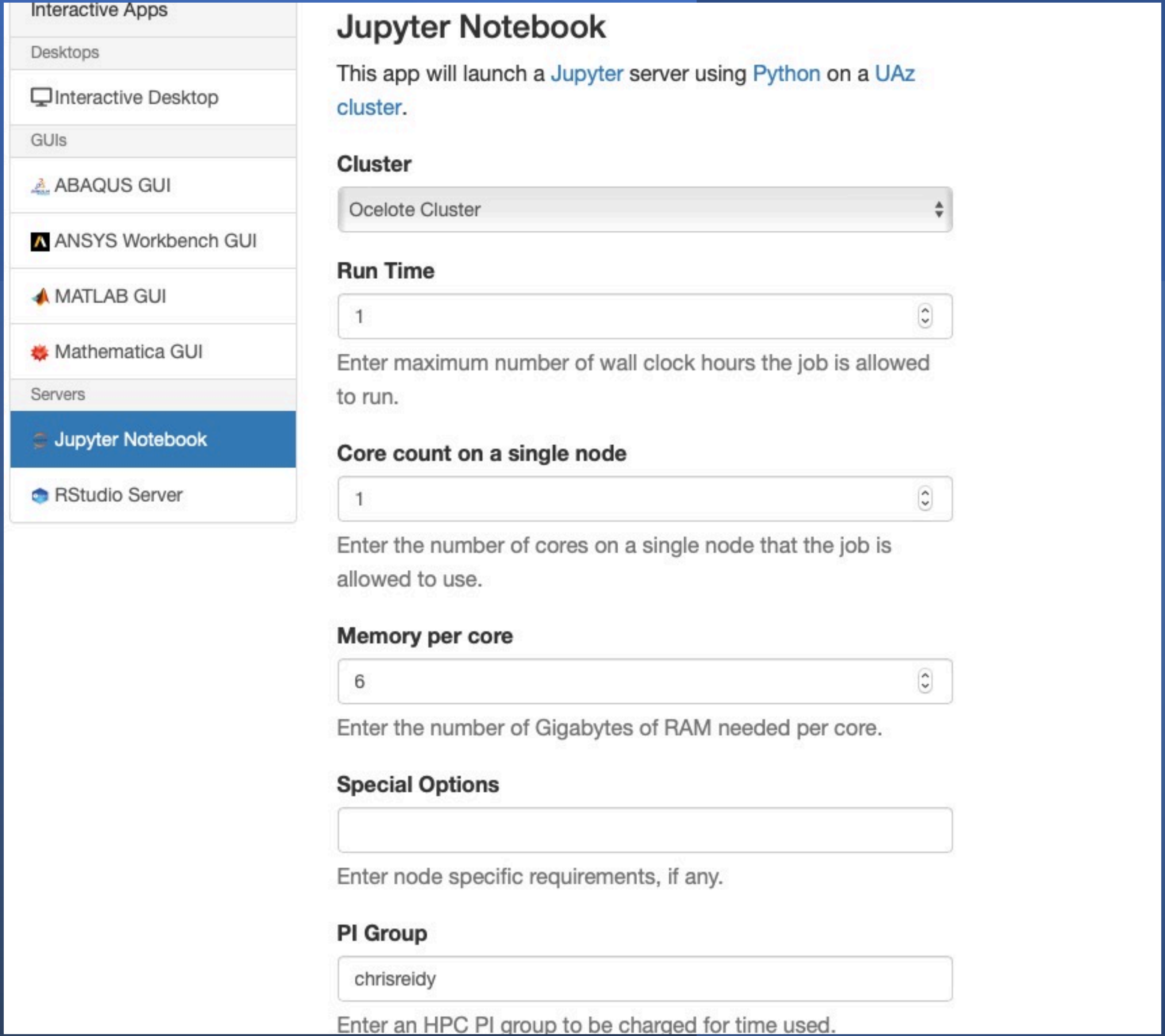
Using RStudio on HPC

Neural Network

We can visualize a Neural Network by using the plot(nn) command. The black lines represent the weighted vectors between the neurons. The blue line represents the bias added.



Using Jupyter Notebooks on HPC



THE UNIVERSITY OF ARIZONA

Using Jupyter Notebooks on HPC

jupyter ML-HPC Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code

In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
import matplotlib.pyplot as plt
```

In [4]:
```
from sklearn.linear_model import LinearRegression
```

In [5]:
```
from sklearn.model_selection import train_test_split
```

In [6]:
```
# Load dat and view the first 5 rows
data = pd.read_excel("king_county_house_data.xls")
```

In [7]:
```
data.head(5)
```

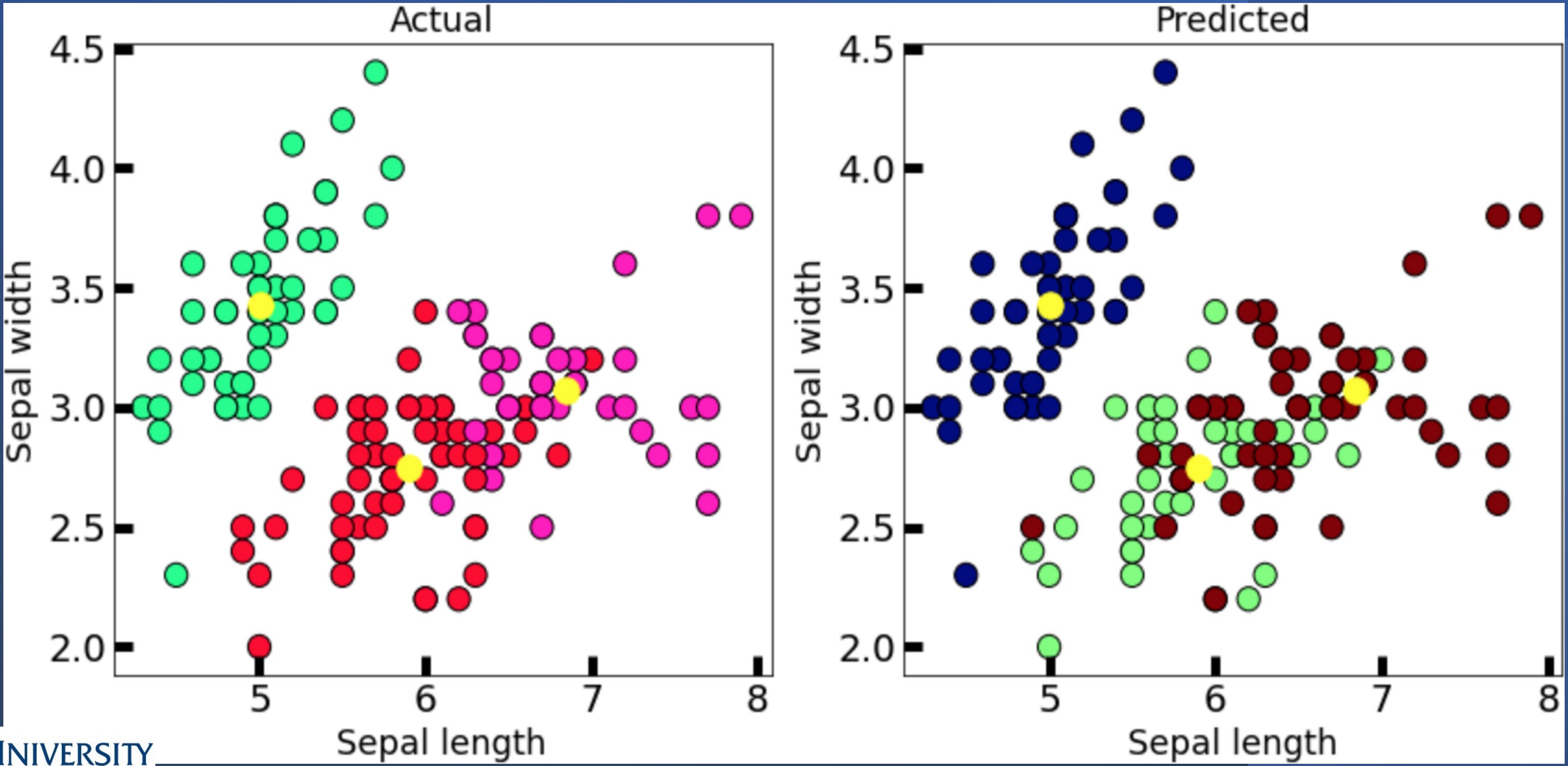
Out[7]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft
0	7129300520	20141013T000000	221900	3	1.00	1180	5650	1.0	0	0	...	7	
1	6414100192	20141209T000000	538000	3	2.25	2570	7242	2.0	0	0	...	7	
2	5631500400	20150225T000000	180000	2	1.00	770	10000	1.0	0	0	...	6	
3	2487200875	20141209T000000	604000	4	3.00	1960	5000	1.0	0	0	...	7	
4	1954400510	20150218T000000	510000	3	2.00	1680	8080	1.0	0	0	...	8	

5 rows × 21 columns

Using Jupyter Notebooks on HPC

Visualizing the Iris Database with matplotlib



THE UNIVERSITY OF ARIZONA

Visualization with Python

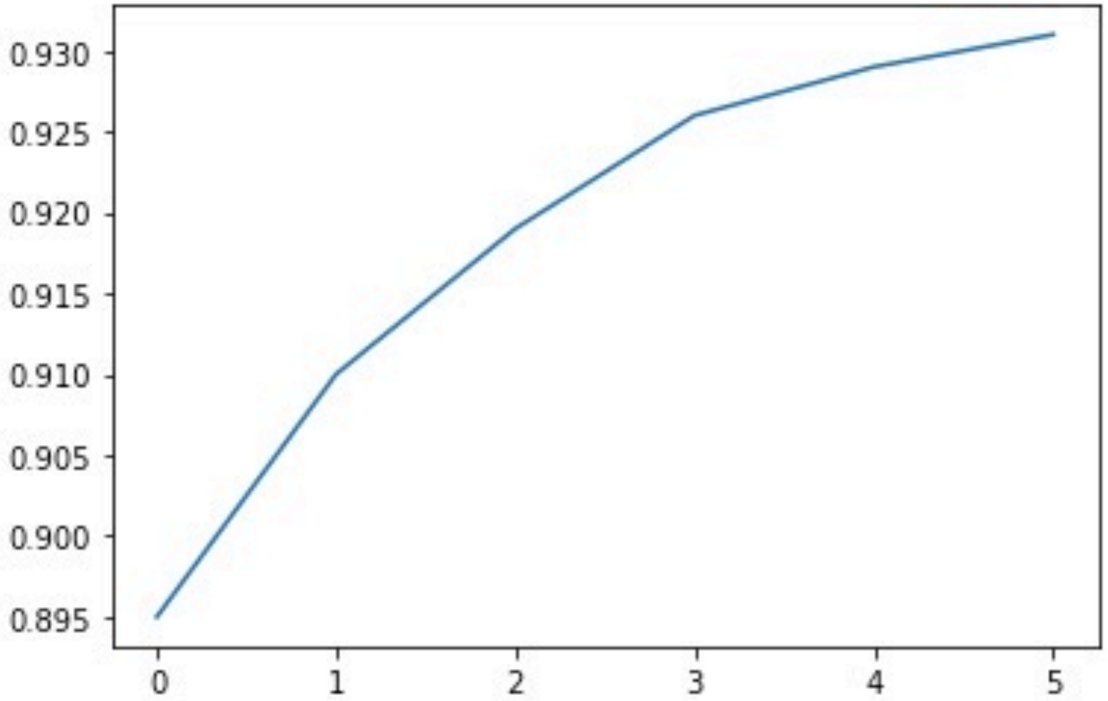
Matplotlib	Seaborn
Used for basic graph plotting like line, bar or pie charts	Mainly used for statistics and performs complex viz with fewer commands
Mainly works with datasets and arrays	Works with entire datasets
Acts productively with data arrays and frames	More organized and functional
Pairs well with Pandas and Numpy	More inbuilt themes

Python using matplotlib and seaborn

```
In [1]: import matplotlib.pyplot as plt
        import seaborn as sns

In [2]: yield_apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931]
        plt. plot(yield_apples)

Out[2]: [<matplotlib.lines.Line2D at 0x7f0f271e6520>]
```

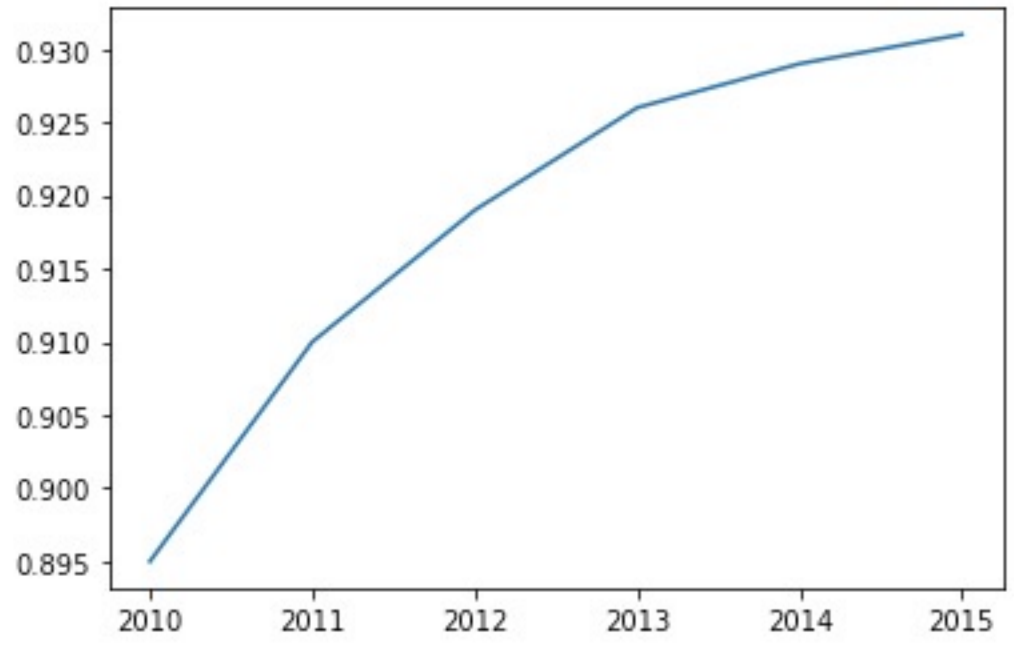


Plotting apple yield

Python: Line Charts

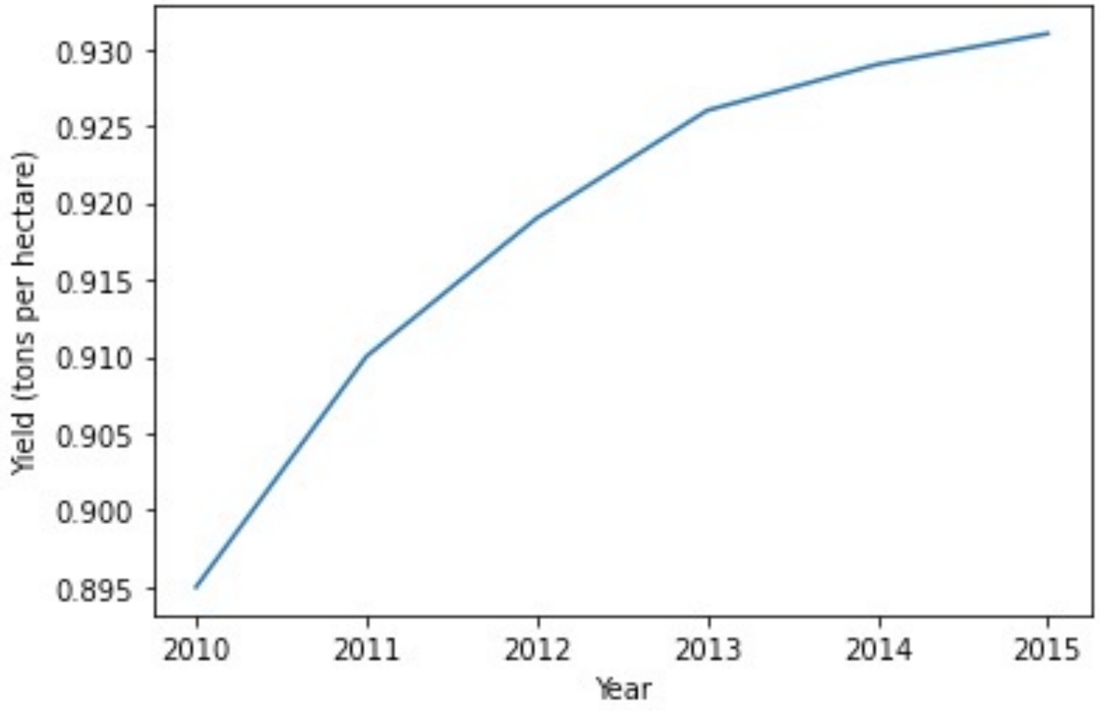
```python
years = [2010, 2011, 2012, 2013, 2014, 2015]
yield_apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931]
plt. plot (years, yield_apples)
```

```
[<matplotlib.lines.Line2D at 0x7f0f2506e340>]
```



Add x axis values

```python
plt. plot (years, yield_apples)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)');
```

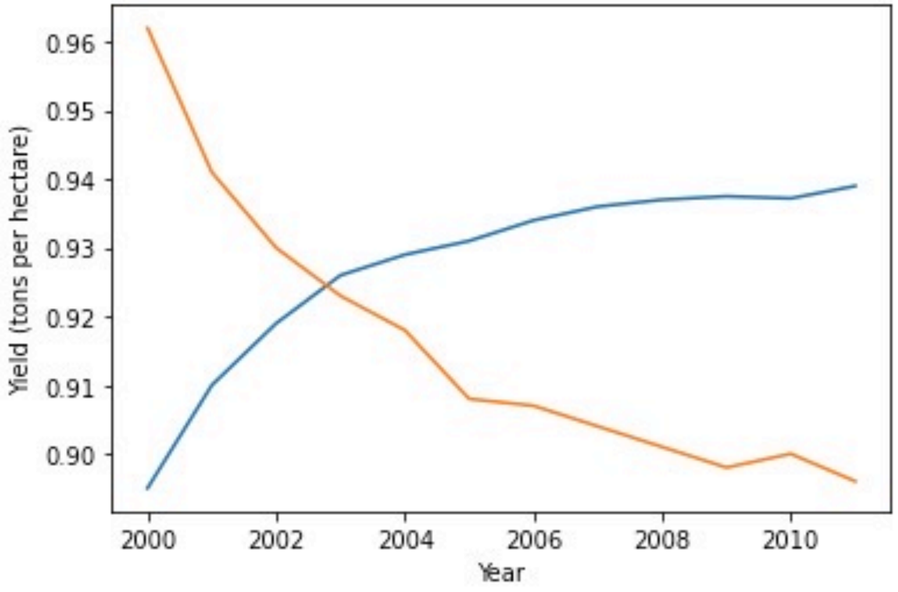


Add labels to the axes

Python: Line Charts

```
In [6]: years = range (2000, 2012)
        apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931, 0.934, 0.936, 0.937, 0.9375, 0.9372, 0.939]
        oranges = [0.962, 0.941, 0.930, 0.923, 0.918, 0.908, 0.907, 0.904, 0.901, 0.898, 0.9, 0.896]
```

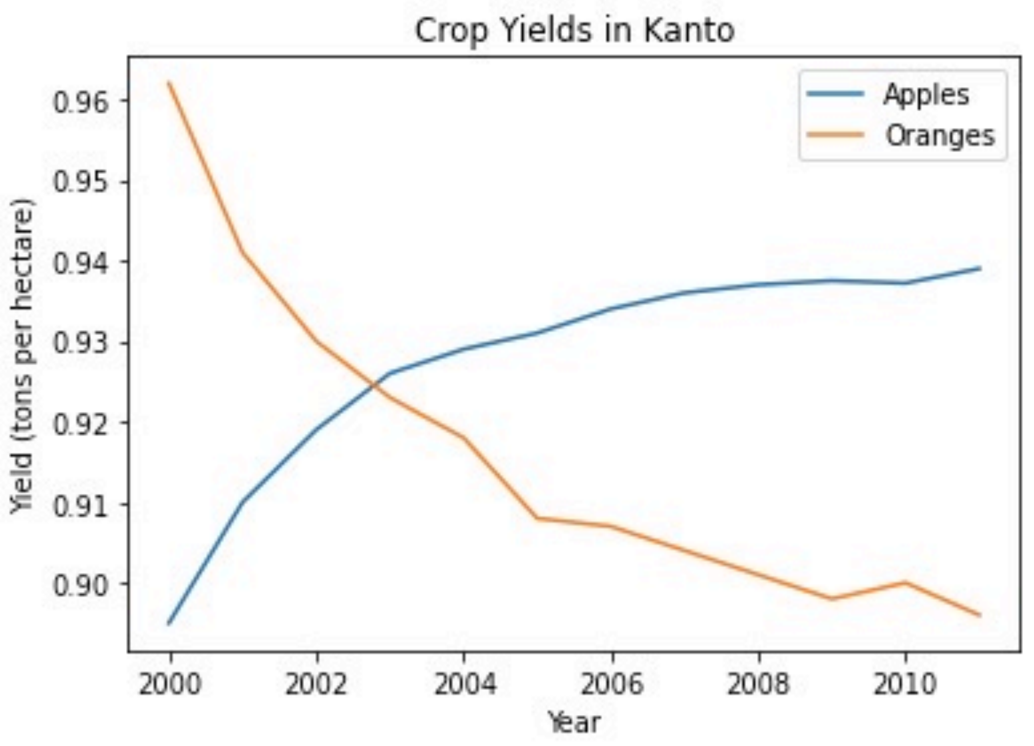
```
In [7]: plt.plot(years, apples)
        plt.plot(years, oranges)
        plt.xlabel('Year')
        plt.ylabel('Yield (tons per hectare)');
```



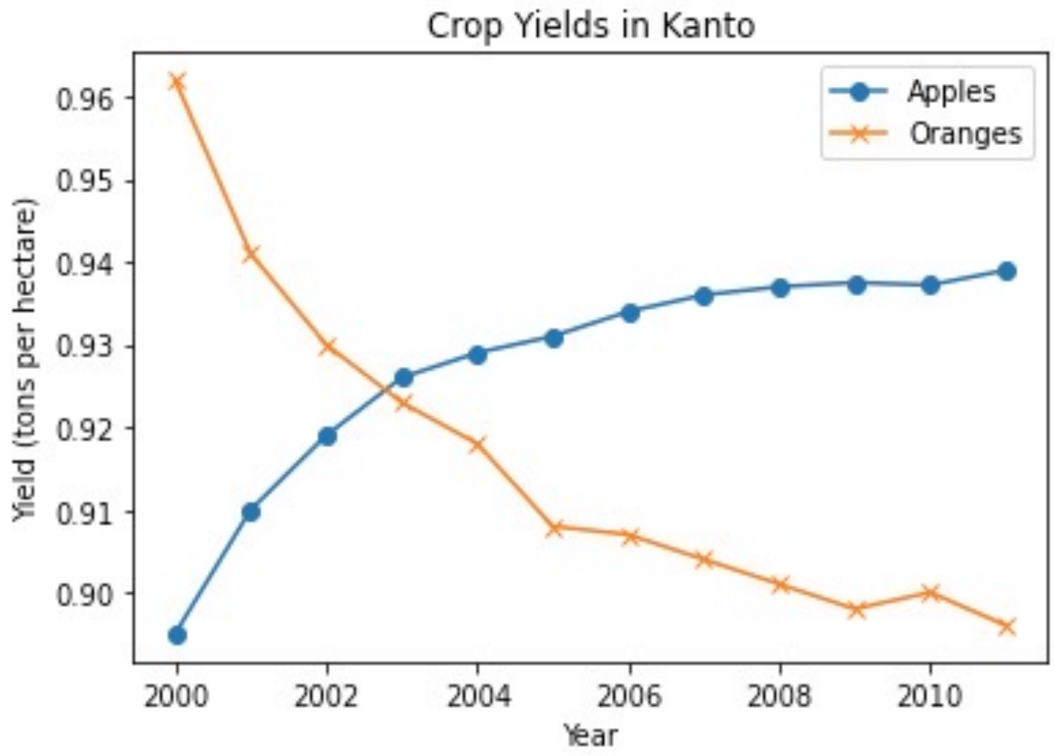
Plotting multiple datasets on the same graph

Python: Line Charts

```
plt.plot(years, apples)
plt.plot(years, oranges)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in Kanto")
plt.legend (['Apples', 'Oranges']);
```




```
plt.plot(years, apples, marker='o')
plt.plot(years, oranges, marker='x')
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in Kanto")
plt.legend (['Apples', 'Oranges']);
```

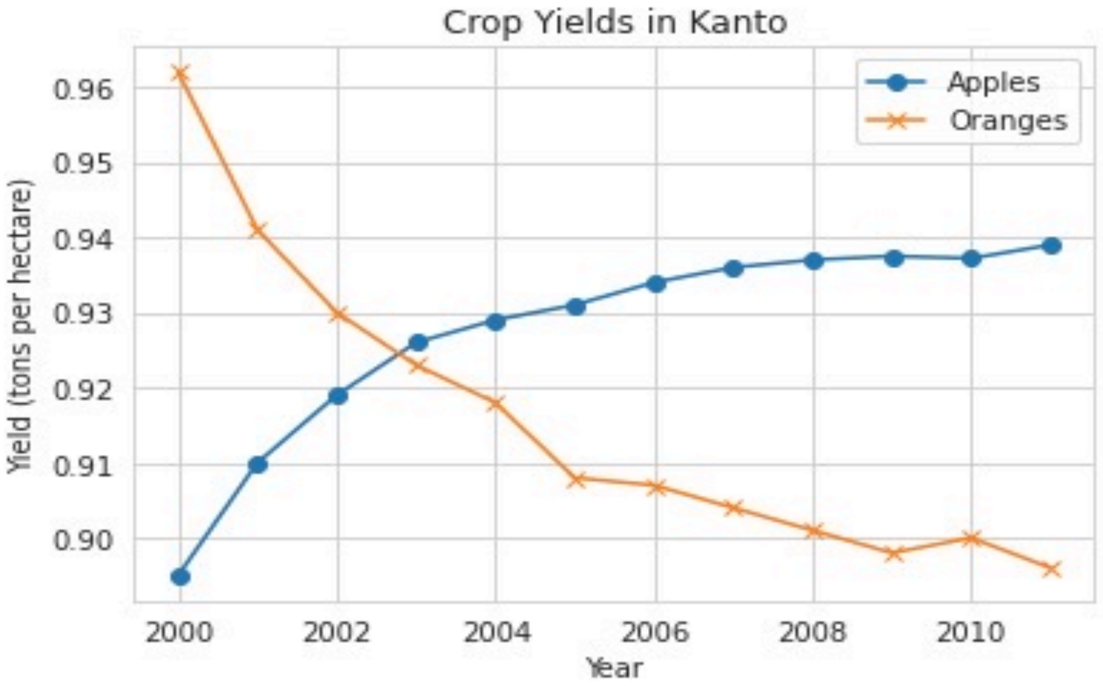


Adding labels and markers

Python: Line Charts

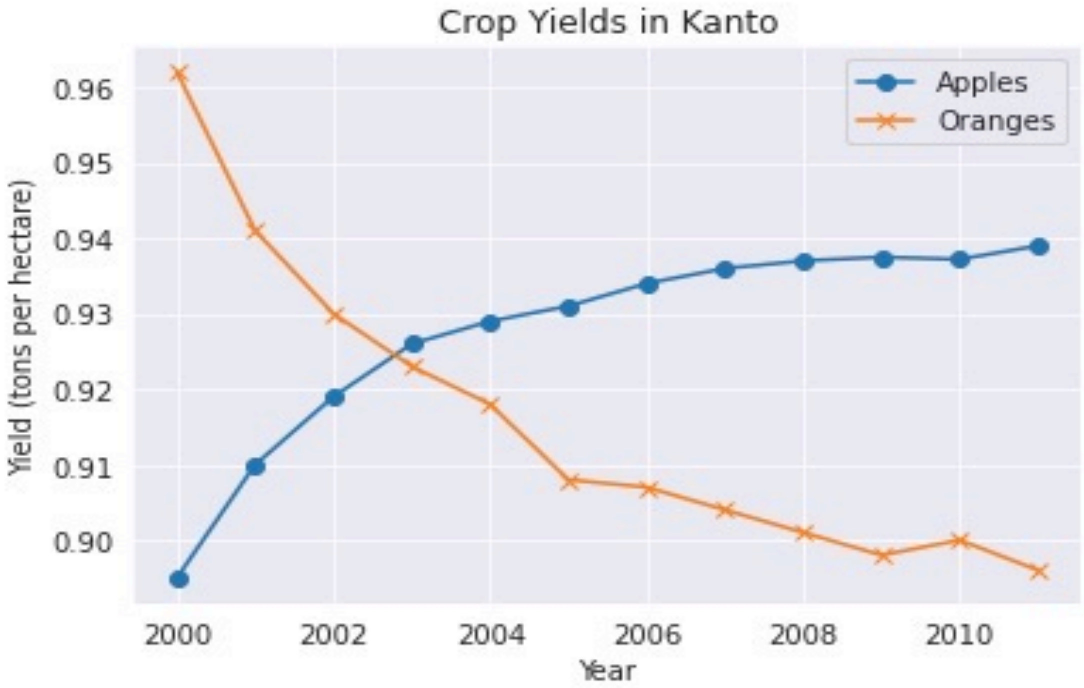
```
sns.set_style("whitegrid")
```

```
plt.plot(years, apples, marker='o')
plt.plot(years, oranges, marker='x')
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in Kanto")
plt.legend (['Apples', 'Oranges']);
```




```
sns.set_style("darkgrid")
```

```
plt.plot(years, apples, marker='o')
plt.plot(years, oranges, marker='x')
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in Kanto")
plt.legend (['Apples', 'Oranges']);
```



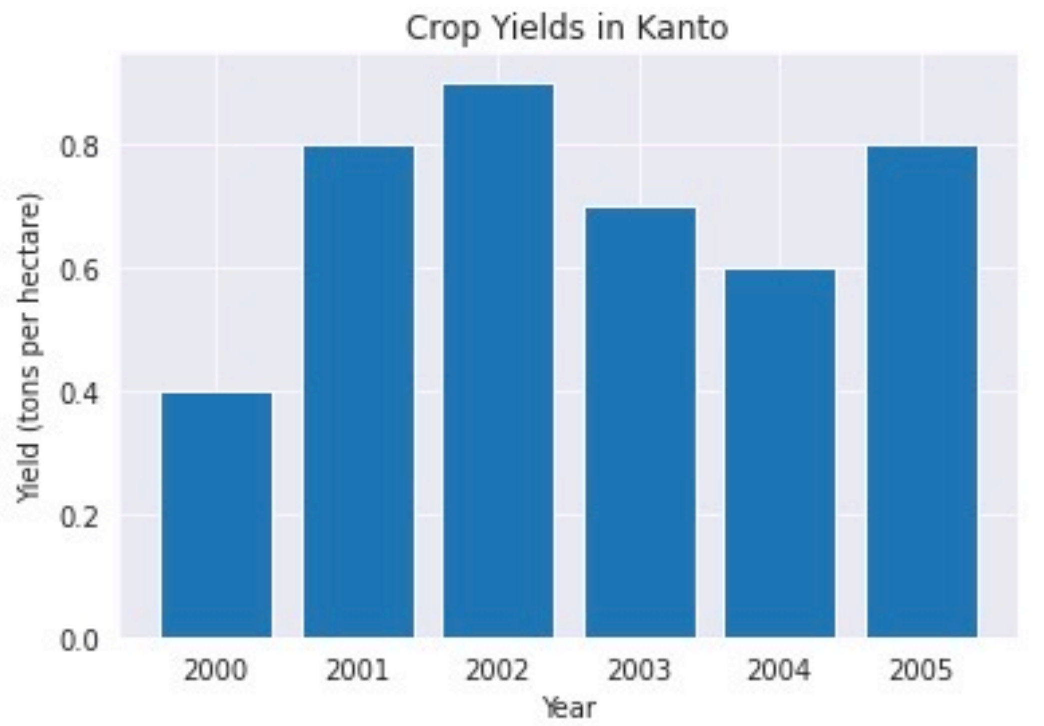
Doing the same with Seaborn

Python: Bar Charts

```python
years = range (2000, 2006)
apples = [0.35, 0.6, 0.9, 0.8, 0.65, 0.8]
oranges = [0.4, 0.8, 0.9, 0.7, 0.6, 0.8]
```

```python
plt.bar (years, oranges)
plt.xlabel('Year')
plt.ylabel('Yield (tons per hectare)')
plt.title("Crop Yields in Kanto")
```

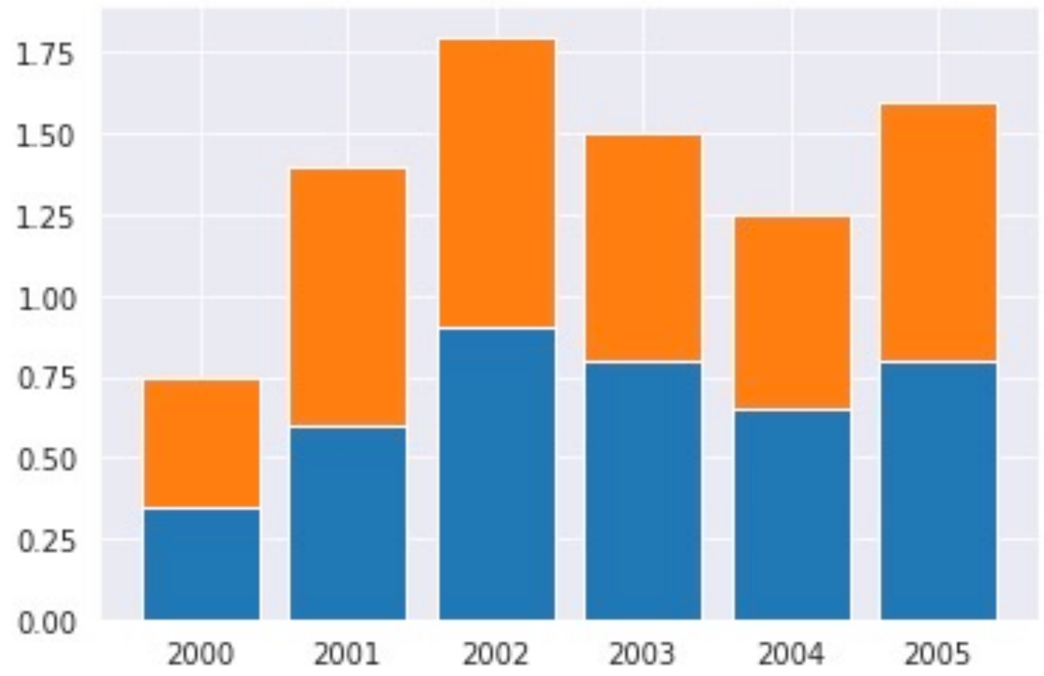
```
Text(0.5, 1.0, 'Crop Yields in Kanto')
```



Bar chart

```python
plt.bar(years, apples)
plt.bar(years, oranges, bottom=apples)
```

```
<BarContainer object of 6 artists>
```



Stacked Bar chart

Python: Histograms

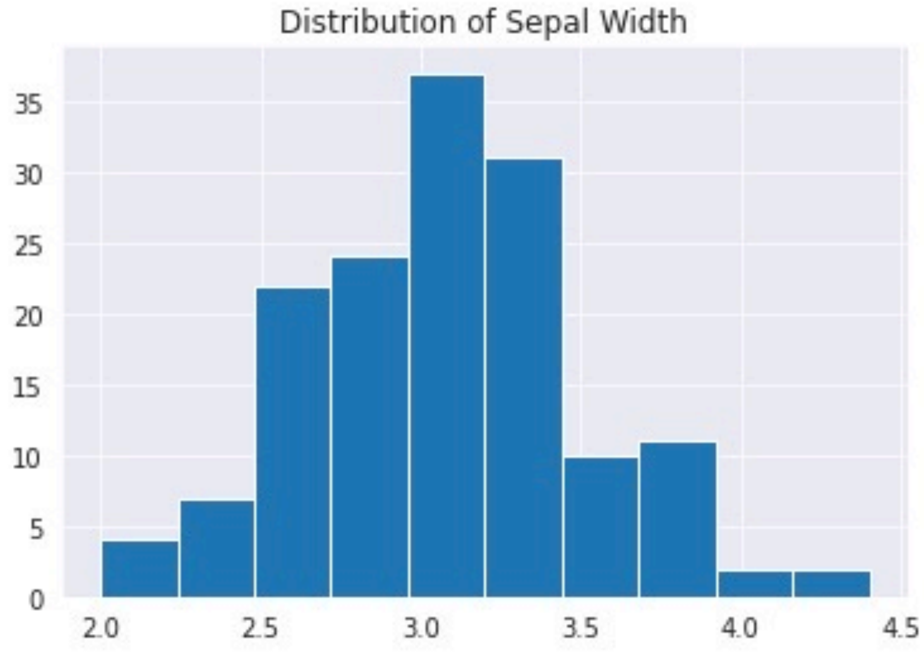
```
flowers_df = sns.load_dataset("iris")
```

```
flowers_df.sepal_width
```

```
0      3.5
1      3.0
2      3.2
3      3.1
4      3.6
      ...
145    3.0
146    2.5
147    3.0
148    3.4
149    3.0
Name: sepal_width, Length: 150, dtype:
```

```
plt.title("Distribution of Sepal Width")
plt.hist(flowers_df.sepal_width)
```

```
(array([ 4.,  7., 22., 24., 37., 31., 10., 11.,  2.,  2.]),
 array([2.  , 2.24, 2.48, 2.72, 2.96, 3.2 , 3.44, 3.68, 3.92, 4.16, 4.4 ]),
 <BarContainer object of 10 artists>)
```

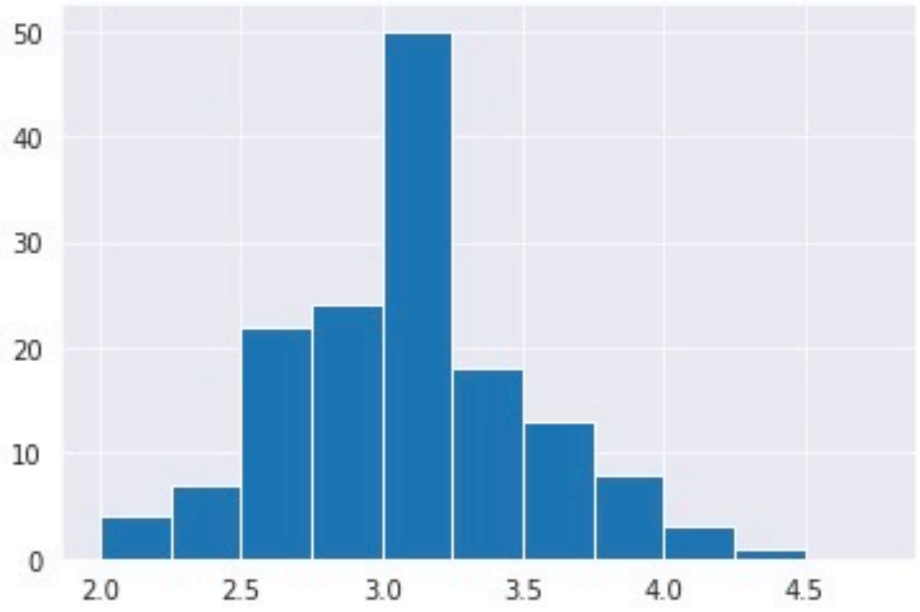


Python: Histograms

```
import numpy as np
```

```
plt.hist(flowers_df.sepal_width, bins=np.arange (2, 5, 0.25))
```

```
(array([ 4.,  7., 22., 24., 50., 18., 13.,  8.,  3.,  1.,  0.]),
 array([2.  , 2.25, 2.5 , 2.75, 3.  , 3.25, 3.5 , 3.75, 4.  , 4.25, 4.5 ,
        4.75]),
 <BarContainer object of 11 artists>)
```

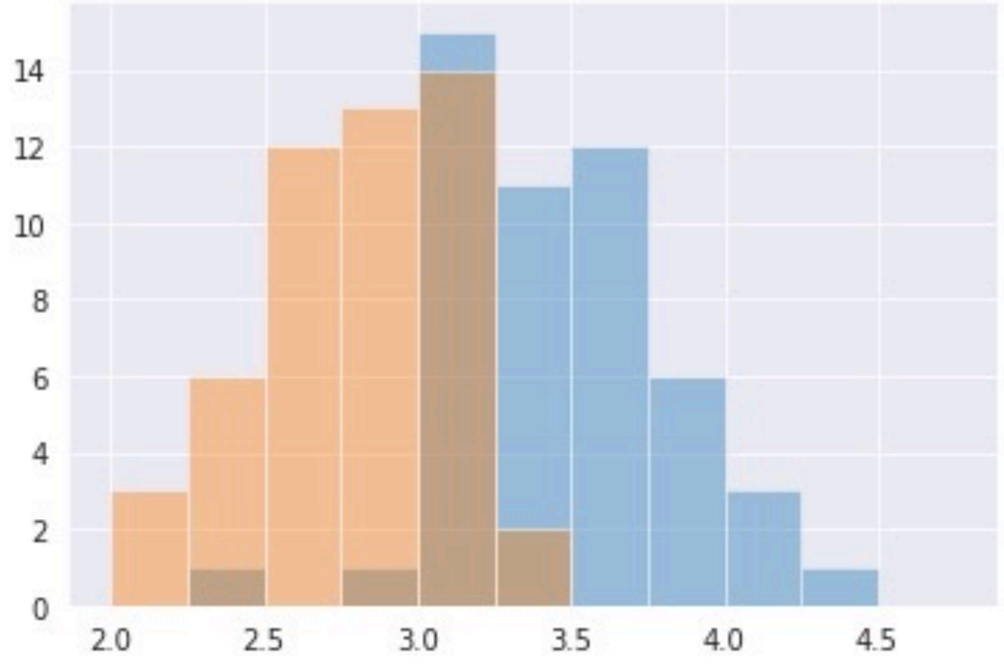


Using numpy to set the bin sizes

Python: Histograms

```
setosa_df = flowers_df[flowers_df.species == 'setosa']
versicolor_df = flowers_df[flowers_df.species == 'versicolor']
virginica_df = flowers_df[flowers_df.species == 'virginica']
```

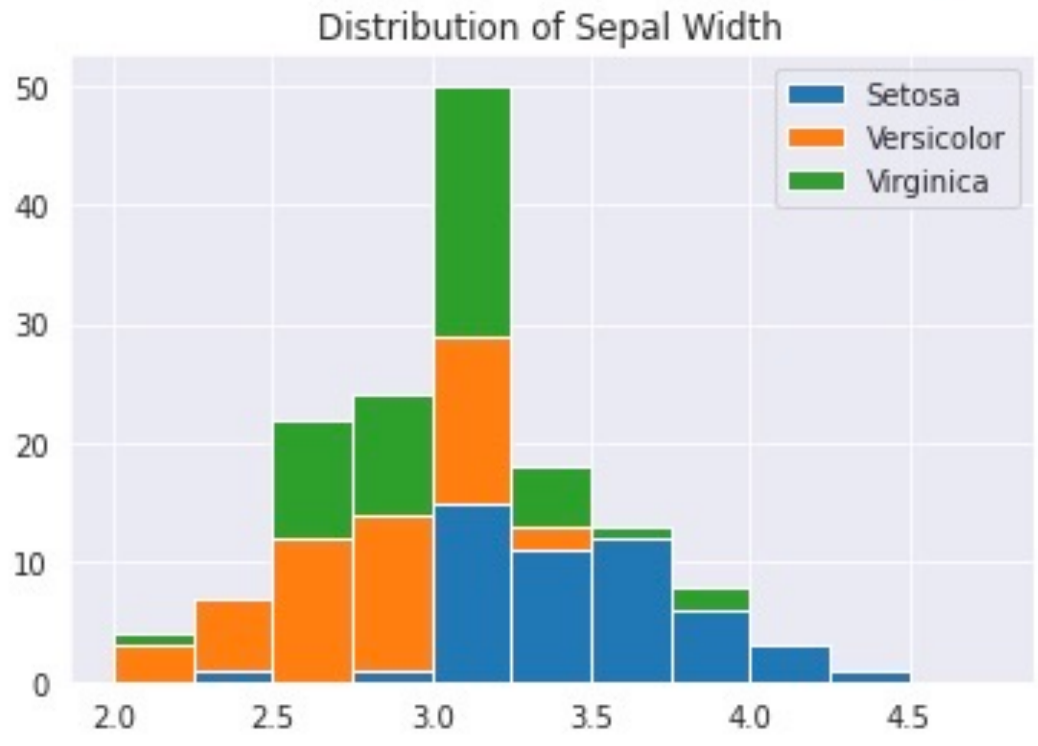
```
plt.hist (setosa_df.sepal_width, alpha=0.4, bins=np.arange(2, 5, 0.25));
plt.hist (versicolor_df.sepal_width, alpha=0.4, bins=np.arange(2, 5, 0.25));
```



Multiple histograms using opacity

Python: Histograms

```
plt.title('Distribution of Sepal Width')
plt.hist([setosa_df.sepal_width, versicolor_df.sepal_width,
          virginica_df.sepal_width],
          bins=np.arange(2, 5, 0.25),
          stacked=True);
plt.legend(['Setosa', 'Versicolor', 'Virginica']);
```



Stacked histograms

Python: Scatter Plots

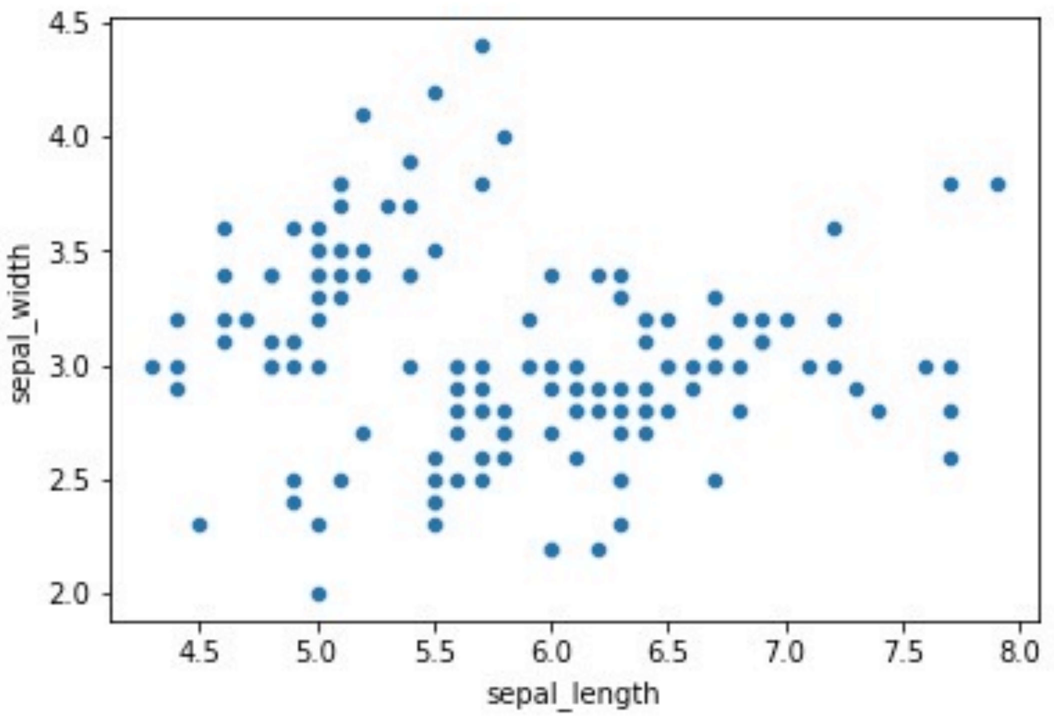
```
flowers_df = sns.load_dataset("iris")
```

```
flowers_df. species.unique()
```

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

```
sns.scatterplot(x=flowers_df.sepal_length, y=flowers_df.sepal_width)
```

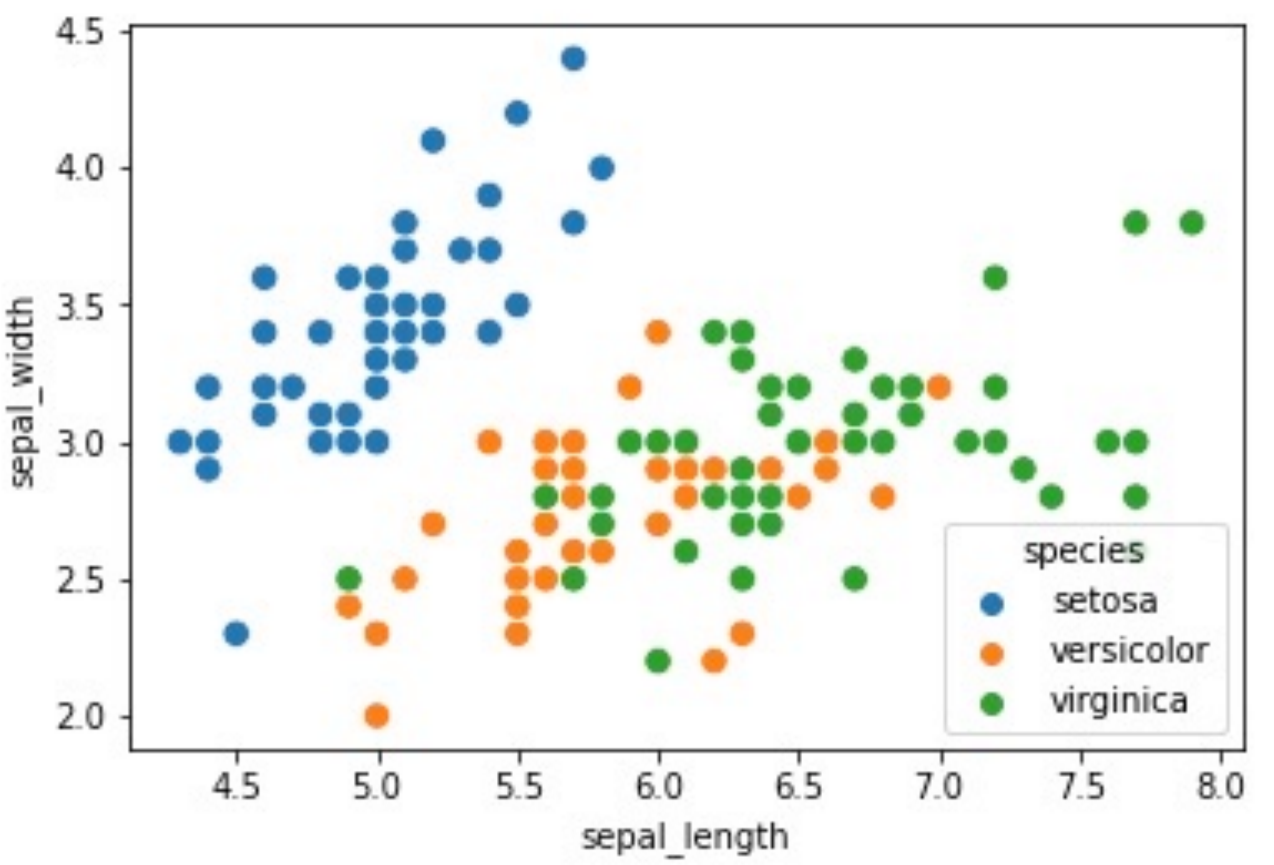
```
<AxesSubplot:xlabel='sepal_length', ylabel='sepal_width'>
```



THE UNIVERSITY OF ARIZONA

Python: Scatter Plots

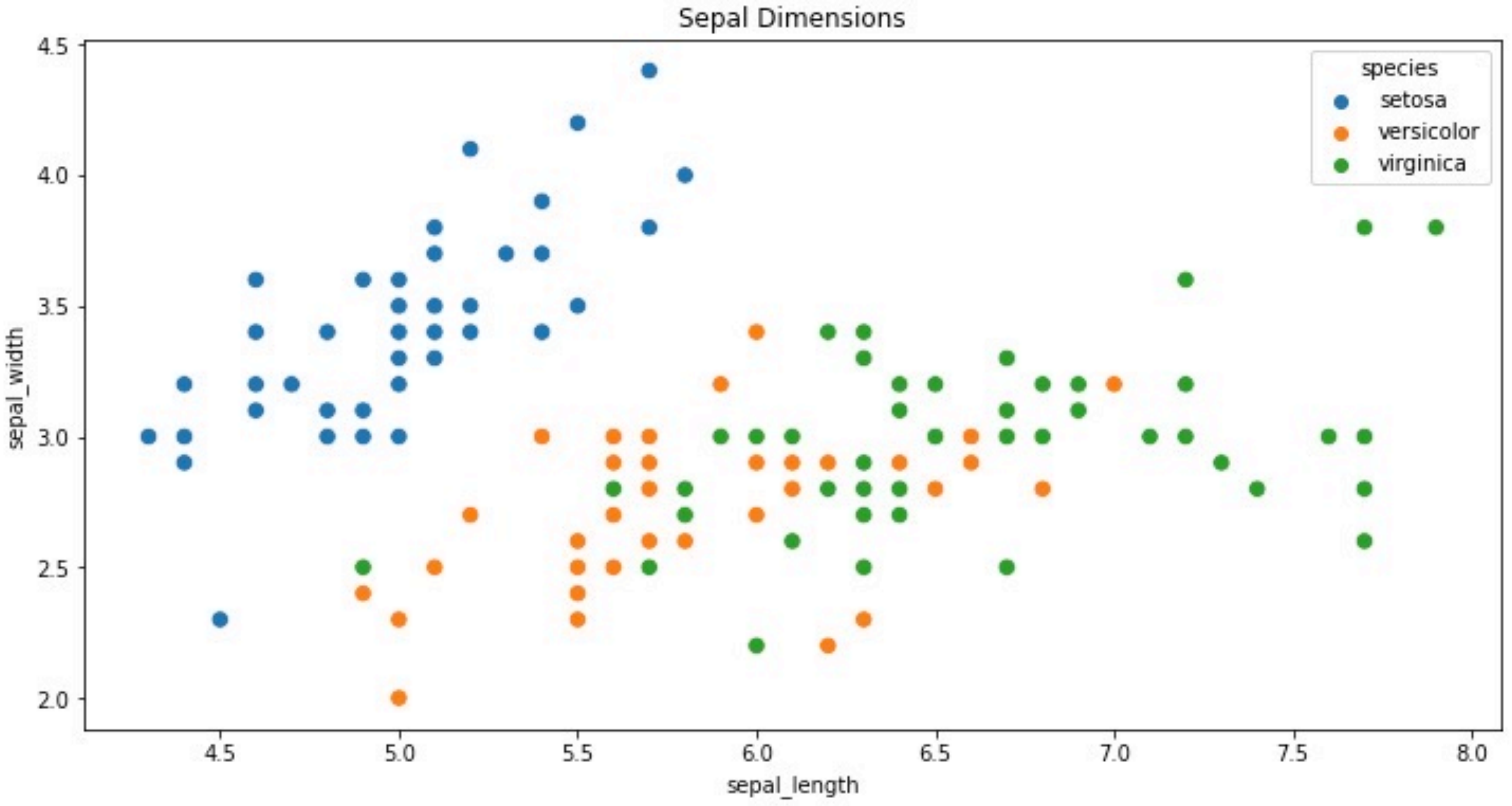
```
sns.scatterplot(x=flowers_df.sepal_length,
                y=flowers_df.sepal_width,
                hue=flowers_df.species, s=70);
```



Add color to distinguish species

Python: Scatter Plots

```python
plt.figure(figsize=(12, 6))
plt.title('Sepal Dimensions')
sns.scatterplot(x=flowers_df.sepal_length,
                y=flowers_df.sepal_width,
                hue=flowers_df.species, s=70);
```



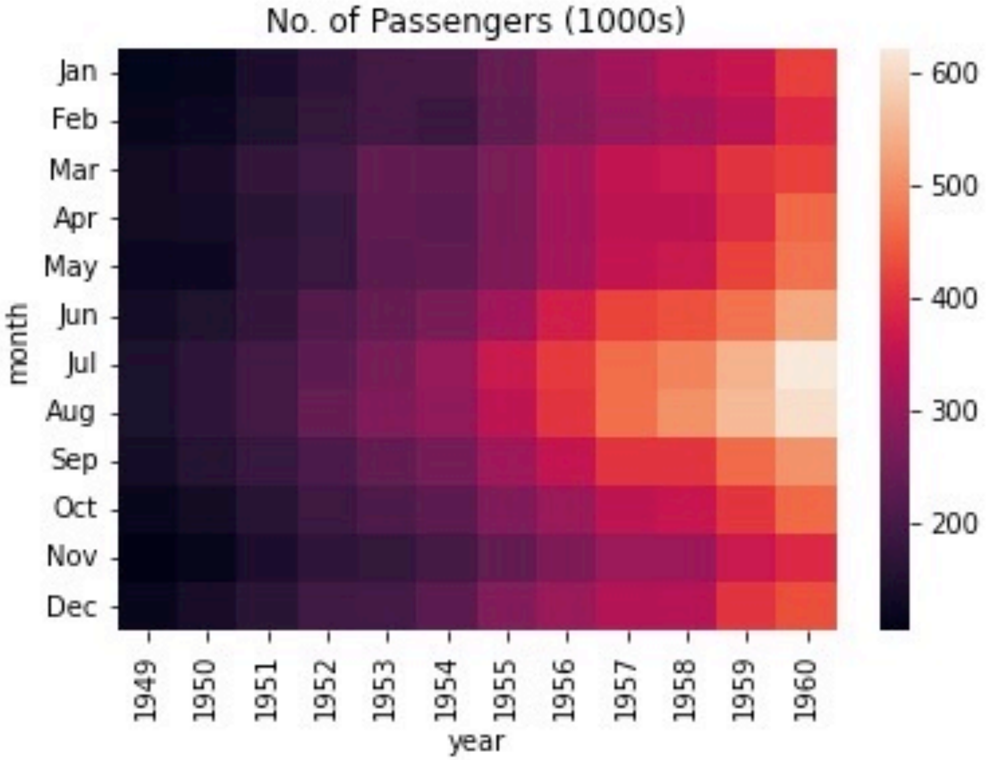
Do the same with seaborn

Python: Heat Maps

```
flights_df = sns.load_dataset("flights").pivot("month", "year", "passengers")
```

```
plt.title("No. of Passengers (1000s)")
sns.heatmap(flights_df)
```

```
<AxesSubplot:title={'center':'No. of Passengers (1000s)'}, xlabel='year', ylabel='month'>
```



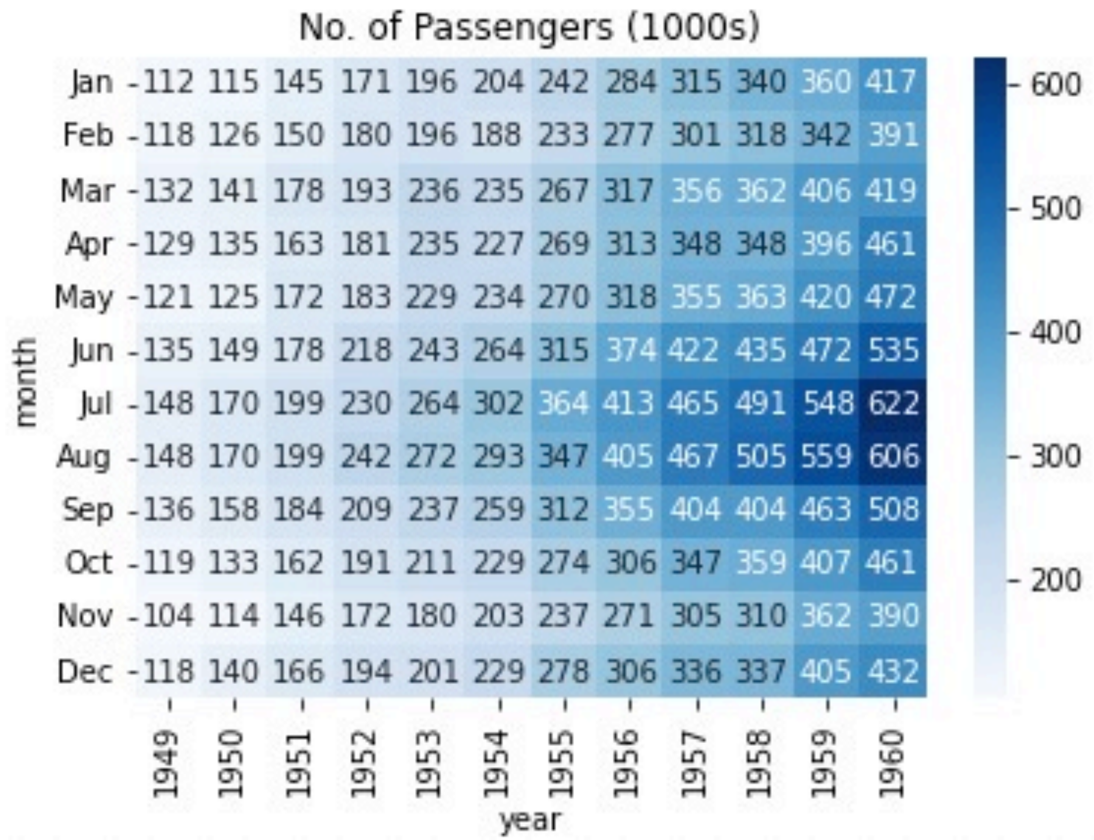
Brighter colors mean more airline traffic
Two conclusions:

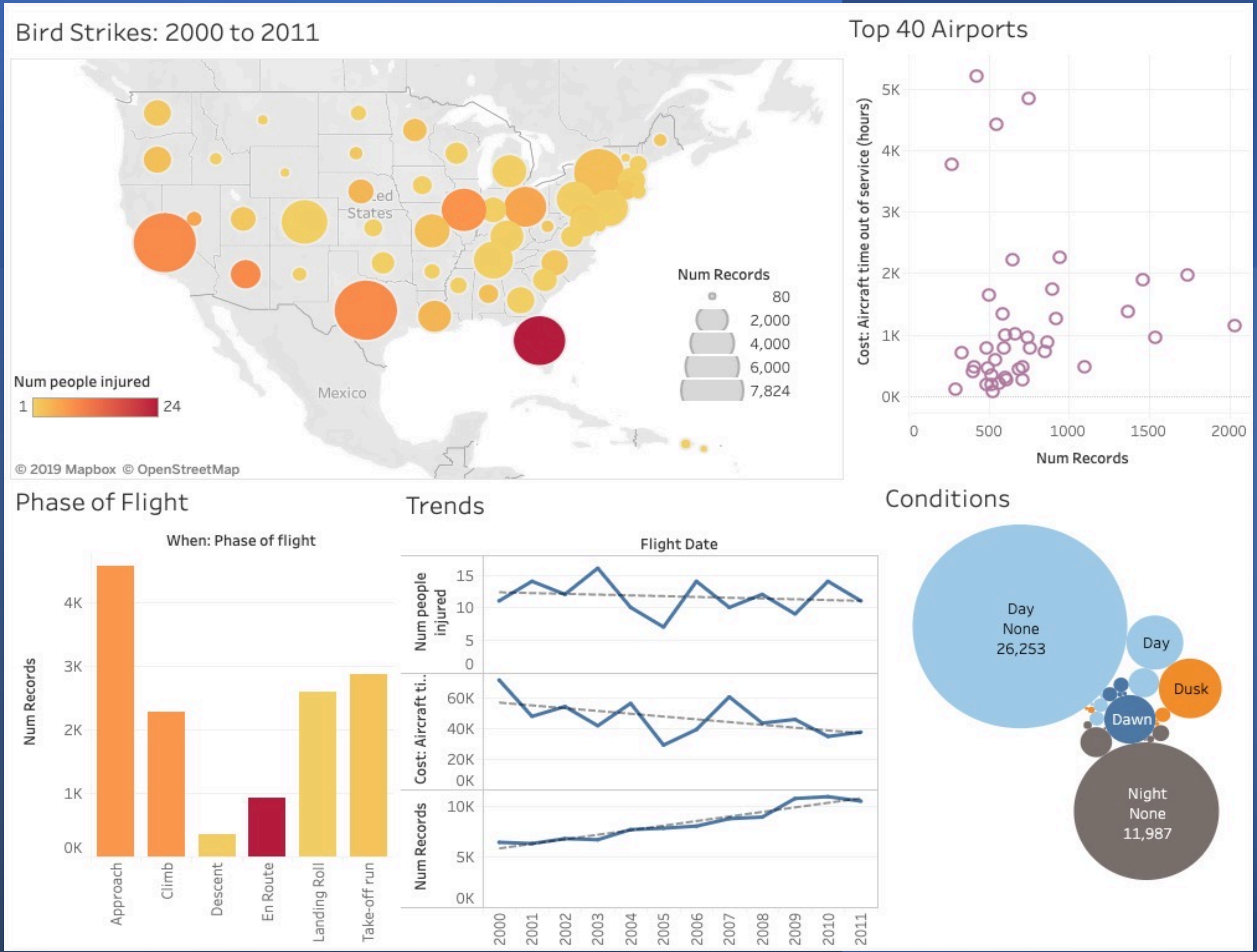
1. There is more traffic in July and August
2. Traffic is generally growing each year

Python: Heat Maps

```python
plt.title("No. of Passengers (1000s)")
sns.heatmap(flights_df, fmt="d", annot=True, cmap='Blues')
```

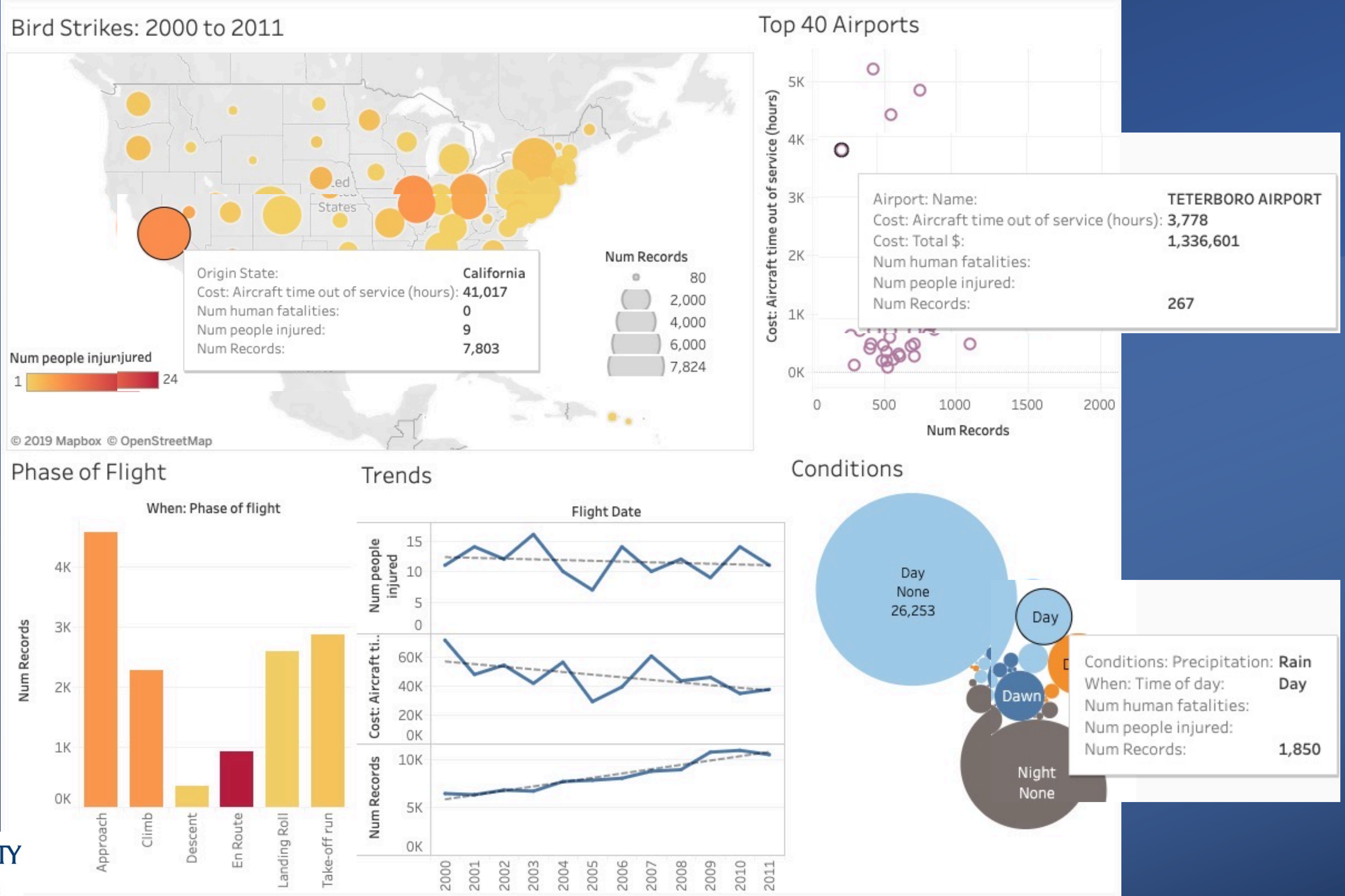
```
<AxesSubplot:title={'center':'No. of Passengers (1000s)'}, xlabel='year',
```




Bird Strikes: 2000 to 2011
Num people injured
1
24
Num Records
80
2,000
4,000
6,000
7,824
Mexico
© 2019 Mapbox © OpenStreetMap
Top 40 Airports
Cost: Aircraft time out of service (hours)
0K
1K
2K
3K
4K
5K
0
500
1000
1500
2000
Num Records
Phase of Flight
When: Phase of flight
Num Records
0K
1K
2K
3K
4K
Approach
Climb
Descent
En Route
Landing Roll
Take-off run
Trends
Flight Date
Num people injured
0
5
10
15
Cost: Aircraft ti..
0K
20K
40K
60K
Num Records
0K
5K
10K
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Conditions
Day
None
26,253
Day
Dusk
Dawn
Night
None
11,987

Underlying Details

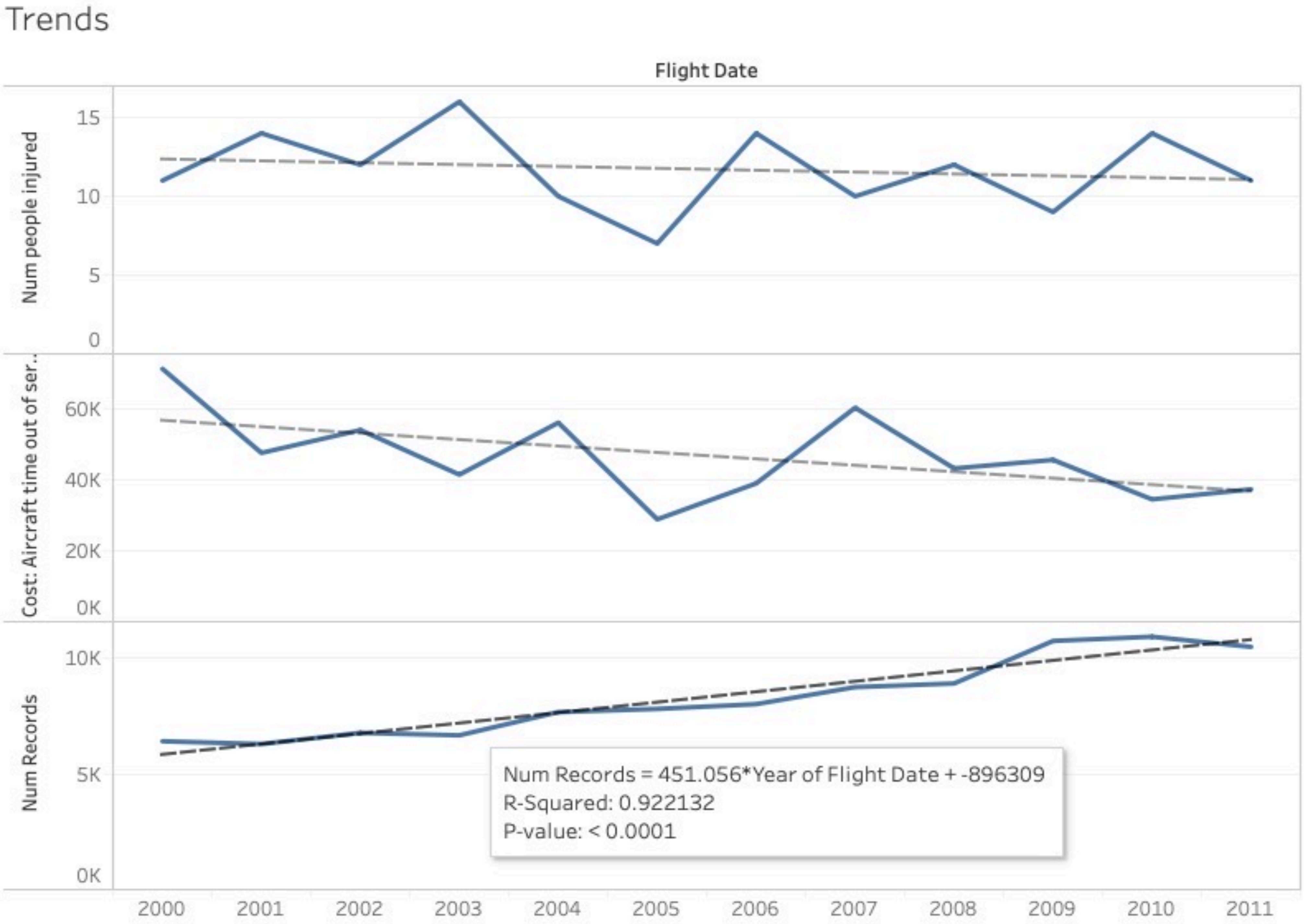
Hover over data points



THE UNIVERSITY OF ARIZONA

Details and Statistics

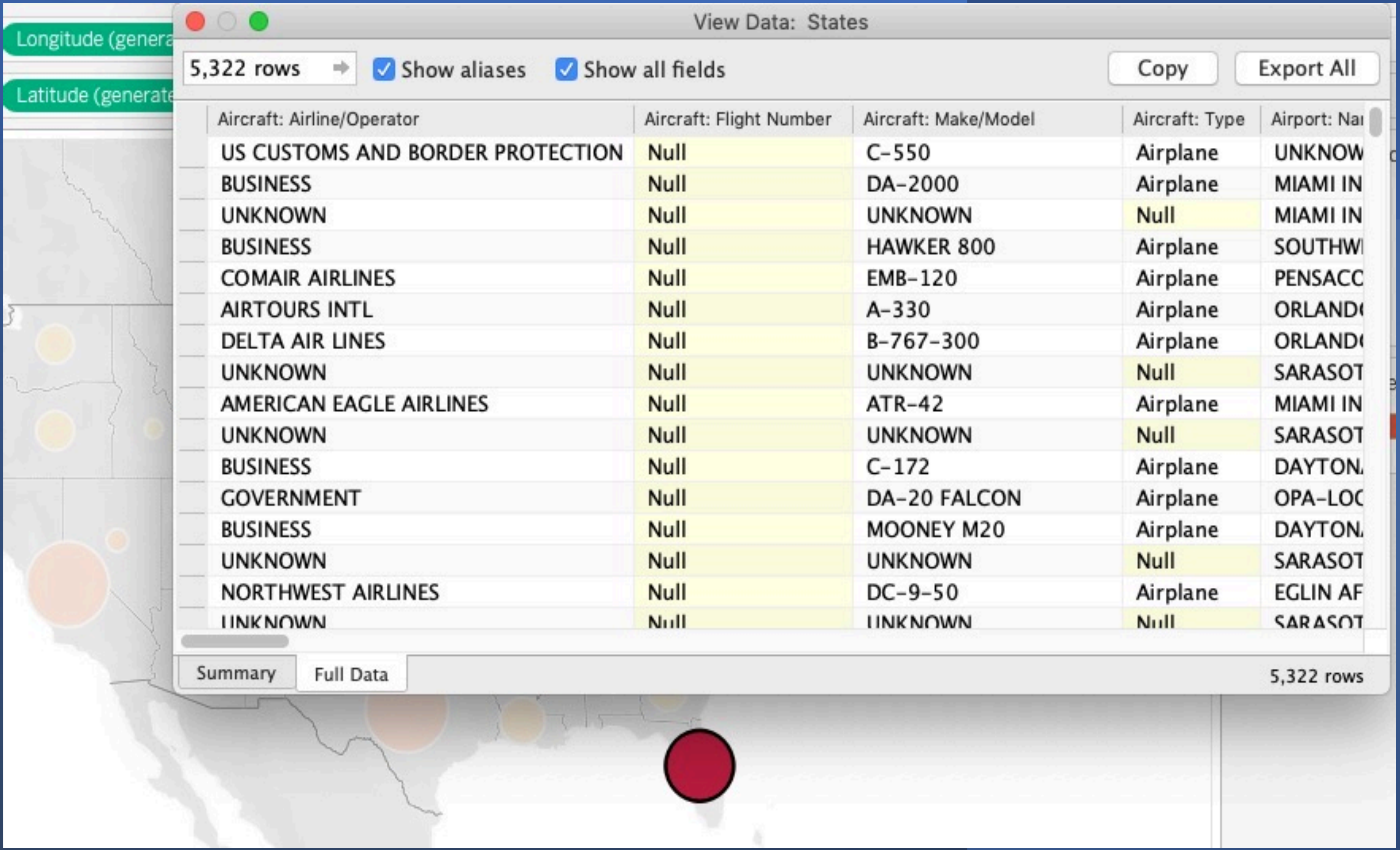
Hover over a data point to see more information



tableau

Show Every Record for Florida

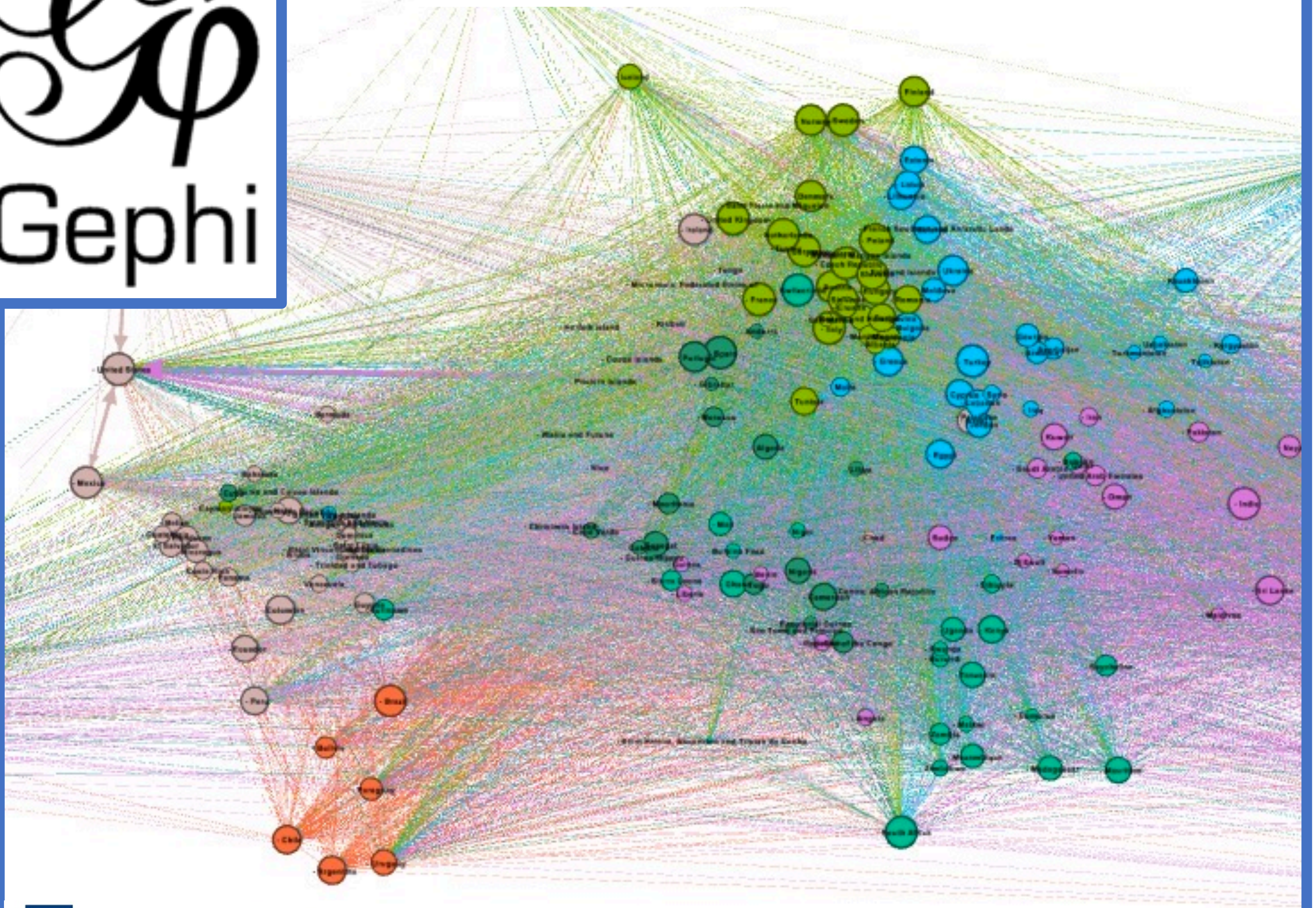
Click on data point and choose View Data option

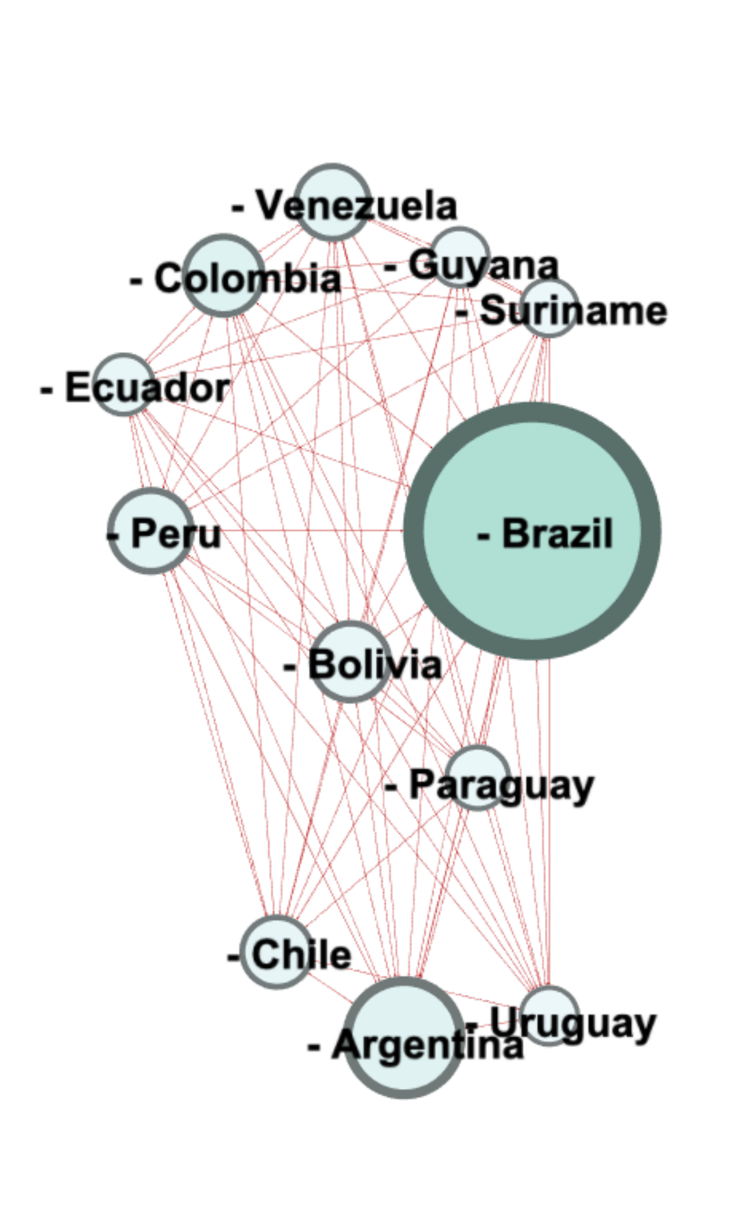


Aircraft: Airline/Operator	Aircraft: Flight Number	Aircraft: Make/Model	Aircraft: Type	Airport: Nai
US CUSTOMS AND BORDER PROTECTION	Null	C-550	Airplane	UNKNOW
BUSINESS	Null	DA-2000	Airplane	MIAMI IN
UNKNOWN	Null	UNKNOWN	Null	MIAMI IN
BUSINESS	Null	HAWKER 800	Airplane	SOUTHWI
COMAIR AIRLINES	Null	EMB-120	Airplane	PENSACC
AIRTOURS INTL	Null	A-330	Airplane	ORLAND(
DELTA AIR LINES	Null	B-767-300	Airplane	ORLAND(
UNKNOWN	Null	UNKNOWN	Null	SARASOT
AMERICAN EAGLE AIRLINES	Null	ATR-42	Airplane	MIAMI IN
UNKNOWN	Null	UNKNOWN	Null	SARASOT
BUSINESS	Null	C-172	Airplane	DAYTON.
GOVERNMENT	Null	DA-20 FALCON	Airplane	OPA-LOC
BUSINESS	Null	MOONEY M20	Airplane	DAYTON.
UNKNOWN	Null	UNKNOWN	Null	SARASOT
NORTHWEST AIRLINES	Null	DC-9-50	Airplane	EGLIN AF
UNKNOWN	Null	UNKNOWN	Null	SARASOT

THE UNIVERSITY OF ARIZONA

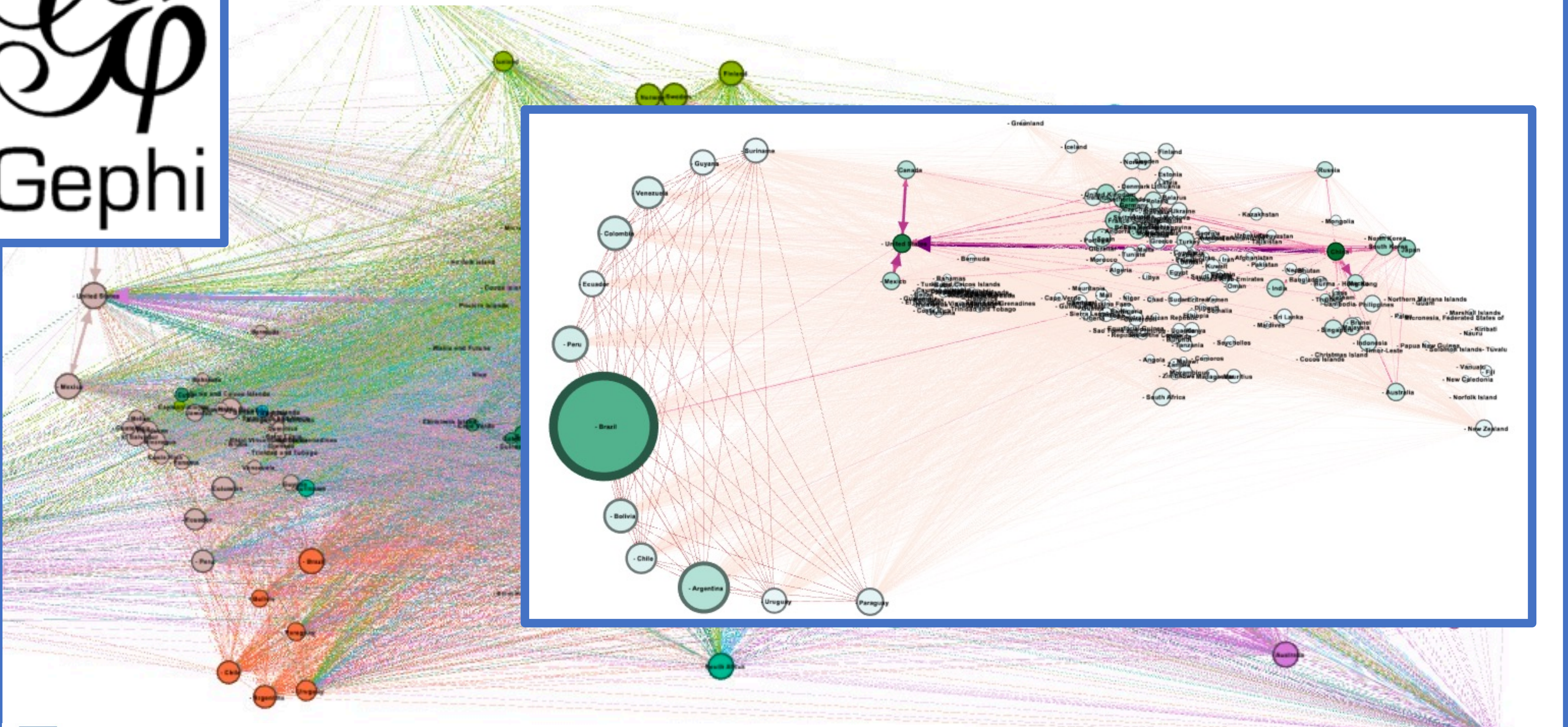
World Trade with South America Insert





World Trade with Insert of South America to the Rest of the World

Gephi



THE UNIVERSITY OF ARIZONA