Hi…

Data Science Enthusiast

My Favorite part about it – Visualizations

Pooja Sharma

PROJECT OBJECTIVE

To create interactive visuals in ipython notebooks.

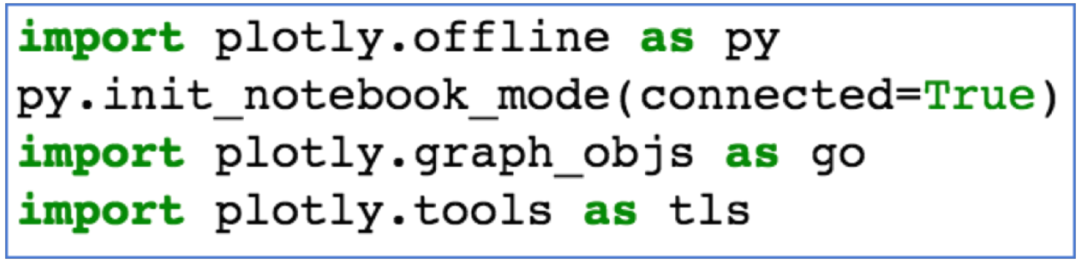
What I did ? - Created graphs using the PLOTLY library –

1. New York City Taxi Trips Data
2. Wine Reviews

What IS PLOTLY

- A Data Visualization library
- Embeddable in jupyter notebooks
- Graphs defined by JSON
- Customizable graphs using key value pairs

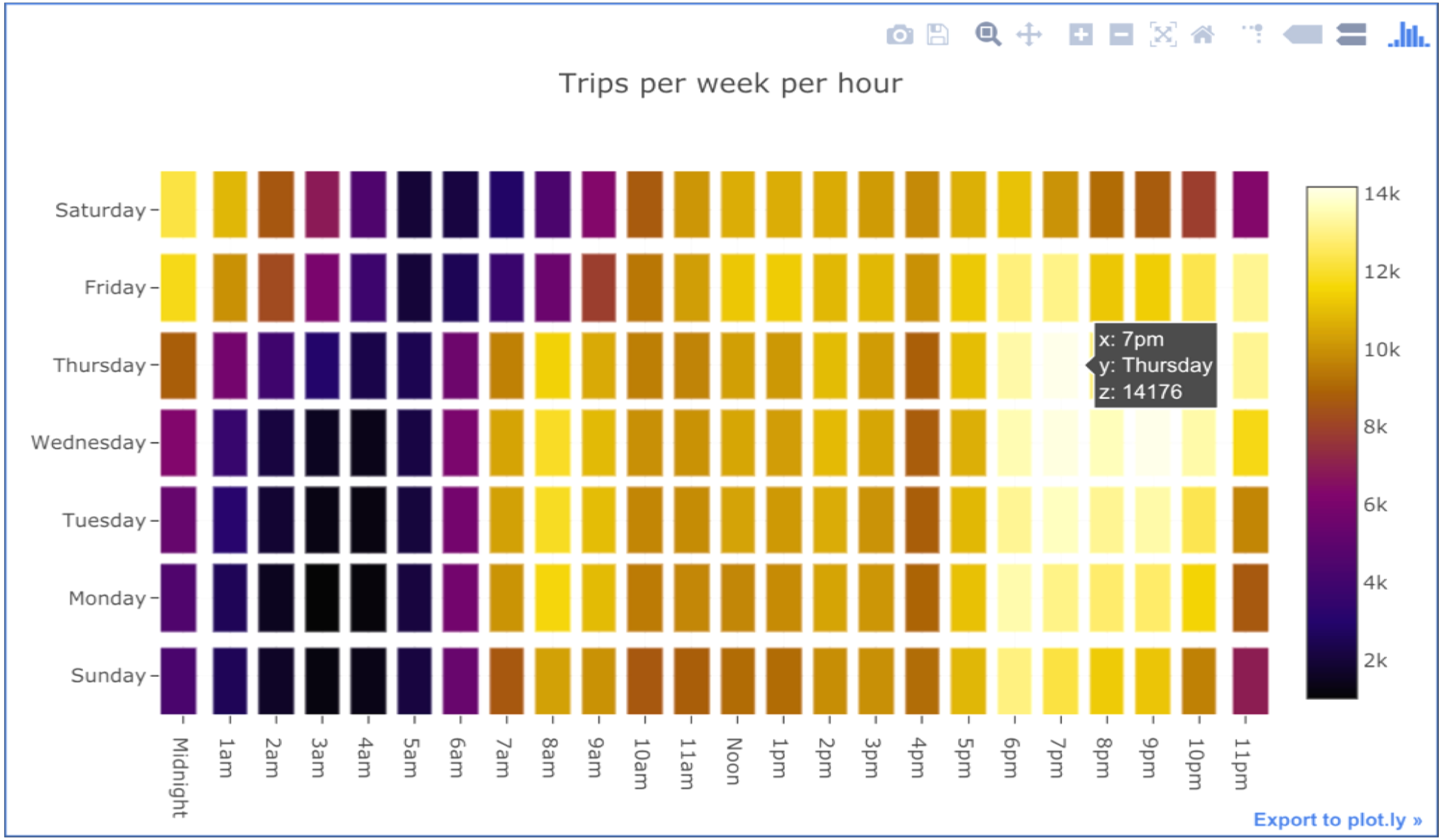
Loading the Plotly Library



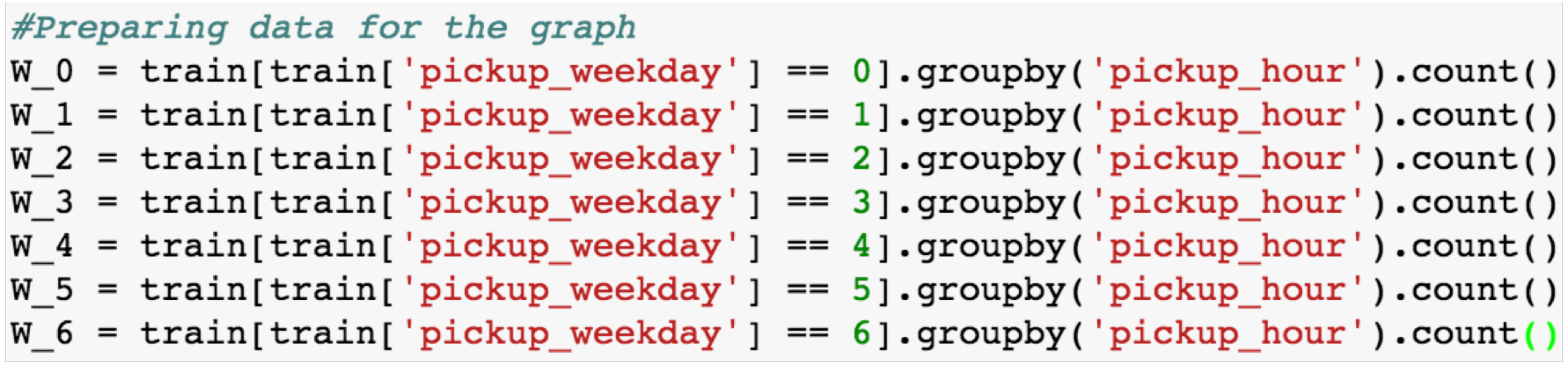
```
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
```

https://plot.ly/python/user-guide/

Heat Map – Showcasing NYC Taxi rides rush hours

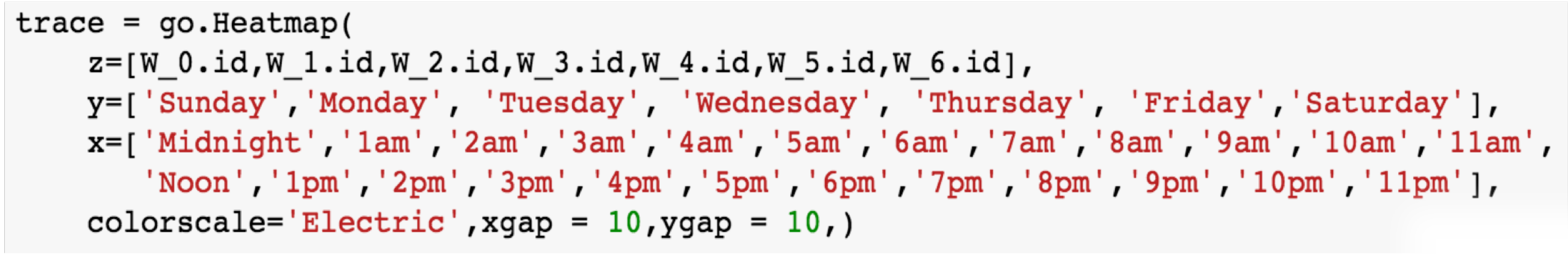


The Plotly Code Recipe – Heat Map



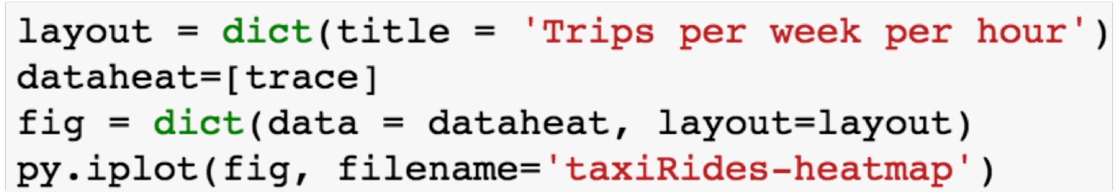
```
#Preparing data for the graph
W_0 = train[train['pickup_weekday'] == 0].groupby('pickup_hour').count()
W_1 = train[train['pickup_weekday'] == 1].groupby('pickup_hour').count()
W_2 = train[train['pickup_weekday'] == 2].groupby('pickup_hour').count()
W_3 = train[train['pickup_weekday'] == 3].groupby('pickup_hour').count()
W_4 = train[train['pickup_weekday'] == 4].groupby('pickup_hour').count()
W_5 = train[train['pickup_weekday'] == 5].groupby('pickup_hour').count()
W_6 = train[train['pickup_weekday'] == 6].groupby('pickup_hour').count()
```

Step1: Prepare Data



```
trace = go.Heatmap(
    z=[W_0.id,W_1.id,W_2.id,W_3.id,W_4.id,W_5.id,W_6.id],
    y=['Sunday','Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday','Saturday'],
    x=['Midnight','1am','2am','3am','4am','5am','6am','7am','8am','9am','10am','11am',
       'Noon','1pm','2pm','3pm','4pm','5pm','6pm','7pm','8pm','9pm','10pm','11pm'],
    colorscale='Electric',xgap = 10,ygap = 10,)
```

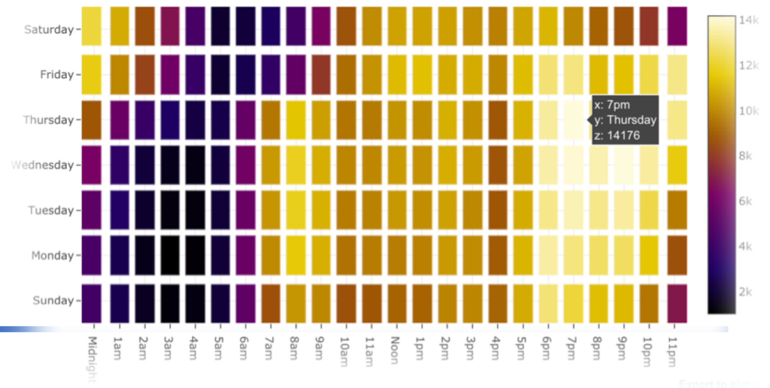
Step2: Define aesthetics



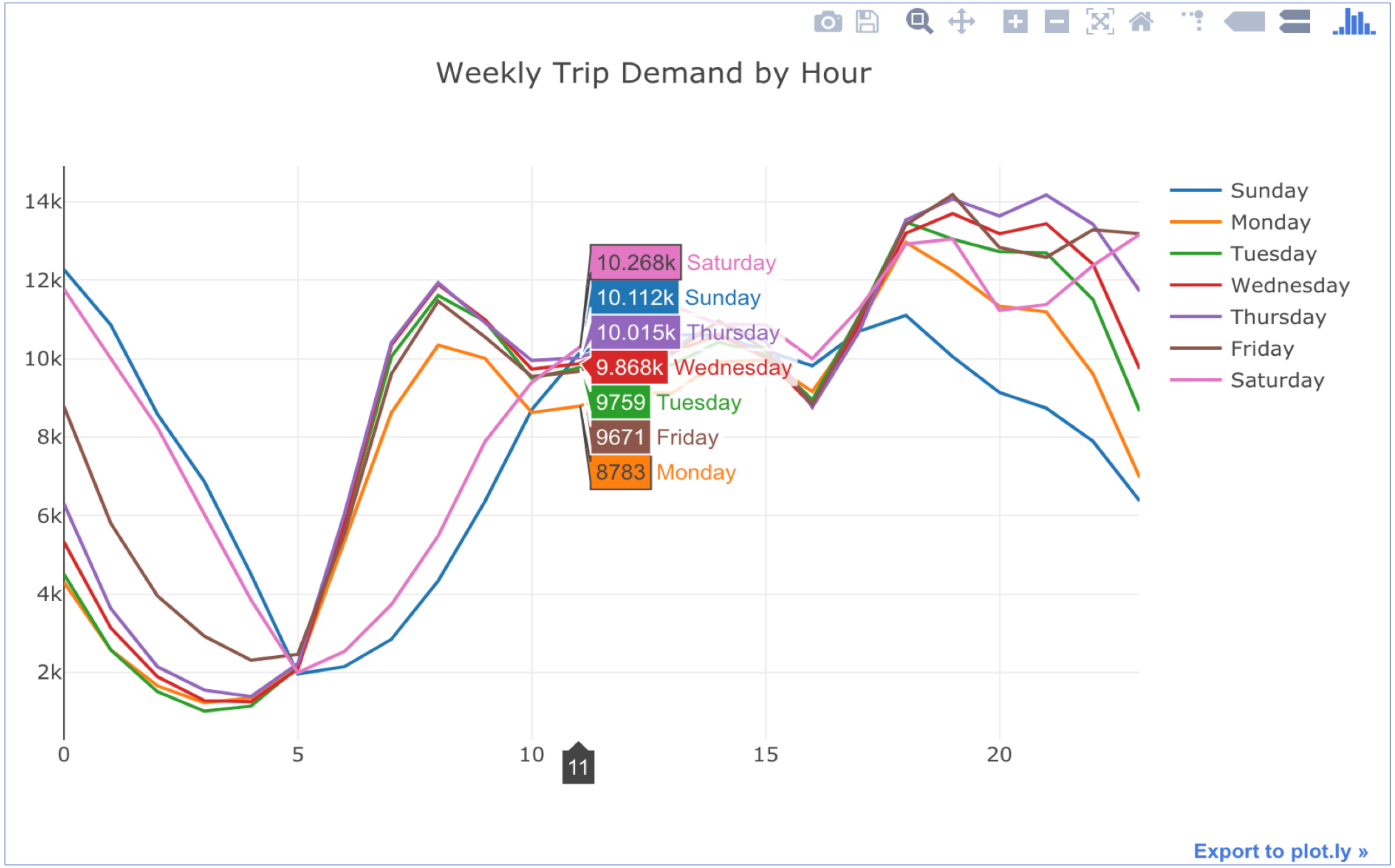
```
layout = dict(title = 'Trips per week per hour')
dataheat=[trace]
fig = dict(data = dataheat, layout=layout)
py.iplot(fig, filename='taxiRides-heatmap')
```

Step3: Lay out and draw

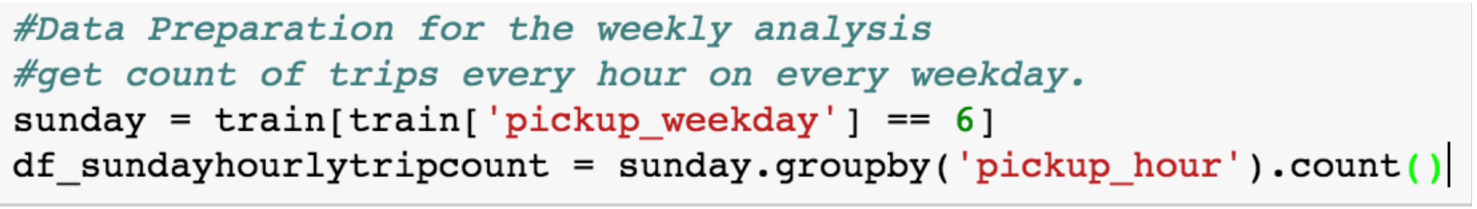
Trips per week per hour



Line Chart - NYC Taxi Trips Weekly Trends

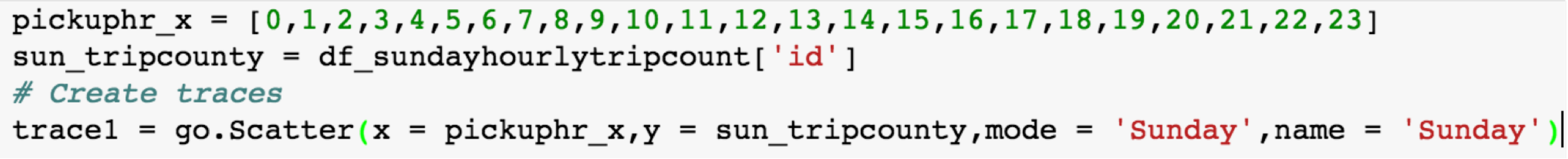


The Plotly Code Recipe – Line Chart



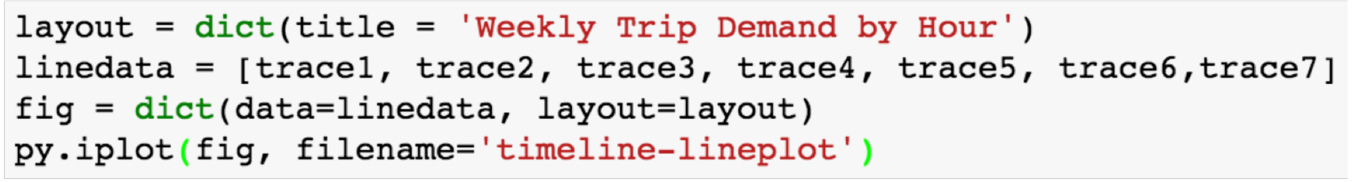
```
#Data Preparation for the weekly analysis
#get count of trips every hour on every weekday.
sunday = train[train['pickup_weekday'] == 6]
df_sundayhourlytripcount = sunday.groupby('pickup_hour').count()
```

Step1: Prepare Data



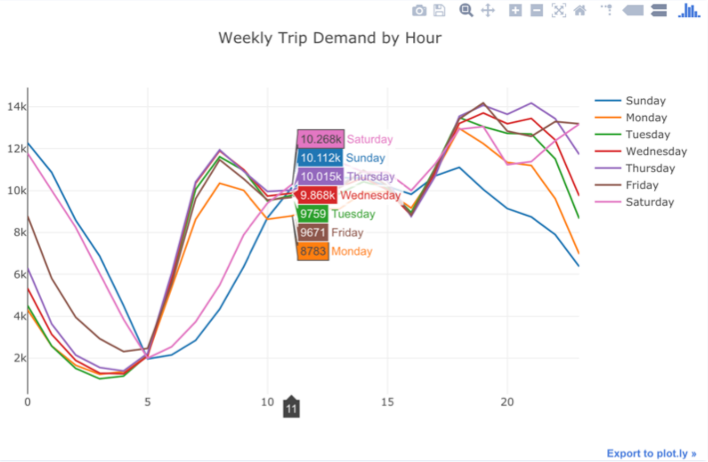
```
pickuphr_x = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23]
sun_tripcounty = df_sundayhourlytripcount['id']
# Create traces
trace1 = go.Scatter(x = pickuphr_x,y = sun_tripcounty,mode = 'Sunday',name = 'Sunday')
```

Step2: Define aesthetics



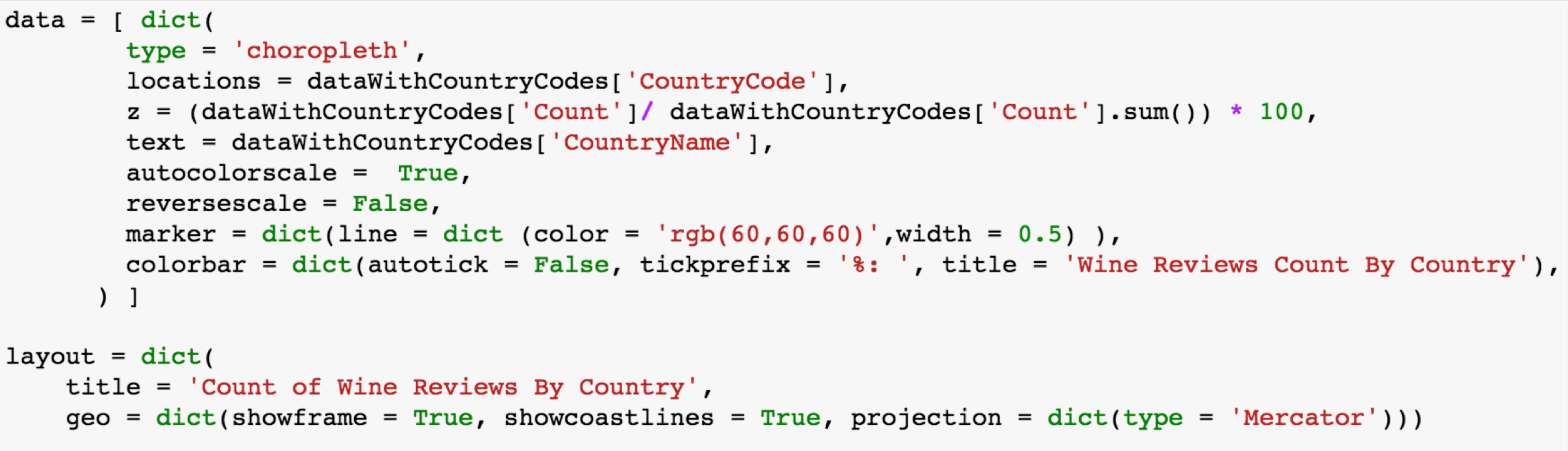
```
layout = dict(title = 'Weekly Trip Demand by Hour')
linedata = [trace1, trace2, trace3, trace4, trace5, trace6,trace7]
fig = dict(data=linedata, layout=layout)
py.iplot(fig, filename='timeline-lineplot')
```

Step3: Lay out and draw

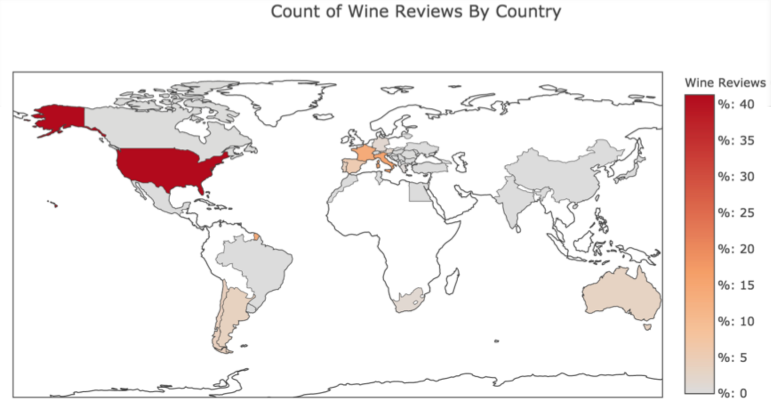


World Map - The Plotly Code Recipe

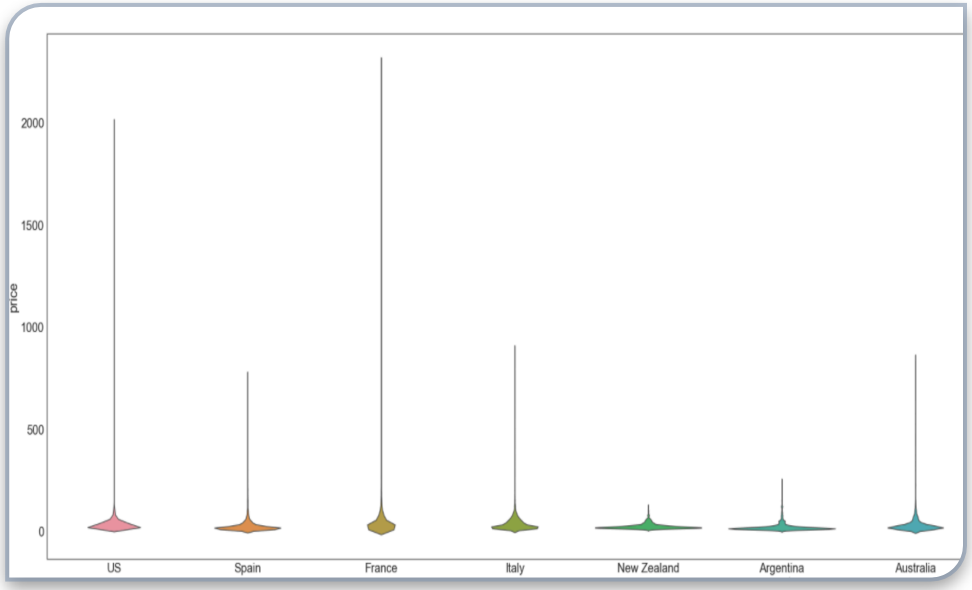
- Prepare Data
- Define aesthetics
- Layout and draw

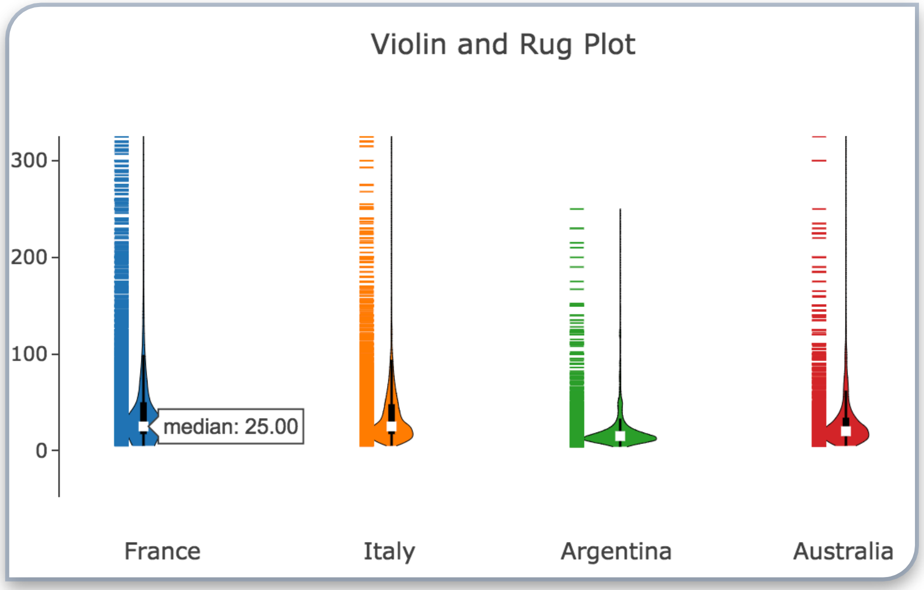


```
data = [ dict(
        type = 'choropleth',
        locations = dataWithCountryCodes['CountryCode'],
        z = (dataWithCountryCodes['Count']/ dataWithCountryCodes['Count'].sum()) * 100,
        text = dataWithCountryCodes['CountryName'],
        autocolorscale =  True,
        reversescale = False,
        marker = dict(line = dict (color = 'rgb(60,60,60)',width = 0.5) ),
        colorbar = dict(autotick = False, tickprefix = '%: ', title = 'Wine Reviews Count By Country'),
      ) ]

layout = dict(
    title = 'Count of Wine Reviews By Country',
    geo = dict(showframe = True, showcoastlines = True, projection = dict(type = 'Mercator')))

fig = dict( data=data, layout=layout )
py.iplot( fig, validate=False, filename='d3-world-map' )
```



Violin Plot – Wines’ Price Range By Country





Matplotlib graph

Accommodate large data points in a single plot
e.g. – plotting all non aggregated points

Plotly graph (Zoomed Out)

Reveals more information about the sample.
e.g. - median, quartiles

Takeaways

- Interactive visuals for ipython notebooks.
- Highly customizable - key components(*data* and *layout*).
- Scope for optimization.

Code Link

https://www.kaggle.com/ps2811/viewing-nyc-taxi-trips-with-plotly