14-12-2020 LECTURE-24

KANAV BANSAL

PLOTLY: SHOWS EXCAT POINTS

It allows users to import, copy and paste or stream data to be analyzed and visualized.

↳ For analysis and styling graphs, Plotly offers a python sandbox (NumPy supported), datagrid and GUI.

↳ The Plotly python graphing library is a scientific graphing library.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns.
```

```
I/p - nyc = pd.read_csv('data/nyc_weather.csv')
  # importing the data
```

I/p - # Scatter plot using seaborn

```
sns.jointplot(x='Temperature', y='Dewpoint',
                                   data=nyc)
```

OBSERVATION FROM THE ABOVE OUTPUT:

* Temperature and Dewpoint are highly correlated.

SCATTER PLOT USING PLOTLY:

Two step process

STEP-1: Importing required library

Plotly offline mode

I/p - from plotly.offline import iplot, init_notebook_mode

```
init_notebook_mode()
```

Importing high-level chart objects.

I/p - import plotly.graphs_objs as go.

STEP-2 : Creating a Scatter Plot

```
I/p - data_scatter = [go.scatter(x = nyc.Temperature.
                      values, y = nyc.Dewpoint.values,
                      mode = 'markers')]

      fig = go.Figure(data = data_scatter)

      iplot(fig)
```

Labeling the Plotly using layout

```
I/p - data_scatter = [go.scatter(x = nyc.Temperature.
                      values, y = nyc.Dewpoint.values,
                      mode = 'markers')]

     layout = {"title": "Temp vs Dewpoint",
               X-axis : {"title": "Temperature"},
               Y-axis : {'title': 'Dewpoint'}}

     fig = go. Figure(data = data_scatter, layout =
                                          layout)

     iplot(fig)
```

* As plotly is a little bit hard to code, we will also look into EXPRESS library which is written on top of plotly.

INTRODUCTION TO PLOTLY EXPRESS

Importing required library

import plotly. express as px

EXPRESS - SCATTER PLOT :

Scatter Plot

I/p - px.scatter(nyc, x = 'Temperature', y = 'Dewpoint')

Scatter Plot with OLS line

I/p - px.scatter(nyc, x = 'Temperature', y = 'Dewpoint', trendline = 'ols')

OLS - ORDINARY LEAST SQUARE : The class estimates a multi-variate regression model and provides a variety of fit - statistics. To see the class in action download the ols.py file and run it (python ols.py). This will estimate a multi-variate regression using simulated data and provide output.

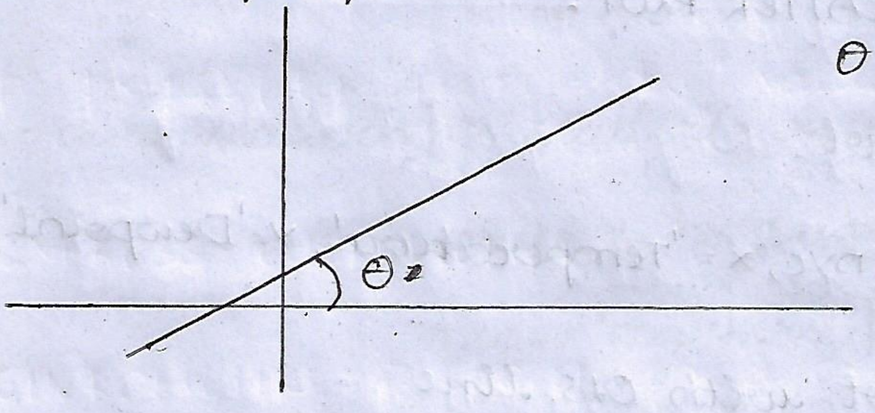
OLS - BEST LINE TO FIT THE DATA (center)

The output of this will be in the form of

$Y = mx + C$ - m = slope, c = intercept

↳ Intercept of Y

$\theta \rightarrow m = \tan\theta$



EXPRESS - BOX PLOT:

Box plot in seaborn

I/p - sns.boxplot(x = 'sex', y = 'age', data = titanic)

OBSERVATION:

↳ There are some outliers in male data points.

↳ Males are having 30 as the average age.

↳ Females have 26 as average age.

Box plot in plotly express:

I/p - px.box(titanic, x = 'sex', y = 'age')

I/p - sns.boxplot(x = 'sex', y = 'age', hue = 'survived',
                   data = titanic)

I/p - px.box(titanic, x = 'sex', y = 'age', color = 'survived')

IN SEABORN - HUE

IN PLOTLY - COLOR

EXPRESS - STRIPPLOT :

I/p - tips = pd.read_csv('data/tips.csv')

I/p - sns.stripplot(x = 'day', y = 'total_bill', data = tips)

I/p - px.strip(tips, x = 'day', y = 'total_bill')

EXPRESS - BARPLOT :

I/p - sns.barplot(x='sex', y='age', data=titanic)

I/p - pvt_df = titanic.pivot_table(values='age',
index='sex')

I/p - px.bar(pvt_df, x=pvt_df.index, y='age')

EXPRESS - PIE CHART :

I/p - tips.head()

I/p - px.pie(tips, name='day', values='total_bill')

EXPRESS - CHOROPLETH :

CHOROPLETHS : A choropleths is a map composed of colored polygons. It is used to represent spatial variations of a quantity.

I/p - ca_df = pd.read_csv('data/countries-aggregated.csv')

ca_df.head()

ca - countries - aggregated

```
I/p - group_df = ca_df.groupby('Date')
      group_df.first()

I/p - temp_df = group_df.get_group("2020-03-15")
      temp_df.head()

I/p - fig = px.choropleth(temp_df, locations='country',
                 color='confirmed', hover name =
                 'confirmed', locationmode = 'country
                 names')
      fig.show()

I/p - fig = px.choropleth(ca_df, locations='country',
                 color = 'confirmed', hover name =
                 'confirmed', locationmode = 'country
                 names', animation_frame = 'Date')
      fig.show()
```

```
I/p - fig = px.choropleth(ca_df, locations='country',
          color='confirmed', hover_name='confirmed
          locationmode='countrynames',
          animation_frame='Date',
          color_continuous_scale = px.colors.
                     sequential.OrRd)
      fig.show()
```

```
I/p - fig = px.choropleth(ca_df, locations='country',
          color='confirmed', hover_name='confirmed',
          location mode='countrynames',
          animation_frame='Date',
          color_continuous_scale = px.colors.sequential.
          OrRd, scope='asia')
      fig.show()
```

OrRd - Orange to Red.