



Aula 37 - Gráficos em Python

☰ Ciclo	Ciclo 05: Visualização
# Aula	37
🕒 Created	@November 21, 2022 7:15 AM
☑ Reviewed	<input type="checkbox"/>
📎 Material PDF	
☑ Finished	<input checked="" type="checkbox"/>

Objetivo da Aula:

1. Criar a visão empresa com os gráficos disponíveis em Python.
2. Na próxima aula

Conteúdo:

▼ A visão: Crescimento

Do lado da empresa:

1. Quantidade de pedidos por dia.
2. Quantidade de pedidos por semana.
3. Distribuição dos pedidos por tipo de tráfego.
4. Comparação do volume de pedidos por cidade e tipo de tráfego.
5. A localização central de cada cidade por tipo de tráfego.

▼ 1. Quantidade de pedidos por dia.

```
# Quantidade de pedidos por dia
df_aux = df2.loc[:, ['ID', 'Order_Date']].groupby( 'Order_Date' ).count().reset_index()
df_aux.columns = ['order_date', 'qtde_entregas']

# gráfico
px.bar( df_aux, x='order_date', y='qtde_entregas' )
```

▼ 2. Quantidade de pedidos por semana.

```
# Quantidade de pedidos por Semana
df2['week_of_year'] = df2['Order_Date'].dt.strftime( "%W" )
df_aux = df2.loc[:, ['ID', 'week_of_year']].groupby( 'week_of_year' ).count().reset_index()
```

```
# gráfico
px.bar( df_aux, x='week_of_year', y='ID' )
```

▼ 3. Distribuição dos pedidos por tipo de tráfego.

```
df_aux = df2.loc[:, ['ID', 'Road_traffic_density']].groupby( 'Road_traffic_density' ).count().reset_index()
df_aux['perc_ID'] = 100 * ( df_aux['ID'] / df_aux['ID'].sum() )

# gráfico
px.pie( df_aux, values='perc_ID', names='Road_traffic_density' )
```

▼ 4. Comparação do volume de pedidos por cidade e tipo de tráfego.

```
df_aux = df2.loc[:, ['ID', 'Road_traffic_density']].groupby( 'Road_traffic_density' ).count().reset_index()
df_aux['perc_ID'] = 100 * ( df_aux['ID'] / df_aux['ID'].sum() )

# gráfico
px.pie( df_aux, values='perc_ID', names='Road_traffic_density' )
```

▼ 5. A quantidade de pedidos por entregador por semana.

```
# Quantidade de pedidos por entregador por Semana
# Quantas entregas na semana / Quantos entregadores únicos por semana
df_aux1 = df2.loc[:, ['ID', 'week_of_year']].groupby( 'week_of_year' ).count().reset_index()
df_aux2 = df2.loc[:, ['Delivery_person_ID', 'week_of_year']].groupby( 'week_of_year' ).nunique().reset_index()
df_aux = pd.merge( df_aux1, df_aux2, how='inner' )

df_aux['order_by_delivery'] = df_aux['ID'] / df_aux['Delivery_person_ID']

# gráfico
px.line( df_aux, x='week_of_year', y='order_by_delivery' )
```

▼ 6. A localização central de cada cidade por tipo de tráfego.

```
data_plot = df2.loc[:, ['City', 'Road_traffic_density', 'Delivery_location_latitude', 'Delivery_location_longitude']].groupby( ['City', 'Road_traffic_density'] ).mean().reset_index()
data_plot = data_plot[data_plot['City'] != 'NaN']
data_plot = data_plot[data_plot['Road_traffic_density'] != 'NaN']

# Desenhar o mapa
map_ = folium.Map( zoom_start=11 )

for index, location_info in data_plot.iterrows():
    folium.Marker( [location_info['Delivery_location_latitude'],
                    location_info['Delivery_location_longitude']],
                  popup=location_info[['City', 'Road_traffic_density']] ).add_to( map_ )

map_
```

▼ Na próxima aula

Aula 38: O conceito de função