09_LB_Average_Sentiment_Tweet_Amount

July 13, 2021

0.1 Imports

[10]:
```
import calendar
import plotly.graph_objects as go
import plotly.express       as px
import pandas               as pd
from   scipy                import signal
```

0.2 Reading Data

[11]:
```
data = pd.read_csv("../data/interim/Overall_data.csv")
```

0.3 Animated Plot with Sentiment per Country

[12]:
```
#Grouping Data by date and country + meaning the Sentiment in the countries and␣
 ↪dates
fig = px.choropleth(data.groupby(['date','country'])['sentiment'].mean().
 ↪reset_index(),
              locations = 'country',
              color='sentiment',
              locationmode='country names',
              animation_frame='date',
              hover_name="country",
              scope='world',
              range_color=(-1,1),
              title='Sentiment over time by country',
              height=600
             )
fig.layout.updatemenus[0].buttons[0].args[1]['frame']['duration'] = 80

#Export as HTML
fig.write_html("../reports/figures/OverallCovidTweets/
 ↪Sentiment_over_time_per_country.html")

fig.show()
```

0.4 Sentiment per Country for whole timeaxis

[13]:
```
#Grouping Data by date and country + meaning the Sentiment in the countries and␣
 ↪dates
fig = px.choropleth(data.groupby(['country'])['sentiment'].mean().reset_index(),
              locations = 'country',
              color='sentiment',
              locationmode='country names',
              hover_name='country',
              scope='world',
              range_color=(-1,1),
              title='Combined Sentiment by country',
              projection='natural earth',
              height=600
             )

#Export as HTML
fig.write_html("../reports/figures/OverallCovidTweets/
 ↪Combined_Sentiment_by_country.html")

fig.show()
```

0.5 Timeplot for Tweets per Day and Sentiment

[14]:
```
#Scatter Plot combined with a Bar Chart with two y axis
fig = go.Figure()

fig.add_trace(go.Bar(
    x= data.groupby(['date']).count().reset_index(drop=False)['date'],
    y= data.groupby(['date'])['id'].count(),
    name='Amount of Tweets',
    marker_color='darkorange',
    yaxis = 'y1',
))

fig.add_trace(go.Scatter(
    x= data.sort_values('created_at')['created_at'],
    y= signal.savgol_filter(
                    data.sort_values('created_at')['sentiment'].dropna(),
                    901,
                    4
                    ),
    name = 'Sentiment over time',
    yaxis = 'y2',
))

fig.update_layout(
```

```
    title_text="Sentiment on Amount of Tweets over time"
)

fig.update(layout=go.Layout(yaxis1 = go.YAxis(title='Amount of Tweets',␣
 ↪titlefont=go.Font(color='orange'), side="right"),
                            yaxis2 = go.YAxis(title='Average Sentiment',␣
 ↪titlefont=go.Font(color='red'), overlaying='y', side="left")))

#Adjusting Axis range
fig.update_layout(yaxis1 = dict(range = [0,2000]))
fig.update_layout(yaxis2 = dict(range = [-0.3,0.3]))

#Export as HTML
fig.write_html("../reports/figures/OverallCovidTweets/
 ↪Sentiment_amount_of_tweets.html")

fig.show()
```

0.6 Amount of Tweets per Month

[15]:
```
#Adjusting Data
data_month = data.groupby(['month','year'])['id'].count().
 ↪reset_index(drop=False)
data_month['month'] = data_month['month'].apply(lambda x: calendar.
 ↪month_abbr[x])
data_month = data_month.drop([6, 7])
```

[16]:
```
#Bar Chart using month and tweet amount
fig = px.bar(
    data_month,
    x= data_month['month'],
    y= data_month['id'],
)

fig.update(layout=go.Layout(yaxis1 = go.YAxis(title='Amount of Tweets',␣
 ↪titlefont=go.Font(color='SteelBlue')),
                            xaxis = go.XAxis(title='Months', titlefont=go.
 ↪Font(color='SteelBlue')))
)

fig.update_layout(
    title_text="Amount of Tweets per Month"
)

#Export as HTML
```

```
fig.write_html("../reports/figures/OverallCovidTweets/Tweet_amount_per_month.
 ↪html")

fig.show()
```

0.7 Average Sentiment per Month

[17]:
```
#Creating new temporary dataset
#Grouping by month and year meaning the Sentiment
#Changing numbers in column 'month' to Month names
#Droping data from 2021
data_senti = data.groupby(['month','year'])['sentiment'].mean().
 ↪reset_index(drop=False)
data_senti['month'] = data_senti['month'].apply(lambda x: calendar.
 ↪month_abbr[x])
data_senti = data_senti.drop([6, 7])
```

[18]:
```
#Bar Chart using mean Sentiment per Month
fig = px.bar(
    data_senti,
    x= data_senti['month'],
    y= data_senti['sentiment'],
    color='sentiment'
)

fig.update(layout=go.Layout(yaxis1 = go.YAxis(title='Average Sentiment',␣
 ↪titlefont=go.Font(color='SteelBlue')),
                            xaxis = go.XAxis(title='Months', titlefont=go.
 ↪Font(color='SteelBlue')))
)

fig.update_layout(
    title_text="Average Sentiment per Month"
)

#Adjusting Axis range
fig.update_layout(yaxis1 = dict(range = [-0.05,0.2]))

#Export as HTML
fig.write_html("../reports/figures/OverallCovidTweets/
 ↪Average_Sentiment_per_month.html")

fig.show()
```