

Tạo ứng dụng khám phá dữ liệu

Trong phần này, chúng ta sẽ sử dụng các tính năng cốt lõi của Streamlit để tạo một ứng dụng tương tác, khám phá tập dữ liệu Uber công khai về các điểm đón và trả khách ở thành phố New York. Khi hoàn tất, bạn sẽ biết cách tìm nạp và lưu dữ liệu vào bộ nhớ cache, vẽ biểu đồ, vẽ thông tin trên bản đồ và sử dụng các tiện ích tương tác, như thanh trượt, thanh lọc kết quả.

Đầu tiên chúng ta cần import một số thư viện cần thiết

```
import streamlit as st
import numpy as np
import pandas as pd
```

Fetch some data

```
DATE_COLUMN = 'date/time'
DATA_URL = ('https://s3-us-west-2.amazonaws.com/'
            'streamlit-demo-data/uber-raw-data-sep14.csv.gz')

def load_data(nrows):
    ''' nrows is number of row that you want to load'''
    data = pd.read_csv(DATA_URL, nrows=nrows)
    lowercase = lambda x: str(x).lower()
    data.rename(lowercase, axis='columns', inplace=True)
    data[DATE_COLUMN] = pd.to_datetime(data[DATE_COLUMN])
    return data

# Create a text element and let the reader know data is loading
data_load_state = st.text('Loading data...')
# Load 10000 rows of data into the dataframe
data = load_data(10000)
# Notify the reader that the data was successfully loader
data_load_state.text('Loading data... done!')
```

Loading data... done!

Sử dụng cache (bộ nhớ đệm)

- Chúng ta thêm **@st.cache** trước hàm **load_data**. Sau đó, lưu tập lệnh và Streamlit sẽ tự động chạy lại ứng dụng của bạn. Vì đây là lần đầu tiên bạn chạy tập lệnh với **@st.cache**, bạn sẽ không thấy bất kỳ điều gì thay đổi. Hãy chỉnh sửa tệp của bạn thêm một chút để bạn có thể thấy sức mạnh của bộ nhớ đệm.
- Thay dòng **data_load_state.text('Loading data... done!')** thành:

```
data_load_state.text('Done! (using st.cache)')
```

- Tốc độ nhanh hẳn vì đã lưu cache

```
DATE_COLUMN = 'date/time'
```

```
DATA_URL = ('https://s3-us-west-2.amazonaws.com/'
            'streamlit-demo-data/uber-raw-data-sep14.csv.gz')

@ st.cache()
def load_data(nrows):
    ''' nrows is number of row that you want to load'''
    data = pd.read_csv(DATA_URL, nrows=nrows)
    lowercase = lambda x: str(x).lower()
    data.rename(lowercase, axis='columns', inplace=True)
    data[DATE_COLUMN] = pd.to_datetime(data[DATE_COLUMN])
    return data

# Create a text element and let the reader know data is loading
data_load_state = st.text('Loading data...')
# Load 10000 rows of data into the dataframe
data = load_data(10000)
# Notify the reader that the data was successfully loader
data_load_state.text('Done! (using st.cache)')

Done! (using st.cache)
```

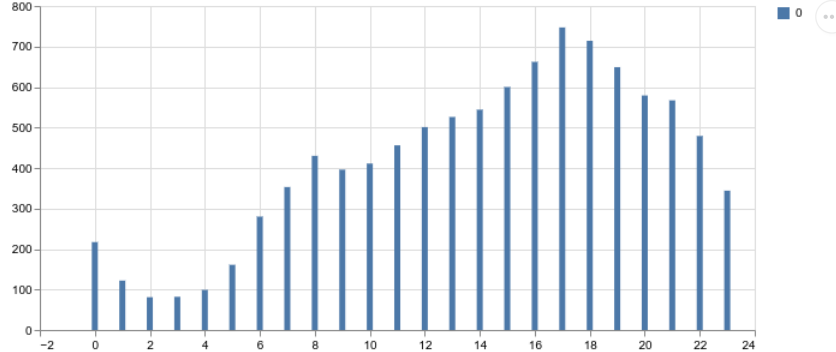
Kiểm tra dữ liệu thô

```
st.write(data)
```

	date/time	lat	lon	base
0	Sep 1, 2014 12:01 AM	40.2201	-74.0021	B02512
1	Sep 1, 2014 12:01 AM	40.7500	-74.0027	B02512
2	Sep 1, 2014 12:03 AM	40.7559	-73.9864	B02512
3	Sep 1, 2014 12:06 AM	40.7450	-73.9889	B02512
4	Sep 1, 2014 12:11 AM	40.8145	-73.9444	B02512
5	Sep 1, 2014 12:12 AM	40.6735	-73.9918	B02512
6	Sep 1, 2014 12:15 AM	40.7471	-73.6472	B02512
7	Sep 1, 2014 12:16 AM	40.6613	-74.2691	B02512
8	Sep 1, 2014 12:32 AM	40.3745	-73.9999	B02512
9	Sep 1, 2014 12:33 AM	40.7633	-73.9773	B02512
10	Sep 1, 2014 12:33 AM	40.7467	-73.6131	B02512

Vẽ biểu đồ

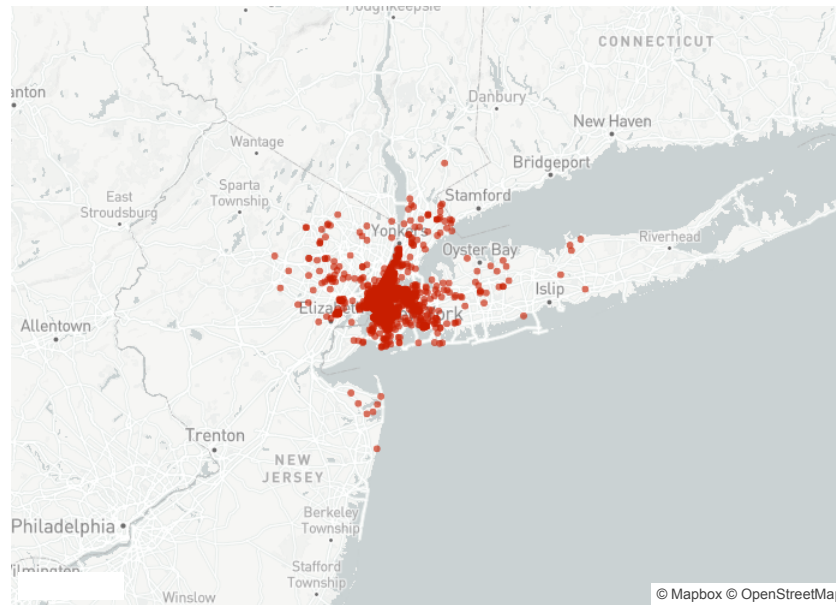
```
hist_values = np.histogram(data[DATE_COLUMN].dt.hour, bins=24, range=(0,24))[0]
st.bar_chart(hist_values)
```



Xem thêm các phương thức hiển thị chart khác tại:
<https://docs.streamlit.io/en/stable/api.html#display-charts>

Vẽ dữ liệu trên map

```
st.map(data)
```



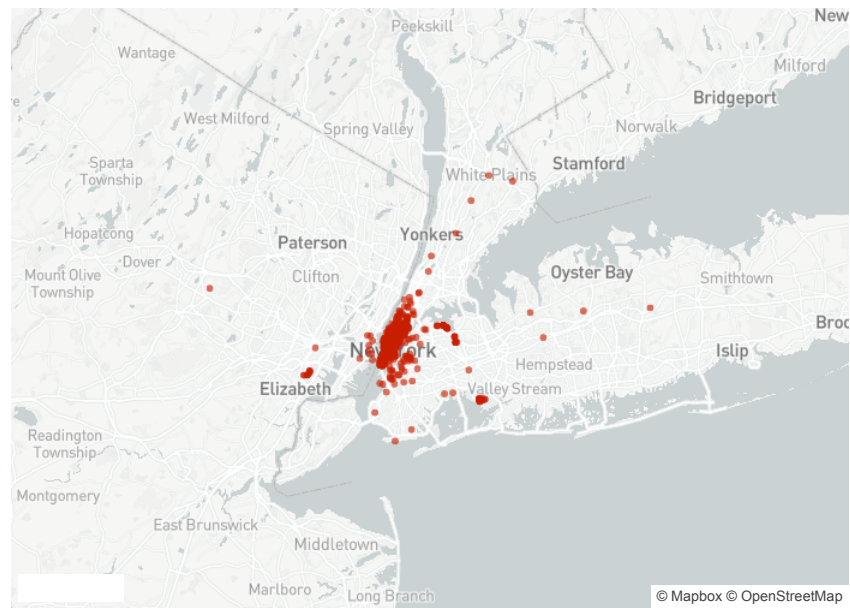
Lọc kết quả với slider

```
hour_to_filter = st.slider('hour', 0, 23, 17) # min: 0h, max: 23h, default: 17h
filtered_data = data[data[DATE_COLUMN].dt.hour == hour_to_filter]
st.subheader(f'Map of all pickups at {hour_to_filter}:00')
st.map(filtered_data)
```

hour

17

Map of all pickups at 17:00



Made with [Streamlit](#)