In [5]:
```
%matplotlib inline
from matplotlib import style
style.use('fivethirtyeight')
import matplotlib.pyplot as plt
```

In [ ]:
```
pip install
```

In [2]:
```
import numpy as np
import pandas as pd
import datetime as dt
```

Reflect Tables into SQLAlchemy ORM

In [3]:
```
# Python SQL toolkit and Object Relational Mapper
import sqlalchemy
from sqlalchemy.ext.automap import automap_base
from sqlalchemy.orm import Session
from sqlalchemy import create_engine, func
```

In [4]:
```
# create engine to hawaii.sqlite
engine = create_engine("sqlite:///Resources/hawaii.sqlite")
```

In [5]:
```
# reflect an existing database into a new model

# reflect the tables
```

In [6]:
```
# View all of the classes that automap found
```

Out[6]:
```
['measurement', 'station']
```

In [7]:
```
# Save references to each table
```

In [8]:
```
# Create our session (link) from Python to the DB
```

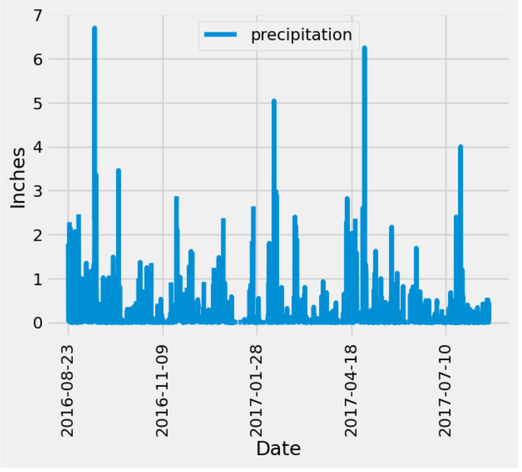
Exploratory Precipitation Analysis

In [9]:
```
# Find the most recent date in the data set.
```

Out[9]:
```
('2017-08-23',)
```

In [10]:
```
# Design a query to retrieve the last 12 months of precipitation data and plot the results.
# Starting from the most recent data point in the database.

# Calculate the date one year from the last date in data set.


# Perform a query to retrieve the data and precipitation scores


# Save the query results as a Pandas DataFrame. Explicitly set the column names


# Sort the dataframe by date


# Use Pandas Plotting with Matplotlib to plot the data
```

Out[10]:
```
Text(0, 0.5, 'Inches')
```



In [3]:
```
# Use Pandas to calculate the summary statistics for the precipitation data
result.session.query(measurement.date, measurement.prcp).filter(measurement.date >= one_year).all()
df = pd.DataFrame(result, columns =["date", "precipitation"])
print(df["precipitation"].describe)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[3], line 2
      1 # Use Pandas to calculate the summary statistics for the precipitation data
----> 2 result.session.query(measurement.date, measurement.prcp).filter(measurement.date >= one_year).all()
      3 df = pd.DataFrame(result, columns =["date", "precipitation"])
      4 print(df["precipitation"].describe)

NameError: name 'result' is not defined
```

Exploratory Station Analysis

In [12]:
```
# Design a query to calculate the total number of stations in the dataset
```

Out[12]:
```
[(9,)]
```

In [13]:
```
# Design a query to find the most active stations (i.e. which stations have the most rows?)
# List the stations and their counts in descending order.
```

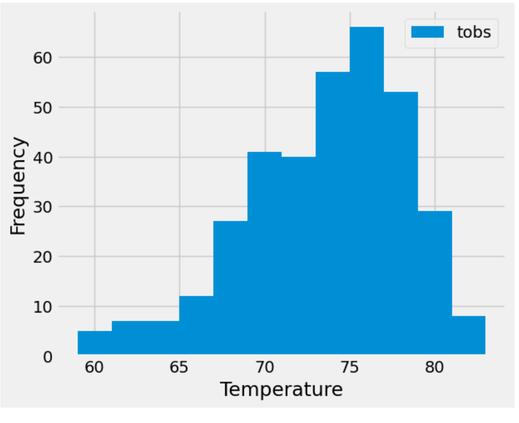
Out[13]:
```
[('USC00519281', 2772),
 ('USC00519397', 2724),
 ('USC00513117', 2709),
 ('USC00519523', 2669),
 ('USC00516128', 2612),
 ('USC00514830', 2202),
 ('USC00511918', 1979),
 ('USC00517948', 1372),
 ('USC00518838', 511)]
```

In [14]:
```
# Using the most active station id from the previous query, calculate the lowest, highest, and average temperature.
```

Out[14]:
```
[(54.0, 85.0, 71.66378066378067)]
```

In [15]:
```
# Using the most active station id
# Query the last 12 months of temperature observation data for this station and plot the results as a histogram
```

Out[15]:
```
Text(0.5, 4.183333333333314, 'Temperature')
```



Close Session

In [16]:
```
# Close Session
session.close()
```

In [ ]:
```

```