

climate_starter

July 22, 2019

```
In [1]: %matplotlib inline
        from matplotlib import style
        style.use('fivethirtyeight')
        import matplotlib.pyplot as plt
```

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

```
In [3]: import datetime as dt
```

1 Reflect Tables into SQLAlchemy ORM

```
In [4]: # 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, inspect
```

```
In [5]: engine = create_engine("sqlite:///Resources/hawaii.sqlite")
```

```
In [6]: # reflect an existing database into a new model
        Base = automap_base()
        # reflect the tables
        Base.prepare(engine, reflect=True)
```

```
In [7]: # We can view all of the classes that automap found
        Base.classes.keys()
```

```
Out[7]: ['measurement', 'station']
```

```
In [8]: # Save references to each table
        Measurement = Base.classes.measurement
        Station = Base.classes.station
```

```
In [9]: # Create our session (link) from Python to the DB
        session = Session(engine)
```

```
In [10]: #inspect measurement data
        inspector = inspect(engine)
        columns = inspector.get_columns('measurement')
        for c in columns:
            print(c['name'], c['type'])
```

```
id INTEGER
station TEXT
date TEXT
prcp FLOAT
tobs FLOAT
```

```
In [11]: #inspect station data
        columns = inspector.get_columns('station')
        for c in columns:
            print(c['name'], c['type'])
```

```
id INTEGER
station TEXT
name TEXT
latitude FLOAT
longitude FLOAT
elevation FLOAT
```

2 Exploratory Climate Analysis

```
In [12]: # Calculate the date 1 year ago from the last data point in the database
        last_data_point = session.query(Measurement.date).order_by(Measurement.date.desc()).first()

        print(last_data_point[0])
```

```
2017-08-23
```

```
In [13]: # Design a query to retrieve the last 12 months of precipitation data and plot the results
        #Last day in data is 8/23/2017. 12 month range defined as 8/24/2016-8/23/2017.
```

```
precipitation_data = session.query(func.strftime("%Y-%m-%d", Measurement.date), Measurement.precipitation)
                        .filter(func.strftime("%Y-%m-%d", Measurement.date) >= dt.date(2016, 8, 24)).all()
```

```
In [14]: # Save the query results as a Pandas DataFrame and set the index to the date column
        # Sort the dataframe by date
        precipitation_df = pd.DataFrame(precipitation_data, columns = ['date', 'precipitation'])
        precipitation_df.set_index('date', inplace = True)
        precipitation_df = precipitation_df.sort_values(by = 'date')
        precipitation_df.head(10)
```