9.5.5   Monthly Temperature Route

You're making progress! You have just two more routes to create and then you'll be ready to share your app with W. Avy and the investors. Now you'll create the temperature observations route.

For this route, the goal is to return the temperature observations for the previous year. As with the previous routes, begin by defining the route with this code:

```
@app.route("/api/v1.0/tobs")
```

Next, create a function called `temp_monthly()` by adding the following code:

```
def temp_monthly():
    return
```

Now, calculate the date one year ago from the last date in the database.

(This is the same date as the one we calculated previously.) Your code should look like this:

```
def temp_monthly():
    prev_year = dt.date(2017, 8, 23) - dt.timedelta(days=365)
    return
```

The next step is to query the primary station for all the temperature observations from the previous year. Here's what the code should look like with the query statement added:

```
def temp_monthly():
    prev_year = dt.date(2017, 8, 23) - dt.timedelta(days=365)
    results = session.query(Measurement.tobs).\
        filter(Measurement.station == 'USC00519281').\
        filter(Measurement.date >= prev_year).all()
    return
```

Finally, as before, unravel the results into a one-dimensional array and convert that array into a list. Then jsonify the list and return our results, like this:

```
def temp_monthly():
    prev_year = dt.date(2017, 8, 23) - dt.timedelta(days=365)
    results = session.query(Measurement.tobs).\
      filter(Measurement.station == 'USC00519281').\
      filter(Measurement.date >= prev_year).all()
    temps = list(np.ravel(results))
```

As we did earlier, we want to jsonify our `temps` list, and then return it. Add the return statement to the end of your code so that the route looks like this:

```python
def temp_monthly():
    prev_year = dt.date(2017, 8, 23) - dt.timedelta(days=365)
    results = session.query(Measurement.tobs).\
      filter(Measurement.station == 'USC00519281').\
      filter(Measurement.date >= prev_year).all()
    temps = list(np.ravel(results))
    return jsonify(temps=temps)
```

Before moving on to the next route, let's test our code to make sure it works.

NOTE

> Testing your code regularly can help prevent any errors or bugs in your code. If errors occur, testing your code often will allow you to pinpoint where and why the error is occurring.

To test your code, navigate to your project folder, surfs_up, for this module in the command line. Then, run the following command:

```
flask run
```

Use the web address to see the output of your code and the various routes you've created. For this route, use the web address (`http://localhost:5000/`) provided by Flask in the command line, which will hold all of our monthly temperature readings. Here's what your output should look like:

```
{"temps":
[77.0,77.0,80.0,80.0,75.0,73.0,78.0,77.0,78.0,80.0,80.0,78.0,78.0,78.0,73
.0,74.0,80.0,79.0,77.0,80.0,76.0,79.0,75.0,79.0,78.0,79.0,78.0,78.0,76.0,
74.0,77.0,78.0,79.0,79.0,77.0,80.0,78.0,78.0,78.0,77.0,79.0,79.0,79.0,79.
0,75.0,76.0,73.0,72.0,71.0,77.0,79.0,78.0,79.0,77.0,79.0,77.0,78.0,78.0,7
8.0,78.0,77.0,74.0,75.0,76.0,73.0,76.0,74.0,77.0,76.0,76.0,74.0,75.0,75.0
,75.0,75.0,71.0,63.0,70.0,68.0,67.0,77.0,74.0,77.0,76.0,76.0,75.0,76.0,75
.0,73.0,75.0,73.0,75.0,74.0,75.0,74.0,75.0,73.0,75.0,73.0,73.0,74.0,70.0,
72.0,70.0,67.0,67.0,69.0,70.0,68.0,69.0,69.0,66.0,65.0,68.0,62.0,75.0,70.
0,69.0,76.0,76.0,74.0,73.0,71.0,74.0,74.0,72.0,71.0,72.0,74.0,69.0,67.0,7
```

2.0,70.0,64.0,63.0,63.0,62.0,70.0,70.0,62.0,62.0,63.0,65.0,69.0,77.0,70.0
,74.0,69.0,72.0,71.0,69.0,71.0,71.0,72.0,72.0,69.0,70.0,66.0,65.0,69.0,68
.0,68.0,68.0,59.0,60.0,70.0,73.0,75.0,64.0,59.0,59.0,62.0,68.0,70.0,73.0,
79.0,75.0,65.0,70.0,74.0,70.0,70.0,71.0,71.0,71.0,69.0,61.0,67.0,65.0,72.
0,71.0,73.0,72.0,77.0,73.0,67.0,62.0,64.0,67.0,66.0,81.0,69.0,66.0,67.0,6
9.0,66.0,68.0,65.0,74.0,69.0,72.0,73.0,72.0,71.0,76.0,77.0,76.0,74.0,68.0
,73.0,71.0,74.0,75.0,70.0,67.0,71.0,67.0,74.0,77.0,78.0,67.0,70.0,69.0,69
.0,74.0,78.0,71.0,67.0,68.0,67.0,76.0,69.0,72.0,76.0,68.0,72.0,74.0,70.0,
67.0,72.0,60.0,65.0,75.0,70.0,75.0,70.0,79.0,75.0,70.0,67.0,74.0,70.0,75.
0,76.0,77.0,74.0,74.0,74.0,69.0,68.0,76.0,74.0,71.0,71.0,74.0,74.0,74.0,7
4.0,80.0,74.0,72.0,75.0,80.0,76.0,76.0,77.0,75.0,75.0,75.0,75.0,72.0,74.0
,74.0,74.0,76.0,74.0,75.0,73.0,79.0,75.0,72.0,72.0,74.0,72.0,72.0,77.0,71
.0,73.0,76.0,77.0,76.0,76.0,79.0,81.0,76.0,78.0,77.0,74.0,75.0,78.0,78.0,
69.0,72.0,74.0,74.0,76.0,80.0,80.0,76.0,76.0,76.0,77.0,77.0,77.0,82.0,75.
0,77.0,75.0,76.0,81.0,82.0,81.0,76.0,77.0,82.0,83.0,77.0,77.0,77.0,76.0,7
6.0,79.0]}

Great work! Next, let's create a route for the statistics analysis.