Surf’s Up:

Planning a vacation to Honolulu, Hawaii

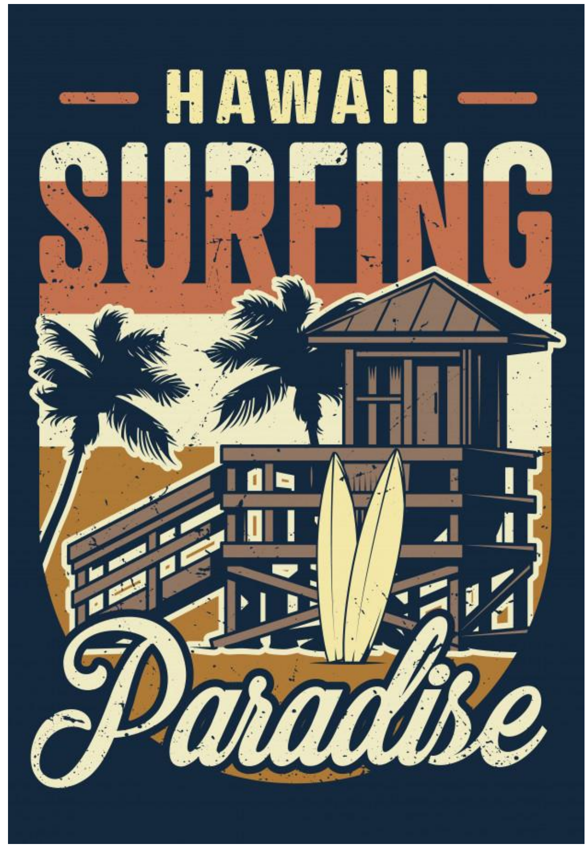

Fig 1. Hawaii Surfing Paradise

Introduction

It’s summer vacation time, which means letting loose and having fun. Hawaii is both the birthplace and the beating heart of surfing. I decided to leave the dark confines of my office cubicle and plan a sunny, surfing vacation to Hawaii. But first, with my newly acquired skills in ***SQLAlchemy***, I decided to do a climate analysis of the ***“Aloha State”.*** Hawaii, which covers the Hawaiian Islands, is tropical but experiences different climates depending on altitude and location. With its warm and gentle trade winds, mild temperatures and sunny skies, Hawaii is an ideal vacation destination throughout the year.

Vacation planning:

I decided to outline my planning project under following headings:

- **Climate Analysis and Exploration:** A basic analysis and exploration of ***“Hawaii. SQLite”*** database was done using SQLAlchemy ORM queries, Pandas and Matplotlib.
- **Precipitation Analysis:** Precipitation over last one year was obtained, sorted by date and loaded into a dataframe. ***Visualization: Bar plot***
- **Station Analysis:** Total number of stations, most active station and observed temperature over past one year from the most active station was retrieved. ***Visualization: Histogram***
- **Temperature Analysis I:** Is there a meaningful difference between the average temperatures in June vs. December from all stations across all available years ? ***Visualization: Box Plot***
- **Temperature Analysis II:** Minimum, maximum and average temperatures from all stations across the trip duration dates. ***Visualization: Box Plot***
- **Daily Rainfall Analysis:** Total precipitation and average observed temperature per weather station across the trip duration dates. ***Visualization: Line Plot***
- **Daily Normal:** Daily minimum, maximum and average temperatures from all stations across the trip duration dates. ***Visualization: Area Plot***
- **Climate App:** A Flask API was designed based on all the queries

Precipitation Analysis:

Using ***func.count()*** we counted the total number of dates in the database. There are 19550 dates with earliest being 2010-01-01 and the latest date in the database is 2017-08-23. Precipitation for past one year i.e. dates > 2016-8-23 (query_date) was obtain using the following code:

Precipitation_one_year = session.query(Measurement.date,Measurement.prcp).filter(Measurement.date >= query_date)

The result was loaded in a pandas dataframe (***df_prcp***). Fig.2 shows the head of the dataframe after setting the date column as the index. Fig. 3 shows a bar chart of daily precipitation from all stations spanning over the last twelve months in the database. From Fig.3 we can see September 2016 was the wettest month followed by more than 6 inches of precipitation in May 2017.

	Precipitation
Date	
2016-08-23	0.00
2016-08-24	0.08
2016-08-25	0.08
2016-08-26	0.00
2016-08-27	0.00

Figure 2. Head of Precipitation Dataframe

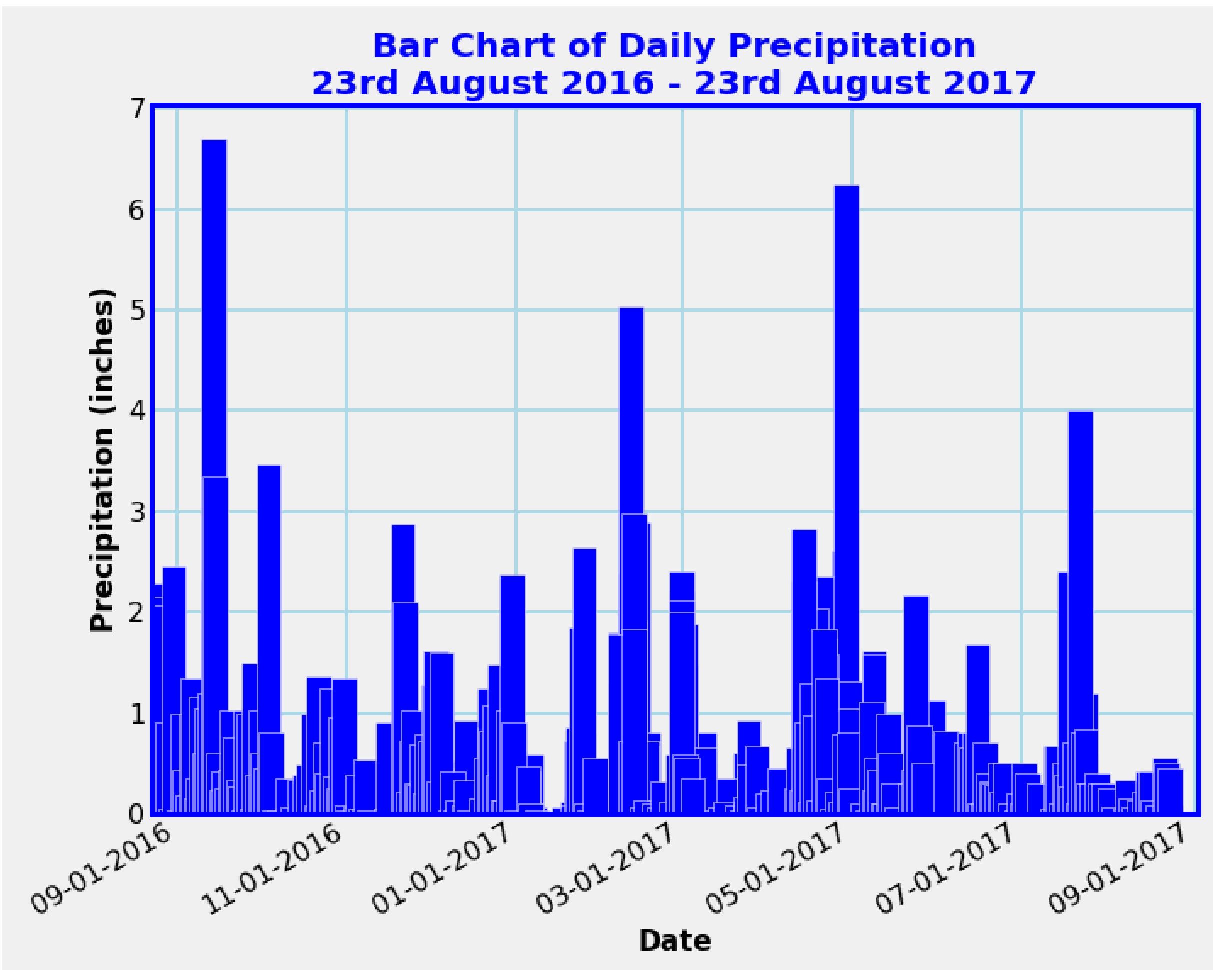

Figure 3. Bar chart of daily precipitation for past one year

Fig.4 shows a summary statistics table of the precipitation data using ***.describe()*** function of pandas.

	One_year_precipitation
count	2021.00
mean	0.18
std	0.46
min	0.00
25%	0.00
50%	0.02
75%	0.13
max	6.70

Figure 4. Statistics table of the precipitation data

Station Analysis:

Using ***func.count()*** we counted the number of distinct weather stations in the dataset. There are total of ***9 stations*** each being a distinct station. The most active station was found by applying the following query: *active_stations = session.query (Measurement.station, func.count(Measurement.station)).group_by (Measurement.station). order_by(func.count(Measurement.station).desc()).all()*

	Station	Count
0	USC00519281	2772
1	USC00519397	2724
2	USC00513117	2709
3	USC00519523	2669
4	USC00516128	2612
5	USC00514830	2202
6	USC00511918	1979
7	USC00517948	1372
8	USC00518838	511

Figure 5. Observation count of the weather stations

Fig.5 shows the observation count of all the stations arranged in a descending order. We can see that station ***# USC00519281*** is the most active station reporting ***2772 observations*** in the database.

The maximum, minimum and average temperatures reported by the most active station was found using the following queries: *Max_Temp = session.query(func.max(Measurement.tobs).label("Max_Temp")).filter(Measurement.station== most_active_station).first()* *Min_Temp = session.query(func.min(Measurement.tobs).label("Min_Temp")).filter(Measurement.station == most_active_station).first()* *Avg_Temp = session.query(func.avg(Measurement.tobs).label("Avg_Temp")).filter(Measurement.station == most_active_station).first()*

Fig.6 shows the results obtained from the queries loaded in a pandas dataframe. The most active station ***“USC00519281”*** also reported the highest number of temperature data.

	Most Active Station	Max. Temp(F)	Min. Temp(F)	Avg. Temp(F)
0	USC00519281	85.0	54.0	71.66

Figure 6. Maximum, minimum and average temperatures from the most active station

The temperature observations from the most active station were found using the following query: *Temp_one_year = session.query(Measurement.date, Measurement.tobs).filter(Measurement.date >= query_date). filter(Measurement.station == most_temp_station)*

Fig.7 shows a histogram of the observed temperatures reported by the most active station spanning over one year . We can see that temperatures mostly vary between 65F – 80F all year round. There are no freezing temperatures or extremely hot temperatures – making Hawaii a perfect vacation spot.

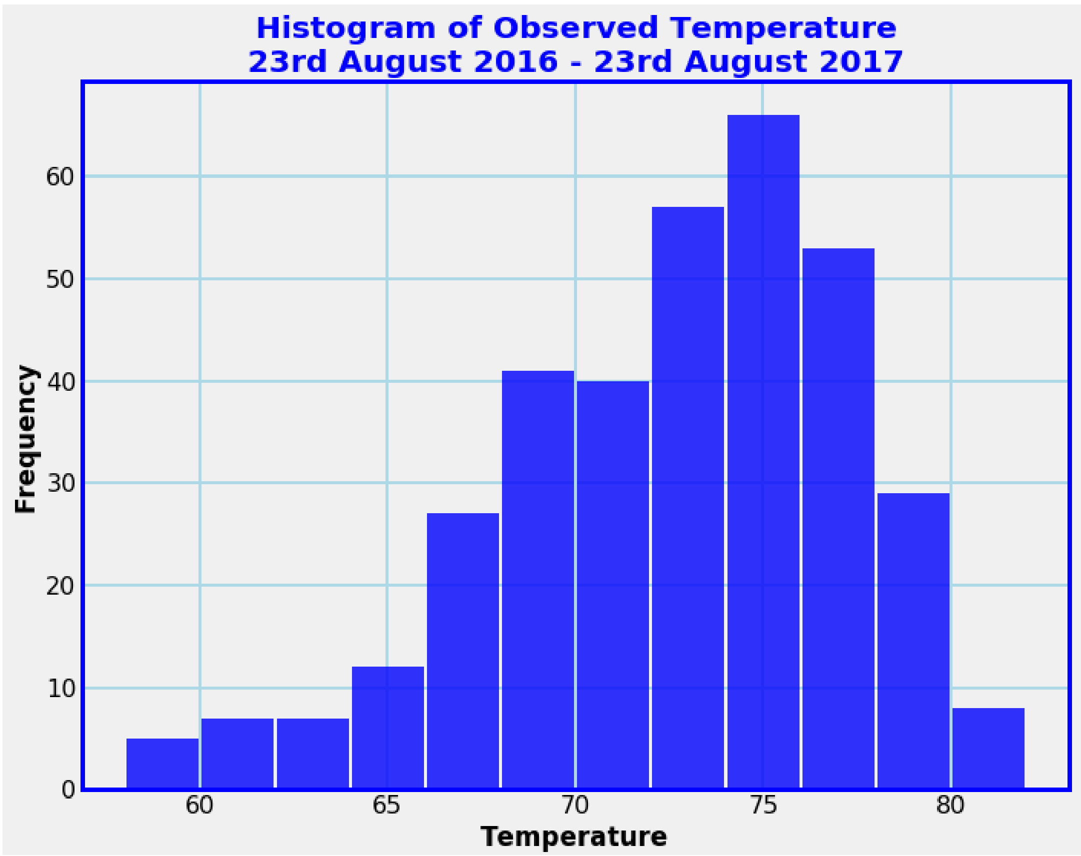

Figure 7. Histogram of temperatures from the most active station spanning over one year

Bonus: Temperature Analysis I

Hawaii is reputed to enjoy mild weather all year. Is there a meaningful difference between the temperature in, for example, June and December ? The following table was constructed using ***func.avg()*** query as follows: *June_weather= session.query (func.avg(Measurement.prcp), func.avg(Measurement.tobs)).filter (func.strftime("%m", Measurement.date) == "06").*

Month	Average Precipitation	Average Temperature
June	0.136	74.9
December	0.217	71.04

As we can see, December is wetter than June with average temperature about 4 degrees lower. June and December temperatures were retrieved from all the stations and loaded in a dataframe. Fig. 8 shows a box plot of the observed temperatures recorded in June and December.

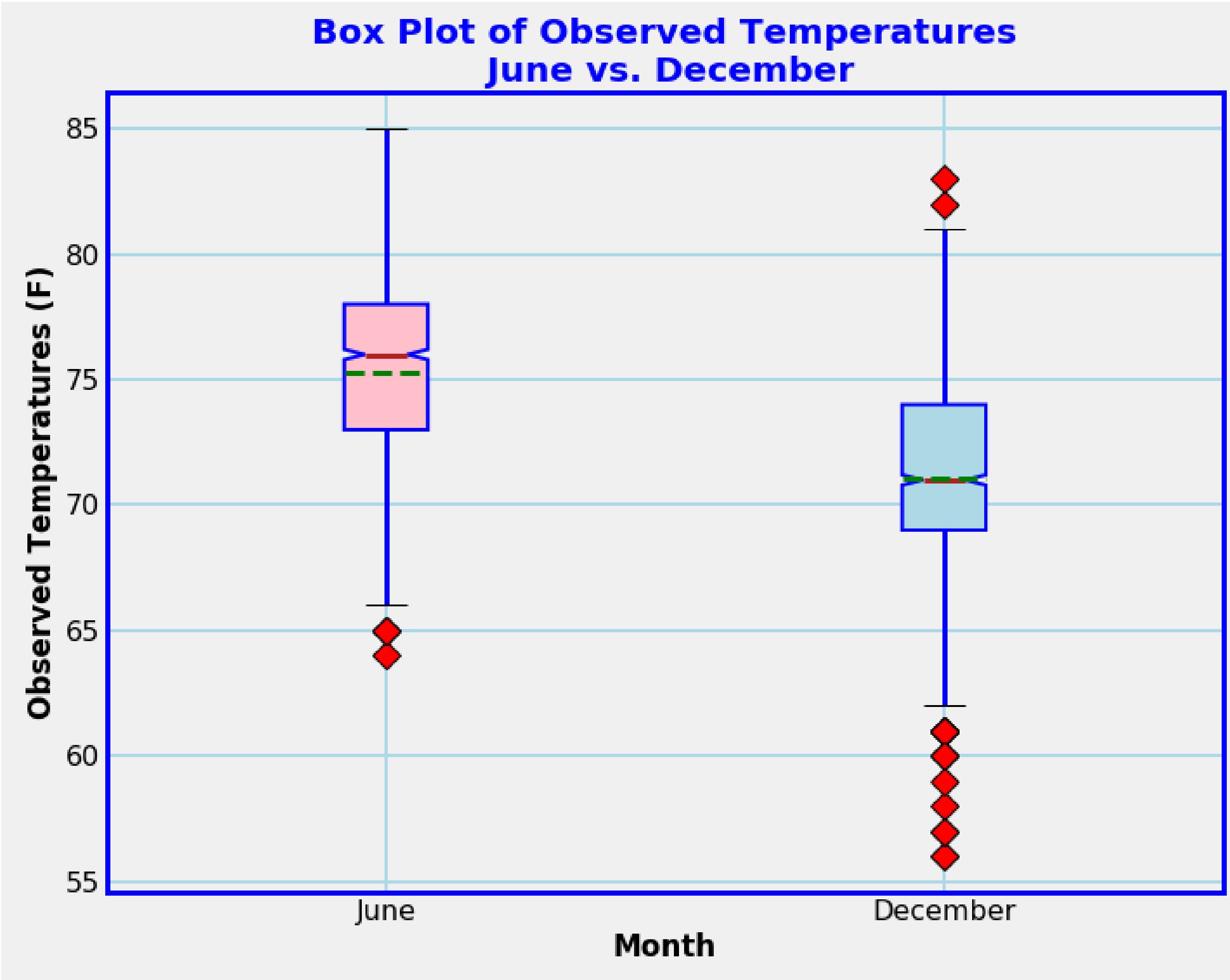

Figure 8. Box plot comparing temperatures recorded in June vs. December

In Fig. 8 the mean is shown as solid red line, median as dashed green line – there are more outliers in December as compared to June. An independent t-test between these two datasets yield a t-statistic as 33.85 and a ***p-value of 2.33e-213***. A p-value as low as this indicates that ***"null hypothesis of equal averages in rejected"***

Bonus: Temperature Analysis II

With precipitation and temperature analysis spanning over one year completed, it's time to fix our trip dates. Start and End dates of our vacation to the "Aloha state" was mapped to 3rd July 2017 and 17th July 2017 respectively. Minimum, maximum and average temperatures during our "mapped" vacation dates were retrieved using the following function:

def calc_temps(start_date, end_date):

return session.query(func.min(Measurement.tobs), func.avg(Measurement.tobs), func.max(Measurement.tobs)).filter(Measurement.date >= start_date).filter(Measurement.date <= end_date).all()

The result was loaded in a pandas dataframe. Fig. 9 shows a boxplot of average temperature during the trip duration (error bar shows the difference between the maximum and minimum temperatures)

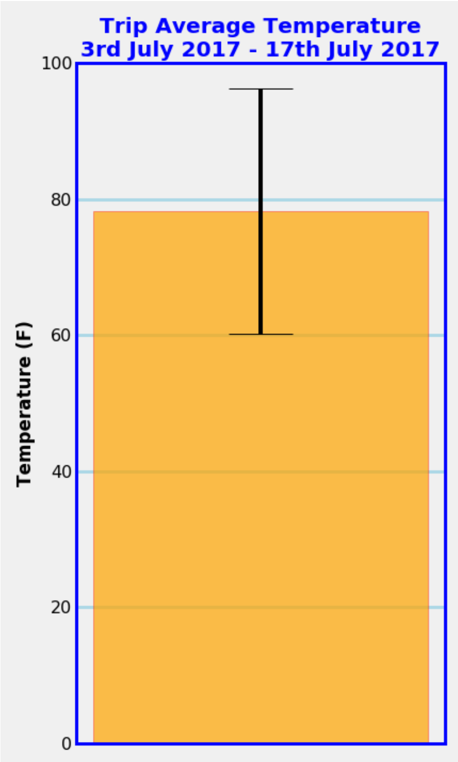

Figure 9. Box plot of average observed temperature during trip duration

Bonus: Daily Rainfall

As shown in Fig.9 the average temperature is 78F, minimum and maximum temperatures are 69F and 87F respectively. Which is perfect for a surfing vacation. But, dealing with rain during our vacation would be a bummer – so let us check the daily rainfall during those days. The following query was designed by joining the measurement and station tables to retrieve the total precipitation during our trip duration.

```
# Let's design a list of query items
query_items = [func.sum(Measurement.prcp),Station.station,Station.name, Station.latitude,Station.longitude,Station.elevation]
# Let's create the query by joining Measurement & Station tables
results = session.query(*query_items).group_by(Measurement.station).order_by(func.sum(Measurement.prcp).desc()).outerjoin(Station, Station.station==Measurement.station). filter(Measurement.date >= start_date) .filter(Measurement.date <= end_date).all()
```

Similarly, querying *avg.(Measurement.tobs)* we retrieved the average temperature during our trip duration. Fig.10 shows the first 5 rows of the two data frames obtained by executing the queries.

	Total_prcp	Station_id	Station_name	Latitude	Longitude	Elevation
0	3.16	USC00516128	MANOA LYON ARBO 785.2, HI US	21.33310	-157.80250	152.4
1	1.72	USC00519281	WAIHEE 837.5, HI US	21.45167	-157.84889	32.9
2	0.91	USC00513117	KANEOHE 838.1, HI US	21.42340	-157.80150	14.6
3	0.67	USC00514830	KUALOA RANCH HEADQUARTERS 886.9, HI US	21.52130	-157.83740	7.0
4	0.22	USC00519397	WAIKIKI 717.2, HI US	21.27160	-157.81680	3.0
5	0.13	USC00519523	WAIMANALO EXPERIMENTAL FARM, HI US	21.33556	-157.71139	19.5
6	NaN	USC00517948	PEARL CITY, HI US	21.39340	-157.97510	11.9

	Avg_Temp	Station_id	Station_name	Latitude	Longitude	Elevation
0	80.625000	USC00517948	PEARL CITY, HI US	21.39340	-157.97510	11.9
1	80.461538	USC00514830	KUALOA RANCH HEADQUARTERS 886.9, HI US	21.52130	-157.83740	7.0
2	80.428571	USC00519523	WAIMANALO EXPERIMENTAL FARM, HI US	21.33556	-157.71139	19.5
3	79.666667	USC00519397	WAIKIKI 717.2, HI US	21.27160	-157.81680	3.0
4	77.000000	USC00513117	KANEOHE 838.1, HI US	21.42340	-157.80150	14.6
5	75.800000	USC00519281	WAIHEE 837.5, HI US	21.45167	-157.84889	32.9

Figure 10. Heads of Total precipitation & Average temperature dataframes.

Are the average temperature and total precipitation related to station elevation?

Fig.11 shows average temperature and total precipitation plotted as function of station elevation.

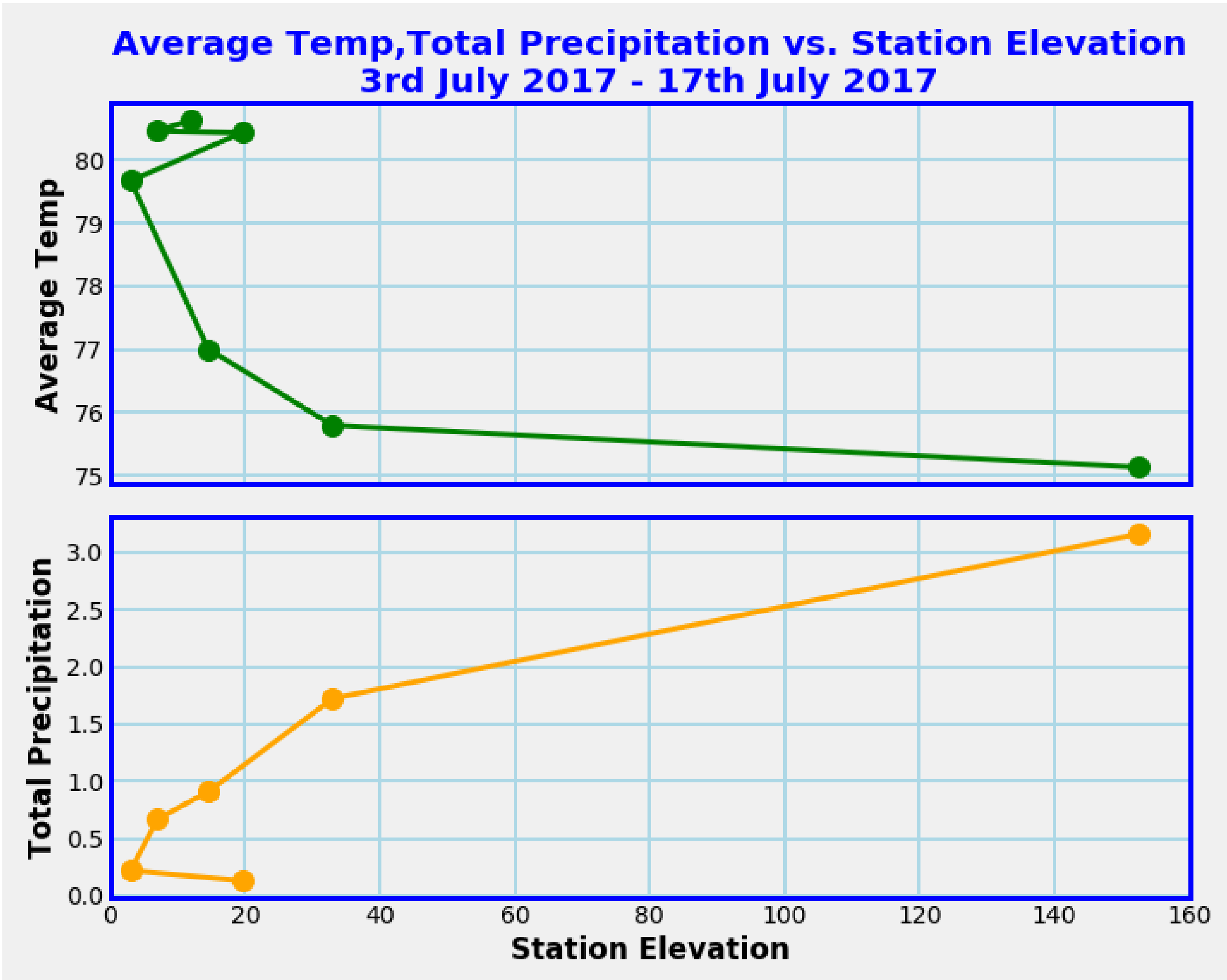

Figure 11. Average temperature and Total precipitation plotted vs. Station elevation

Fig. 11 shows a very interesting trend – total precipitation ***increases*** with increasing elevation while average temperature ***decreases*** with increasing station elevation.

Bonus: Daily Temperatures: We completed a daily analysis of precipitation; our next step would be to analyze daily temperatures during our trip duration. The following function returns a tuple of minimum, average and maximum observed temperature for a particular date.

```
def daily_normals(date):
    sel = [func.min(Measurement.tobs), func.avg(Measurement.tobs), func.max(Measurement.tobs)]
    return session.query(*sel).filter(func.strftime("%m-%d", Measurement.date) == date).all()
```

After creating a list of dates corresponding to our trip duration, results were retrieved for each date using the above function.

Fig. 12 shows a dataframe comprising of minimum, average and maximum temperatures for each day from 3rd July 2017 to 17th July 2017. An area plot of the dataframe is shown in Fig.13.

Conclusions: Our analysis using SQLAlchemy and python resulted in following insights:

- Precipitation analysis over 12 months showed September is the wettest month
- Observations from the most active station shows that temperature remains close to 75F all year round.
- Small yet noticeable difference between the mean temperatures from June vs. mean temperature from December
- Average temperature decreases while total precipitation increases with increasing station elevation
- Avg. temperature is close to 75F, min. temperature averages 68F and max. temperatures averages close to 82F during the span of the vacation dates.

	Min_Temp	Avg_Temp	Max_Temp
2017-07-03	68.0	75.320755	87.0
2017-07-04	70.0	76.571429	81.0
2017-07-05	66.0	75.000000	81.0
2017-07-06	69.0	75.000000	81.0
2017-07-07	69.0	74.910714	82.0
2017-07-08	69.0	76.083333	83.0
2017-07-09	68.0	76.192982	83.0
2017-07-10	68.0	75.620690	82.0
2017-07-11	70.0	75.789474	81.0
2017-07-12	67.0	75.964912	82.0
2017-07-13	69.0	76.189655	81.0
2017-07-14	61.0	76.228070	82.0
2017-07-15	68.0	76.254237	82.0
2017-07-16	70.0	76.344828	83.0
2017-07-17	70.0	76.301887	83.0

Figure 12. Temperature data for each date spanning vacation duration

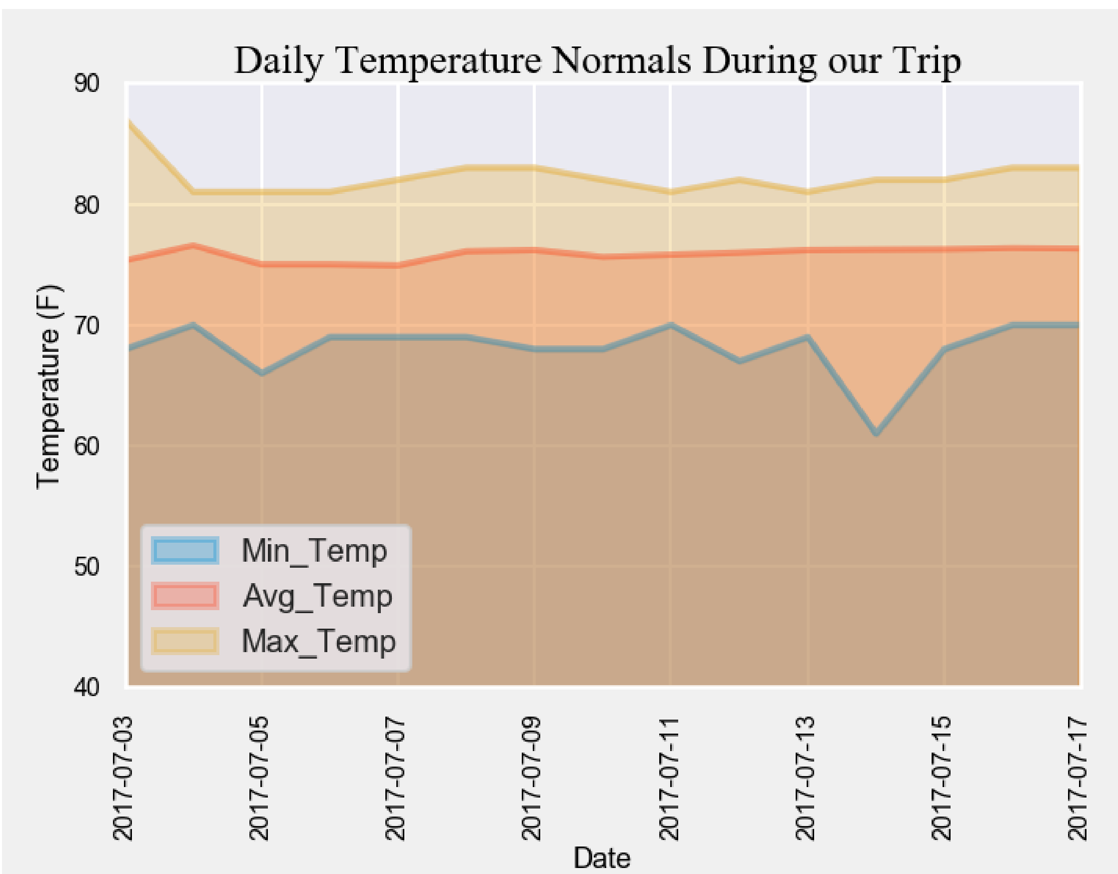

Figure 13. Area plot of daily temperatures spanning vacation duration