

6.5.3

Get Vacation Criteria

Now, the cherry on top of the project: a feature on the app that allows customers to search for locations they want to travel based on their temperature preferences.

Your clients will enter their preferences. Your code will tell them where to travel. It's beautiful, simple, and just needs a few more lines of code to actually work.

For the app we are creating, we need to prompt the user to enter the minimum and maximum temperature ranges as floating-point decimal numbers to filter the `city_data_df` DataFrame.



REWIND

Recall that you use the `input()` statement to prompt a user to enter information. To convert the input to a floating-point decimal number, wrap the `input()` statement with the `float()` method, like this:

```
float(input())
```

In a new cell, we'll write two input statements for the app that will prompt the customer to add minimum and maximum temperature values and convert these values to floating-point decimals.

The statement can be whatever you want it to be, but for our app, we'll prompt the customer with the following two input statements.

```
er to add a minimum and maximum temperature value.  
(input("What is the minimum temperature you would like for your trip? "))
```

```
(input("What is the maximum temperature you would like for your trip? "))
```

Add these two input statements to a new cell and run the cell. When we run this cell, the customer will be prompted to enter the minimum temperature.

```
# Ask the customer to add a minimum and maximum temperature value.
min_temp = float(input("What is the minimum temperature you would like for your trip? "))
max_temp = float(input("What is the maximum temperature you would like for your trip? "))
```

What is the minimum temperature you would like for your trip?

Add 75 as the minimum temperature. After the customer enters the minimum temperature and presses Enter, he or she will be prompted to enter a maximum temperature.

```
# Ask the customer to add a minimum and maximum temperature value.
min_temp = float(input("What is the minimum temperature you would like for your trip? "))
max_temp = float(input("What is the maximum temperature you would like for your trip? "))
```

What is the minimum temperature you would like for your trip?

For the maximum temperature, add 90 and press Enter.

```
# Ask the customer to add a minimum and maximum temperature value.
min_temp = float(input("What is the minimum temperature you would like for your trip? "))
max_temp = float(input("What is the maximum temperature you would like for your trip? "))
```

What is the minimum temperature you would like for your trip? 75

What is the maximum temperature you would like for your trip? 90

Next, filter the maximum temperature column in the `city_data_df` DataFrame using logical operators to create a new DataFrame with the cities that meet the customer's criteria.



REWIND

Recall that you use the `loc[]` method on the current DataFrame to create a new DataFrame from a current DataFrame. Inside the brackets, add the conditional filter.

The conditional filter will be `city_data_df["Max Temp"] <= max_temp) & (city_data_df["Max Temp"] >= min_temp)`. Also, filter the `city_data_df` DataFrame using the following statement.

```
# Filter the dataset to find the cities that fit the criteria.
preferred_cities_df = city_data_df.loc[(city_data_df["Max Temp"] <= max_t
                                         (city_data_df["Max Temp"] >= min_t
preferred_cities_df.head(10)
```

The `preferred_cities_df` DataFrame will contain all the cities that meet the temperature criteria.

	City_ID	City	Country	Date	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed
7	7	Avarua	CK	2019-08-08 22:26:22	-21.21	-159.78	77.00	78	90	6.93
13	13	Vaini	IN	2019-08-08 22:26:24	15.34	74.49	77.30	91	100	8.88
16	16	Dingle	PH	2019-08-08 22:26:24	11.00	122.67	75.43	90	99	11.18
21	21	Vinh Yen	VN	2019-08-08 22:26:26	21.31	105.60	77.00	100	40	3.36
28	28	Mazagao	BR	2019-08-08 22:26:28	-0.12	-51.29	84.20	79	75	5.82
29	29	Marsaxlokk	MT	2019-08-08 22:26:28	35.84	14.54	82.00	88	0	4.70
30	30	Portland	US	2019-08-08 22:24:00	43.66	-70.25	82.00	78	1	6.93
32	32	Camacha	PT	2019-08-08 22:26:29	33.08	-16.33	77.00	88	40	2.24
34	34	Prata	BR	2019-08-08 22:26:29	-19.31	-48.92	76.75	41	0	2.64
45	45	Albany	US	2019-08-08 22:21:44	42.65	-73.75	75.99	68	75	11.41

Before moving on, determine if the `preferred_cities_df` DataFrame has any null values for any of the rows.

How would you determine if there are any null values for the rows in the `preferred_cities_df` DataFrame? (Select all that apply)

- ☒ `preferred_cities_df.count()` ✓
- ☐ `preferred_cities_df.counts()`
- ☒ `preferred_cities_df.isnull().sum()` ✓
- ☒ `preferred_cities_df.notnull().sum()` ✓

Feedback

Correct. Nice work!

Finish ►

Add the following code to a new cell and run the cell to determine if there are any null values.

```
preferred_cities_df.count()
```

For the above DataFrame, there are 180 cities and no null values.

```
preferred_cities_df.count()
```

```
City_ID      180
City         180
Country      180
Date         180
Lat          180
Lng          180
Max Temp     180
Humidity     180
Cloudiness   180
Wind Speed   180
dtype: int64
```

IMPORTANT

Consider the following guidance:

1. Depending on the time of year and the seasons, you might have to adjust the minimum and maximum temperature to get enough cities.
2. It is a good idea to keep the number of cities to fewer than 200 to make it easier to plot the markers on the heatmap.
3. If you have some rows with null values, you'll need to drop them using the `dropna()` method at the end of your filtering statement when you are creating the new DataFrame.

Now that we have all the cities the customer wants to travel to, they will need to find a hotel to stay in the city.

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.