

911 Calls Project

Following are the columns for the dataset

String variable, Latitude

- lng: String variable, Longitude
- desc: String variable, Description of the Emergency Call
- zip: String variable, Zipcode
- title: String variable, Title
- timeStamp: String variable, YYYY-MM-DD HH:MM:SS
- twp: String variable, Township
- addr: String variable, Address
- e: String variable, Dummy variable (always 1)

1) What are the top 5 zipcodes for 911 calls?

2) ** In the titles column there are "Reasons/Departments" specified before the title code. These are EMS, Fire, and Traffic. Use `.apply()` with a custom function to create a new column called "Reason" that contains this string value.**

**For example, if the title column value is EMS: BACK PAINS/INJURY , the Reason column value would be EMS. **

3) What is the most common Reason for a 911 call based off of this new column?

4) Now use seaborn to create a countplot of 911 calls by Reason

5) What is the data type of the objects in the timeStamp column?

6) You should have seen that these timestamps are still strings. Use [pd.to_datetime](#) to convert the column from strings to DateTime objects.

7) ** You can now grab specific attributes from a Datetime object by calling them. For example:**

```
time = df['timeStamp'].iloc[0]
time.hour
```

You can use Jupyter's tab method to explore the various attributes you can call. Now that the timestamp column are actually DateTime objects, use `.apply()` to create 3 new columns called Hour, Month, and Day of Week. You will create these columns based off of the timeStamp column.

8) Notice how the Day of Week is an integer 0-6. Use the `.map()` with this dictionary to map the actual string names to the day of the week:

```
dmap = {0:'Mon',1:'Tue',2:'Wed',3:'Thu',4:'Fri',5:'Sat',6:'Sun'}
```

9) Now use seaborn to create a countplot of the Day of Week column with the hue based off of the Reason column

10) You should have noticed it was missing some Months, let's see if we can maybe fill in this information by plotting the information in another way, possibly a simple line plot that fills in the missing months, in order to do this, we'll need to do some work with pandas.

Now create a groupby object called `byMonth`, where you group the DataFrame by the month column and use the `count()` method for aggregation. Use the `head()` method on this returned DataFrame

11) Now create a simple plot off of the dataframe indicating the count of calls per month

12) Now see if you can use seaborn's `lplot()` to create a linear fit on the number of calls per month. Keep in mind you may need to reset the index to a column.

13) *Create a new column called 'Date' that contains the date from the timeStamp column. You'll need to use apply along with the .date() method.*

14) Now groupby this Date column with the `count()` aggregate and create a plot of counts of 911 calls

15) Now recreate this plot but create 3 separate plots with each plot representing a Reason for the 911 call

16) Now let's move on to creating heatmaps with seaborn and our data. We'll first need to restructure the dataframe so that the columns become the Hours and the Index becomes the Day of the Week