HW 3 Manually Graded: Upload PDF Here

● Graded

Student

Rey Stone

Total Points

16 / 17 pts

Question 1

Question 2c

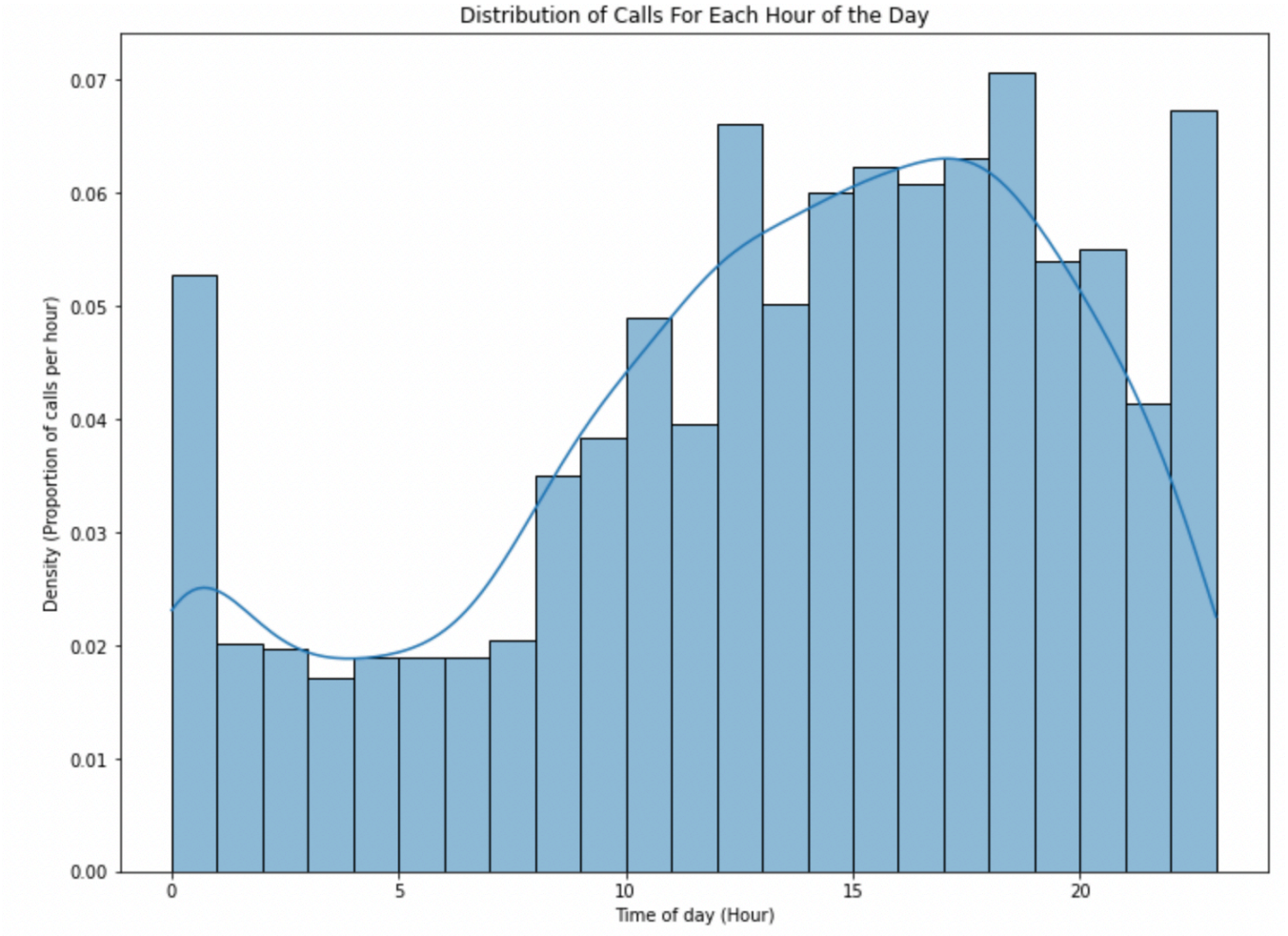
3 / 3 pts

2c (3pts)

Correct:

```
# BEGIN SOLUTION

ax = sns.histplot(calls, x = 'Hour', stat='density',kde=True, label='casual', bins=23)
ax.set_xlabel("Time of day (Hour)")
ax.set_ylabel("Density (Proportion of calls per hour)")
ax.set_title("Distribution of Calls For Each Hour of the Day")
```



✔ **– 0 pts** Correct answer

Question 2

Question 2e

3.5 / 4 pts

2e) i. (1pts)

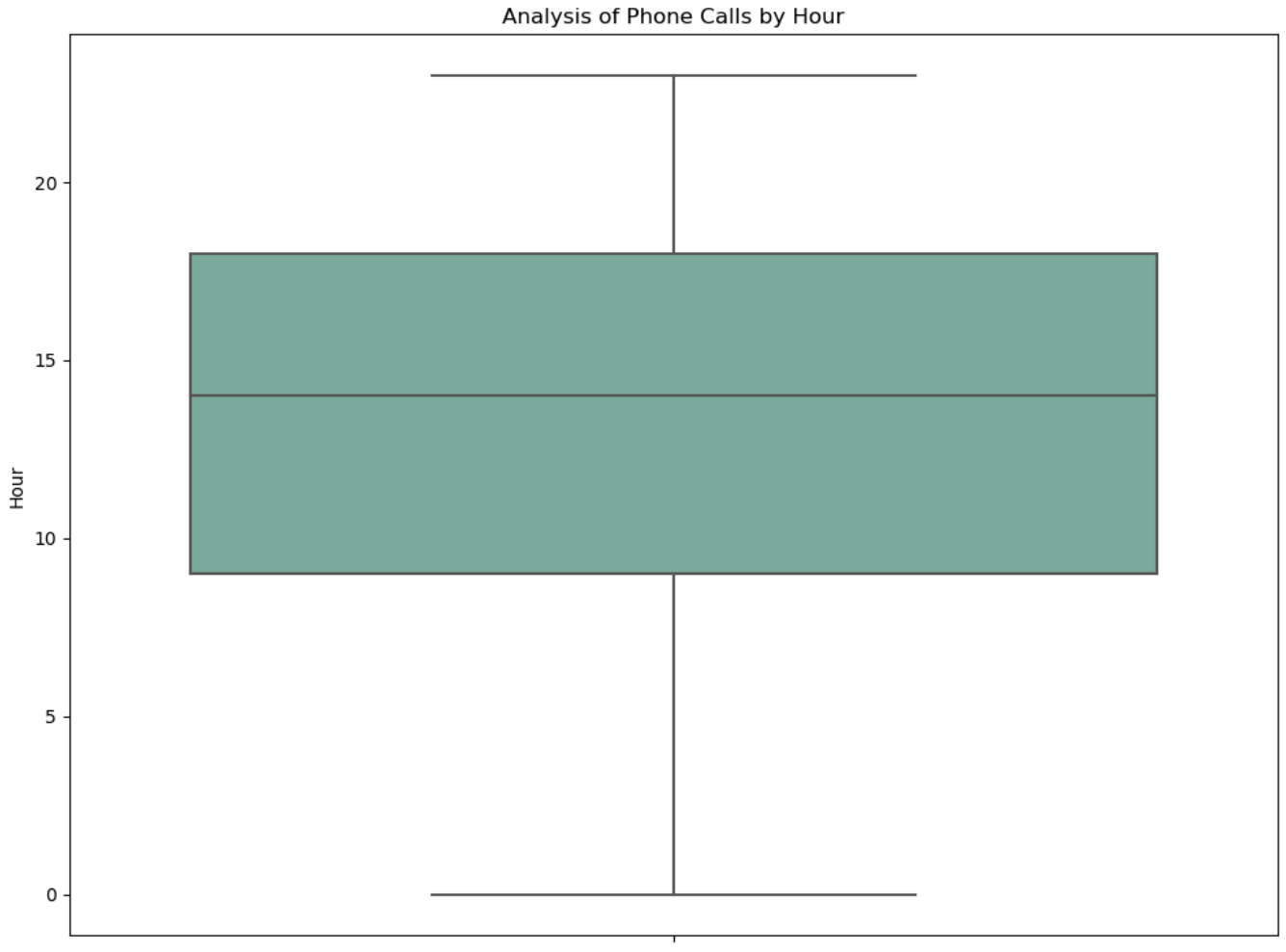
Correct:

```
# BEGIN SOLUTION

ax = sns.boxplot(data=calls,
                 y="Hour",
                 saturation=0.5, palette="Set2")
ax.set_title("Analysis of Phone Calls by Hour");



# END SOLUTION
# Your code for boxplot above this line
```



✔ **– 0 pts** Correct answer

2e) ii. (3pts)

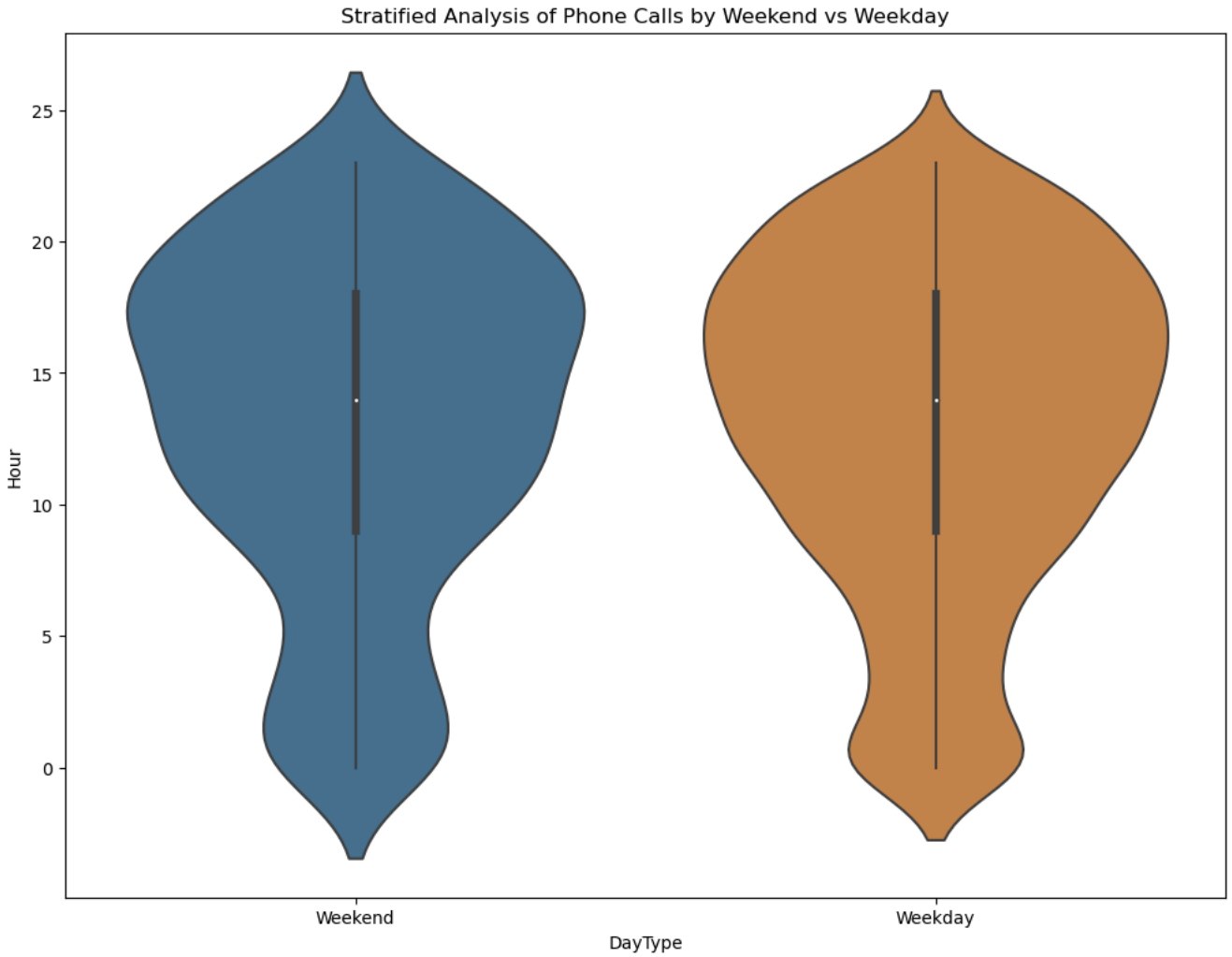
Correct:

```
# BEGIN SOLUTION


ax = sns.violinplot(data=calls.sort_values("CVDOW"),
                    x="DayType", y="Hour",
                    saturation=0.5)
ax.set_title("Stratified Analysis of Phone Calls by Weekend vs Weekday");



# END SOLUTION
# Your code for side-by-side violin plots above this line
```



✔ **– 0.5 pts** Missing title

Question 3

Question 2f

3 / 3 pts

2f) (2pts)

Correct:

- Are there more calls in the day or at night?
  --There are more calls in the day.
- What are the most and least popular times?
  --The most popular times are around 15-16 (i.e. 3pm-4pm). The least popular times are early morning (4am-5am)
- Do call patterns and/or IQR vary by weekend vs weekday?
  --No, not by much.

✔ **– 0 pts** Correct answer

Question 4

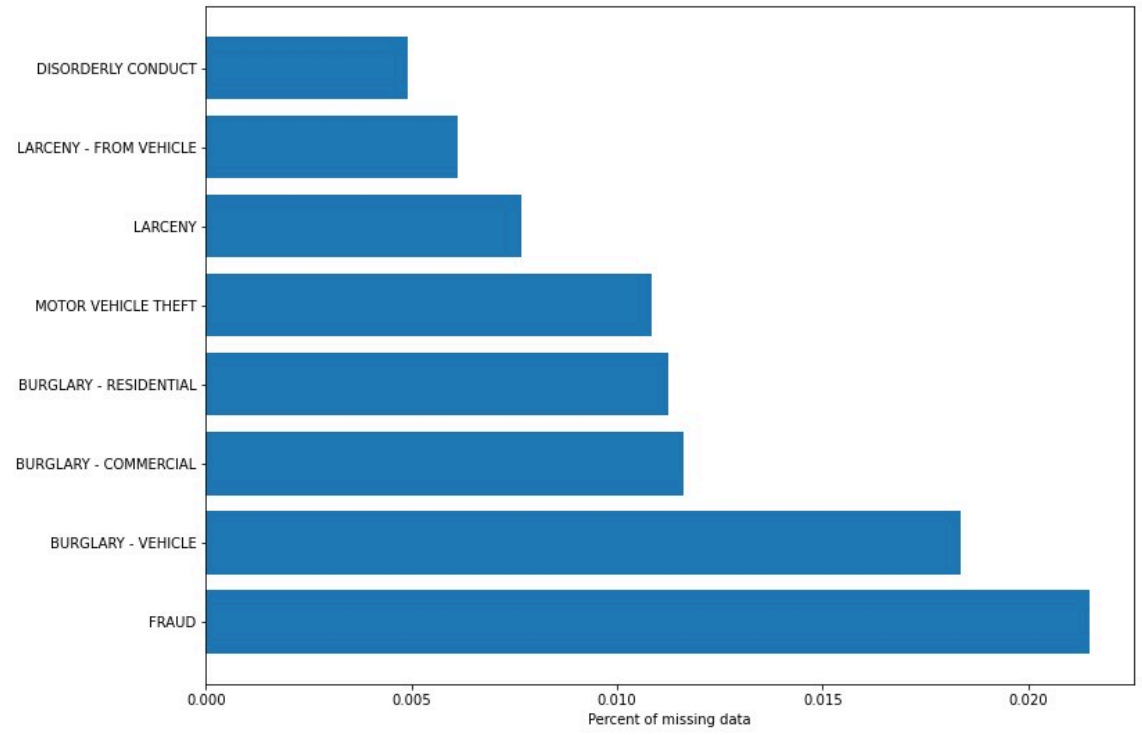
Question 3c

1.5 / 2 pts

3c ii) (2 pts)

```
1
2 plt.barh(missing_by_crime.index, missing_by_crime)
3 plt.xlabel("Percent of missing data")
4
5 # Your code to create the barplot above this line
```

```
Text(0.5, 0, 'Percent of missing data')
```



✔ **– 0.5 pts** Missing or incorrect label on x-axis

Question 5

Question 3d

3 / 3 pts

3d

SOLUTION: While some dates have more unlabeled data than others, it seems that a small percentage of Burglary and Fraud calls don't have GPS coordinates.

Because these account for such a small percentage of overall crimes, it seems reasonable to drop this data.

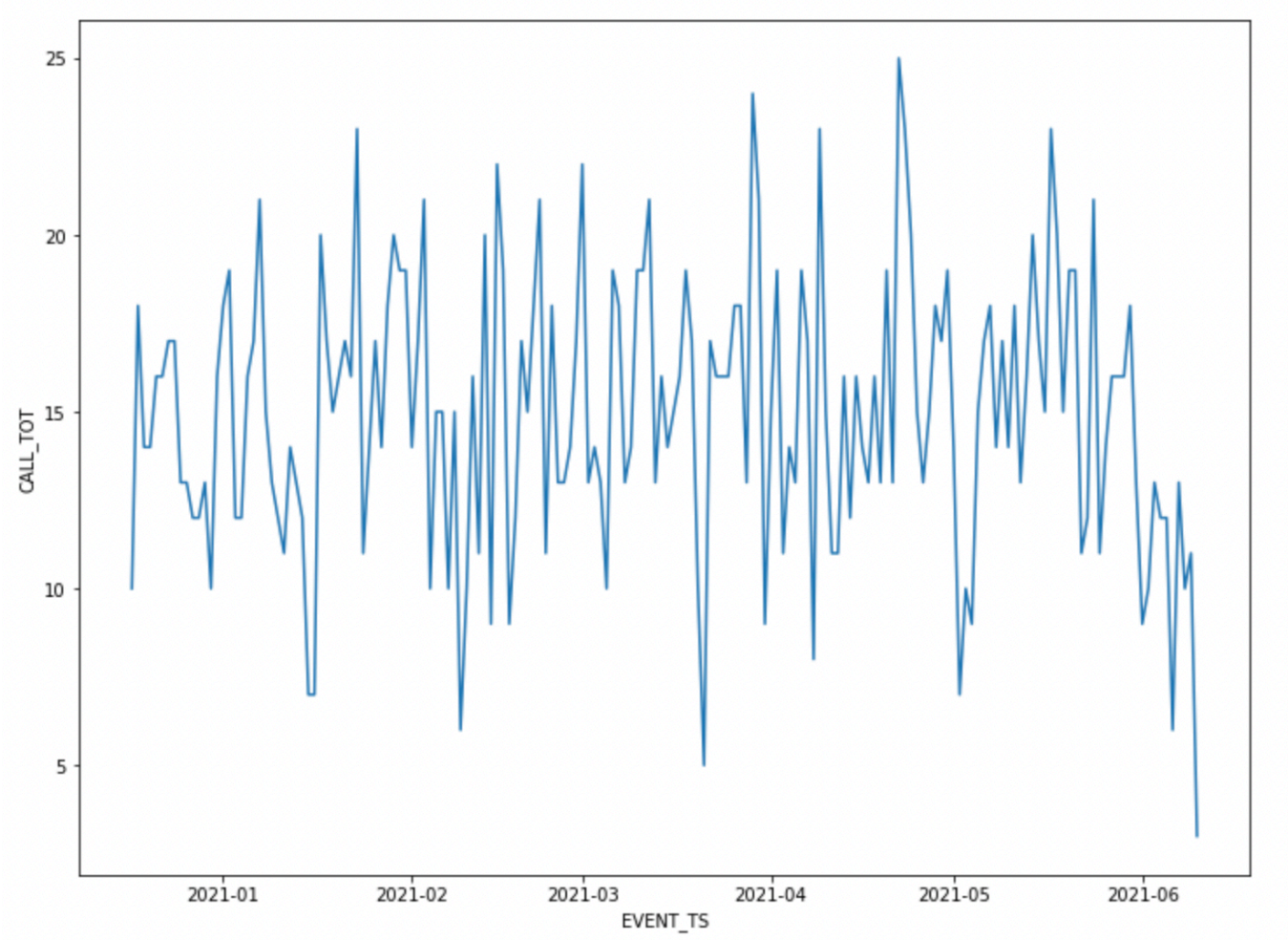
✔ **– 0 pts** Correct

Question 6

Question 3e

2 / 2 pts

3e (2 pts):



✔ **– 0 pts** Correct

Question assigned to the following page: 1

0.1 Question 2c (3 pts)

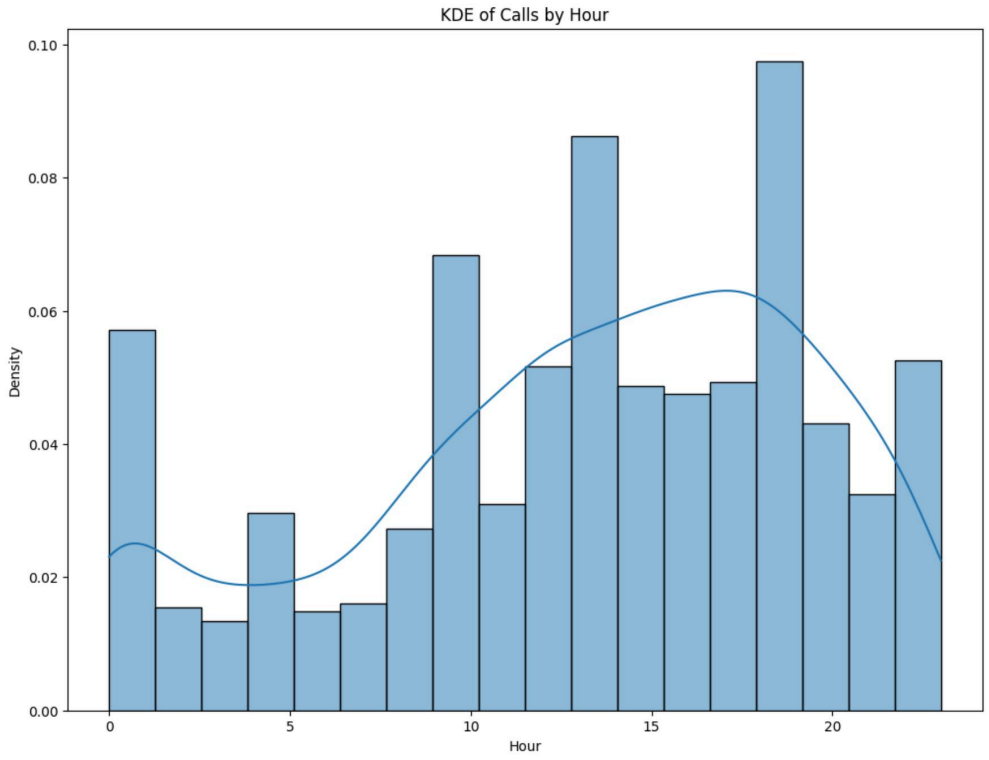
Use seaborn to create a **density** histogram showing the distribution of calls by hour.
Include the Kernal Density Estimate (KDE) graph on your histogram.

Be sure that your axes are labeled and that your plot is titled.

```
In [80]: sns.histplot(data=calls, x="Hour", stat="density", kde=True).set_title("KDE of Calls by Hour")

         # Your code above this line

         # Leave this for grading purposes
         ax_3d = plt.gca()
```



Question assigned to the following page: 2

0.2 Question 2e (4 pts)

i). Use seaborn to construct a box plot showing the distribution of calls by hour.

ii). To better understand the time of day a report occurs we could **stratify the analysis by DayType (i.e. by weekday vs weekends).**

To do this, use seaborn to create side-by-side violin plots comparing the distribution of calls by hour on the weekend vs weekday (here's a reminder of how to create violin plots in Seaborn: https://seaborn.pydata.org/generated/seaborn.violinplot.html)

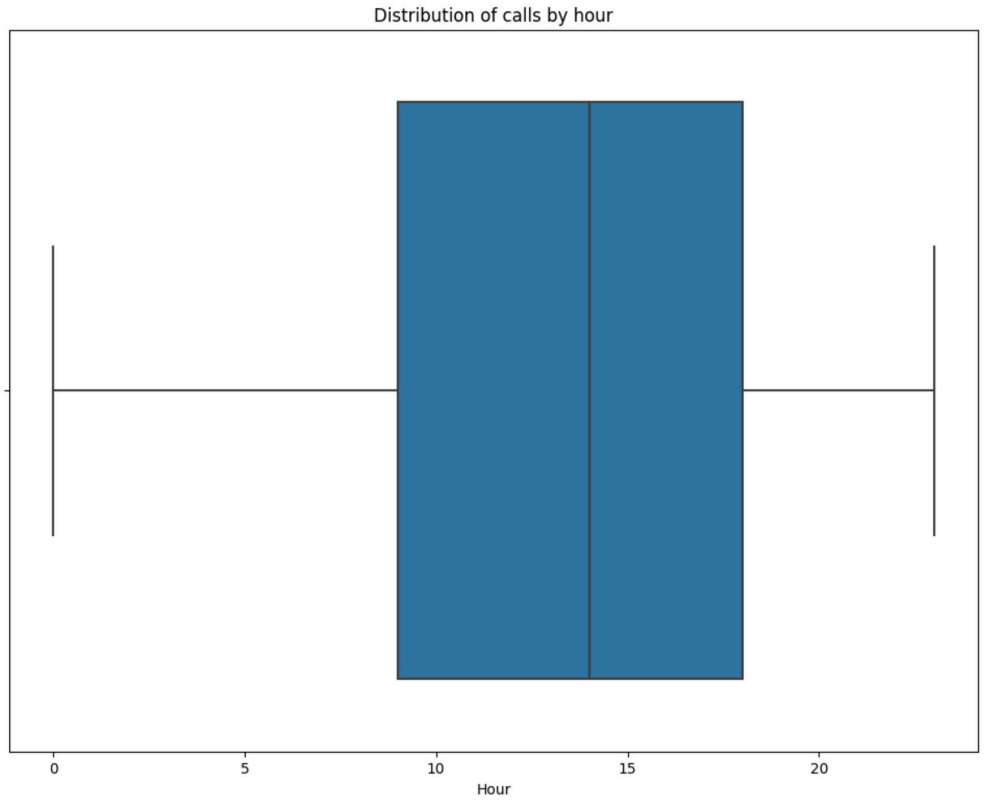
Note: For aesthetic purposes only the violin plot continues past the end of the whiskers (i.e. past 0 and 24 hours); however it is not possible to get data points outside of the whiskers for this distribution.

In [83]:
```
sns.boxplot(data=calls, x='Hour').set_title("Distribution of calls by hour")

        # Your code for boxplot above this line
        # Include a title and label both axes
```

Out[83]:
```
Text(0.5, 1.0, 'Distribution of calls by hour')
```

Question assigned to the following page: 2

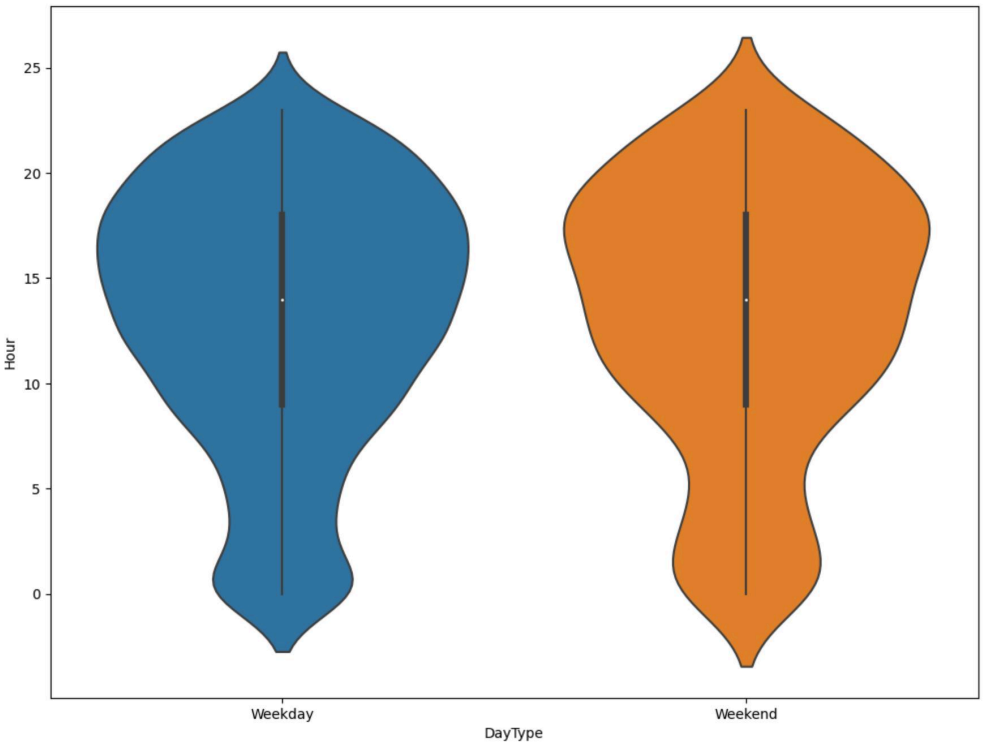



```
In [84]: sns.violinplot(data=calls,x='DayType', y='Hour')

         # Your code for side-by-side violin plots above this line
         # Include a title and label both axes
```

```
Out[84]: <Axes: xlabel='DayType', ylabel='Hour'>
```

Question assigned to the following page: 2


25
20
15
10
5
0
Hour
Weekday
Weekend
DayType

Question assigned to the following page: 3

0.3 Question 2f (3 pts)

Based on your histogram, boxplot, and violin plots above, what observations can you make about the patterns of calls? Answer each of the following questions:

i). Are there more calls in the day or at night?

ii). Approximately which hour of the day is most popular? (for example 8-9 in the morning)

iii). Approximately which hour of the day is least popular?

iv). Is there a major difference in call patterns and/or IQR between weekends and weekdays? Explain your answer.

Answers

i.) There are more calls in the afternoon between the times of 10-15 (10am-3pm).

ii.) The hour of the day that is the most popular is around late 2pm.

iii.) The time of day that is the least popular seems to be 3am.

iv.) There are no major differences between weekdays and weekends. The most calls still happen around 10am-3pm with the early morning (3am) being the least popular.

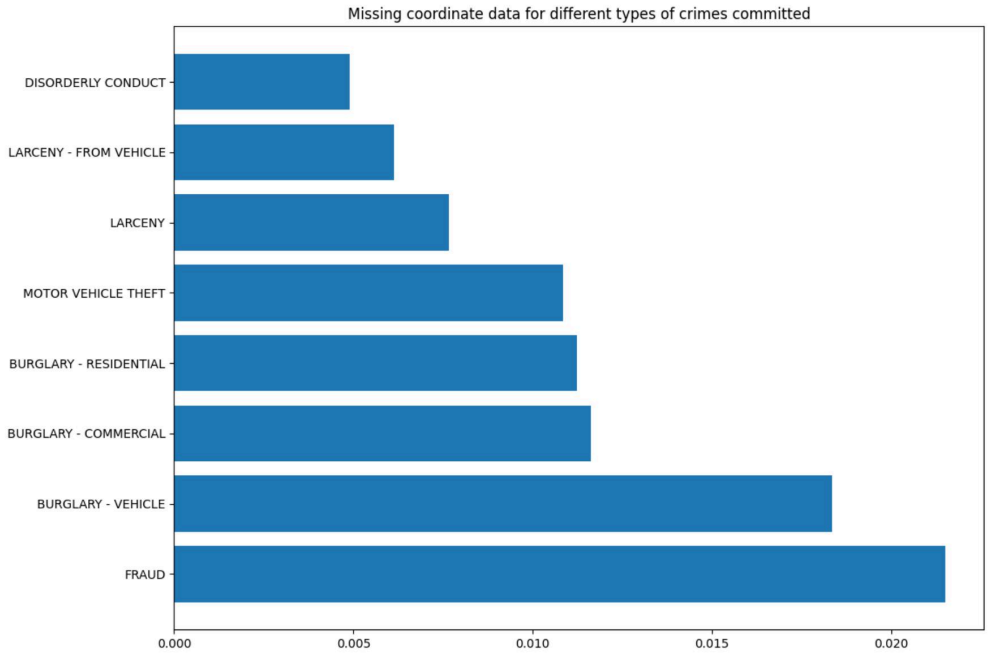
Question assigned to the following page: 4

```
In [94]: missing_by_crime = missing_lat_lon.groupby('CVLEGEND').size()/calls.groupby('CVLEGEND').size()
         missing_by_crime = missing_by_crime.dropna()
         missing_by_crime = missing_by_crime.sort_values(ascending=False)
         # Your code above this line
         missing_by_crime
```

```
Out[94]: CVLEGEND
         FRAUD                       0.021505
         BURGLARY - VEHICLE          0.018349
         BURGLARY - COMMERCIAL       0.011628
         BURGLARY - RESIDENTIAL      0.011236
         MOTOR VEHICLE THEFT         0.010830
         LARCENY                     0.007673
         LARCENY - FROM VEHICLE      0.006135
         DISORDERLY CONDUCT          0.004902
         dtype: float64
```

```
In [100]: plt.barh(missing_by_crime.index,missing_by_crime.values);
          plt.title("Missing coordinate data for different types of crimes committed")
          # Your code to create the barplot above this line
```

```
Out[100]: Text(0.5, 1.0, 'Missing coordinate data for different types of crimes committed')
```



Question assigned to the following page: 4

```
In [101]: grader.check("q3c")
```

```
Out[101]: q3c results: All test cases passed!
```

Question assigned to the following page: 5

0.4 Question 3d (3 pts)

Based on the plots above, give your recommendation as to how we should handle the missing data, and justify your answer:

Option 1). Drop rows with missing data

Option 2). Set missing data to NaN

Option 3). Impute data

Answer

Option 2.

Explanation:

I think that the most reasonable thing to do is to have the missing data as NaN. While the coordinates of where a specific crime occured is integral information, it's more important to at least have a general record of crimes that happen even if you cannot pinpoint it to a specific location. The other data such as crime type, time of crime, etc. is still very important to keep. Thus these records should not only be kept on the basis of coordinates.

Question assigned to the following page: 6

0.5 Question 3e (3 pts)

We'll end by visualizing the number of calls as a function of time, to see if there are any seasonal patterns in call volume during the time period the data spans.

Start by grouping the `calls` dataframe to create a new dataframe called `count_by_date`, with index `EVENT_TS` and column `CALL_TOT` that gives the number of calls on that specific date, sorted in chronological order.

The first 5 rows of your `count_by_date` dataframe should be:

EVENT_TS	CALL_TOT
2020-12-17	10
2020-12-18	18
2020-12-19	14
2020-12-20	14
2020-12-21	16

Then create a lineplot displaying this data (with x-axis equal to `EVENT_TS` and y-axis equal to `CALL_TOT`). Be sure to label your axes.

```
In [103]: count_by_date = calls.set_index("EVENT_TS").sort_values(by='EVENT_TS',ascending=True)

          count_by_date = count_by_date.assign(CALL_TOT=count_by_date.groupby('EVENT_TS').size())

          count_by_date = count_by_date.groupby("EVENT_TS")["CALL_TOT"].size()

          count_by_date = count_by_date.to_frame()

          count_by_date.head(5)
```

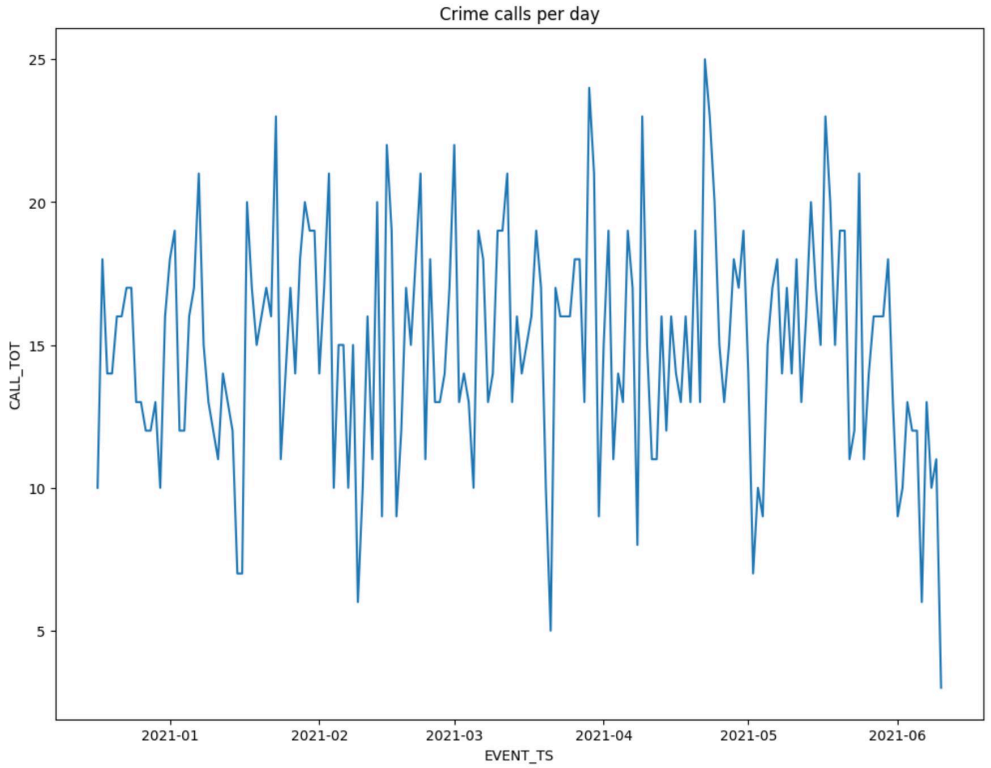
```
Out[103]:             CALL_TOT
          EVENT_TS
          2020-12-17        10
          2020-12-18        18
          2020-12-19        14
          2020-12-20        14
          2020-12-21        16
```

```
In [108]: sns.lineplot(data=count_by_date, x="EVENT_TS", y="CALL_TOT").set_title("Crime calls per day")

          # Code to create a lineplot above this line
```

```
Out[108]: Text(0.5, 1.0, 'Crime calls per day')
```

Question assigned to the following page: 6



In [109]:
```
grader.check("q3e")
```

Out[109]:
```
q3e results: All test cases passed!
```