At the beginning of the course, we saw that there are two types of data visualization:

* Exploratory data visualization: we build graphs for *ourselves* to explore data and find patterns.
* Explanatory data visualization: we build graphs for *others* to communicate and explain the patterns we've found through exploring data.

Throughout this course, we focused only on exploratory data visualization and learned the following:

* How to visualize time series data with line plots.
* How to visualize correlations with scatter plots.
* How to visualize frequency distributions with bar plots and histograms.
* How to speed up our exploratory data visualization workflow with the pandas library.
* How to compare graphs using grid charts.

To make learning smoother and more efficient, we learned about each of these topics in isolation. In this guided project, we'll go one step further and learn to combine these skills.

We're going to analyze a dataset about the westbound traffic on the [I-94 Interstate highway](https://en.wikipedia.org/wiki/Interstate_94).

John Hogue made the dataset available, and you can download it from the [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume).

The goal of our analysis is to determine a few indicators of heavy traffic on I-94. These indicators can be weather type, time of the day, time of the week, etc. For instance, we may find out that the traffic is usually heavier in the summer or when it snows.

You can find the solution notebook [at this link](https://github.com/dataquestio/solutions/blob/master/Mission524Solutions.ipynb). Let's start!

Instructions

1. To help readers gain context into your project, use the first Markdown cell of the notebook to do the following:
   * Add a title.
   * Write a short introduction that explains the following in no more than two paragraphs:
     + What the project is about
     + Your goal with this project
   * Side note: the title and the introduction are tentative at this point — you can come back at the end of your work to change them.
2. Read in the Metro\_Interstate\_Traffic\_Volume.csv file using Pandas.
3. Examine the first and the last five rows.
4. Use DataFrame.info() to find more information about the dataset.

On the previous screen, we read in the dataset and set our analysis goal: determine indicators of heavy traffic on I-94.

The [dataset documentation](https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume) mentions that a station located approximately midway between Minneapolis and Saint Paul recorded the traffic data. Also, the station only records westbound traffic (cars moving from east to west).



This means that the results of our analysis will be about the westbound traffic in the proximity of that station. In other words, we should avoid generalizing our results for the entire I-94 highway.

In the next exercise, we're going to plot a histogram to visualize the distribution of the traffic\_volumecolumn.

When we use Matplotlib inside Jupyter, we also need to add the %matplotlib inline magic — this enables Jupyter to generate the graphs.

import matplotlib.pyplot as plt

%matplotlib inline

​

plt.plot()

plt.show()

We only need to run %matplotlib inline once inside a notebook. If we have a notebook with ten cells, and we plot a graph in each cell, it's enough to add %matplotlib inline in the first cell.

Instructions

1. Add the %matplotlib inline magic.
2. Plot a histogram to examine the distribution of the traffic\_volume column. Use a Pandas method.
3. Use Series.describe() to look up a few statistics about the traffic\_volume column.
4. Note a few observations about the distribution of traffic\_volume based on the histogram and the result of Series.describe().
5. Do you think daytime and nighttime influence the traffic volume?

Previously, we analyzed the distribution of traffic\_volume and found the following:

* About 25% of the time, there were 1,193 cars or fewer passing the station each hour — this probably occurs during the night, or when a road is under construction.
* About 25% of the time, the traffic volume was four times as much (4,933 cars or more).

This possibility that nighttime and daytime might influence traffic volume gives our analysis an interesting direction: comparing daytime with nighttime data.

We'll start by dividing the dataset into two parts:

* Daytime data: hours from 7 a.m. to 7 p.m. (12 hours)
* Nighttime data: hours from 7 p.m. to 7 a.m. (12 hours)

While this is not a perfect criterion for distinguishing between nighttime and daytime, it's a good starting point.

Instructions

1. Transform the date\_time column to datetime by using the function pd.to\_datetime().
2. Use the Series.dt.hour property to get the hour of every instance of the date\_time column and do the following:
   * Isolate the daytime data.
   * Isolate the nighttime data.

Previously, we divided the dataset into two parts:

* Daytime data: hours from 7 AM to 7 PM (12 hours)
* Nighttime data: hours from 7 PM to 7 AM (12 hours)

Now we're going to compare the traffic volume at night and during day.

Instructions

1. Plot the histograms of traffic\_volume for both day and night. Organize the two histograms side-by-side on a grid chart.
2. For both histograms, do the following:
   * Add a title.
   * Add x- and y-labels.
   * Bring the x- and the y-axis to the same ranges (this will help you compare the histograms correctly).
3. Use Series.describe() to look up a few statistics for traffic\_volume for both day and night.
4. Analyze the results:
   * What shape do the histograms have and what does that indicate?
   * If the traffic is light at night, and our goal is to find indicators of heavy traffic, should we still be using the nighttime data?

Previously, we determined that the traffic at night is generally light. Our goal is to find indicators of heavy traffic, so we decided to only focus on the daytime data moving forward.

One of the possible indicators of heavy traffic is time. There might be more people on the road in a certain month, on a certain day, or at a certain time of the day.

We're going to look at a few line plots showing how the traffic volume changed according to the following parameters:

* Month
* Day of the week
* Time of day

The fastest way to get the average traffic volume for each month is by using the [DataFrame.groupby()method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.groupby.html" \t "_blank). If you haven't yet learned this method, don't worry: we'll write out the code for you, and you'll only have to focus on the visualization code.

In the code below, we'll do the following:

* Use day['date\_time'].dt.month to create a new column where each value describes the month when the traffic volume measurement was taken.
* Use day.groupby('month').mean() to group the dataset by the month column with the mean as an aggregate function.

day['month'] = day['date\_time'].dt.month

by\_month = day.groupby('month').mean()

by\_month['traffic\_volume']

month

1     4495.613727

2     4711.198394

3     4889.409560

4     4906.894305

5     4911.121609

6     4898.019566

7     4595.035744

8     4928.302035

9     4870.783145

10    4921.234922

11    4704.094319

12    4374.834566

Name: traffic\_volume, dtype: float64

Let's start with generating the plot for the first time unit mentioned above — the month.

Instructions

1. Copy the code above and run it into your notebook to get the monthly traffic volume averages.
2. Generate a line plot to visualize how the traffic volume changed each month on average.
3. Analyze the line plot. Do you notice any interesting exception?

In the previous screen, we generated a line plot showing how the traffic volume changed each month on average.

We'll now continue with building line plots for another time unit: day of the week.

To get the traffic volume averages for each day of the week, we'll need to use the following code:

day['dayofweek'] = day['date\_time'].dt.dayofweek

by\_dayofweek = day.groupby('dayofweek').mean()

by\_dayofweek['traffic\_volume']  # 0 is Monday, 6 is Sunday

dayofweek

0    4893.551286

1    5189.004782

2    5284.454282

3    5311.303730

4    5291.600829

5    3927.249558

6    3436.541789

Name: traffic\_volume, dtype: float64

Let's now generate the plot.

Instructions

1. Generate a line plot to visualize how the traffic volume changed each day of the week on average.
2. Analyze the line plot. Do you notice any difference between business days and the weekend?

On the previous screen, we found that the traffic volume is significantly heavier on business days compared to the weekends.

We'll now generate a line plot for the time of day. The weekends, however, will drag down the average values, so we're going to look at the averages separately. To do that, we'll start by splitting the data based on the day type: business day or weekend.

Below, we show you how to split the dataset so you can focus on plotting the graphs. While your variable names may vary, the logic of the code should be the same.

day['hour'] = day['date\_time'].dt.hour

bussiness\_days = day.copy()[day['dayofweek'] <= 4] # 4 == Friday

weekend = day.copy()[day['dayofweek'] >= 5] # 5 == Saturday

by\_hour\_business = bussiness\_days.groupby('hour').mean()

by\_hour\_weekend = weekend.groupby('hour').mean()

​

print(by\_hour\_business['traffic\_volume'])

print(by\_hour\_weekend['traffic\_volume'])

hour

7     6030.413559

8     5503.497970

9     4895.269257

10    4378.419118

11    4633.419470

12    4855.382143

13    4859.180473

14    5152.995778

15    5592.897768

16    6189.473647

17    5784.827133

18    4434.209431

Name: traffic\_volume, dtype: float64

​

hour

7     1589.365894

8     2338.578073

9     3111.623917

10    3686.632302

11    4044.154955

12    4372.482883

13    4362.296564

14    4358.543796

15    4342.456881

16    4339.693805

17    4151.919929

18    3811.792279

Name: traffic\_volume, dtype: float64

Instructions

1. Plot two line plots on a grid chart to visualize how the traffic volume changes by time of the day.
   * One plot shows how traffic volume changes during business days and the other shows how it changes during weekends.
   * Add a title to each graph.
   * Bring both graphs to the same x- and y-axis range.
2. Analyze both charts. How do they compare? When are the rush hours on business days?
3. Summarize all your findings regarding time indicators for heavy traffic.

So far, we've focused on finding time indicators for heavy traffic, and we reached the following conclusions:

* The traffic is usually heavier during warm months (March–October) compared to cold months (November–February).
* The traffic is usually heavier on business days compared to weekends.
* On business days, the rush hours are around 7 and 16.

Another possible indicator of heavy traffic is weather. The dataset provides us with a few useful columns about weather: temp, rain\_1h, snow\_1h, clouds\_all, weather\_main, weather\_description.

A few of these columns are numerical so let's start by looking up their correlation values with traffic\_volume.

Instructions

1. Find the correlation values between traffic\_volume and the numerical weather columns.
2. Find the weather column with the strongest correlation with traffic\_volume and plot a scatter plot for this weather column and traffic\_volume.
3. Do any of these weather columns seem like reliable indicators for heavy traffic?

Previously, we examined the correlation between traffic\_volume and the numerical weather columns. However, we didn't find any reliable indicator of heavy traffic.

To see if we can find more useful data, we'll look next at the categorical weather-related columns: weather\_main and weather\_description.

We're going to calculate the average traffic volume associated with each unique value in these two columns. We've already calculated the values for you — we grouped the data by weather\_main and weather\_description while using the mean as an aggregate function.

by\_weather\_main = day.groupby('weather\_main').mean()

by\_weather\_description = day.groupby('weather\_description').mean()

Let's create a bar plot for the traffic\_volume column of by\_weather\_main and by\_weather\_description.

Instructions

1. Plot a horizontal bar plot for the traffic\_volume column of by\_weather\_main.
2. Analyze the bar plot. Is there any traffic volume exceeding 5,000 cars? Can you find any weather type as a heavy traffic indicator?
3. Plot a horizontal bar plot for the traffic\_volume column of by\_weather\_description. This column has many unique values, so you'll need to enlarge the figure size to make it readable. You can use plt.figure(figsize=(width,height)) or Series.plot.barh(figsize=(width,height)).
4. Analyze the bar plot. Is there any traffic volume exceeding 5,000 cars? Can you find any weather type as a heavy traffic indicator?
5. Write up your conclusions for the entire project — this was the last step.

In this project, we tried to find a few indicators of heavy traffic on the I-94 Interstate highway. We managed to find two types of indicators:

* Time indicators
  + The traffic is usually heavier during warm months (March–October) compared to cold months (November–February).
  + The traffic is usually heavier on business days compared to the weekends.
  + On business days, the rush hours are around 7 and 16.
* Weather indicators
  + Shower snow
  + Light rain and snow
  + Proximity thunderstorm with drizzle

Next steps include the following:

* Use the nighttime data to look for heavy traffic indicators.
* Find more time and weather indicators.
* Make your project portfolio-ready by following this [style guide](https://www.dataquest.io/blog/data-science-project-style-guide/).

Curious to see what other students have done on this project? [Head over to our Community to check them out](https://community.dataquest.io/tags/c/social/share/49/524). While you are there, please share your own feedback!

And of course, we welcome you to share your own project and show off your hard work. Head over to our Community to [share your finished Guided Project](https://community.dataquest.io/tags/c/social/share/49/524)!