Data Science and Visualization, S2026

Data visualization

2026-03-03

Exercise 1: Importing and loading

In this exercise set, you will use the pandas and Plotnine libraries to visualize features of a dataset of taxi trips in New York City. The data contains the following columns:

- **pickup**: Date and time of the beginning of the trip
- **dropoff**: Date and time of end of the trip
- **passengers**: The number of passengers in the taxis
- **distance**: The distance of the trip (miles)
- **fare**: The base fare of the trip (USD)
- **tip**: The tip amount paid (USD)
- **tolls**: The fee charged for use of roads (USD)
- **total**: The total amount charged (USD)
- **color**: The color of the taxi (yellow, green)
- **payment**: The payment method (credit card, cash)
- **pickup_zone**: The zone where the passengers were picked up
- **dropoff_zone**: The zone where the passenger were dropped off
- **pickup_borough**: The boroughs where the passengers were picked up
- **dropoff_borough**: The borough where the passengers were dropped off

There two important things to keep in mind throughout this exercise:

1. Your plots should abide to the principles of data visualization, i.e, *show the data*, *reduce the clutter*, and *integrate the text and the graph*. One approach to controlling the layout of your plots is to specify a global theme with the `theme()` function. You can then modify the theme for each plot you create as needed
2. Remember to think about what each plot tells you about the data once you have made it

1.1 Use the empty code cell below to import the pandas and Plotnine libraries. Download the file *taxis.csv* from Moodle and save it in the same folder as this notebook. Then read it as DataFrame object and assign it to the variable `df`. Finally, use a DataFrame method to find out how many rows and columns the data has and print the results

```
# Import libraries

# Load the taxis data

# How many rows and columns does df have?
```

1.2 Use the empty code cell below to specify a global theme to use in all plots. You can return to this code cell, edit it, and rerun it, if want to change the global theme

```
# Specify a global theme to use in all plots
```

Exercise 2: Visualizing amounts

2.1 Use the empty code cells below to create a bar plot of the frequencies of taxi trips starting in each of the boroughs of New York City. Do this by following these 2 steps:

1. Create a new pandas DataFrame containing the frequencies of taxi trips starting in each of the 5 boroughs. You will need to use the pandas method `value_counts()` to do this
2. Create the bar plot using Plotnine. Use the function `geom_col()` to do this

```
# Step 1: Create a DataFrame containing the data to be plotted
```

```
# Step 2: Create the plot
```

```
# Output the plot
```

2.2 Use the empty code cells below to create a bar plot of the frequencies of taxi trips in yellow and green taxis across each of the boroughs of New York City. Use the column pickup_borough to assign trips to boroughs. Do this by following these 2 steps:

1. Create a pandas DataFrame containing the frequencies of taxi trips by taxi color and the borough in which the trip started. You will need to use the pandas methods `.value_counts()` and reset_index() to do this
2. Create the bar plot using Plotnine. Use the function `geom_col()`, map the color column to the `fill` aesthetic, use the `position` argument to make the bars appear side by side, and select colors to use with the function `scale_fill_manual()`

```
# Step 1: Create a DataFrame containing the data to be plotted
```

```
# Step 2: Create the plot
```

```
# Output the plot
```

Exercise 3: Visualizing distributions

3.1 Use the empty code cells below to create a boxplot of taxi trip travel time. Do this by following these 2 steps:

1. Use the columns pickup and dropoff in `df` to construct a new column containing the travel time of each taxi trip **in minutes**. You will need to use the pandas function pd.to_datemite() and the methods .dt and .total_seconds() to do this. Name the column travel_time
2. Create the boxplot. Use the function `geom_boxplot()` and the column you just created to do this

```
# Step 1: Create a new column containing travel time in minutes
```

```
# Step 2: Create the plot
```

```
# Output the plot
```

3.2 Use the empty code cells below to create a histogram of the distribution of taxi trip travel time. Create the histogram by using the function `geom_histogram()` and the column travel_time you created above to do this. Remember to select an appropriate binwidth

```
# Create the plot
```

```
# Output the plot
```

3.3 Use the empty code cells below to create a density plot that shows the distribution of taxi trip travel time across by payment across each of the boroughs of New York City. Use the column pickup_borough to assign trips to boroughs. Do this by following these 3 steps:

1. Construct a new DataFrame with the columns travel_time, payment, and pickup_borough. Make sure there are no missing values in the new DataFrame
2. Create the density plot. Use the function `geom_density()`, map the column payment to the `fill` aesthetic, and map the column pickup_borough to small multiples using the `facet_wrap()` function. Remember to select an appropriate bandwidth
3. Output the plot

```
# Step 1: Create a DataFrame containing the data to be plotted
```

```
# Step 2: Create the plot
```

```
# Output the plot
```

Exercise 4: Visualizing proportions

Use the code cell below to create a stacked bar plot that compares the proportions of taxi trips with a total fare below 10 USD, between 10 and 20 USD, and greater than 20 USD across the boroughs of New York City. Use the column pickup_borough to assign trips to boroughs. You can do this by following these 2 steps:

1. Start by constructing a new categorical column that groups the column total into the 3 categories
   (1) less than or equal to 10 USD,
   (2) greater than 10 USD and less than or equal to 20 USD, or (3) greater than 20 USD. Use the pandas function `pd.cut()` for this. Then compute the relative frequencies of trips by total fare category and borough. Use the pandas methods `.groupby()` and `.value_counts()`. Assign the results to a new DataFrame. Finally, transform the proportions to percentages and round the values to one decimal place

2. Create the plot. Use the function `geom_col()`, map percentages and the column pickup_borough to the position scales of the coordinate system, map the column total_categorical to the `fill` aesthetic, and select colors to use with the function `scale_fill_manual()`. Label each slice of the bars with the percentage it represents

```
# Step 1: Create a DataFrame containing the data to be plotted
```

```
# Step 2: Create the plot
```

```
# Output the plot
```

Exercise 5: Visualizing associations

5.1: Use the empty code cells below to create a bubble chart of taxi trip distance, total fare, and travel time. You can create the plot by using the function `geom_point()`. Map the columns distance and total to the position scales of the coordinate system and map the column travel_time to the `size` aesthetic

```
# Create the plot
```

```
# Output the plot
```

5.2: Create a scatterplot of travel time and total fare that groups data points by whether the taxi trip began and ended in the same borough. You can do this by following these 2 steps:

1. Start by creating a categorical column that takes on the value 'Inter-borough trip' for trips starting and ending in different boroughs and 'Intraborough trip' for trips starting and ending in the same borough. Use the pandas method .mask() to do this. Name the column inter_intra. Then create a new DataFrame with the columns travel_time, total, and inter_intra. Make sure there are no missing values in the new DataFrame
2. Create the plot. Use the function `geom_point()`, map the columns travel_time and total to the position scales of the coordinate system and map the column inter_intra to the `fill` aesthetic

```
# Step 1: Create a DataFrame containing the data to be plotted
```

```
# Step 2: Create the plot
```

```
# Output the plot
```

Exercise 6: Visualizing time series

Create a line graph that shows the number of taxi trip on all dates present in the data. Use the variable pickup to assign trips to dates. You can do this using the following two steps:

1. Start by creating a new column that contains the date of each trip. As all trips in the data took place in 2019, we can do this by excluding the year and time from the column pickup and formatting the result as a string. One possible approach is to use the pandas function `pd.to_datetime()` and the methods `dt` and strftime(). Then use the `value_counts()` method to compute trip frequencies by dates
2. Create the plot. Use the functions `geom_point()` and `geom_line()` to plot the data. Consider how to make the horizontal axis representing the dates readable. This could be by formatting the axis or wrangling the underlying Series of dates

```
# Step 1: Create a DataFrame containing the data to be plotted
```

```
# Step 2: Create the plot
```

```
# Output the plot
```