

V2_exploring_visualizations

August 23, 2025

1 Activity: Exploring with Visualizations

1.1 Introduction

In this activity you will practice using Pandas functionality to create and explore visualizations.

This activity will cover the following topics: - Compare single values against one another. - Compare multiple values against one another. - Use different methods to change how you see the data.

```
[3]: import pandas as pd
import matplotlib.pyplot as plt

# Data from https://github.com/mwaskom/seaborn-data/blob/
↳ 2b29313169bf8dfa77d8dc930f7bd3eba559a906/mpg.csv
df = pd.read_csv('mpg.csv')
```

Question 1 Create a new column in the DataFrame `df` called `horsepower_per_cylinder` that gives the value of horsepower per cylinder.

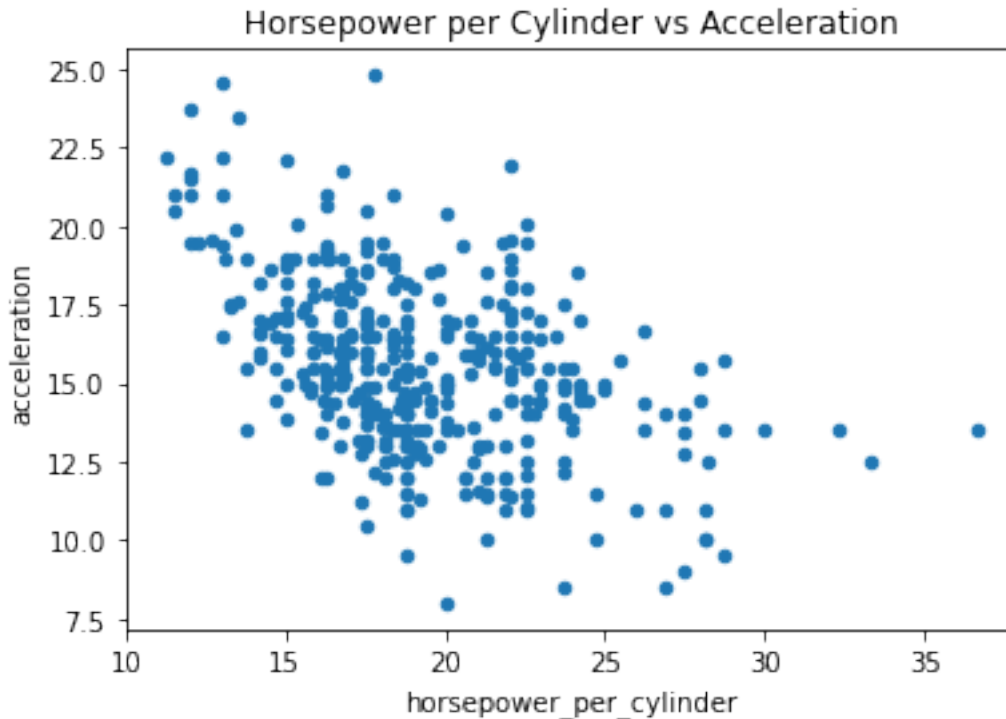
Then, create a scatter plot of `horsepower_per_cylinder` vs `acceleration` (`horsepower_per_cylinder` on the x-axis and `acceleration` on the y-axis). Does `acceleration` tend to *increase* or *decrease* as `horsepower_per_cylinder` *increases*?

Assign the boolean value `True` to the variable `acc_decreases` if `acceleration` decreases as `horsepower_per_cylinder` increases. Otherwise, assign the boolean value `False` to the variable `acc_decreases`.

```
[5]: df["horsepower_per_cylinder"] = df["horsepower"] / df["cylinders"]

ax = df.plot.scatter(x="horsepower_per_cylinder", y="acceleration",
↳ title="Horsepower per Cylinder vs Acceleration")

plt.show()
acc_decreases = True
```



```
[ ]: # Question 1 Grading Checks

assert 'horsepower_per_cylinder' in df.columns, 'Did create a column called_
↳`horsepower_per_cylinder` in the DataFrame?'
assert isinstance(acc_decreases, bool), 'Did you assign the either True or_
↳False to acc_decreases?'
```

Question 2 Create a single visualization where `horsepower_per_cylinder` and `horsepower` are on the y-axis vs `mpg` on the x-axis in a scatter plot. Make each set of points a different color.

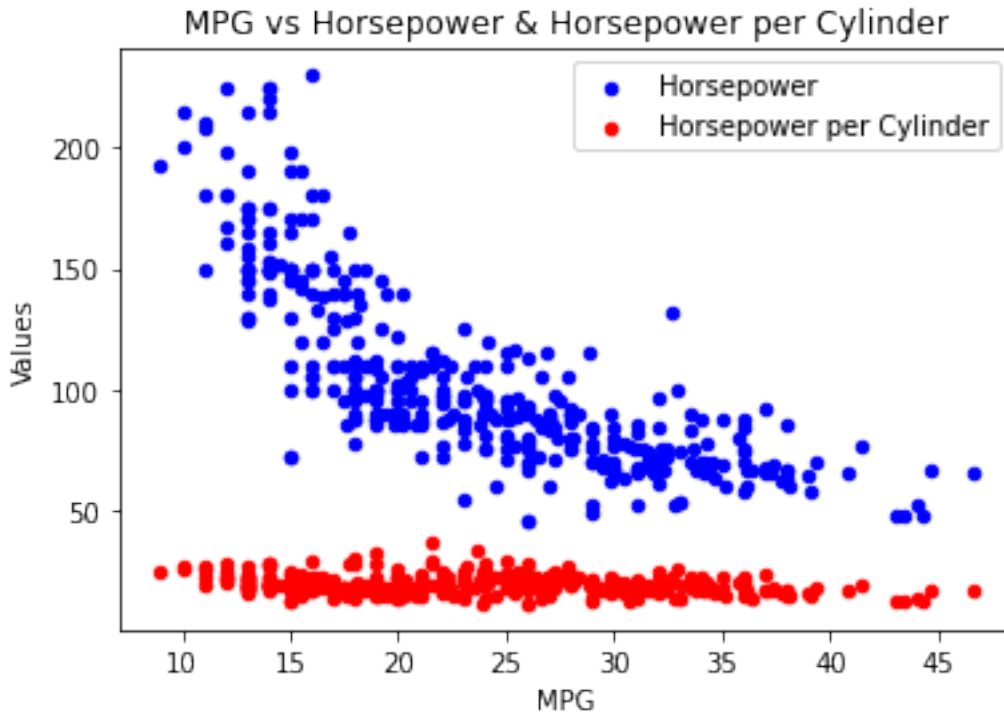
Set the result of the plot to the variable `ax`. Your code will look something like:

```
ax = # code to create a scatter plot
# ... other code

[7]: ax = df.plot.scatter(x="mpg", y="horsepower", color="blue", label="Horsepower")

df.plot.scatter(x="mpg", y="horsepower_per_cylinder", color="red",
↳label="Horsepower per Cylinder", ax=ax)

plt.title("MPG vs Horsepower & Horsepower per Cylinder")
plt.xlabel("MPG")
plt.ylabel("Values")
plt.show()
```



```
[ ]: # Question 2 Grading Checks

assert isinstance(ax, plt.Axes), 'Did you assign the plot result to the_
→variable ax?'
```

Question 3 Create a histogram of the `mpg` column with the number **50** bins. Change the x-axis of the visualization to *'Miles Per Gallon'*

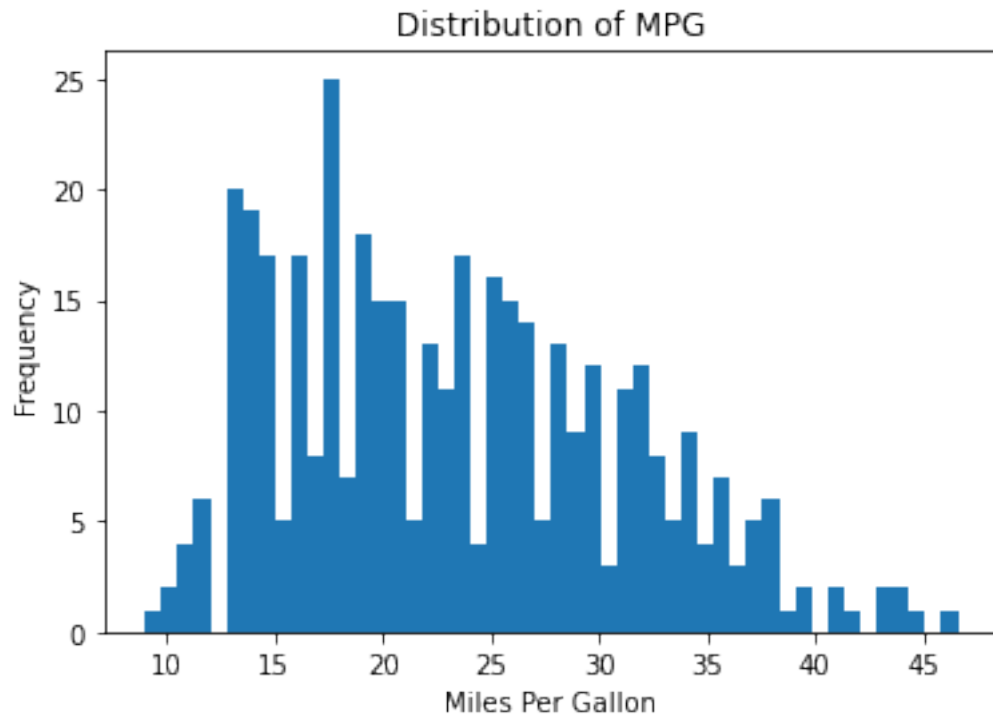
Set the result of the plot to the variable `ax`. Your code will look something like:

```
ax = # code to create a scatter plot
# ... other code
```

```
[8]: ax = df["mpg"].plot.hist(bins=50)

ax.set_xlabel("Miles Per Gallon")

plt.title("Distribution of MPG")
plt.show()
```



```
[ ]: # Question 3 Grading Checks
```

```
assert isinstance(ax, plt.Axes), 'Did you assign the plot result to the_  
→variable ax?'
```