Case one

```
import pandas as pd

df = pd.read_csv('winequality-red.csv', sep=';')
```

find the mean of density:

```
mean=df['density'].mean()

print(mean)
```

```
import numpy as np

import pandas as pd

load red and white wine datasets #

red_df = pd.read_csv('winequality-red.csv')

white_df = pd.read_csv('winequality-white.csv')
```

rename column:

```
red_df.rename(columns={'total_sulfur-dioxide':'total_sulfur_dioxide'}, inplace=True)
```

add to array color:

```
red_df['color'] = color_red

red_df.head()
```

append data frames:

```
wine_df = red_df.append(white_df)
```

save dataset or save change:

```
wine_df.to_csv('winequality_edited.csv', index=False)
```

```
import pandas as pd
```

import matplotlib:

```
import matplotlib.pyplot as plt

df = pd.read_csv('winequality_edited.csv', sep=';')

df.head()
```

histogram for pH:

```
df.pH.hist();
```

Scatterplots for pH:

```
df.plot(x='pH', y='quality', kind='scatter');
```

find the mean quality of groupby color:

```
df.groupby('color').mean().quality
```

view:

```
df.describe()
```

Labels for the four acidity level groups:

```
bin_names = ['high', 'mod_high', 'medium', 'low']
```

Creates acidity_levels column:

```
df['acidity_levels'] = pd.cut(df['pH'], bin_edges, labels=bin_names)
```

Checks for successful creation of this column:

```
df.head()
```

get the median amount of alcohol content:

```
df.alcohol.median()
```

select samples with alcohol content greater than or equal to the median:

```
high_alcohol = df.query('alcohol >= 10.3')
```

ensure these queries included each sample exactly once:

```
num_samples = df.shape[0]

num_samples == low_alcohol['quality'].count() + high_alcohol['quality'].count()
```

get mean quality rating for the low sugar and high sugar groups:

```
low_sugar.quality.mean(), high_sugar.quality.mean()
```

Creating a Bar Chart Using Matplotlib:

```
import matplotlib.pyplot as plt
% matplotlib inline
```

Bar function:

```
plt.bar([1, 2, 3], [224, 620, 425]);
```

naming the bar chart:

```
plt.bar([1, 2, 3], [224, 620, 425], tick_label=['a', 'b', 'c'])
plt.title('Some Title')
plt.xlabel('Some X Label')
plt.ylabel('Some Y Label');
```

Use query to select each group and get its mean quality:

```
median = df['residual_sugar'].median()
low = df.query('residual_sugar < {}'.format(median))
high = df.query('residual_sugar >= {}'.format(median))
mean_quality_low = low['quality'].mean()
mean_quality_high = high['quality'].mean()
```

***

Create a bar chart with proper labels:

```
locations = [1, 2]
heights = [mean_quality_low, mean_quality_high]
labels = ['Low', 'High']
plt.bar(locations, heights, tick_label=labels)
plt.title('Average Quality Ratings by Residual Sugar')
plt.xlabel('Residual Sugar')
plt.ylabel('Average Quality Rating')
```

Use groupby to get the mean quality for each acidity level:

```
acidity_level_quality_means = df.groupby('acidity_levels').quality.mean()
```

***

Create a bar chart with proper labels:

```
locations = [4, 1, 2, 3]  # reorder values above to go from low to high\n",
heights = acidity_level_quality_means
labels = ['Low', 'Medium', 'Moderately High', 'High']
labels = acidity_level_quality_means.index.str.replace('_', ' ').str.title()
plt.bar(locations, heights, tick_label=labels)
plt.title('Average Quality Ratings by Acidity Level')
plt.xlabel('Acidity Level')
plt.ylabel('Average Quality Rating')
```

get total counts for each color:

```
color_totals = wine_df.groupby('color').count()['pH']
```

get proportions by dividing white rating counts by total # of white samples:

```
white_proportions = color_counts['white'] / color_totals['white']
```

legend:

```
plt.legend()
```