import numpy as np

import pandas

import matplotlib.pyplot as plt

def entries\_histogram(turnstile\_weather):

'''

Before we perform any analysis, it might be useful to take a

look at the data we're hoping to analyze. More specifically, let's

examine the hourly entries in our NYC subway data and determine what

distribution the data follows. This data is stored in a dataframe

called turnstile\_weather under the ['ENTRIESn\_hourly'] column.

Let's plot two histograms on the same axes to show hourly

entries when raining vs. when not raining. Here's an example on how

to plot histograms with pandas and matplotlib:

turnstile\_weather['column\_to\_graph'].hist()

'''

plt.figure()

rainy = turnstile\_weather[turnstile\_weather['rain']==1]['ENTRIESn\_hourly']

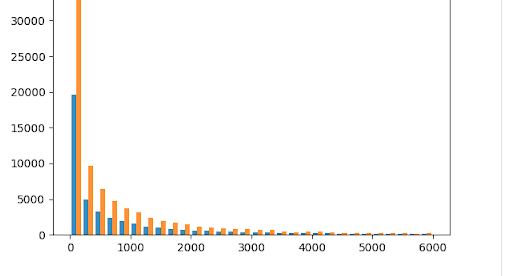
clear = turnstile\_weather[turnstile\_weather['rain']==0]['ENTRIESn\_hourly']

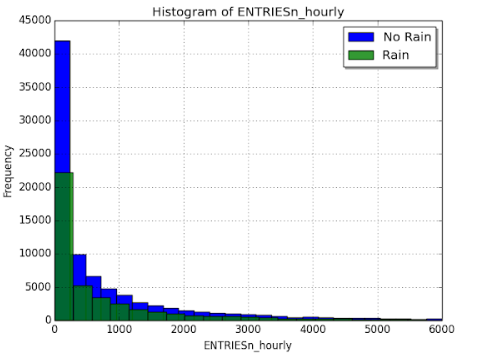
#range=(x.min(), x.max()), Set the alpha value used for blending - not supported on all backends.

plt.hist([rainy.values,clear.values], bins=30, range = (0, 6000), alpha=0.85, ec='None',)

plt.show()

return plt





Does the data seem normally distributed?

Do you think we would be able to use Welch's t-test on this data?

No because it is not normally distributed, so the Welch t-test should not be used

import numpy as np

import scipy

import scipy.stats

import pandas

def mann\_whitney\_plus\_means(turnstile\_weather):

'''

This function will consume the turnstile\_weather dataframe containing

our final turnstile weather data.

You will want to take the means and run the Mann Whitney U-test on the

ENTRIESn\_hourly column in the turnstile\_weather dataframe.

This function should return:

1) the mean of entries with rain

2) the mean of entries without rain

3) the Mann-Whitney U-statistic and p-value comparing the number of entries

with rain and the number of entries without rain

'''

### YOUR CODE HERE ###

rain = (turnstile\_weather['rain']==1).values

with\_rain\_mean = turnstile\_weather['ENTRIESn\_hourly'].loc[rain].mean()

without\_rain\_mean = turnstile\_weather['ENTRIESn\_hourly'].loc[~rain].mean()

U,p = scipy.stats.mannwhitneyu(turnstile\_weather['ENTRIESn\_hourly'].loc[rain],turnstile\_weather['ENTRIESn\_hourly'].loc[~rain])

return with\_rain\_mean, without\_rain\_mean, U, p

Here's your output: (1105.4463767458733, 1090.278780151855, 1924409167.0, 0.024940392294493356)

The distribution of the number statistically different between rainy and non-rainy days.