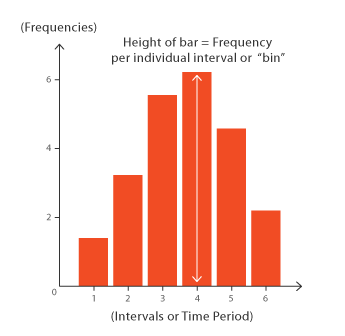
Histogram in Python

Introduction

When analyzing data, you often need to study the characteristics of a single group of numbers, observations, or measurements. You might want to know the center and the spread about this central value. You might want to investigate extreme values (referred to as outliers) or study the distribution or pattern of the data values. Several plots are available to allow you to study the distribution. One such plot is the histogram.

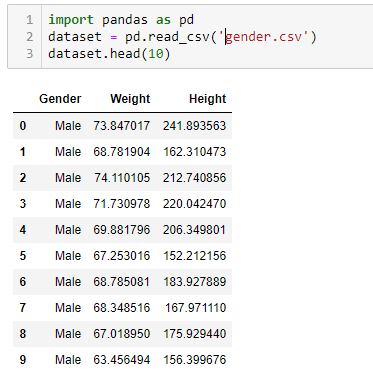


Description

A Histogram visualizes the distribution of data over a continuous interval or certain time period. Each bar in a histogram represents the tabulated frequency at each interval/bin.

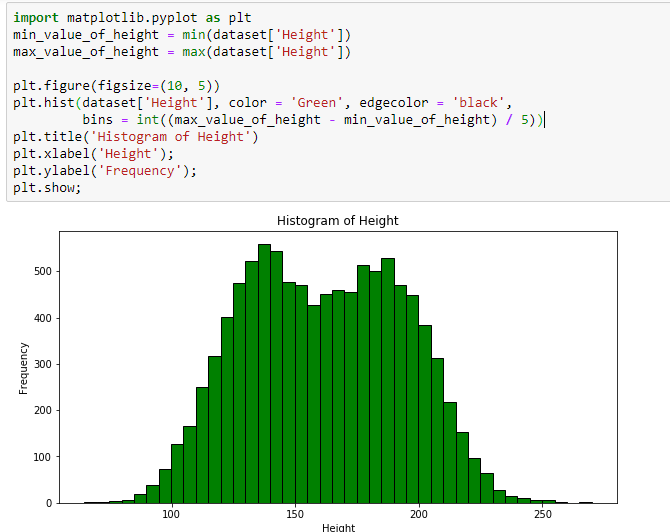
This article will take a comprehensive look at using histograms in Python using the [matplotlib](https://matplotlib.org/" \t "_blank) library. Throughout, we will explore a test dataset “gender”. We will visualize the [gender data](https://cran.r-project.org/web/packages/nycflights13/nycflights13.pdf), which contains over 10000 observations. We will focus on displaying a single variables, the height of population.

It’s always a good idea to examine our data before we get started plotting. We can read the data into a pandas dataframe and display the first 10 rows.

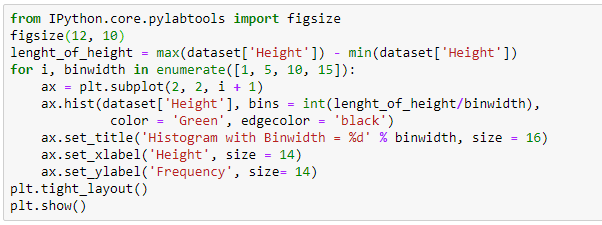


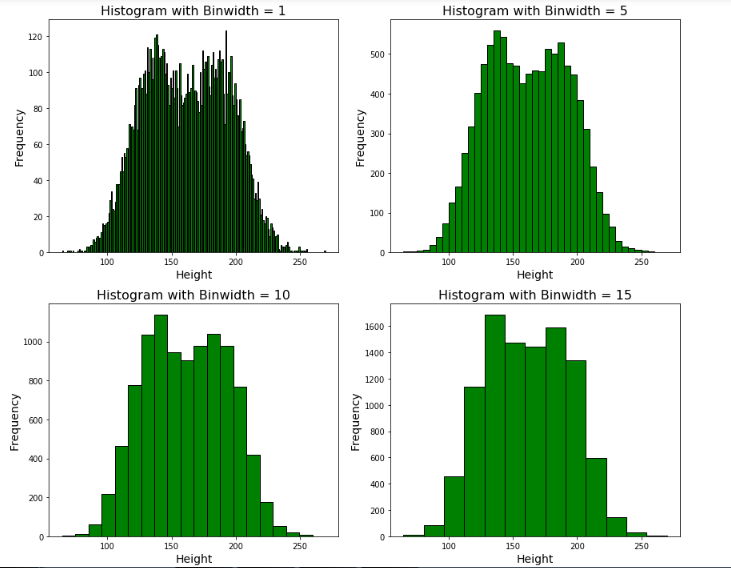
A histogram divides the variable into bins, counts the data points in each bin, and shows the bins on the x-axis and the counts on the y-axis. In our case, the bins will be an interval of heights and the count will be the number of people Height falling into that interval. The binwidth is the most important parameter for a histogram and we should always try out a few different values of binwidth to select the best one for our data.

The code below shows function calls in matplotlib library that create a simple histogram. For the plot calls, we specify the binwidth by the number of bins. For this plot, I will use bins that are 5, which means that the number of bins will be the range of the data (from min(Height) to max(Height)) divided by the binwidth.



How did I come up with 5 santimetre for the binwidth? The only way to figure out an optimal binwidth is to try out multiple values! Below is code to make the same figure in matplotlib with a range of binwidths. Ultimately, there is no right or wrong answer to the binwidth, but I choose 5 santimetre because I think it best represents the distribution.

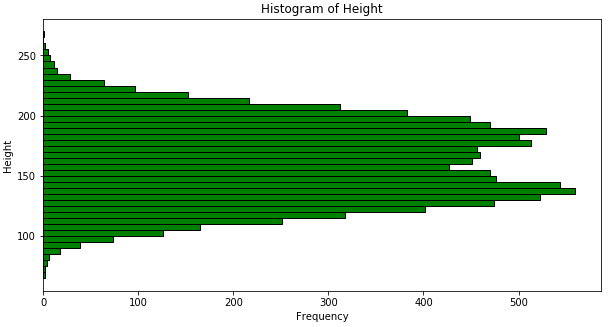




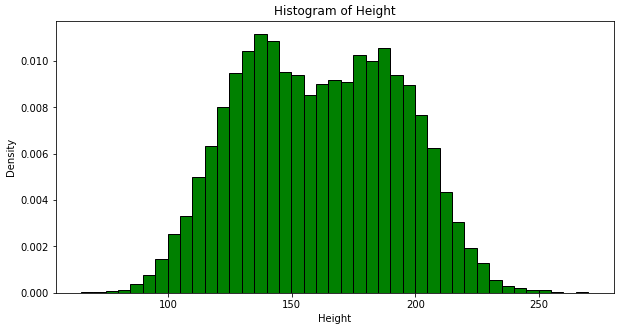
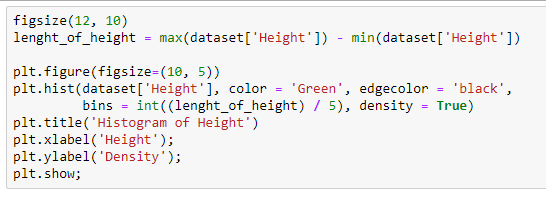
The choice of binwidth significantly affects the resulting plot. Smaller binwidths can make the plot cluttered, but larger binwidths may obscure nuances in the data. Matplotlib will automatically choose a reasonable binwidth for you, but I like to specify the binwidth myself after trying out several values. There is no true right or wrong answer, so try a few options and see which works best for your particular data.

Another important parameter for a histogram is “orientation”, which is by default vertical but we can change it with horizontal and we will obtain another histogram.



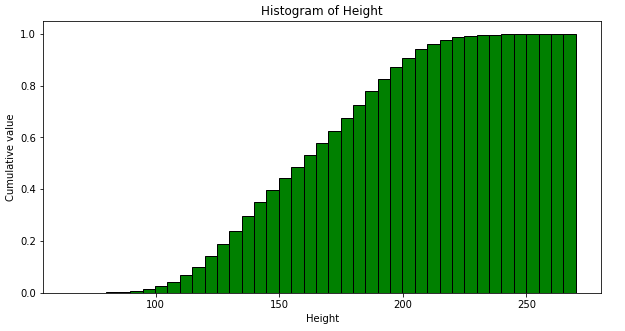


The optional parameter “density”. If True, the first element of the return tuple will be the counts normalized to form a probability density, i.e., the area (or integral) under the histogram will sum to 1. This is achieved by dividing the count by the number of observations times the bin width and not dividing by the total number of observations. Default is False.



The optional parameter cumulative helps us to attain cumulative histogram. This is a Boolean parameter. If True, then a histogram is computed where each bin gives the counts in that bin plus all bins for smaller values. The last bin gives the total number of data points. If normed or density is also True then the histogram is normalized such that the last bin equals 1. If cumulative evaluates to less than 0 (e.g., -1), the direction of accumulation is reversed. In this case, if normed and/or density is also True, then the histogram is normalized such that the first bin equals 1. Default is False.





There are other parameters in the matplotlib histogram, but the above parameters are the most useful and important. If you want to know more about histograms, you can read the matplotlib histogram documentation at the following link:

<https://matplotlib.org/api/_as_gen/matplotlib.pyplot.hist.html>