Louise Bassog

Dr. Bemley

CTEC294

11/10/2022

**Chapter 4**

**Exploratory Data Analysis**

EDA is used for a variety of purposes, the most important of which are the following: the identification of errors; the validation of hypotheses; the preliminary selection of appropriate models; the determination of relationships among the explanatory variables; and the evaluation of the direction and approximate magnitude of relationships between explanatory variables and outcome variables. People aren't great at analyzing a column of numbers or a spreadsheet to find crucial info. Numbers are tiresome, uninteresting, or overpowering. Exploratory data analysis approaches can help. Most of these strategies hide some data while revealing others. Exploratory data analysis is usually cross classified. Each approach is non- or pictorial. Each approach is univariate or multivariate (usually just bivariate). Non-graphical approaches entail calculating summary statistics, whereas graphical methods summarize data visually. Univariate approaches examine one variable (data column) at a time, whereas multivariate methods examine two or more. Sometimes our multivariate EDA will involve three or more variables. Before doing multivariate EDA, perform univariate EDA on each component. Non-graphical and graphical approaches work together.

Non-graphical approaches are quantitative and objective, but they don't convey a whole picture of the facts; thus, qualitative, and subjective graphical methods are also needed. Histograms are bar plots that show the frequency (count) or proportion (count/total count) of instances given a range of values. Count (or percentage) axis runs vertically with bars. To manually generate a histogram, define the data range for each bar (a bin), and draw the bars high enough to show the count. You may gain a broad idea of the distribution's form by reading facts like "two instances have data values between 1 and 2" and "nine cases have data values between 2 and 3." Generally, values that fall directly on the boundary between two bins are put in the lower bin. Depending on the quantity of data and distribution shape, use 5 to 30 bins. Seeing the histogram to determine distribution shape may be an iterative procedure. Changing bin sizes/numbers can modify the contour of the histogram, especially with tiny data. Differences are typically modest. A stem and leaf plot replaces a histogram. Stem-and-leaf plots are easier to draw by hand than histograms and don't hide information. A histogram is better for comprehending a sample distribution's form than a stem and leaf plot. The boxplot is another univariate graphical approach. We'll discuss the boxplot in its most frequent vertical arrangement. Formatting might be horizontal.

**Matplotlib Tutorial: Python Plotting**

Text

Description automatically generated Text

Description automatically generated

Matplotlib is Python's data visualization package. Secondly, matplotlib's pyplot module. Matplotlib.pyplot is widely used in programming. The module's interface creates figures and axes implicitly and automatically for plotting. This is useful when you wish to plot rapidly without instantiating Figures or Axes, like in the first portion of this lesson. You haven't specifically stated these components, yet you've designed a plot! Any adjustments you make use the current Figure and Axes. pylab is another module included with matplotlib. It bulk imports pyplot and numpy and is ideal for array operations, interactive math, and graphing. Older matplotlib tutorials and examples may still utilize this, but it's no longer encouraged, especially when using the IPython kernel in Jupyter. Here's more: Use %matplotlib magic with the correct backend, such inline, qt, etc. Inline ensures that graphs are incorporated in the notebook. Data Camp’s Guide to Jupiter Notebook explains more. When you are not working in a Jupiter notebook, you will still be required to select a different side, dependent on the circumstances of your project. To put it another way, if you do not want to embed plots within a notebook but rather want to embed them into graphical user interfaces, batch scripts, or web application servers, etc., you will need to specify the backend that you want to use. This is the case even if you do not want to embed plots inside of notebooks. This subject, however, is not going to be covered in this article. Instead, the course is going to operate under the assumption that you will use Matplotlib to save your pictures to your local file system.