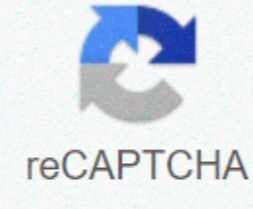




I'm not robot



Continue

Data visualization in r cheat sheet

The book is fanatical, who likes to encode pythonData visualization as a form of visual communication. This includes creating and studying the visual representation of the data. We will perform different data visualization techniques on the iris dataset. Different types of analysis:Univariate (U) : In single-variable analysis, we use a single function to analyze its properties. Bivariate (B): When comparing data between exactly 2 characteristics, it is called bivariate analysis. Multi-variable (M): A comparison of more than 2 variables is called multi-variable analysis. The most common sample types used in the data visualization are:Scatter pattern (B)Pair plot (M)Box plot (U)Fiddle Pattern(U)Distribution Pattern (U)Common Plot (U) & amp; (B)Bar Chart (B)Line drawing (B)Let's look at some of these plots used for data visualization one by one: Import libraries data visualizationReleare two important directories for data visualization -Matplotlib is a python library widely used for visualization data. While seaborn is a python library based on matplotlib. Seaborn provides a high level interface drawing attractive and informative statistical graphics.import matplotlib.pyplot as plt import seaborn as snsLoad file with a dataframeiris = pd.read_csv(iris.csv)1. Scatter chart:This is one of the most commonly used plots for easy data visualization. This gives us a representation of where each point of the entire dataset is present, given any 2 or 3 functions (or columns). These are available in both 2D and 3D.# Here we plan sepal_length vs sepal_width # setosa - red; versicolor - blue; virginica - green n(0,150): if iris[species][n] == setosa: plt.scatter(iris[sepal_length][n], iris[sepal_width][n], color = 'red') plt.xlabel('sepal_length') plt.ylabel('sepal_width') elif iris[species][n] == versicolor: plt.scatter(iris[sepal_length][n], iris[sepal_width][n], color = 'blue') plt.xlabel('sepal_length') plt.ylabel('sepal_width') elif iris[species][n] == virginica: plt.scatter(iris[sepal_length][n], iris[sepal_width][n], color = 'green') plt.xlabel('sepal_length') plt.ylabel('sepal_width') 2. Couple PlotLets say that n number of functions per data. Couple plots will help us create a (n x n) number where the diagonal plots will have histogram plots with the function corresponding to the row and the other plots a combination of function in each row y axis and function of each column x axis. Code snippet of the even plot implemented on the Iris dataset:3. PlotA field drawing (or boxed and mustache sample) shows the distribution of quantitative data in such a way as to facilitate comparison between levels of variables or categorical variables. The field shows the quartiles in the dataset, while the mustache extends to show the rest of the distribution. Code to represent features using Box plots :# A plt.style.use('ggplot') plt.subplot(2,2,1) sns.boxplot(x = 'species', y = 'sepal_length', data = iris) plt.subplot(2,2,2) sns.boxplot(x = 'species', y = 'sepal_width', data = iris) plt.subplot(2,2,3) sns.boxplot(x = 'species', y = 'petal_length', data = iris)

