ADS ASSIGNMENT 1

Univariate analysis

It is the simplest form of data analysis where the data being contains only one variable. Since it's a single variable it doesn't deal with causes or relationships. The main purpose of analysis is to describe the data and find patterns that exist within it

You can think of the variable as a category that your data falls into. One example of a variable in analysis might be "age". Another might be "height". not look at these two variables at the same time, nor would it look at the relationship between them.

Bivariate analysis

It is used to find out if there is a relationship between two different variables. Something as simple as creating a scatterplot by plotting one variable against another on a Cartesian plane (think X and Y axis) can sometimes give you a picture of what the data is trying to tell you. If the data seems to fit a line or curve then there is a relationship or correlation between the two variables. For example, one might choose to plot caloric intake versus weight.

Multivariate analysis

It is the analysis of three or more variables. There are many ways to perform multivariate analysis depending on your goals. Some of these methods include Additive Tree, Canonical Correlation Analysis, Cluster Analysis, Correspondence Analysis / Multiple Correspondence Analysis, Factor Analysis, Generalized Procrustean Analysis, MANOVA, Multidimensional Scaling, Multiple Regression Analysis, Partial Least Square Regression, Principal Component Analysis / Regression / PARAFAC, and Redundancy Analysis.

df=pd.read_csv(r'https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv')

df.head()

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

df.shape

(150, 5)

Univariate Analysis

Classifying based on the different species

df_setosa=df.loc[df['species']=='setosa']

df_virginica=df.loc[df['species']=='virginica']

df_versicolor=df.loc[df['species']=='versicolor']

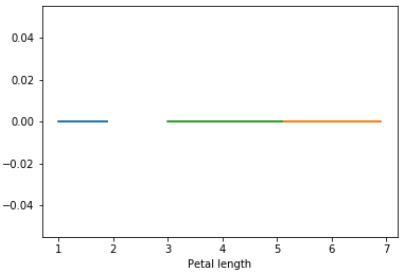
Create a univariate diagram with 'petal_length' and keep yaxis= 0 with same length as species petal length

plt.plot (df_setosa['petal_length'],np.zeros_like(df_setosa['petal_length']))

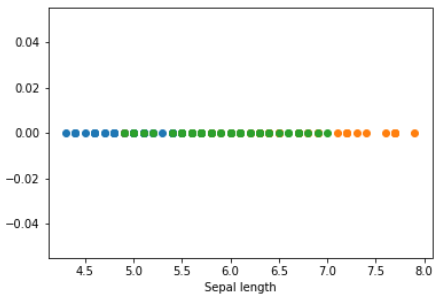
plt.plot (df_virginica['petal_length'],np.zeros_like(df_virginica['petal_length']))

plt.plot (df_versicolor['petal_length'],np.zeros_like(df_versicolor['petal_length']))

plt.xlabel('Petal length')



plt.show ()



Create a univariate diagram with 'sepal_length' and keep yaxis= 0 with same length as species sepal_length

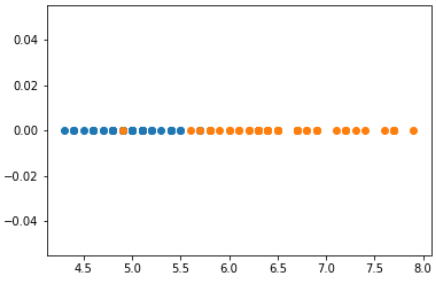
'o' makes the points little bit bigger

plt.plot (df_setosa['sepal_length'],np.zeros_like(df_setosa['sepal_length']),'o')

plt.plot (df_virginica['sepal_length'],np.zeros_like(df_virginica['sepal_length']),'o')

plt.plot (df_versicolor['sepal_length'],np.zeros_like(df_versicolor['sepal_length']),'o')

plt.xlabel('Sepal length')



plt.show()

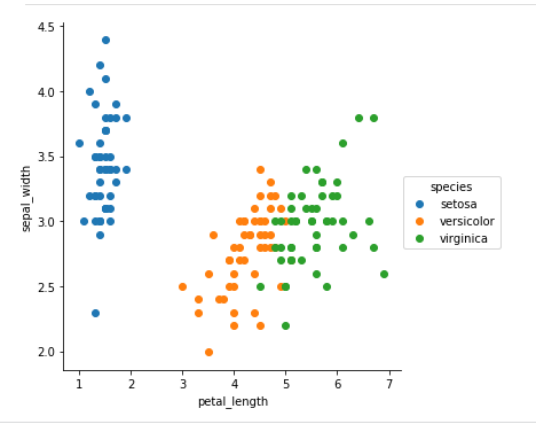
#Analysis - All the virginica sepal length are more than 5.5 but only one has less than 5 value

plt.plot (df_setosa['sepal_length'],np.zeros_like(df_setosa['sepal_length']),'o')

plt.plot (df_virginica['sepal_length'],np.zeros_like(df_virginica['sepal_length']),'o')

plt.xlabel('Sepal length')

plt.show()



BIVARIATE ANALYSIS

Selecting

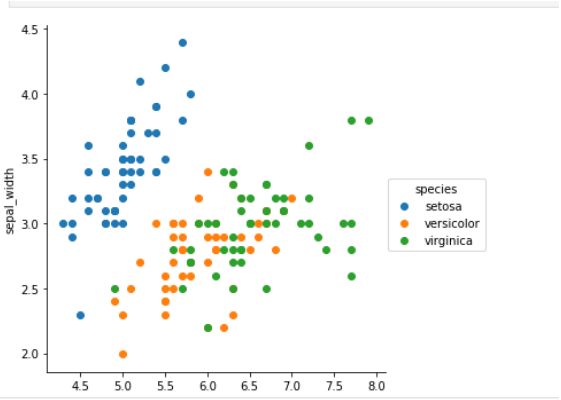
Using FacetGrid from seaborn library to take two features 'sepal_length' and 'sepal_width'

Parameter we use 'hue'='species' as it is the output feature and size ='5' to see the output in a 5*5 figure

We are using .map to map it to a plot (Scatter plot here) and mapping just 2 features

Adding legend for clear distinction of the species

sns.FacetGrid(df,hue='species',size=5).map(plt.scatter,"sepal_length","sepal_width").add_legend()



plt.()

#Analysis - for setosa , we can easily distinguish . It can be distinguished by drawing a straight line but other 2 species

as it is overlapped , it can't be distinguised by a straight line

sns.FacetGrid(df,hue='species',size=5).map(plt.scatter,"petal_length","sepal_width").add_legend()

plt.show()

Multivariate analysis

Using Pairplot from sea born for multiple features. This will also help us to identify the most prominent feature and the correlations.

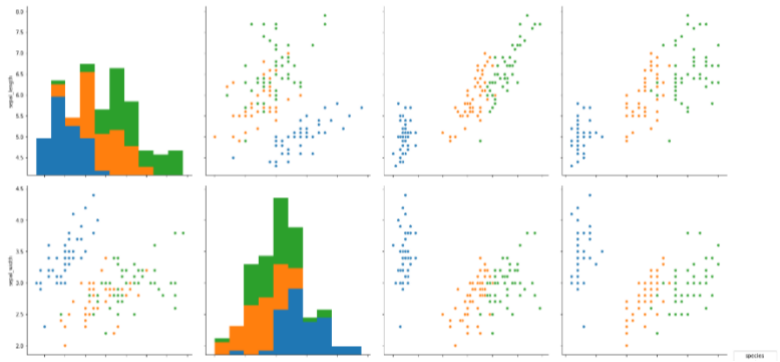
Using Pairplot from seaborn library to take all features from our dataframe

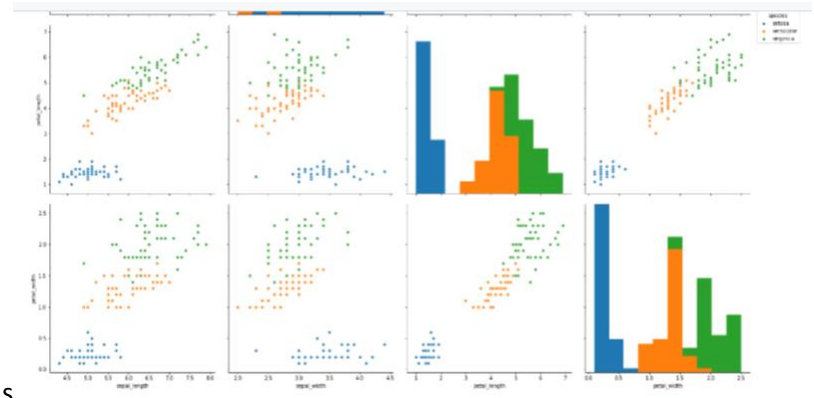
Parameter we use 'hue'='species' as it is the output feature

Adding legend

sns.pairplot(df,hue='species',size=5)

<seaborn.axisgrid.PairGrid at 0xb9c8978>





s

df.corr()

sepal_length	sepal_width	petal_length	petal_width	
sepal_length	1.000000	-0.109369	0.871754	0.817954
sepal_width	-0.109369	1.000000	-0.420516	-0.356544
petal_length	0.871754	-0.420516	1.000000	0.962757
petal_width	0.817954	-0.356544	0.962757	1.000000