To: Ian Mcloughlin, Course Lecturer, GMIT Data Analytics.

From: André Roche, Student.

Date: 27-April-2018

Subject: Programming and Scripting Project 2018 – Exploration of the Iris Data Set using Python.

Summary

Following a review of Fisher's iris data set, this technical report incorporates review data and conclusions from that data using the python language.

The review criterion includes:

- Investigation of the Iris data set and its significance.
- Data analysis and supporting statistical methods to calculate solution indicators.
- Supporting python code used for investigation and stored in the relevant GitHub repository.
- Summary of investigations and conclusions.
- Potential for further analysis.
- Other applications for the python language.
- Full list of references.

Background – Investigation of the Iris Data set

Introduced by Ronald Fisher in a 1936 paper titled 'The use of multiple measurements in taxonomic problems', it is an example of linear discriminant analysis. The data set itself is a multivariate data set, that is to say it contains two or more variable quantities. Specifically it contains data of three Iris flower species. They are Setosa, Versicolor and Virginica. Four geometric features were recorded from the three species and 50 samples of each were gathered. They were Petal Length, Petal Width, Sepal Length and Sepal Width. Linear discriminant analysis is a statistical method used for classification and pattern recognition (to distribute things into groups, classes and categories). Today it is commonly used in applications such as machine learning. Linear discriminant analysis is a transformation technique and a way to reduce dimensionality (Euclidean Space). Feature sets of data can be transformed onto the same axis (or dimension -1) for better separation as opposed to trying to interrogate a scatterplot of multiple features sets on an XY plane/graph. In this, LDA attempts to separate the means of the distributions of datasets as much as possible while maintaining the smallest spread or variance possible of each dataset (Fisher's ratio). In terms of machine learning, if there were hundreds of features sets then applying a technique like this can greatly reduce computational time.

The dataset is sometimes referred to as Anderson's Iris Data Set because it was Edgar Anderson who collected the data to quantify the morphologic variation (variation in form, size, colour etc.) in the species. Perhaps today this might be done through DNA. Research notes that the data was collected using the same apparatus on the same day by the same person and measured at the same time but no data is available on the method used to take the data. It suggests that less variation in data gathering would be present based on the fact that it was the same person, same apparatus, same day etc. but the method of how they were picked, how the samples were inspected, what they were inspected with and

was accuracy and resolution correctly considered is not documented. It does open the question as how close the actual values recorded are to the conventional true values and perhaps how repeatable the method of gathering the data was. If any method is being used to catergorise and classify, the data being used to do so should be of sufficient quality and precision in order to classify correctly.

Understanding the Dataset:
A good approach to an investigation like this is to understand what is required beforehand. It is known that the project brief is to investigate a data set with python but what is actually involved in that?
A simple approach might look like the following:

Item no.	**Problem Description**	**Additional Information**
1	What is the Iris Dataset	Open in txt or csv, view it
2	What size is it	Is it hundreds or millions of data points
3	Integrity	Data missing from set
4	How is the data structured	Is it presented in a meaningful way
5	Is the data static / periodical	When and how will I be receiving this data
6	What is the purpose / what is the problem	What am I looking for in this data
7	Libraries / Modules	What tools do I need
8	Information specific to this problem	Any other additional information

Fig 1.1 – *Understanding the problem to better determine what is required*

Using an approach like this it can be determined that this is a classification problem.
The data set contains 150 measurements of four geometrical characteristics across 3 species of flowers. The problem is can a species be classified using the results from the measurement of the four characteristics. Anderson himself wanted to use this data to quantify morphological variation in the species in a time where computers and DNA analysis was not possible.
The data set is not large, relatively speaking and so analysis can easily be accomplished on a standard computer without high end components.
The data is stored in a less than desirable format, which is the species is a fifth column meaning the data for three species is combined. Python libraries are needed to handle this more efficiently if the data is required to be dissected to characteristics and species.
Finally, some statistical methods will be required to describe the data, make comparisons and then how to use those methods within the python environment.

Statistical Methods:

Mean:
It is often referred to as the average. It is commonly used in statistics to describe the average of a set of numbers. Given a set of numbers, the mean of that set is the sum of all the elements divided by the number of elements in the set.

Median:
The middle number of the data set**,** it separates the lower and upper half of data set or population. In addition to the mean can tell you more about the data set.

Range and Max/Min:
The range is distance between the max value and min value in a data set. Max and min the maximum and minimum values contained in a dataset.

Variance:
Variance measures the width or spread of data in a set from the average value.

Standard Deviation:
The standard deviation is also a measure of the spread of data about the mean. It is the square root of variance. It is linearised in order to turn it into units we can understand so we can quantify the spread about the mean.

Skew:
This measure represents the lateral deviation or the deviation from the centre, or the symmetry (or lack of) of a probability distribution. Typically if skewness is negative, it implies a deviation to the right and if it is positive, it implies a deviation to the left.

Kurtosis:
The peakedness or flatness of the graph of a distribution especially with respect to the concentration of values near the mean as compared with the normal distribution.

Normality:
Normality tests are carried out to determine if a data set is well modelled by a normal distribution.

Hypothesis testing:

Student's t-test:
Typically used to compare two sets of data or specifically their means and determine if they are significantly different or not.

Anova:
Analysis of variance can used to compare multiple sets of data to each other to determine if their means are significantly different.

Other potential useful methods (Prediction / classification):

Logistic Regression:
Logistic Regression is a statistical method for analysing a dataset in which there are one or more independent variables that determine an outcome.

KNN:
K nearest neighbours is an algorithm that finds the most similar observations to the one you have to predict and from which you derive a good intuition of the possible answer by averaging the neighbouring values.

Python and Libraries:

Python:
Python is a general-purpose scripting language, created by the Dutch programmer Guido Van Rossum in 1989. It possesses a very simple syntax with great extensibility, thanks to its numerous extension libraries, making it a very suitable language for prototyping and general coding. Because of its native C bindings, it can also be a candidate for production deployment. The language is actually used in a variety of areas, ranging from web development to scientific computing, in addition to its use as a general scripting too.

Libraries:

NumPy:
A library that contains a powerful statistical package and algebra routines.

Matplotlib:
Matplotlib is an extensively used plotting library, especially designed for 2D graphs and contains the pyplot module.

SciPy:
SciPy is a stack of very useful scientific Python libraries, including NumPy, pandas, matplotlib, and others, but it also the core library of the ecosystem, with which we can also perform many additional fundamental mathematical operations, such as integration, optimization, interpolation, signal processing, linear algebra, statistics, and file I/O.

Pandas:
Library which is excellent for many statistical and data mangling methods, such as I/O, for many different formats, such as slicing, sub setting, handling missing data, merging, and reshaping.
The DataFrame object is one of the most useful features of the whole library, providing a special 2D data structure with columns that can be of different data types.

Scikit-learn:
A useful library with tools for data analysis and machine learning. It comes with the iris data set built in. It's built on NumPy, SciPy and matplotlib. Useful for classification, regression, clustering, dimensional reduction and model selection**.**

Seaborn:
Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

All of the above libraries come pre-packaged/bundled with Anaconda and so no additional installations were required.

Initial Evaluation

The content of the data set is explained in a previous section, however it is useful to examine the data contained within the file before commencing investigation. It is useful to know what information the file contains, if it is missing any data or if for example the file integrity is compromised.

Some basic interrogation of the file using supporting python code returns the following information:

```
The features values are (sample first 5 rows):
 [ 5.1  3.5  1.4  0.2]
 [ 4.9  3.   1.4  0.2]
 [ 4.7  3.2  1.3  0.2]
 [ 4.6  3.1  1.5  0.2]
 [ 5.   3.6  1.4  0.2]

The feature names and units of measurement are:
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']

The species names are as follows:
['setosa' 'versicolor' 'virginica']

Shape and storage type:
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
(150, 4)
(150,)
```

Fig 1.2 – *Summary of file contents*

Using simple python commands, it can be determined with relative ease what the content of the file is structured like. It shows that there are a total of 5 columns, 4 of which contain data and 1 contains species information. The data is stored in a numpy array (150 lines by 4 columns) and the species is contained in 1 column with 150 lines.
The features or data are Sepal Length, Sepal Width, Petal Length and Petal Width while the species are Setosa, Versicolor and Virginica. The units of measurement are centimetres.

Summary of the Data Set: Available in supporting python code.

	sepal_length	sepal_width	petal_length	petal_width
Count	150.000000	150.000000	150.000000	150.000000
Mean	5.843333	3.057333	3.758000	1.199333
Std	0.828066	0.435866	1.765298	0.762238
Min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
Max	7.900000	4.400000	6.900000	2.500000

Fig 1.3 – *Summary of Data*

The summary provides a quick overview and puts some values on the dataset.
It can be seen that the means for each feature are different although sepal width and petal length are close. Petal length has a large standard deviation compared to everything else which might suggest some outliers are present in the data. Looking at the minimum and maximum, all features are different. One point to note is that at this point the values are not separated by species or 'classes'.

Further Analysis: Sepal Length, Sepal Width, Petal Length and Petal Width

Column1	Column2
The Summary for Sepal Length is as follows:	The Summary for Sepal Width is as follows:
Max = 7.9	Max = 4.4
Min = 4.3	Min = 2.0
Mean = 5.84333333333	Mean = 3.054
Standard Deviation = 0.825301291785	Standard Deviation = 0.432146580071
Median = 5.8	Median = 3.0
Skew = 0.3117530585022963	Skew = 0.330702812773315
Kurtosis = -0.5735679489249765	Kurtosis = 0.24144329938318343
Normality = NormaltestResult(statistic=5.7355842362357334, pvalue=0.0	Normality = NormaltestResult(statistic=3.5766421600696949, pvalue=0.1

Fig 1.4 – *Further Analysis*

Column3	Column4
The Summary for Petal Length is as follows:	The Summary for Petal Width is as follows:
Max = 6.9	Max = 2.5
Min = 1.0	Min = 0.1
Mean = 3.75866666667	Mean = 1.19866666667
Standard Deviation = 1.75852918341	Standard Deviation = 0.760612618588
Median = 4.35	Median = 1.3
Skew = -0.2717119501716388	Skew = -0.10394366626751729
Kurtosis = -1.3953593021397128	Kurtosis = -1.3352456441311857
Normality = NormaltestResult(statistic=221.33178660723647, pvalue=8.6	Normality = NormaltestResult(statistic=136.77701788227716, pvalue=1.9

Fig 1.5 – *Further Analysis*

In addition to the summary above, skew, kurtosis and a normality test were performed to describe the data more. All results are taken from supporting python code.

Sepal Length:
The distribution is skewed positively suggesting a deviation to the left. The kurtosis value is negative suggesting lighter tails and a flatter peak.
Normality testing returned a P-value <0.05 (alpha) suggesting that we will reject the null hypothesis and conclude that the alternate hypothesis is true at 95% confidence (it is not normal).

Sepal Width:
The distribution is skewed positively suggesting a deviation to the left. The kurtosis value is positive suggesting heavier tails and a sharper peak.
Normality testing returned a P-value >0.05 (alpha) suggesting that we will fail to reject the null hypothesis.

Petal Length:
The distribution is skewed negatively suggesting a deviation to the right. The kurtosis value is negative suggesting lighter tails and a flatter peak.
Normality testing returned a P-value >0.05 (alpha) suggesting that we will fail to reject the null hypothesis.

Petal Width:
The distribution is skewed negatively suggesting a deviation to the right. The kurtosis value is negative suggesting lighter tails and a flatter peak.
Normality testing returned a P-value >0.05 (alpha) suggesting that we will fail to reject the null hypothesis.

Distributions for each feature:

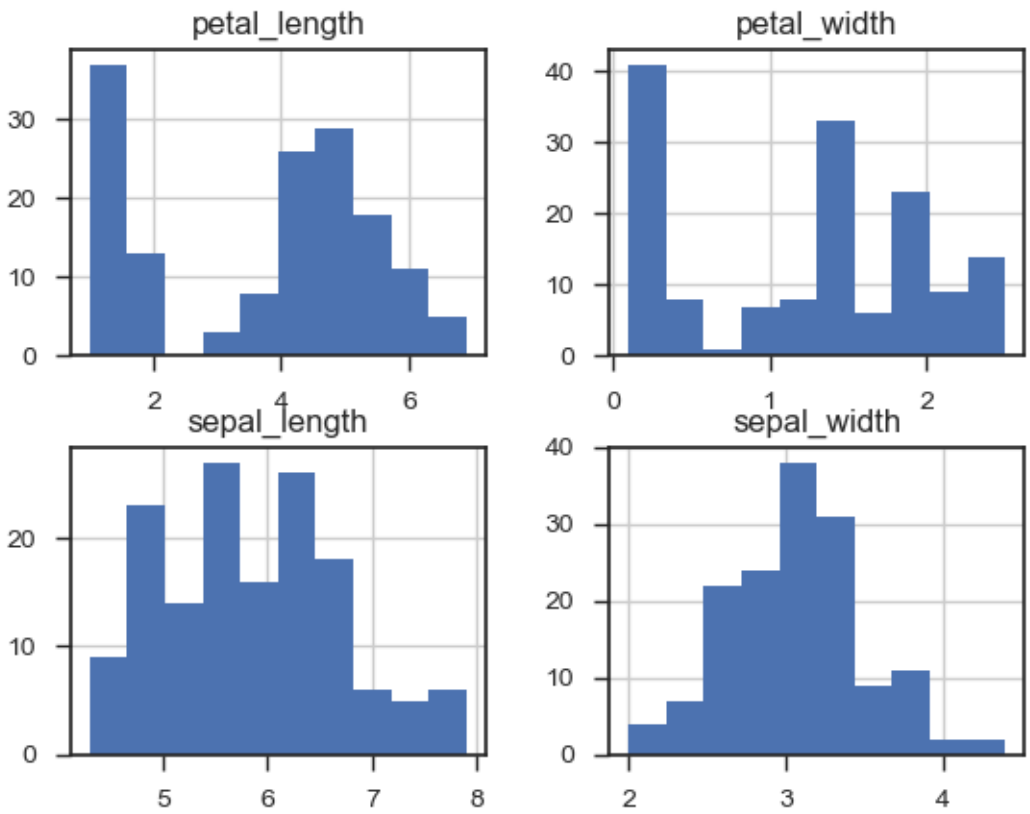


Fig 1.6 – *Distribution Graphs for Iris Characteristics*

The distribution graphs show that some kind of subset is present within petal length and petal width.

This does not provide enough information to allow any kind of categorisation. The data will have to be sorted by species to get a better insight.

Analysing the data by species:

The mean of each feature across species:

	sepal_length	sepal_width	petal_length	petal_width
species				
setosa	5.006	3.428	1.462	0.246
versicolor	5.936	2.77	4.26	1.326
virginica	6.588	2.974	5.552	2.026

Fig 1.7 – *Mean of features across species*

This is definitely more informative from a categorisation point of view.
Petal length for setosa is significantly different from that of versicolor and virginica and should help to discriminate more easily. The differences for versicolor and virginica does appear to be large enough to use as a discriminator but the distributions may overlap.
Petal width for setosa is also quite small relative to the other two species so this may also be a good indicator for discrimination.
Finally, the sepal length mean for virginica is large compared to setosa. It might be possible to use this also.

The median of each feature across species:

	sepal_length	sepal_width	petal_length	petal_width
species				
setosa	5.0	3.4	1.5	0.2
versicolor	5.9	2.8	4.35	1.3
virginica	6.5	3.00	5.55	2

Fig 1.8 – *Median of features across species*

Very similar to the mean values above and confirms that there isn't any strange average being derived from one value being very different to the rest in the set.

The Standard Deviation of each feature across species:

	sepal_length	sepal_width	petal_length	petal_width
species				
setosa	0.352490	0.379064	0.173664	0.105386
versicolor	0.516171	0.313798	0.469911	0.197753
virginica	0.635880	0.322497	0.551895	0.274650

Fig 1.9 – *STD of features across species*

Relatively large standard deviations imply there is a possibility of distributions overlapping.

The max and min of each feature across species:

	sepal_length	sepal_width	petal_length	petal_width
species				
setosa	5.8	4.4	1.9	0.6
versicolor	7.0	3.4	5.1	1.8
virginica	7.9	3.8	6.9	2.5

Fig 1.10 – *Max of features across species*

	sepal_length	sepal_width	petal_length	petal_width
species				
setosa	4.3	2.3	1.0	0.1
versicolor	4.9	2.0	3.0	1.0
virginica	4.9	2.2	4.5	1.4

Fig 1.10 – *Min of features across species*

This shows useful data. Looking at the sepal length, if the value is greater than 7 then it must be a virginica species flower. For petal width, if the value is between 0.1 and 0.6 it must be a setosa species flower. Petal length also has some useful information. If the value found is between 1.0 and 1.0 then it must be setosa species flower.

Box plot by Species:

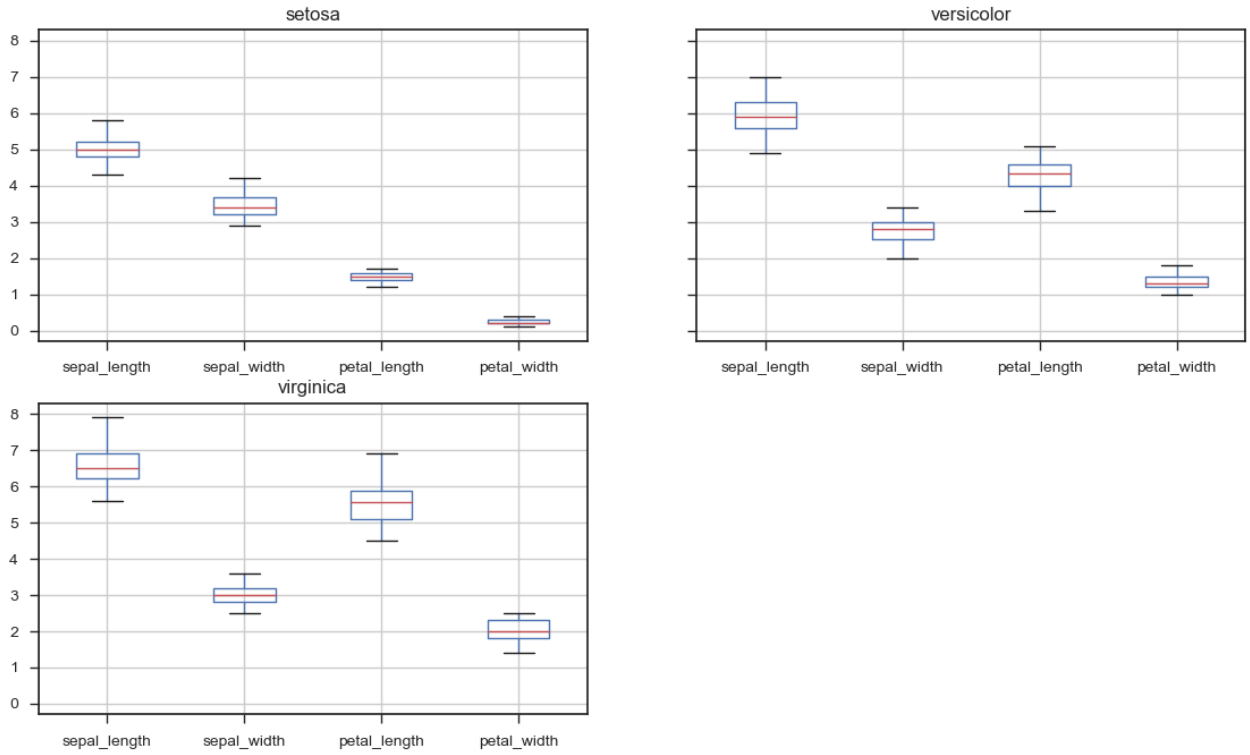


Fig 1.11 – *Boxplot of features across species*

Supporting earlier information, the boxplots show that for setosa, petal width and petal length are different enough to discriminate against the other two species. Sepal width is very similar across all species.

Distributions of features across species:

Setosa:

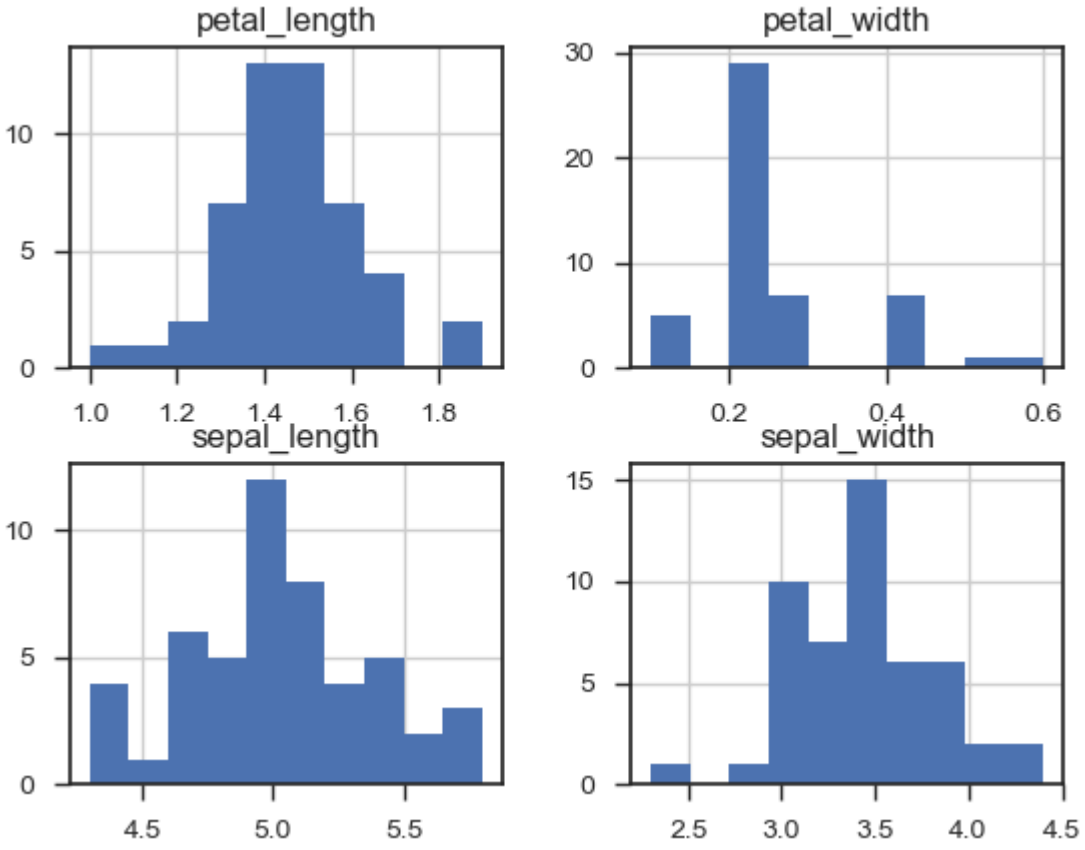


Fig 1.12 – *Distributions of features across species*

Distributions of features across species:

Versicolor:

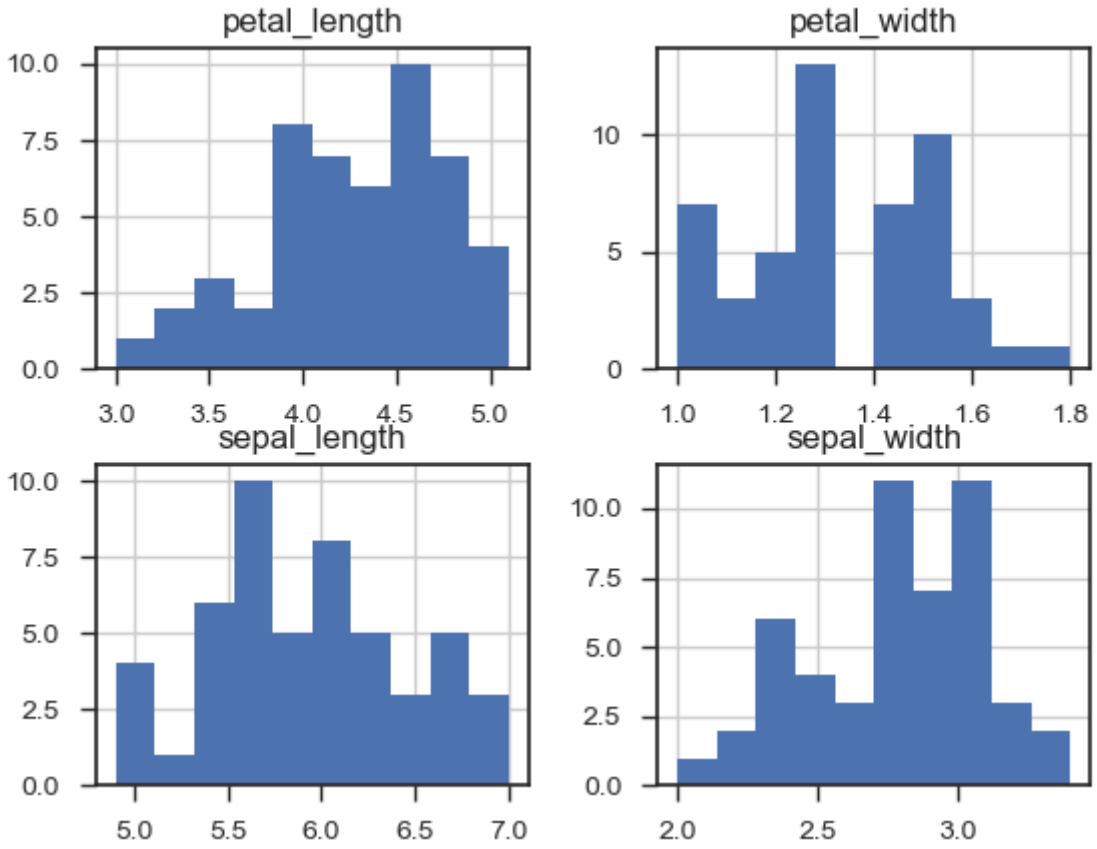


Fig 1.13 – *Distributions of features across species*

Distributions of features across species:

Virginica:

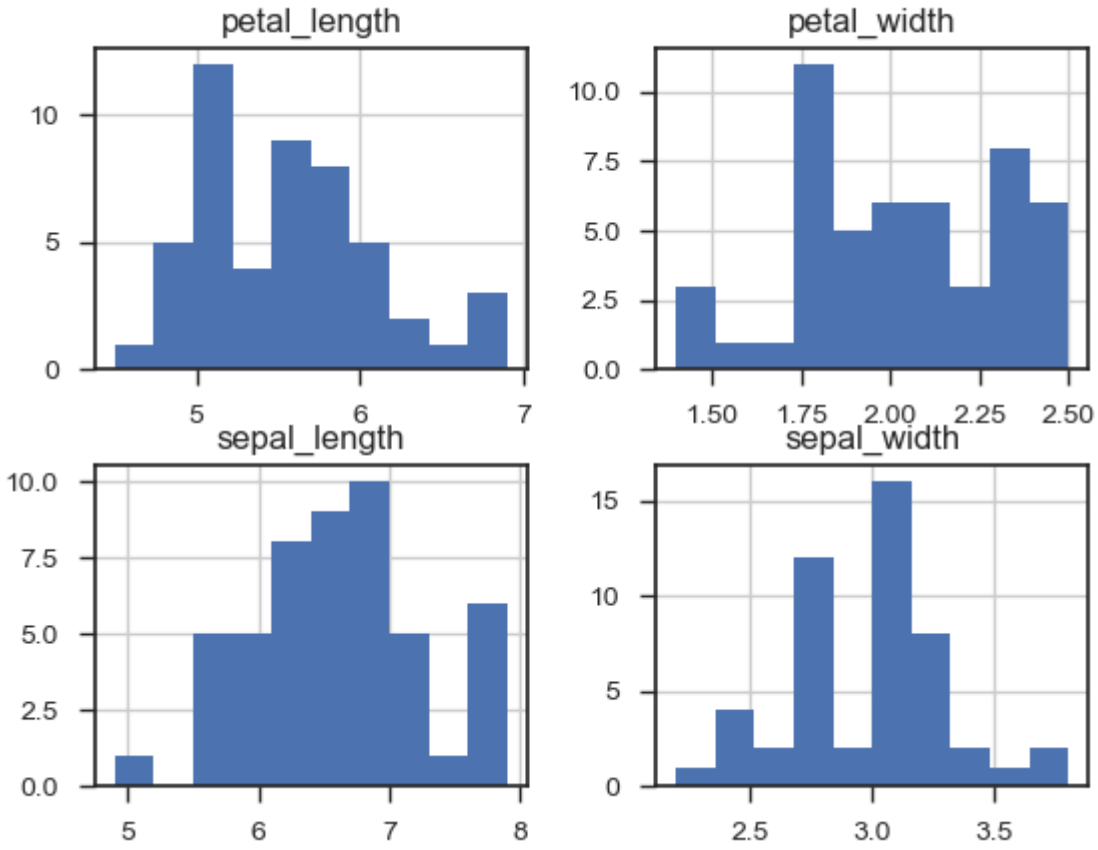
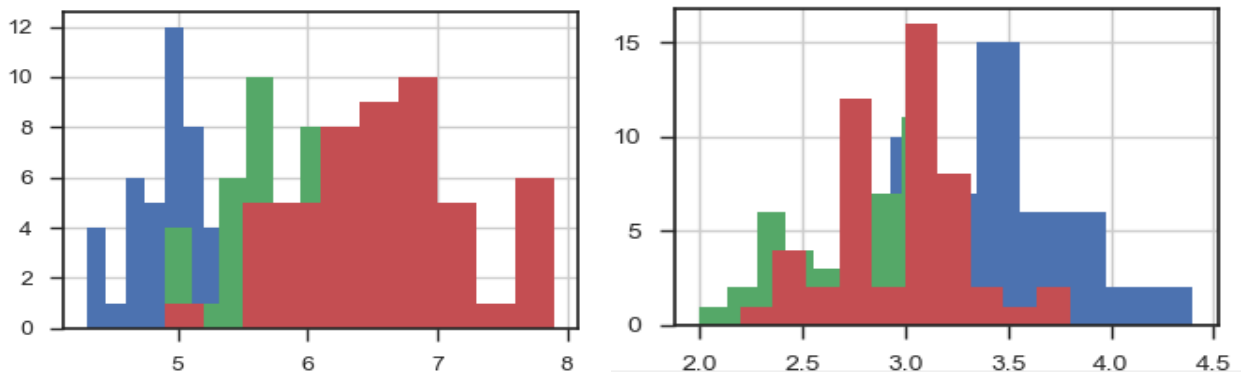


Fig 1.14 – *Distributions of features across species*

Overlapping Distributions:

Sepal Length and Sepal Width:



Petal Length and Petal Width:

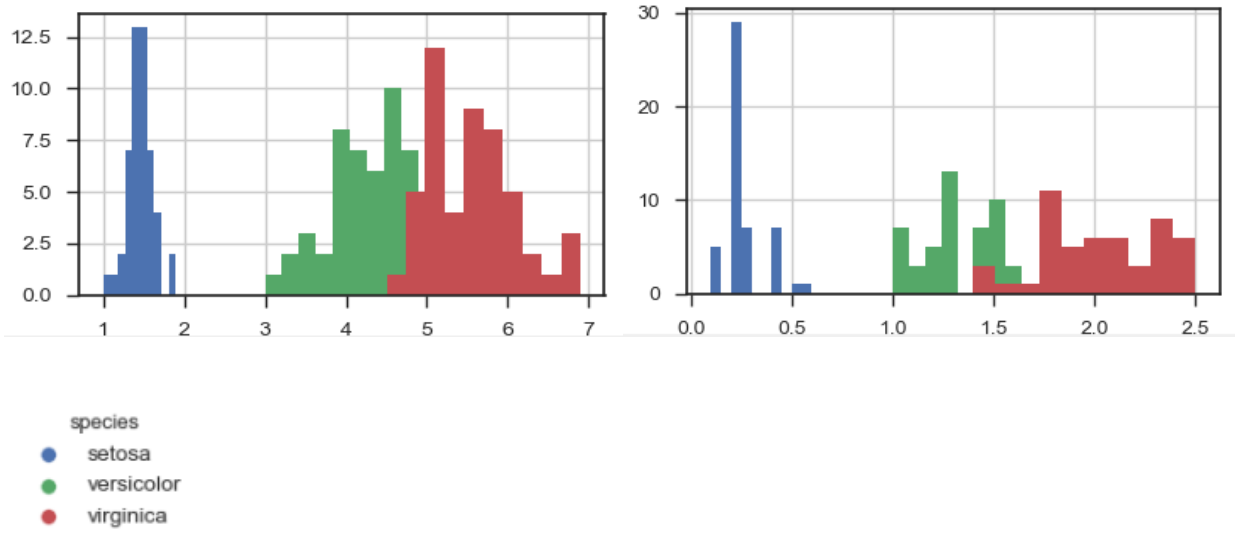


Fig 1.15 – *Overlapping distributions of features across species*

Correlation matrix across the entire dataset:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

Fig 1.16 – *Correlation of features*

The correlation matrix across the data set shows strong positive correlation between petal length and width at 0.96. Also, petal length and sepal length have the second highest correlation at 0.87. It is not known if this is the case across species or if there are any additional factors that may influence the results.

species		petal_length	petal_width	sepal_length	sepal_width
setosa	**petal_length**	1.000000	0.331630	0.267176	0.177700
	petal_width	0.331630	1.000000	0.278098	0.232752
	sepal_length	0.267176	0.278098	1.000000	0.742547
	sepal_width	0.177700	0.232752	0.742547	1.000000
Versicolor	**petal_length**	1.000000	0.786668	0.754049	0.560522
	petal_width	0.786668	1.000000	0.546461	0.663999
	sepal_length	0.754049	0.546461	1.000000	0.525911
	sepal_width	0.560522	0.663999	0.525911	1.000000
Virginica	**petal_length**	1.000000	0.322108	0.864225	0.401045
	petal_width	0.322108	1.000000	0.281108	0.537728
	sepal_length	0.864225	0.281108	1.000000	0.457228
	sepal_width	0.401045	0.537728	0.457228	1.000000

Fig 1.17 – *Correlation of features across species*

It's a different story when evaluating across species. It clear that the strong correlation between petal length and petal width across the set is not true across the species. It's actually a lot less for setosa (0.33) and virginica (0.32) but remains high for versicolor (0.78). Sepal length and petal length had the second highest correlation across the set but across species we can see that it is again not the case for setosa (0.26) but remains high for both versicolor (0.75) and virginica (0.86). Looking through the tables it's clear that other features display the same within flower variations. There is another factor or 'within species variation' here that could be explored.

Scatterplot across species:

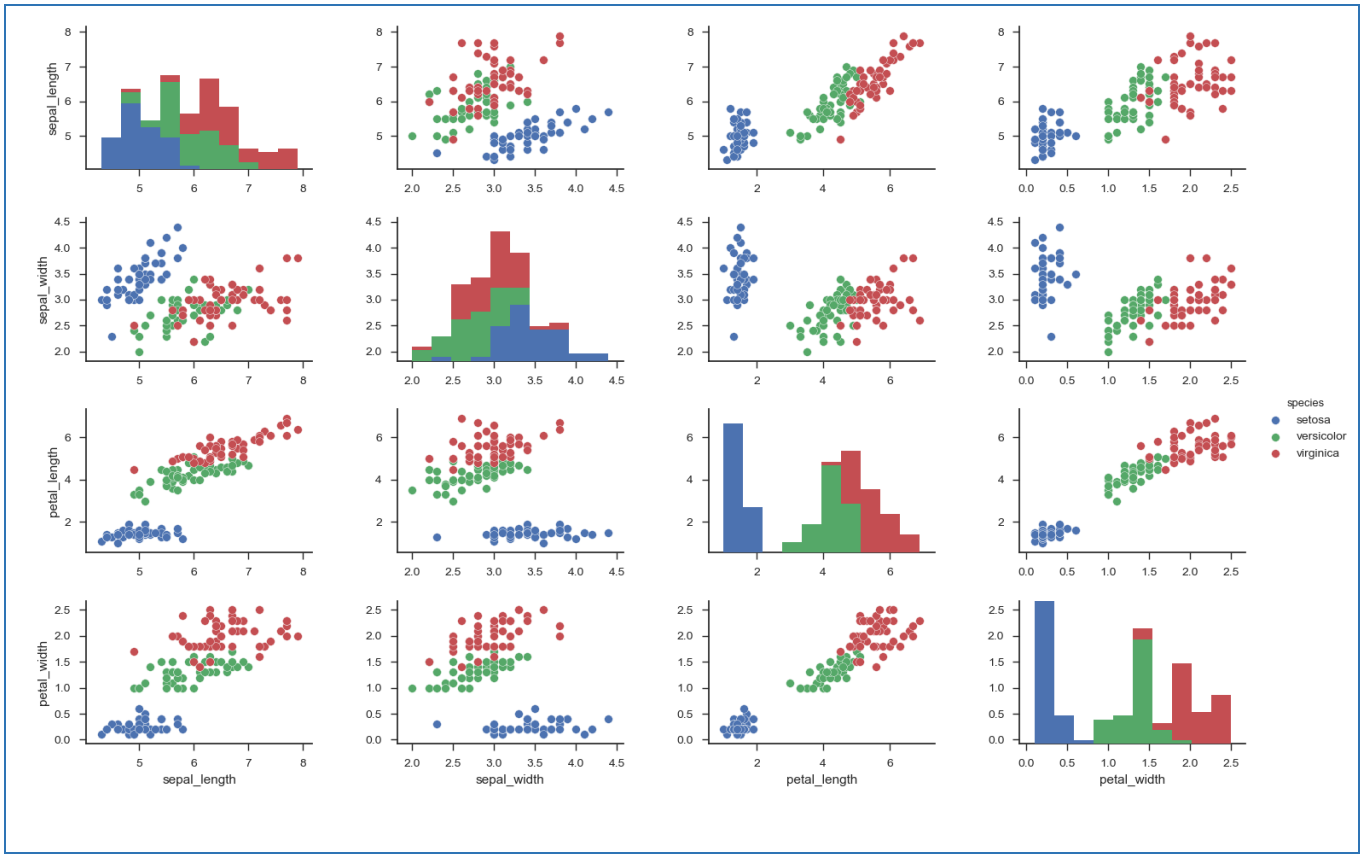


Fig 1.18 – *Scatterplot of features across species*

The scatterplot shows that setosa is probably the easiest to categorise as in most cases for each feature it separated from versicolor and virginica. Both virginica and versicolor tend to overlap across features.

Classification:
Since we have what Fisher termed a supervised learning problem, we can learn the relationship between the iris measurements and the species.

Using the scikitlearn library, which has a built in load_iris function, we can load the iris data set and store the values of the features and the species in numpy arrays. We can then store features in one variable and the species in another. Scikitlearn library requires that the features and response must be stored as numeric and so a number will be assigned to the each of the species.
Using K-nearest neighbours, we can predict the species from a set of specified features.

Please refer to accompanying python program for details.

By inputting the unknown iris (1, 2, 3, 4) using k=1, the model predicts the species as a 2 which is Virginica.
By inputting the unknown Iris (3,5,4,2) and (5,4,3,2) the model predicts a 2 and 1 which is a Virginica and a Versicular.
By inputting the same unknown Iris but changing K to K=3, the model predicts both iris as Versicolor.
The K number can have an effect on the overall prediction and parameters for the k-nearestneighbors classifier are left at default. Both of these are not fully understood at this point and are future learnings.

Finally, using a logistic regression techniques to train the model and by inputting the same unknown iris flowers (3,5,4,2) and (5,4,3,2), the model predicts a 2 and 0 which is a Virginica and a Setosa.
It's clear some fine tuning is required here and for any prediction model. Also, all prediction models would require validation to ensure accuracy of model.
This is future learning and outside the scope of this report.

Conclusion:
Python, with its vast library's and modules is a powerful tool for data analysis. It has powerful tools for I/O and reporting, generating meaningful reports within minutes. It is an incredibly powerful tool for statistics and can automate quite a lot of tasks that would be otherwise mundane in a typical statistics package.

References:

Machine learning for developers 2017 ISBN 978-1-78646-987-8

Python Data Science Essentials - Second Edition 2016 ISBN 978-1-78646-213-8

An Introduction to Statistics with Python First Edition 2016 ISBN 978-3-319-28316-6 (e-book)

Machine Learning For Absolute Beginners Second Edition 2017

https://stackoverflow.com/questions/38583738/get-descriptive-statistics-of-numpy-ndarray

http://nullege.com/codes/search/numpy.skew

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.normaltest.html

https://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.stats.skew.html

https://stackoverflow.com/questions/45483890/how-to-correctly-use-scipys-skew-and-kurtosis-functions

https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

https://github.com/PyCQA/pylint/issues/1161

https://www.isixsigma.com/tools-templates/analysis-of-variance-anova/how-compare-data-sets-anova/

https://warwick.ac.uk/fac/sci/moac/people/students/peter_cock/r/iris_lm/

http://www.pythonforfinance.net/2016/04/04/python-skew-kurtosis/

https://machinelearningmastery.com/quick-and-dirty-data-analysis-with-pandas/

https://www.youtube.com/channel/UCnVzApLJE2ljPZSeQylSEyg

www.dataschool.io

http://scikit-learn.org/stable/tutorial/basic/tutorial

https://seaborn.pydata.org/