Assignment 1: Datasets, Exploratory Data Analysis and Data Preprocessing

In this assignment, you'll examine datasets, perform statistical analysis and generate visualizations to understand the data better, and perform pre-processing steps to prepare the data for analysis.

In order to complete this assignment, you will need Jupyter (formerly iPython) - an interactive python environment that runs in your web browser. There are many ways to install Jupyter, but the easiest way is probably to download Anaconda (https://www.continuum.io/downloads), which includes Python, Jupyter, and the most important data analysis tools in one installer.

Once you finish the installation process, you can run the command `jupyter notebook` at the command prompt (Windows) or in a terminal (Mac OS X, Linux) to start the notebook environment. You can then open the notebooks you've downloaded from eCommons in iPython and complete the assignment.

When you complete each question, I expect you to show your work as well as answer the question. If the question asks for statistics, include a cell containing the code you used to compute the statistics and anoter cell in Markdown (https://daringfireball.net/projects/markdown/basics) format with your answer.

Once you've completed the assignment, you'll need to turn in:

- Three PDFs, one for each part, generated by converting the iPython notebook to a PDF. You can do this by first generating a print preview and saving the preview as a PDF, or you can use the more complicated approach of installing pandoc and LaTeX, then generating a PDF using the "Download As" menu item.
- Three Jupyter notebooks (ipynb format) corresponding to your solutions for each part. Include these in case there's an issue with your solution and we want to assign partial credit.

Submit these six files in eCommons by the deadline, Monday, 1/18/16, at 11:59pm

Part 1: Instance, Attributes and Attribute Value Types (20 points)

For this part of the assignment, we'll explore how to load data - first using built-in datasets from the scikit-learn package, and then from a simple CSV flatfile.

In [1]:
```
## Preliminaries

#Show plots in the notebook
%matplotlib inline

# To start we import some prerequisites
from sklearn import datasets
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import urllib2
```

In [2]:
```
# Let's load the famous Iris dataset in scikit-learn!

iris = datasets.load_iris()

# The datasets in scikit-learn come with lots of metadata documenting
#Let's look at a full description of this dataset

print iris.DESCR
```

```
Iris Plants Database

Notes
-----
Data Set Characteristics:
    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the
class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica
    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:    0.1  2.5   1.20  0.76     0.9565  (high!)
    ============== ==== ==== ======= ===== ====================

    :Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988

This is a copy of UCI ML iris datasets.
http://archive.ics.uci.edu/ml/datasets/Iris (http://archive.ics.uc
i.edu/ml/datasets/Iris)

The famous Iris database, first used by Sir R.A Fisher

This is perhaps the best known database to be found in the
pattern recognition literature.  Fisher's paper is a classic in the
field and
is referenced frequently to this day.  (See Duda & Hart, for exampl
```

```
e.)  The
data set contains 3 classes of 50 instances each, where each class r
efers to a
type of iris plant.  One class is linearly separable from the other
2; the
latter are NOT linearly separable from each other.

References
----------
   - Fisher,R.A. "The use of multiple measurements in taxonomic prob
lems"
     Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contribut
ions to
     Mathematical Statistics" (John Wiley, NY, 1950).
   - Duda,R.O., & Hart,P.E. (1973) Pattern Classification and Scene
Analysis.
     (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page 21
8.
   - Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New S
ystem
     Structure and Classification Rule for Recognition in Partially
Exposed
     Environments".  IEEE Transactions on Pattern Analysis and Machi
ne
     Intelligence, Vol. PAMI-2, No. 1, 67-71.
   - Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule".  IEEE T
ransactions
     on Information Theory, May 1972, 431-433.
   - See also: 1988 MLC Proceedings, 54-64.  Cheeseman et al"s AUTOC
LASS II
     conceptual clustering system finds 3 classes in the data.
   - Many, many more ...
```

In [3]:
```
# Now let's see what attributes are available

print iris.feature_names
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'peta
l width (cm)']
```

In [4]:
```
# We can look at the raw data too. First let's get the dimensions of t

print iris.data.shape

#This tells us there are 150 rows, each of which contains 4 feature va
```

```
(150, 4)
```

In [5]:
```
# Now let's see the rows and columns

print iris.data
```

```
[[ 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]
 [ 5.4  3.9  1.7  0.4]
 [ 4.6  3.4  1.4  0.3]
 [ 5.   3.4  1.5  0.2]
 [ 4.4  2.9  1.4  0.2]
 [ 4.9  3.1  1.5  0.1]
 [ 5.4  3.7  1.5  0.2]
 [ 4.8  3.4  1.6  0.2]
 [ 4.8  3.   1.4  0.1]
 [ 4.3  3.   1.1  0.1]
 [ 5.8  4.   1.2  0.2]
 [ 5.7  4.4  1.5  0.4]
 [ 5.4  3.9  1.3  0.4]
 [ 5.1  3.5  1.4  0.3]
 [ 5.7  3.8  1.7  0.3]
```

In [6]:
```
# That's just the data, but since this is a dataset where the goal is
# labels, we can look at those labels. First we'll look at the shape,

print iris.target_names

print iris.target.shape

print iris.target
```

```
['setosa' 'versicolor' 'virginica']
(150,)
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2
 2 2]
```

Question 1 (10 points)

Okay, now that we've looked at the data, time for some questions:

1. How many objects or instances are in the dataset?

2. How many attributes are in the dataset?
3. What type is each attribute (nominal, ordinal, interval or ratio?)

1. How many objects or instances are in the dataset?
150 objects

2. How many attributes are in the dataset?
4 attributes/features

3. What type is each attribute (nominal, ordinal, interval or ratio?)
They are all ratio attributes

Loading a different dataset

Now let's try something a bit more difficult - loading a dataset from the web (which won't have all the pretty metadata that scikit gives us). We'll look at a dataset from the Data.gov website. This dataset contains preliminary accident and incident data from the Federal Aviation Administration, and can be downloaded from the Data.gov webpage (http://catalog.data.gov/dataset/preliminary-accidentincident-data-daily-data-file), but we'll download it using Python as part of this notebook

In [7]:
```
#Let's download the data using the link on the page above:
faa_data = urllib2.urlopen("http://www.asias.faa.gov/pls/apex/f?p=100:

#We can use a module called pandas to parse and manipulate this data
faa_dataset = pd.read_csv(faa_data, quotechar='"', skipinitialspace=Tr

# By default, only a few columns are shown. Setting this option allows
pd.set_option('display.max_columns', None)
# Let's look at the first ten rows
print faa_dataset.head(10)
```

```
  UPDATED ENTRY_DATE EVENT_LCL_DATE EVENT_LCL_TIME     LOC_CITY_NAME
\
0      No  19-JAN-16      18-JAN-16      22:41:00Z  COLORADO SPRINGS
1      No  19-JAN-16      18-JAN-16      21:45:00Z             VENICE
2      No  19-JAN-16      18-JAN-16      21:02:00Z         SCOTTSDALE
3      No  19-JAN-16      18-JAN-16      19:13:00Z          PENSACOLA
4      No  19-JAN-16      18-JAN-16      16:45:00Z             DUBLIN
5      No  19-JAN-16      16-JAN-16      18:05:00Z        BROOKSVILLE
6      No  19-JAN-16      16-JAN-16      01:00:00Z             AUBURN
7      No  19-JAN-16      12-JAN-16      22:30:00Z           BAUDETTE
8      No  19-JAN-16      15-JAN-16      21:07:00Z            SEATTLE
9      No  19-JAN-16      18-JAN-16      00:45:00Z            MADISON

  LOC_STATE_NAME LOC_CNTRY_NAME  \
0       Colorado            NaN
1        Florida            NaN
2        Arizona            NaN
3        Florida            NaN
4        Georgia            NaN
5        Florida            NaN
```

In [8]:
```
#We can also convert this into the same sort of data matrix we used in
print faa_dataset.values
```

```
[['No' '19-JAN-16' '18-JAN-16' ..., nan nan nan]
 ['No' '19-JAN-16' '18-JAN-16' ..., nan nan nan]
 ['No' '19-JAN-16' '18-JAN-16' ..., nan nan nan]
 ...,
 ['Yes' '04-JAN-16' '02-JAN-16' ..., nan nan nan]
 ['Yes' '04-JAN-16' '02-JAN-16' ..., nan nan nan]
 ['Yes' '04-JAN-16' '02-JAN-16' ..., nan nan nan]]
```

Question 2 (10 points)

Looking at the data above, answer the following questions.

1. How many objects or instances are in the dataset?
2. How many attributes are in the dataset?
3. Can you name a:

    - nominal attribute

- ordinal attribute
- interval attribute
- ratio attribute

in the dataset?

1. How many objects or instances are in the dataset?
63

2. How many attributes are in the dataset?
42

3. Can you name a:
nominal attribute - LOC_STATE_NAME
ordinal attribute - ACFT_DMG_DESC
interval attribute - ENTRY_DATE
ratio attribute - FLT_CRW_INJ_NONE

In [9]:
```
# Qustion 2 code
# Number of objects and attributes
print faa_dataset.shape
```

```
(63, 42)
```