EDA in a Nutshell

You often jump from problem-to-problem in data science and you need to get up to speed on a new dataset, fast. A classical and under-utilized approach that you can use to quickly build a relationship with a new data problem is **Exploratory Data Analysis (EDA)**.

In this article, you will discover Exploratory Data Analysis, the techniques and tactics that you can use and why you should be performing EDA on your next problem.

1. Build a Relationship with the Data

The process of classical statistics is to test **hypotheses** already held about the problem. This is done by fitting specific models and demonstrating specific relationships in the data. It's an effective approach, but it assumes you already have hypotheses about the problem, that you already understand the data. This is rarely the case in applied machine learning.

Before you can model the data and test your hypotheses, you need to build a **relationship** with the data. You can build this relationship by spending time **summarizing**, **plotting** and **reviewing** actual real data from the domain.

This approach of analysis before modeling is called *Exploratory Data Analysis*. In spending time with the data up-front you can build an **intuition** with the data formats, values, and relationships that can help to explain observations and modeling outcomes later.

It is called exploratory data analysis because you are **exploring your understanding of the data**, building an intuition for how the underlying process that generated it works and provoking questions and ideas that you can use as the basis for your modeling.

The process can be used to **sanity check** the data, to identify **outliers** and come up with specific strategies for handling them. In spending time with the data, you can spot corruption in the values that may signal a fault in the data logging process.

2. Origin of Exploratory Data Analysis

Exploratory Data Analysis was developed by **John Tukey** at Bell Labs as a way of systematically using the tools of statistics on a problem before a hypotheses about the data were developed. It is an alternative or opposite approach to "*confirmatory data analysis*". The seminal description of the process was in Tukey's *1977* book *Exploratory Data Analysis*.

The **objective** is to understand the problem in order to generate testable hypotheses. As such, the outcomes like the graphs and summary statistics are only for you to improve *your* understanding, not to demonstrate a relationship in the data to a general audience. This gives the agile flavor to the process.

The **S language** was developed in the same laboratory and was used as the tool for EDA. The use of scripts to generate data summaries and views is a natural and intentional fit for the process.

Wikipedia provides a nice short list of the objectives of EDA:

- **Suggest hypotheses about the causes of observed phenomena**
- **Assess assumptions on which statistical inference will be based**
- **Support the selection of appropriate statistical tools and techniques**
- **Provide a basis for further data collection through surveys or experiments**

3. Techniques for Exploratory Data Analysis

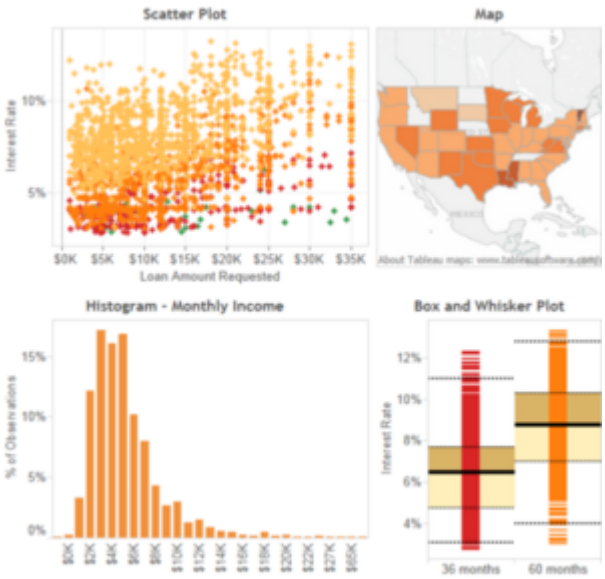
Exploratory Data Analysis is often performed with a representative sample of the data. You do not need to use all data available nor big data infrastructure. Spend time with the raw data.

Starting with **eyeballing tables** of numbers is wise. **Skimming** through tables can quickly highlight the form of each data attribute, obvious perversions and large outliners in the values and start to suggest candidate relationships to explore between attributes. Take notes.

Simple univariate and multivariate methods that give a view on the data can be used. For example, five methods that I would consider must have are:

- **Five number summaries (mean/median, min, max, q1, q3)**
- **Histogram graphs**
- **Line Charts**
- **Box and Whisker plots**

- **Pairwise Scatterplots (scatterplot matrices)**



In addition to summaries, also look at transforms of the data and re-scalings of the data. Flush out interesting structures that you can describe. Take notes. Take lots of notes.

Ask lots of **questions** of the data, for example:

- What values do you see?
- What distributions do you see?
- What relationships do you see?
- What relationships do you think might benefit the prediction problem?
- What ideas about the domain does the data spark?

4. Focus on Understanding

You are not creating a report, you are trying to understand the problem. The results are ultimately throw-away, and all that you should be left with is a greater understanding and intuition for the data and a long list of hypotheses to explore when modeling.

The code does not need to be beautiful (but they need to be correct). Use reproducible scripts and standard packages. You do not need to dive into advanced statistical methods or plots. Keep it simple and spend time with the data.

A query interface like SQL can help you play a lot of what-if scenarios very quickly with a sample of your data.

The models will only be as good as the questions and understanding you have of the data and the problem.