# DATA ANALYSIS AND VISUALIZATION USING PYTHON: PANDAS, NUMPY, SEABORN AND MATPLOTLIB

# 

# Author:

# Sourabh P Joshi

12/09/2018

# **Table of Contents**

# **Revision History**

# **Introduction**

# Purpose

# Document Conventions

# Project Scope

# **Overall Description**

# Project Perspective

# Operating Environment: Packages

# Design and Implementation Constraints

# New vs Deprecated functions

# **External Requirements**

# Hardware and Software Requirements

# Communications requirements

# **Data Features**

# Data Set Attributes

# Seaborn

# Matplotlib

* 1. Monte Carlo Analysis

**Appendix A: Future Scope**

**Appendix B: References**

# **1. INTRODUCTION**

# **1.1 PURPOSE**

# This project deals with analyzing data with respect to Python libraries namely Pandas and NumPy and create a visual understanding using Seaborn and matplotlib. The purpose of this document is to organize all findings with the use of Python libraries and present them for further use and advanced analysis. Project deals with datasets containing all types of data in different environments with variety of ranges. Analyzing this data (raw data) and generating information that gives further insights about the data serves the purpose. Along with this processing visual eye-catching information on the go is an add-on.

# **1.2 DOCUMENT CONVENTIONS**

# This document uses the following conventions.

|  |  |
| --- | --- |
| Np | NumPy import |
| Pd | Pandas import |
| Sns | Seaborn Import |
| Plt | Matplotlib import |
| Si | Stock Info import |
| Def | Defining a new function |

# **1.3 PROJECT SCOPE**

# The scope of this project is to explore and implement features from Python libraries to extract and analyze data to create information. The following are the main four libraries in use: Pandas, NumPy for data manipulation while Seaborn, Matplotlib for data visualization.

The Project makes use of 4 datasets of which 2 datasets are imported from csv files while others are extracted from internet. This data is analyzed with inbuilt functionalities of the libraries. In case the data is not readily available, the data is modified in an expected format and then analysis is performed. All the coding aspects lie within the Python coding language. The scope of project includes some other imports for reading csv files and reading any column containing datetime variable.

# **2. OVERALL DESCRIPTION**

# **2.1 PROJECT PERSPECTIVE**

# Analyze Datasets to produce information and present this information in the port of visual display or plotting using different types of graphs. Using statistical methods to find trends, patterns and Series.

# **Titanic Dataset:** This dataset has the Realtime data for the passengers on board of a ship titanic. The data is for passenger gender, position, survival, ticket range, occupation etc.

# **Stock Market Analysis:** In this dataset the Realtime stock values for Amazon, Google, Microsoft and Apple are under analysis. Data extracted is for a period of a year starting today’s date to last year same date. The columns are as stock price, open, high, low, volume etc.

# **Election Dataset:** This dataset contains information about the 2012 elections for Obama and Romney as Democratic and Republican leaders and other participants for the same. The prime focus of the dataset is with the above mentioned 2 entities. This dataset contains values Pollster, start date, end date, number of votes(observations), affiliation and so on.

# **Election Donor Dataset:** This contains data for the donations received for all the parties for campaigning the election. This data is a huge dataset with million entries in it. Attributes consist of contributor id, candidate name, contributor name, city, zipcode, occupation, amount etc.

# **2.2 OPERATING ENVIRONMENT: PACKAGES**

# Data cleaning and manipulation was mostly done using Pandas and NumPy packages in Python. Other packages were used for certain tasks. The list and their function is mentioned below.

1. Requests – Package is used for extracting information from a URL in text format.

Use: source = requests.get(url).text

1. StringIO – Package is used for converting input output in string format to other formats without data loss.

Use: poll\_data = StringIO(source)datetime – Package used for getting date time values in a specified format. The date can be manipulated with 2 digit year, 4 digit year, timestamp values etc.

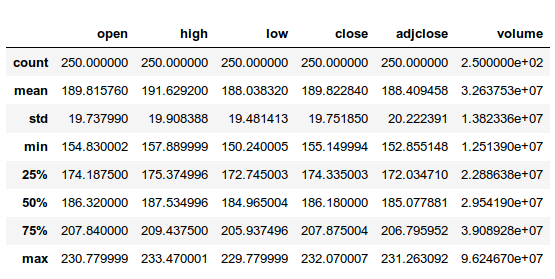
Use: start = datetime(end.year – 1,end.month,end.day)

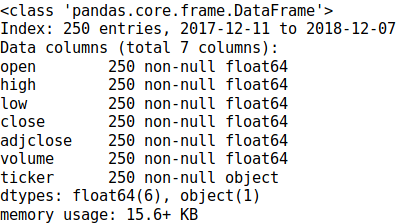
1. from yahoo\_fin import stock\_info as si – A public project for importing stock data from internet. Stock info can be extracted given a specific company name, start date and end date.

Use: globals()[stock] = si.get\_data(stock, start, end)

**2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS**

# The design of the analysis involves data to be present in a manner to be easily manipulated. The data was extracted and stored in a pandas DataFrame in a tabular format to easily visualize eahc column of the data.



Pandas DataFrame also allows to display general information about the data which also helps design functions if needed.

# Implementation constraints only require to have Jupyter Notebook with Python installed and/or running the code on a python command-line or in a file.

# **2.4 NEW VS DEPRECATED FUNCTIONS**

# The implementations used in project have changed a lot as far as Python version 2.7 and 3.6 are concerned. Many packages used in the previous version of python are deprecated and new ways to implement these functions are mentioned below.

Some packages were deprecated because of safety of data concerns like finance packages for google and yahoo stock analysis. Public projects that have packages that can be used to extract this data are used.

1. from pandas\_datareader import wb, data:

Functionality is deprecated because the data extraction was done with unsafe methods. Other types of intrusions could also be used for exploiting this package.

1. from pandas.io.data import DataReader:

The whole package for pandas.io was deprecated because of security reasons. This package was used to extract data form google/yahoo financial websites. Since these websites stopped sharing data directly this package was deprecated in the new version of Python.

A replacement for both the above packages was a public project for data extraction.

from yahoo\_fin import stock\_info as si

This package uses safe protocol to extract data.

1. pd.rolling\_mean:

Rolling mean calculation functionality in Pandas has been deprecated. This functionality is extended to pandas data structures to directly incorporate. Series and DataFrames in pandas can directly calculate Rolling mean on the data or a particular column.

Use:

AAPL[column\_name] = AAPL['adjclose'].rolling(ma).mean()

1. from StringIO import StringIO:

This package was deprecated to merge in another package. The StringIO can still be used from package io.

Use: from io import StringIO

1. Seaborn.corrplot():

The Correlation Plot method from seaborn war depricated. This plot type depicted with less clarity. Plus, heatmaps present better understanding of data than correlation plot.

# **3. EXTERNAL REQUIREMENTS**

# **3.1 HARDWARE AND SOFTWARE REQUIREMENTS**

# Laptop with decent RAM (processing power): 8GB. Election Donor Dataset is large about 15MB containing 1 million data values. Processing this will become slow with less RAM.

# A browser to support HTML, CSS and Jupyter Notebook or Anaconda installed with python 3.6 version. Flash preferable. Operating System: Windows, Linux, OS X (Mac), Python version: 3.6 or any version 3 python

# **3.2 COMMUNICATION REQUIREMENTS**

# This project supports all kinds of Browsers with HTML capacity. The communication to the project has less user interference. Datasets should be present in the same directory to extract data for analysis. Two of the data sets extract data from web.

Stock Market Analysis uses web scraping using yahoo finance website to extract stock rates for certain companies. Python inbuilt code using get data function internally implemented.

Election Analysis extracts data from a URL using a request data function, requires http protocol support for handling the request of data. This data is extracted from a website for which the URL is mentioned.

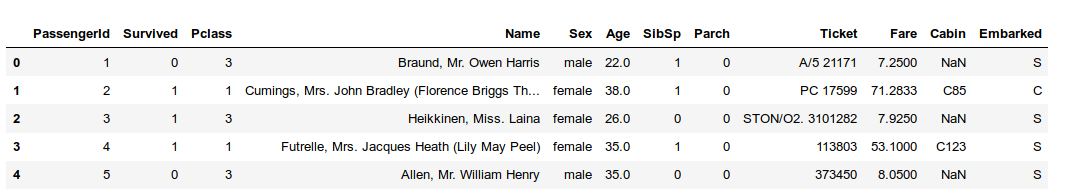
# **4. DATA FEATURES**

**4.1 DATASET ATTRIBUTES**

The project uses 4 datasets extracted in 3 different formats. 2 datasets use csv file that are downloaded and stored in the project folder.

1. Titanic dataset:

Contains dataset for titanic data for passengers, positions, fares, family details, survival etc. A brief image of the dataset is given below.



1. Stock dataset:

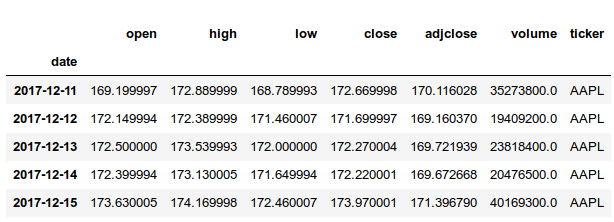
These stock details were extrected using yahoo\_fin library in Python. This is live data for recent dates. This dataset was used for realtime stock data analysis.

Figure 1Apple stock details

details

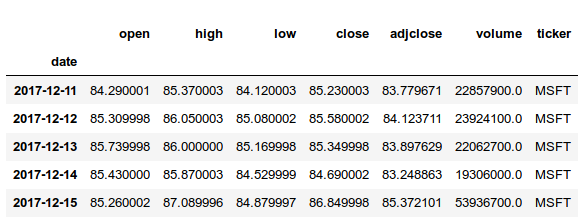
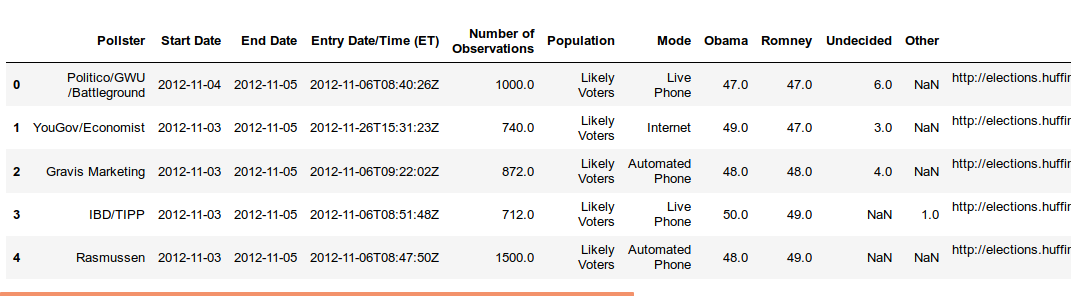


Figure 2Microsoft stock

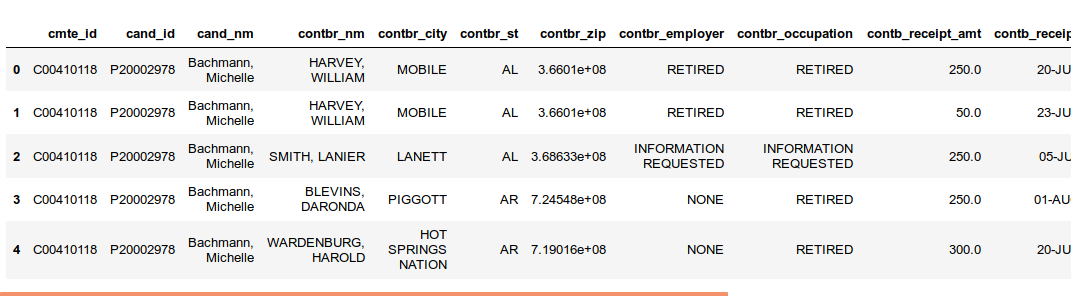
1. Election Dataset

This dataset has many column for analysis namely pollster, start date, end date, republic/democrat leader etc. This data is for 2012 US elections (Obama vs Romney)



1. Election Donor datasets

This dataset has more than a million rows, it represents the donors to the election for campaigning. This dataset has donor name, amount, person’s occupation, sex, pin code etc. This dataset test performance of the implementation as well as detailed analysis for very small aspects as well.

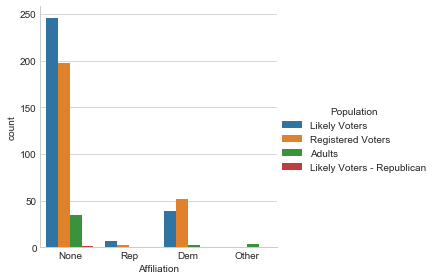


**4.2 SEABORN**

Seaborn is a data visualization library that displays data in visual format easy to interpret and make certain decisions. This library is inspired from Matplotlib, although has many advanced functions than matplotlib. Data munching and simple calculations were possible using Matplotlib. The statistical aspect of data visualization was done using Seaborn. It also has advanced statistical functions pre-implemented in the library to support advanced mathematical implementations.

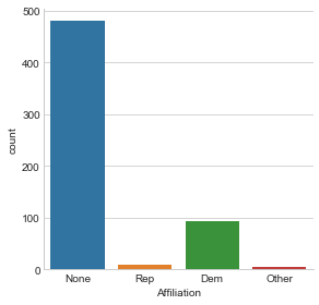
As a part of this project the following plotting techniques were explored and implemented using Seaborn:

**Factorplot**

 This function provides access to several axes-level functions that show the relationship between a numerical and one or more categorical variables using one of several visual representations.

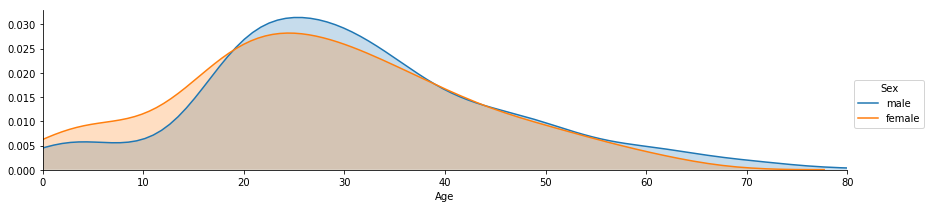
**FacetGrid**

This class maps a dataset onto multiple axes arrayed in a grid of rows and columns that correspond to levels of variables in the dataset. The plots it produces are often called “lattice”, “trellis”, or “small-multiple” graphics.



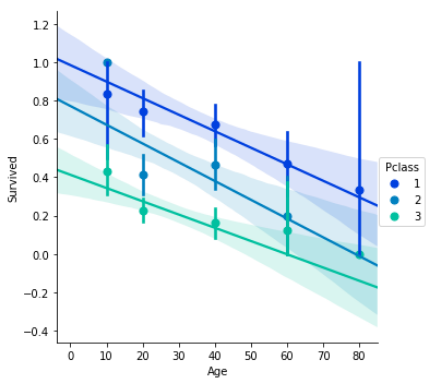
# **KDEplot**

This plot fits and plot a univariate or bivariate kernel density estimate.



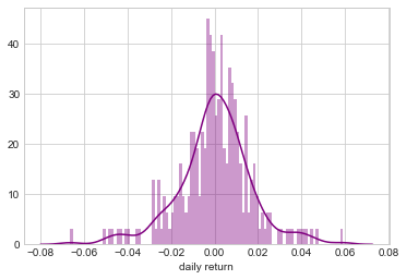
**Lmplot**

This function combines regplot and [FacetGrid](https://seaborn.pydata.org/generated/seaborn.FacetGrid.html#seaborn.FacetGrid). It is intended as a convenient interface to fit regression models across conditional subsets of a dataset.



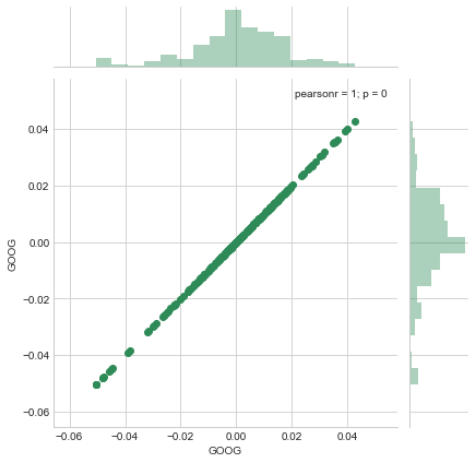
**Displot**

This function combines the matplotlib hist function (with automatic calculation of a good default bin size) with the seaborn kdeplot() and rugplot() functions. It can also fit scipy.stats distributions and plot the estimated PDF over the data.



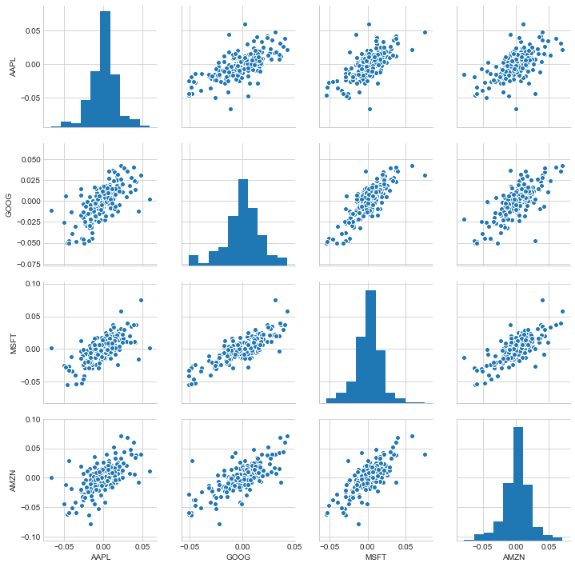
**Jointplot**

This function provides a convenient interface to the JointGrid class, with several canned plot kinds. This is intended to be a lightweight wrapper; if you need more flexibility, you should use JointGrid directly.



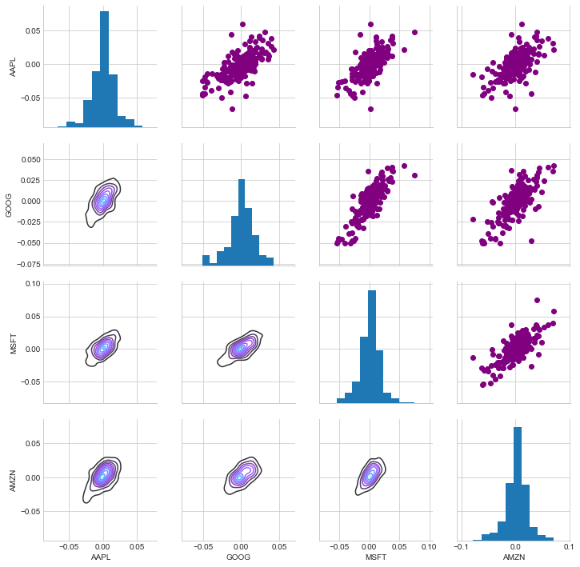
**Pairplot**

By default, this function will create a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column. The diagonal Axes are treated differently, drawing a plot to show the univariate distribution of the data for the variable in that column.



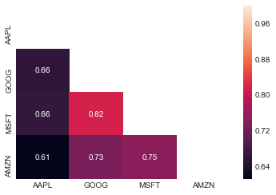
**PairGrid**

This class maps each variable in a dataset onto a column and row in a grid of multiple axes. Different axes-level plotting functions can be used to draw bivariate plots in the upper and lower triangles, and the the marginal distribution of each variable can be shown on the diagonal.



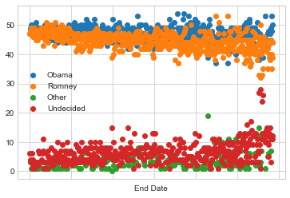
**Heatmaps**

This is an Axes-level function and will draw the heatmap into the currently-active Axes if none is provided to the ax argument. Part of this Axes space will be taken and used to plot a colormap, unless cbar is False or a separate Axes is provided to cbar\_ax.



**ScatterPlot**

The relationship between x and y can be shown for different subsets of the data using the hue, size, and style parameters. These parameters control what visual semantics are used to identify the different subsets. It is possible to show up to three dimensions independently by using all three semantic types, but this style of plot can be hard to interpret and is often ineffective. Using redundant semantics (i.e. both hue and style for the same variable) can be helpful for making graphics more accessible.



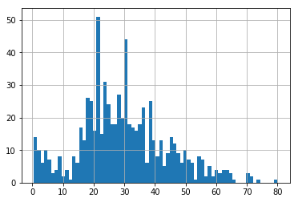
**4.3 MATPLOTLIB**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

The following plot are build pertaining to this project

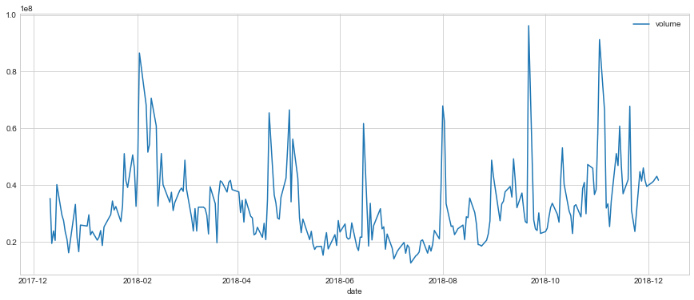
**Histogram**

2D plot of continuous data iteams.



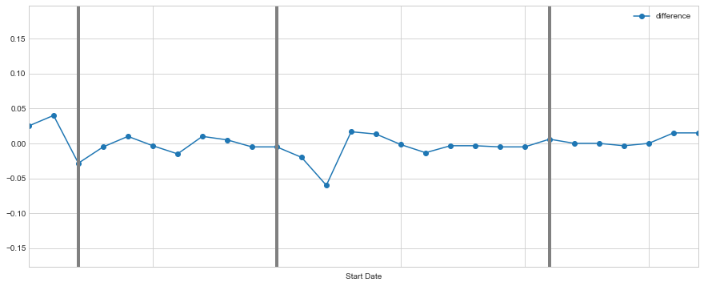
**Subplots.Axesplots**

This plot depicts a digram pertaining to the specified axes.



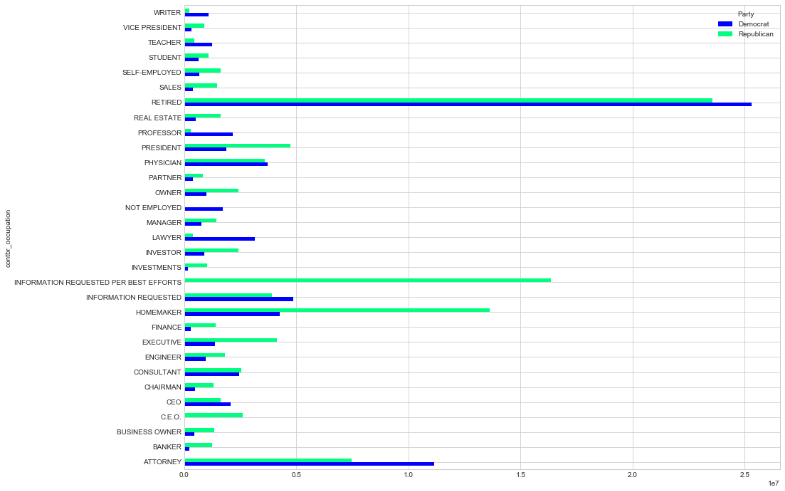
**Pyplot.avxline**

This particular function plots a line along Y-Axis at specific intervals to either specify a certain point or distribute the graph in to parts.



**Barplot**

Plotting horizontal or vertical bars for the entities and their corresponding value in the other dimension.



**4.4 MONTE CARLO ANALYSIS**

This project makes use of Monte Carlo analysis for predicting stock values given a span and starting price of a stock.

Formula for calculations: ΔS=S(μΔt+σϵ√Δt)

Here the S (stock price is multiplied by 2 terms)

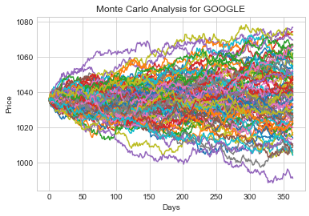
1st term: Drift:

Drift is forward movement of values based on average daily return multiplied by the change of time.

2nd term: Shock:

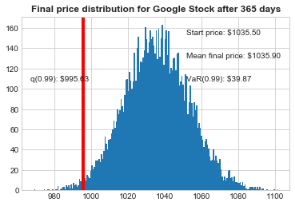
This is a vertical movement of price (up or down) randomly.

For predicting stock every time, the stock price will Drift and experience a SHock either up or down. Multiple simulations of these will generate a histogram of lines that will predict stock price at a certain point in time. Plotting different combinations of predictions for one data value over 100 days the prediction can be plotted as:



Risk analysis on this stock value reveals that the stock’s dropping value can be predicted for 95% confidence. This is a scenario, if happens the least amount to be lost for a stock bought at ‘x’ value, its margins can be predicted.

The plot for the same is presented below:



Monte Carlo is basic prediction algorithm predicting value based on Drift and Shock values.

**APPENDIX A: FUTURE SCOPE**

The project sheds light on certain aspects of data analytics and data visualization with a Python perspective. The project intends to show simple yet elegant data visualizations for faster information processing.

Upcoming data analytics tools like Tableau, Google Analytics might be able present better information combined with python or other programming language. With Jupyter Notebook, there is always a header for performance since rendering the data online and processing takes it out of the computer’s processing capacity. Some distributed environments like ‘Apache Spark’ can help analyze and visualize big data better than the tools used in this course.

This project can be extended to have a standardized procedure, hence analyzing multiple datasets at the same time. These standard procedures can be implemented at backend with simple GUI for easy implementation. IBM Cognos has capacity and can be used for this purpose.

**APPENDIX B: REFERENCES**

* 1. <https://web.archive.org/web/20171018181046/https://spotlessdata.com/blog/exploring-data-analysis>
  2. <https://www.udemy.com/learning-python-for-data-analysis-and-visualization/>
  3. <https://en.wikipedia.org/wiki/Data_visualization>
  4. <https://matplotlib.org/api/pyplot_api.html>
  5. <https://seaborn.pydata.org/api.html>
  6. <https://pypi.org/project/yahoo-fin/>
  7. <https://docs.python.org/2/library/stringio.html>
  8. <https://www.investopedia.com/university/stocks/>
  9. <https://transition.fec.gov/pubrec/fe2012/federalelections2012.shtml>