**Capstone Project** **Andrejs Zujevs**

Machine Learning Engineer Nanodegree August 15, 2016

Riga

**Quadcopter Landing Detection Problem**

# I. Definition

(approx. 1-2 pages)

## Project Overview

Drones are more formally known as unmanned aerial vehicles. Essentially, a drone is a flying robot and a quadcopter is a specific type of drone, propelled by four motors (see Figure 1). Quadcopters and other multicopters can fly autonomously for different purposes, for example, to deliver a pizza. Usually an autonomous drone’s take-off is from a home position (turn-on the motors and rise up into the air – in this project described as drone’s status – ‘Take-off’), flying to a defined destination point (described as drone’s status – ‘Flight’), then landing (described as drone’s status – ‘Land’) and returning back to the home location. In this project the word ‘landing’ means the act of drone coming to land/surface and then turning off the motors in time (immediately before the surface/land is touched).





**Figure 1.** Quadcopter (from: http://www.hobbyflip.com) (left), drone’ states (right)

In reality, the autonomous landing isn’t easy for a drone due to the noise in sensors. Usually drones use one single sensor or a combination of them to measure the actual distance to the land/surface and then turn-off the motors if it touches the land/surface or is very close to it. For example, ultrasound, [LIDAR](https://en.wikipedia.org/wiki/Lidar) and optical sensor (stereo vision) is mostly used for measuring the distance to land on a surface. There exist common problems of the drone autonomous landing:

1. noise in the sensor(-s) due to the impact of the other electronics in a drone;
2. different surfaces of the landing location; sensors may measure different altitude, although in reality the drone is at the same altitude;
3. slope of the landing surface and the drifting of the drone to other location;
4. inaccuracy in detecting the location of the landing caused by the inaccuracy of GPS sensor or low GPS signal;
5. dirt on the sensors (dust, water, etc.) due to environment conditions.

The above mentioned problems will cause a headache for drone developers and users:

1. drone’s propellers may be damaged, due to itsdrifting to a side and collision with an obstacle or landing in a high grass which in turn damage the propellers;
2. drifting of a drone (propellers not turned-off in time) may cause an injury (cut, bruise, etc.) to the user, other people or animals;
3. a drone may cause damage to the property of a third party if the drone landing position is detected with inaccuracy due to GPS signal problem or noise in sensors.

The database available for this project consists of 25 short time flights with one or many landing events during a flight. The total size of the dataset is ~88K records with 21 feature (different sensors’ data and derived data from sensors as well as data from motors). Large size of the dataset is explained with high frequency of data recording (approximately for every 30 millisecond). Each flight has three categories of data records which belong to the drone’s states: Take-off, Flight and Land. The sequence of the particular drone statuses may be repeated many times if a drone landed more than once during a flight. Also land/surface touch moment is manually labeled for each landing event than drone’s status was Land.

## Problem Statement

**The aim of this project** – to create an algorithm (with one or many alternatives) which can detect the time moment of a drone touching a surface during the landing and then send immediately a command to turn-off the motors. The final algorithm should use only actual/current data from sensors and motors **without using** data from the external sensors such as LIDAR, ultrasonic and vision sensor due to the previously mentioned problems with them. Also, the final algorithm should be used in real time on ARM processor architecture with single core at 166Mhz and limited <1MB RAM.

The defined problem may be solved with supervised learning due to the touch time moment which is labeled and we only need to construct a model that can predict the touch moment with the land or surface. First, we need to explore the dataset, detect outliers and remove theirs if necessary, then try to construct the model. For this reason, such methods as Support Vector Machine, Decision Trees, Linear Regression would be used, but we also need to take into account the complexity of the algorithm (how many support vectors to use or what depth of decision tree is adequate with respect to the memory usage and model implementation easiness in drone infrastructure).

On the other hand, if we drop touch moment label feature, then it would be interesting to explore dataset and obtain clusters of data which can represent touch moment and other opposite events. In this case unsupervised learning is more appropriate because we need to distinguish correctly two things: did or did not the drone touch the surface/land and we know about the real touch time moment only approximately. The Independent components analysis with K-means clustering is more preferred in the unsupervised learning case. The first one is for finding differences in the dataset with transformation and the second one is for data clustering after transformation with the aim to identify a cluster of data which belongs to the touch time moment. Then we can use it for predicting the touching.

## Metrics

In the supervised learning case we need to take into account FP (False Positives) and FN (False Negatives), because FP will prematurely turn-off motors and the drone may crash from high altitude, then again FN will delay motors turn-off and drone may drift and/or cause injury to people or animals. Finally, more appropriate metric is F1-score due to it’s a weighted average of the precision and recall and F1 takes into account FN and FP.

In the unsupervised learning case we only try to separate data in two or many clusters and then for prediction identify cluster which represents the touch moment. For this case also F1-score is used due to the problem that it is sensitive for FN and FP.

# II. Analysis

(approx. 2-4 pages)

## *Data Exploration*

The dataset consists of 88084 records with 21 feature:

BAR float64 # Barometer’s sensor data

MOTOR1 int64 # Rotation speed of motor 1, rpm

MOTOR2 int64 # Rotation speed of motor 2, rpm

MOTOR3 int64 # Rotation speed of motor 3, rpm

MOTOR4 int64 # Rotation speed of motor 4, rpm

LIDAR float64 # LIDAR’s sensor data, m

THRUST float64 # Thrust of motors (mixed), rate

ACC\_X float64 # Accelerometer’s sensor data (acceleration in X axis) m/s2

ACC\_Y float64 # Accelerometer’s sensor data (acceleration in Y axis) m/s2

ACC\_Z float64 # Accelerometer’s sensor data (acceleration in Z axis) m/s2

MAG\_Z float64 # Magnetometer’s sensor data (magnetic flow in Z axis), rate

GYRO\_X float64 # Gyroscope’s sensor data (orientation in X axis), rate

GYRO\_Y float64 # Gyroscope’s sensor data (orientation in Y axis), rate

GYRO\_Z float64 # Gyroscope’s sensor data (orientation in Z axis), rate

PITCH float64 # Drone pitch – internally estimated using different sensors

ROLL float64 # Drone roll – internally estimated using different sensors

YAW float64 # Drone yaw – internally estimated using different sensors

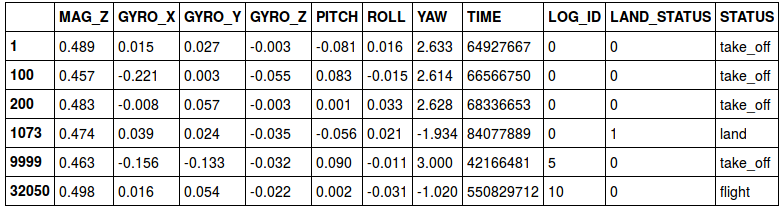
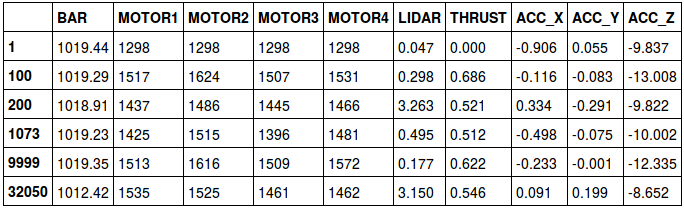
TIME float64 # Log’s record time in microseconds from logging started

LOG\_ID int64 # Log file ID

LAND\_STATUS int64 # Status– is drone actually touched a surface or not (in 0.5s time frame)

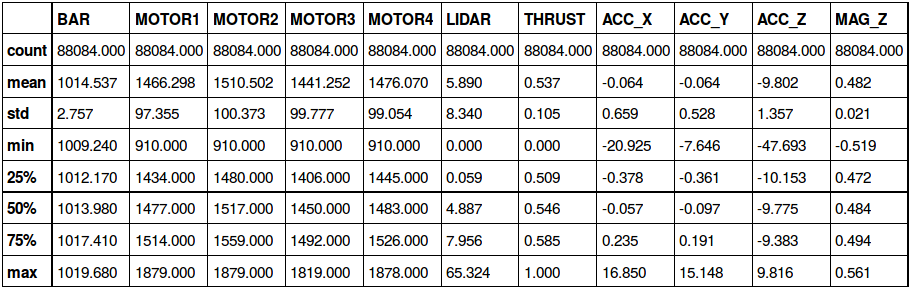
STATUS object # Drone’s state

Samples of the dataset:



The dataset consists from data time series where each belongs to the particular drone flying event. The feature’s TIME values are in microseconds. The records’ TIME when the drone is in status TAKE-OFF is with some shift due to a log file which started writing a little earlier before the take-off. Also you can see that the speed of the motors is represented with integer numbers (rpm), LIDAR (0 m – 10 m calibrated) and BAR sensor provides their raw scalar values. Thrust is scalar and in range [0.0, 1.0], there 0.0 is minimum value and motors’ rotation speed for that is 910 rpm.

Descriptive statistics of the Dataset:

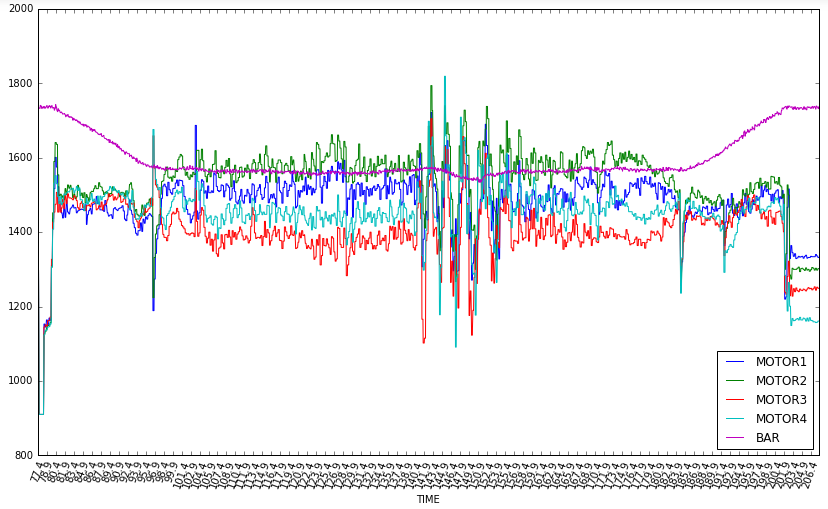


rom

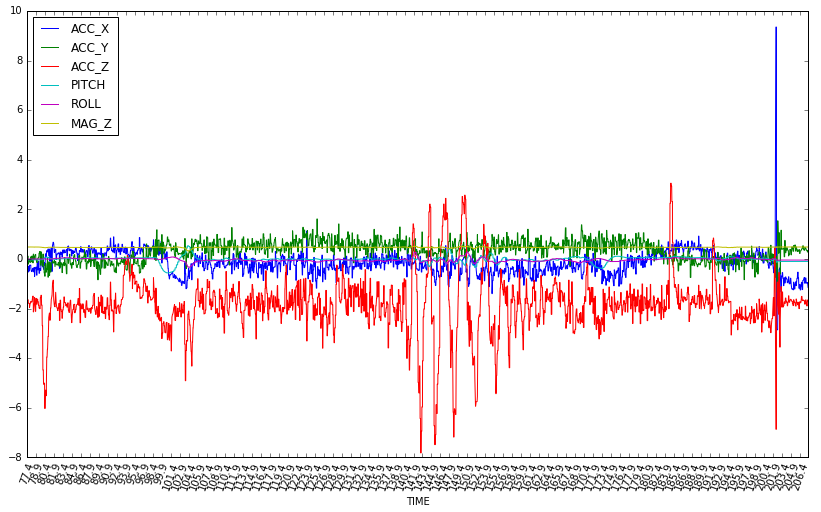
From the descriptive statistics you can see that BAR sensor’s values are in the range 1009-1019, meaning that flight evens recorded inon the soame one place and thissensor’s value is relative to place where drone fliedit. LIDAR sensor’ values are in the range 0.0-65.32, although sensor precisely can measure altitude in range 0.2 – 10 m and the smaller/larger values are out of range of sensor’s sensitivity. The feature ACC\_Z values are mostly negative due to difference of drone acceleration in Z axis and gravity constant g=~9.8m/s2.

*Exploratory Visualization*

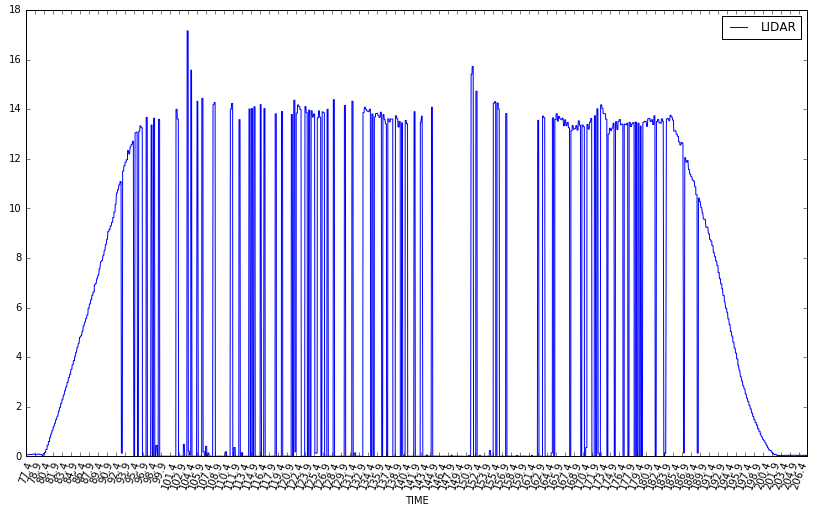
In the figures below depicted sensors’ values from the log ID=1. The figure below represents motors’ and barometer sensor values. As you can see barometer’s value significantly decreased after the 80 second (drone is taking-off) and increased after the 185 second (drone is landing). In the 201 second the drone touched the land surface and all four motors after that time point were in relatively low rotation speed. For the visualization reasons the BAR sensor’s values shifted by -1.002+5e and scaled by 100.0.



In the figure below values of the feature ACC\_Z (accelerometer’ value for the z-axis) are shifted by 8.0 (for the visualization reason). The land touching time moment happened in the 201 second and where you can see high spikes for ACC\_X and ACC\_Y.



In the next figure LIDAR sensor’s data presented (also for log ID=1), where sensor’s values scaled using 1.5 rate (for visualization resons). Abundance of spike lines explained by sensor calibration range, where altitude above 12 m interpreted by sensor incorectly.



In the figures below represented scatter plots with estimation of each feature values’ frequencies in the diagonal and data correlation between different features.

## 

In the figure above some correlation of motors exists due to motors rotations are synchronized by flight controller.

## 

In the figure above some correlation present between the THRUST and ACC\_X, ACC\_Y, ACC\_Z feature.

## 

In the figure above some correlation exists between all features due to PITCH, ROLL and YAW controlled by the drone flight controller and the gyroscope state depends on it.

* *Have you visualized a relevant characteristic or feature about the dataset or input data?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

## Algorithms and Techniques

* *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?*

1. *K-means clustering – divide data for flight, take-off and landing.*
2. *ICA – detect independed features*
3. DecisionTreeRegressor – check relevant features
4. PCA – detect principal components
5. Factor Analysis – detect differences in the features

* *Are the techniques to be used thoroughly discussed and justified?*

*no*

* *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

Firstly, data cleared from noise using outlier detection technique and data visual analysis. Also data normalized using logarithm function, due to it some features were scaled to positive values range. Secondly, using features relevant analysis, were more relevant features used for further analysis. Due to the need to find approach, how to detect the drone touch moment with the land surface then the drone is in state – landing, the PCA, ICA and Factor Analysis were used. Finally, the Factor Analysis gave more acceptable (the time moment when drone touched the land surface become more different as other data in the drone state - landing) results and using transformed data by FA the K-Means clustering was applied to detect the drone touch moment in the time.

## Benchmark

* *Has some result or value been provided that acts as a benchmark for measuring performance?*

*As our dataset also includes testing logs’ data with manually marked time moment then the drone touched the land surface, we use this subset to measure performance for the final decision rule (where used FA and clustering) accuracy using F1-score metric. Please see the results explanation in the section – SECTION.*

* *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

# III. Methodology

(approx. 3-5 pages)

## Data Preprocessing

During this project we need to select relevant features for further analysis and data transformations. Also we need to care about the final solution performance on the drone for the real time processing of the data, hence we should use as smaller features number as possible. For PCA, ICA and FA the dataset was normalized using logarithm function. Since the logarithm function requires values larger than zero, then selected features were scaled to the positive values’ range (if in origin they were negative), also outliers detection was applied on some features using the dataset visual analysis.

Feature - “Status” (drone’s state) transformed into the separate binary features: STATUS\_flight, STATUS\_land and STATUS\_take\_off.

Linear regression (using DecisionTreeRegressor) was applied to determine relevant features (see Table X):

Table X. Relevant features using linear regression

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr.** | **Feature** | **Score value** | **Nr.** | **Feature** | **Score value** |
| **1** | BAR | 0.981 | **11** | MAG\_Z | 0.972 |
| **2** | MOTOR1 | 0.977 | **12** | GYRO\_X | 0.718 |
| **3** | MOTOR2 | 0.983 | **13** | GYRO\_Y | 0.755 |
| **4** | MOTOR3 | 0.979 | **14** | GYRO\_Z | 0.914 |
| **5** | MOTOR4 | 0.983 | **15** | PITCH | 0.910 |
| **6** | LIDAR | 0.759 | **16** | ROLL | 0.895 |
| **7** | THRUST | 0.948 | **17** | YAW | 0.953 |
| **8** | ACC\_X | 0.823 | **18** | STATUS\_flight | 1.000 |
| **9** | ACC\_Y | 0.831 | **19** | STATUS\_land | 1.000 |
| **10** | ACC\_Z | 0.871 | **20** | STATUS\_take\_off | 1.000 |

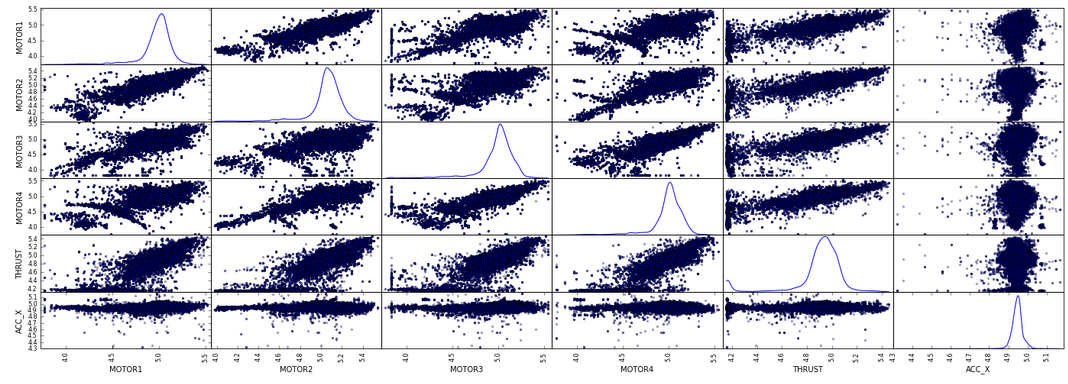
For the further analysis were used: MOTOR1, MOTOR2, MOTOR3, MOTOR4, THRUST, ACC\_X, ACC\_Y, ACC\_Z, MAG\_Z, PITCH and ROLL. All those features selection mostly based on data exploratory visualization and some relevant assumptions from experience or export knowledge, due to the linear regression analysis provides relatively the same results for all features and only several features such a LIDAR, ACC\_\*, GYRO\_\* have smallest correlation values, but not significant to say they are not impacted by other features.

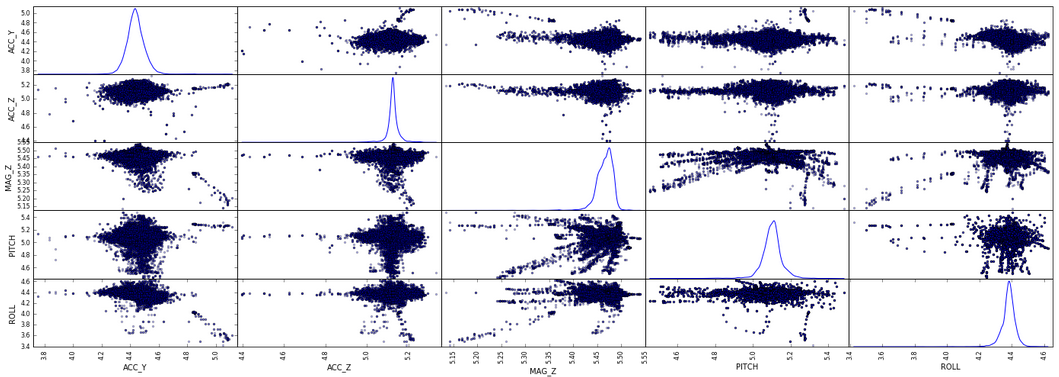
The feature BAR not used for further analysis because their values depends on a place where the drone flying. The LIDAR sensor’s values are too much noised and we can`t really trust them in near a one meter (due to specification of the sensor). GYRO\_\* features’ values also not used due to pitch, roll and yaw features are more precisely estimated and depends on mentioned features.

Features STATUS\_\* are needed only for the data separation and results validation but they are not primary used for the data analysis.

All features values scaled to range 0.0 – 255.0.

Alter all the outliers detected using such rules: 'MOTOR1': (40.868, 250.447), 'MOTOR2': (53.579, 264.316), 'MOTOR3': (23.536, 274.104), 'MOTOR4': (40.489, 259.557), 'THRUST': (34.542, 235.443), 'ACC\_X': (75.0, 190.0), 'ACC\_Y': (40.0, 175.0), 'ACC\_Z': (75.0, 220.0), 'MAG\_Z': (150.0, None), 'PITCH': (80.0, 245.0), 'ROLL': (25.0, 125.0), where in parenthesis given actual range of values. After all the scaled data looks like:





* *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented? - ok*
* *Based on the* ***Data Exploration*** *section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected? - ok, outliers filtered.*

## Implementation

* *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
* *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
* *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

## Refinement

* *Has an initial solution been found and clearly reported?*
* *Is the process of improvement clearly documented, such as what techniques were used?*
* *Are intermediate and final solutions clearly reported as the process is improved?*

# IV. Results

(approx. 2-3 pages)

## Model Evaluation and Validation

* *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
* *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
* *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
* *Can results found from the model be trusted?*

## Justification

* *Are the final results found stronger than the benchmark result reported earlier?*
* *Have you thoroughly analyzed and discussed the final solution?*
* *Is the final solution significant enough to have solved the problem?*

# V. Conclusion

(approx. 1-2 pages)

## Free-Form Visualization

* *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

## Reflection

* *Have you thoroughly summarized the entire process you used for this project?*
* *Were there any interesting aspects of the project?*
* *Were there any difficult aspects of the project?*
* *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

## Improvement

* *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
* *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
* *If you used your final solution as the new benchmark, do you think an even better solution exists?*

**Before submitting, ask yourself. . .**

* Does the project report you’ve written follow a well-organized structure similar to that of the project template?
* Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
* Would the intended audience of your project be able to understand your analysis, methods, and results?
* Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
* Are all the resources used for this project correctly cited and referenced?
* Is the code that implements your solution easily readable and properly commented?
* Does the code execute without error and produce results similar to those reported?