Big Data Analytics
Prof. Dr. Martin Theobald & Dr. Vinu Venugopal
Summer Semester 2020 – University of Luxembourg
Exercise Sheet #4 – Due on June 3, 2020 – 8:00am

Please upload your solutions on or before the due date to the provided Moodle assignment as a single zip file using your group id as the file name. Provide some brief instructions on how to run your solution to each problem in a file called `Problem_X.txt`*, and report also the most important results (such as number of results, runtimes, etc.) of your solutions to that problem within this file. Remember that all solutions may be submitted in groups of up to 3 students.*

Geo-Spatial & Temporal Data Analysis — 12 Points

Problem 1. Consider the NYC taxi trips and NYC boroughs GeoJson datasets which are available from Moodle. Also consider the sample script `RunTaxiTrips.scala` (and related classes) on Moodle as a basis to perform the following analytical tasks.

(a) Compute the number of taxi trips which (i) *started* and (ii) *ended* in *each of the NYC boroughs* over the entire period of time recorded in the data set. **2 Points**

(b) Compute the number of taxi trips which (i) *started* and (ii) *ended* in *each of the NYC boroughs* and at *each day of the week* over the entire period of time recorded in the data set. **2 Points**

(c) Modify the provided script such that it computes the *average duration between two subsequent trips* conducted by the same taxi driver *per borough* and *per hour-of-the-day* (at which the second trip starts). **2 Points**

(d) Detect potential outliers by finding taxi trips whose duration is longer than the 95% quantile of the durations of all taxi trips. **2 Points**

(e) Detect potential outliers by finding taxi trips whose duration is longer than the 95% quantile of all taxi trips normalized by the distances of the respective trips.

That is, for each taxi trip, compute its duration (e.g. in seconds) and divide this duration by the direct distance (e.g. in miles or kilometres) between the start and end point of this trip. Then sort all trips in ascending order of these ratios and cut-off all trips which are above 95% of this list to find the outliers. **4 Points**

Please make sure that you properly make use of the Spark infrastructure by loading the taxi trips and their various transformations into RDDs and by computing the requested tasks as much as possible via parallel RDD transformations.

Linking Traffic Safety & Taxi Trips Datasets — 12 Points

Problem 2. For this exercise, besides the two data sets used in Problem 1, we will additionally consider the "Motor Vehicle Collisions" open data collection (available from `https://data.cityofnewyork.us/` and via Moodle) from the New York Police Department, a CSV file which consists of records with information about collisions that occurred in New York City in 2013. Each line of this data set contains (amongst others) the following fields:

```
DATE, TIME, BOROUGH, LATITUDE, LONGITUDE, LOCATION, ON STREET NAME, CROSS
STREET NAME, OFF STREET NAME, NUMBER OF PERSONS INJURED, NUMBER OF PERSONS KILLED,
CONTRIBUTING FACTOR VEHICLE 1, CONTRIBUTING FACTOR VEHICLE 2,
VEHICLE TYPE CODE 1, VEHICLE TYPE CODE 2
```

(a) Implement a parser for the collisions data that extracts the lines (including all of the above fields) of this file into an initial RDD. Filter out useless or meaningless lines from your data set: remove all records which have no coordinates information or where the `ON STREET NAME` or `CROSS STREET NAME` is empty. **2 Points**

(b) Find the most dangerous street crossings according to the number of people that are injured or even killed in collisions which are recorded within the dataset. Return pairs of `(ON STREET NAME, CROSS STREET NAME)` together with the number of people involved (injured or killed) and a list of up to the 10 most common contributing factors (of either one of the two vehicles involved in a collision) for each such crossing. Sort all crossings in descending order of the total number of people involved in accidents. **4 Points**

(c) Based on the results of (b), analyze whether vehicles of type `SPORT UTILITY` are more frequently among the top contributing factors of accidents than vehicles of type `PASSENGER VEHICLE`. **2 Points**

(d) Finally, let us aim to find taxi trips that likely were affected by (or even possibly involved in) a collision. To do so, we first define the *affecting area* of a collision to be the area in a radius of 50 meters around the collision. Since we are not getting information about the exact trip directions, we will assume that trips follow linear surface trajectories. Thus, a taxi trip is assumed to be affected by a collision if (1) a *linear surface trajectory* from the start to the end point of the trip crosses an affecting area of a collision, and (2) the *time* of the collision also interleaves with the start to end time of the taxi trip.

The result of this query should contain all taxi trips which are found to likely be affected by a collision, together with the date, time, coordinates and contributing factors of each affecting collision. **4 Points**

Hint: See the Esri geometry API (`http://esri.github.io/geometry-api-java/javadoc/`) for a reference on the necessary distance operators.

Financial Risk Analysis **12 Points**

Problem 3. Consider the stocks and factors time-series dataset available from Moodle to solve this exercise. Also consider the sample script `RunMontoCarlo.scala` on Moodle as a basis to perform the following analytical tasks.

(a) Compute the Var and CVaR for each of the four stocks in the provided dataset by directly computing the two-week returns over only the *historical recordings* of these stocks. Compare your results with the simulated VaR and CVaR values for each of the four stocks individually.

- Which stock do you think is the safest investment according to each of the two methods?
- What is the advantage of simulating the VaR and CVar values as it is done in the provided Scala class?

4 Points

(b) Based on the provided script, analyze how the Var and CVaR values change when we consider (i) *1-day*, (ii) *1-week*, (iii) *2-week* and (iv) *1-month* returns (instead of just the 2-week returns computed by the script).

Print the respective Var and CVaR values, and plot the underlying distributions of all four series of returns you obtain from this setting via the `breeze-viz` package (similarly to how the second plot of the provided script is created). **4 Points**

(c) Assume we wish to *drop the analysis of correlations among market factors* (which is currently implemented via the Multivariate Normal Distribution in the provided script), and instead assume that all market factors are independent of each other.

Modify the provided script to accommodate this independence assumption, i.e., sample the factors f_j from each of the three distributions of historical factor returns *independently* from each other, and feed these factors as features into the linear-regression model to predict the respective stock returns r_i.

Based on the averages of the predicted returns among the four stocks in our portfolio, compute the VaR and CVaR values now under this independence assumption. **4 Points**