**Project classes detailed description**

**Function name:** AverageInsuranceAvailedLocationWise

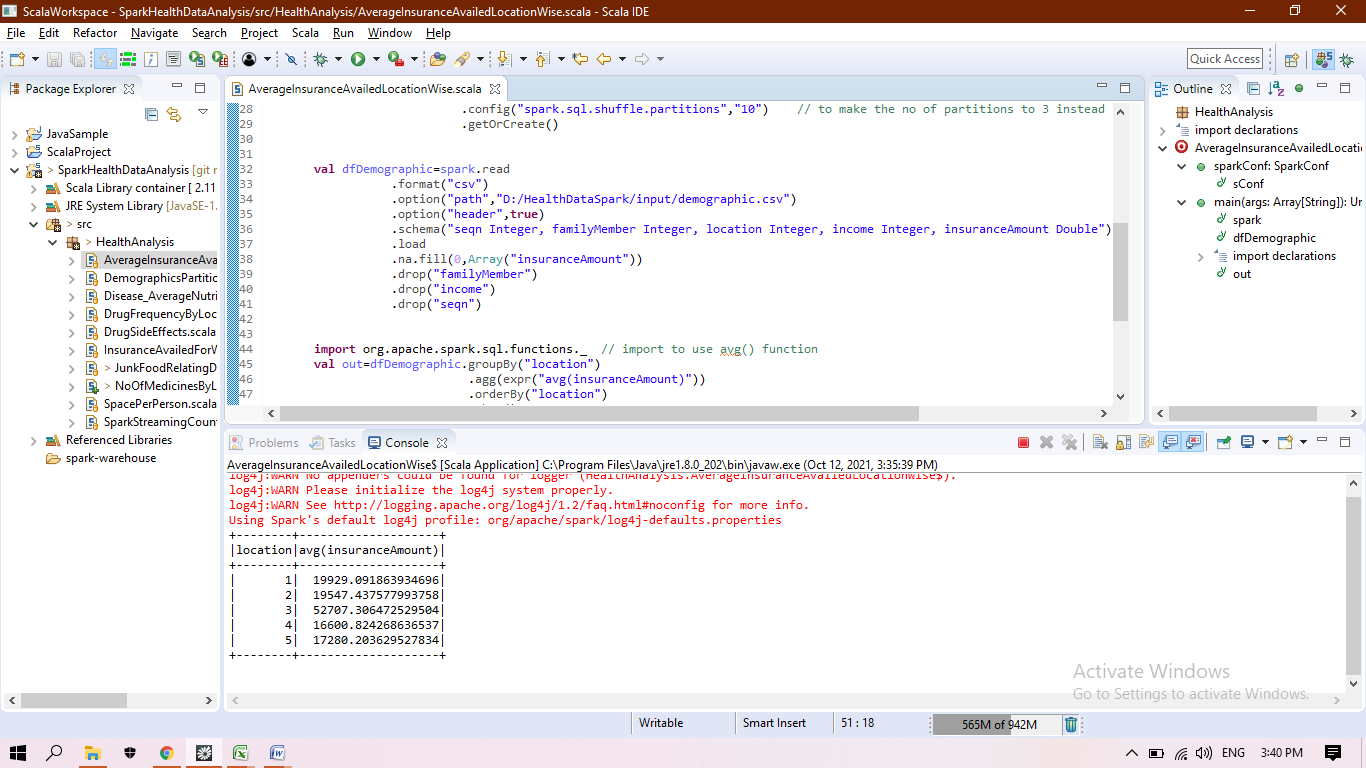
**Input file used:** demographic.csv

**Analysis result:** displaying the average amount of insurance availed by people at different locations. Helpful for insurance companies to plan their investments.

**Purpose (practice or big data analysis):** big data analysis

**Technique used:** spark, aggregation

**Screenshot:**

****

**Function name:** DemographicsPartitionByLocation

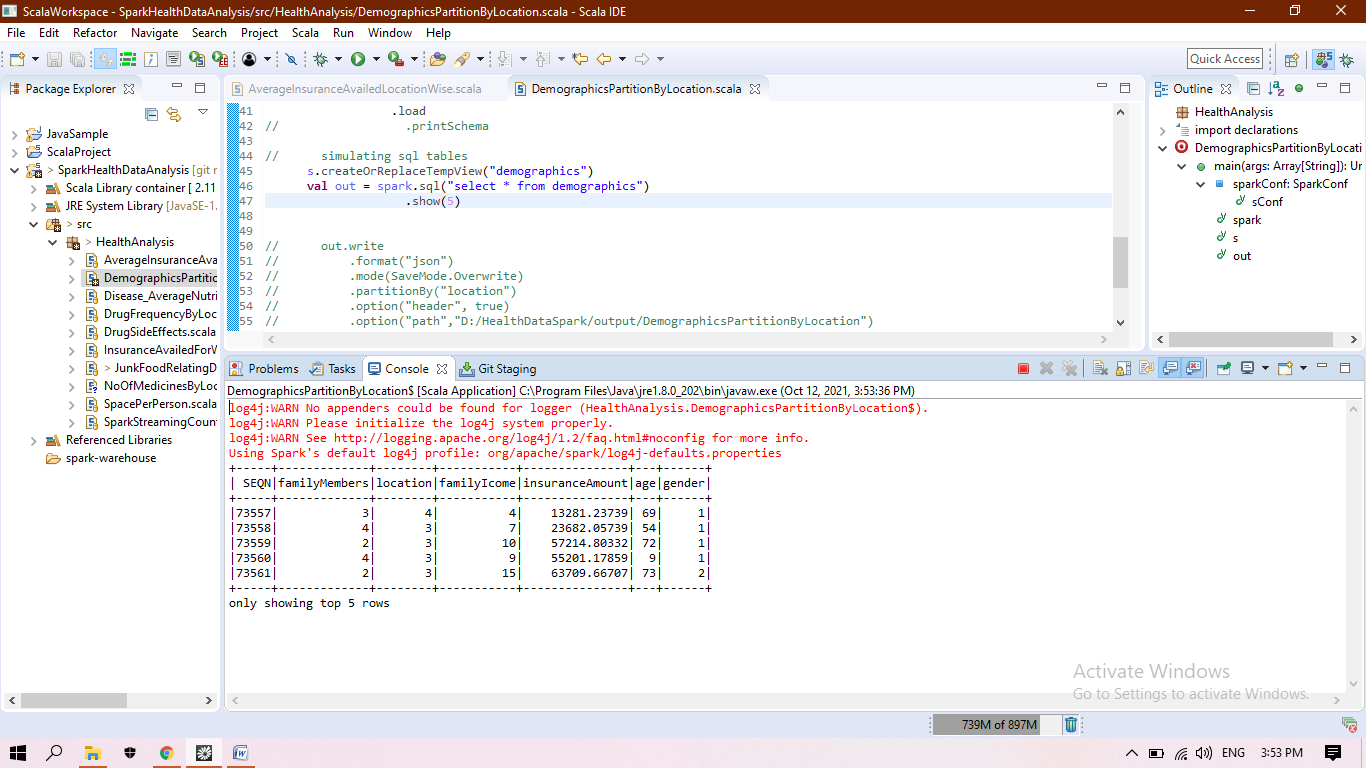
**Input file used:** demographic.csv

**Analysis result:** partitioning the whole data based on the location, so the further queries can be optimized

**Purpose (practice or big data analysis):** practice

**Technique used:** spark, spark sql, partitionBy

**Screenshot:**

****

**Function name:** Disease\_AverageNutrientIntake.

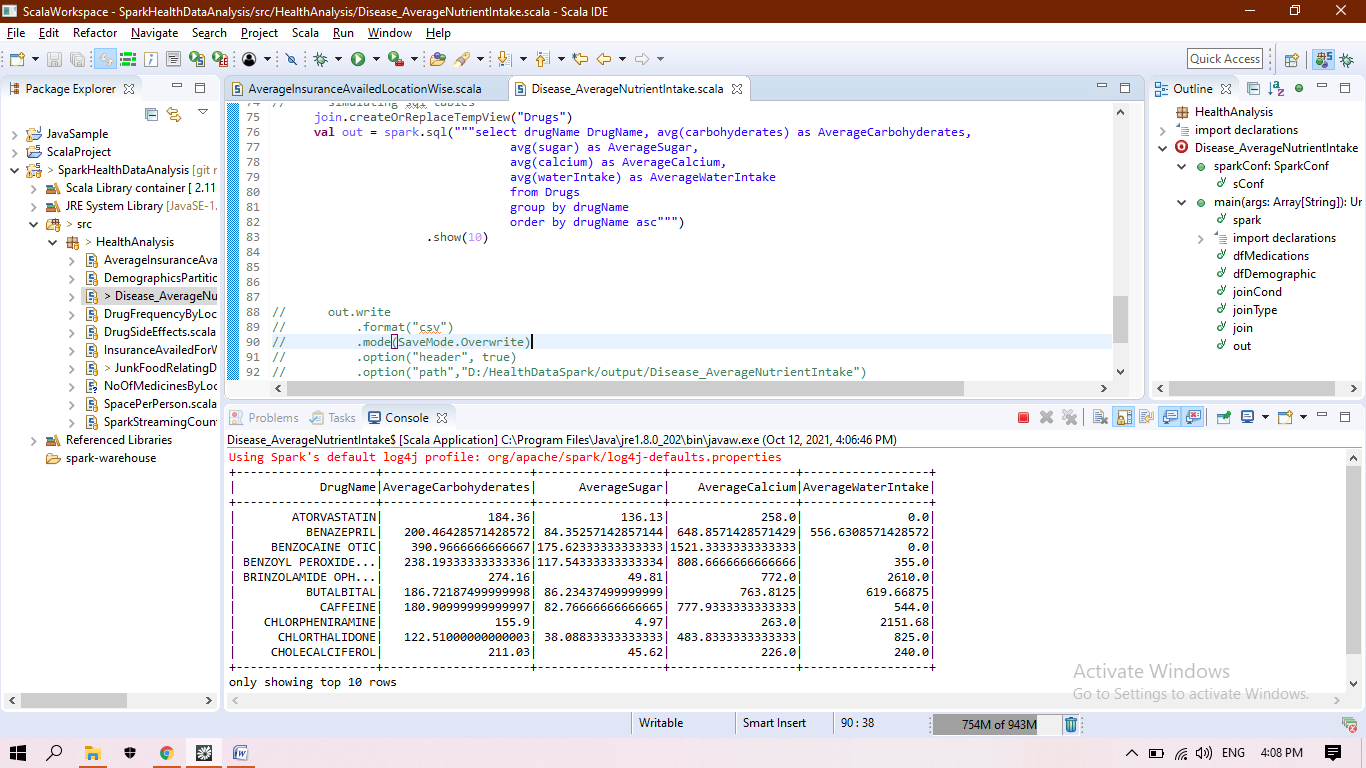
**Input file used:** medications.csv, diet.csv

**Analysis result:** relating the disease with the nutrients intake of people, grouped by disease so that we can find that if a person having nutrients value around the found values then that person may have that disease.

**Purpose (practice or big data analysis):** big data analysis

**Technique used:** spark, join, aggregation

**Screenshot:**

****

**Function name:** DrugFrequencyByLocation

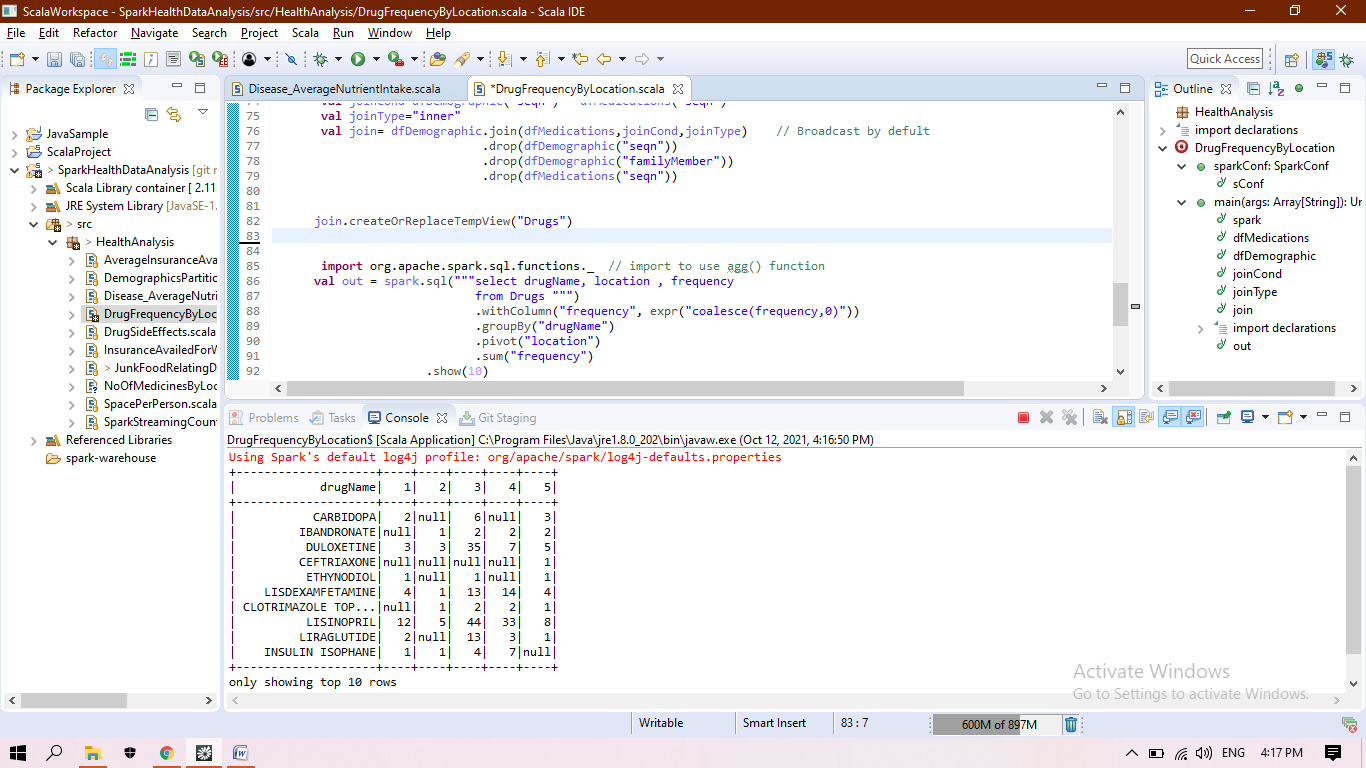
**Input file used:** demographic.csv, medications.csv

**Analysis result:** displaying the frequency of different drugs required at different locations. Useful or MR’s or opening new medical store

**Purpose (practice or big data analysis):** big data analysis

**Technique used:** spark, spark sql, join, pivot, coalesce, withColumn(adding new column)

**Screenshot:**

****

**Function name:** DrugSideEffects

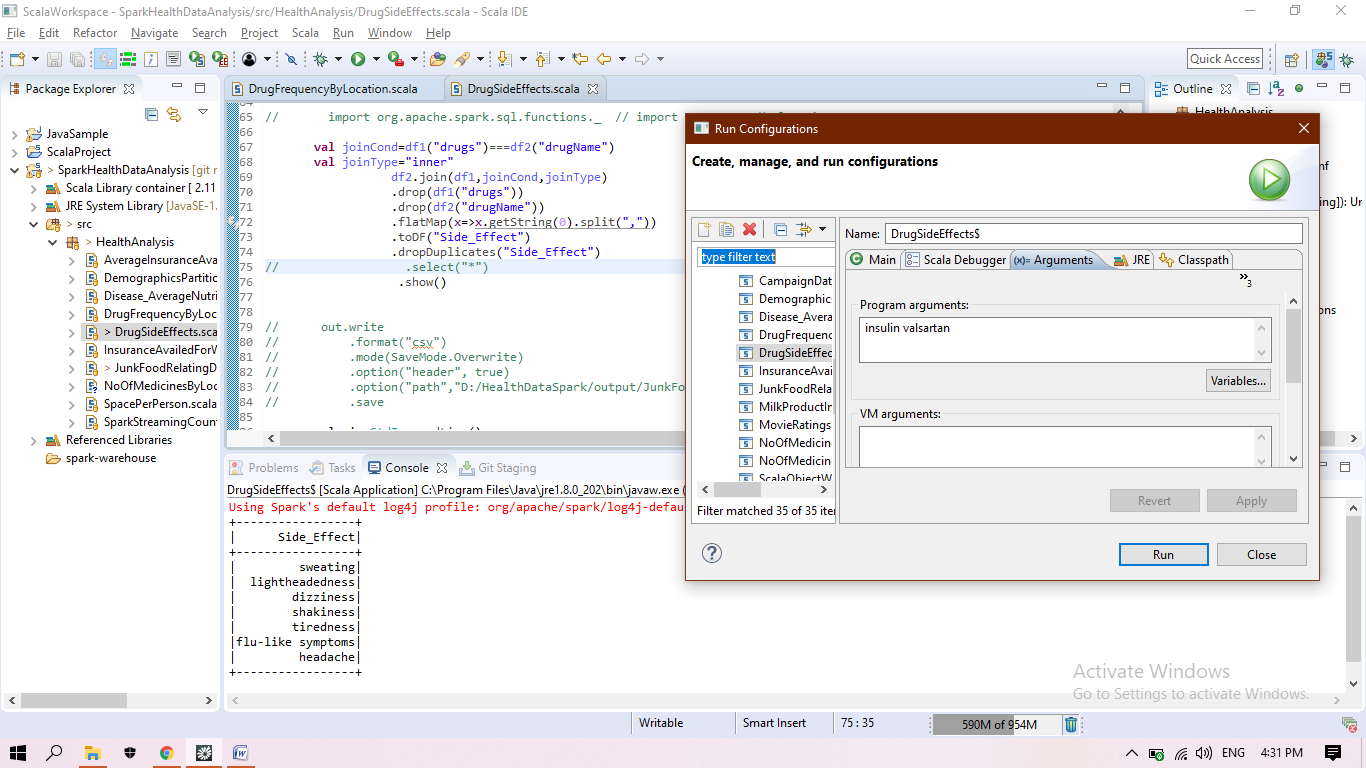
**Input file used:** drugSideEffect.csv

**Analysis result:** displaying the side effects of the drug taken by the person

**Purpose (practice or big data analysis):** practice

**Technique used:** spark, parellalize, using program arguments, join

**Screenshot:**

****

**Function name:** InsuranceAvailedForWindow3

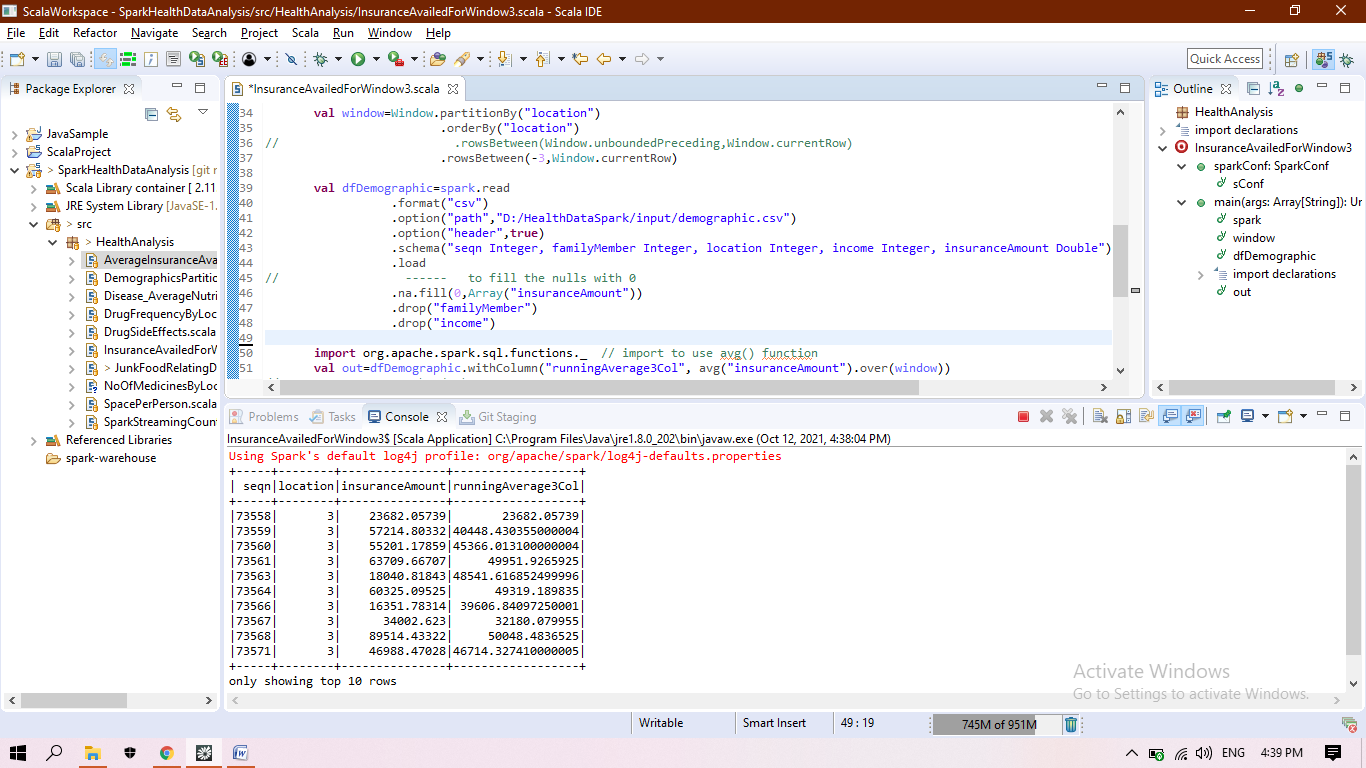
**Input file used:** demographic.csv

**Analysis result:** calculating average of insurance availed within the window size = 3

**Purpose (practice or big data analysis):** practice

**Technique used:** spark, window function, input data filtering

**Screenshot:**

****

**Function name:** JunkFoodRelatingDiseaseCount

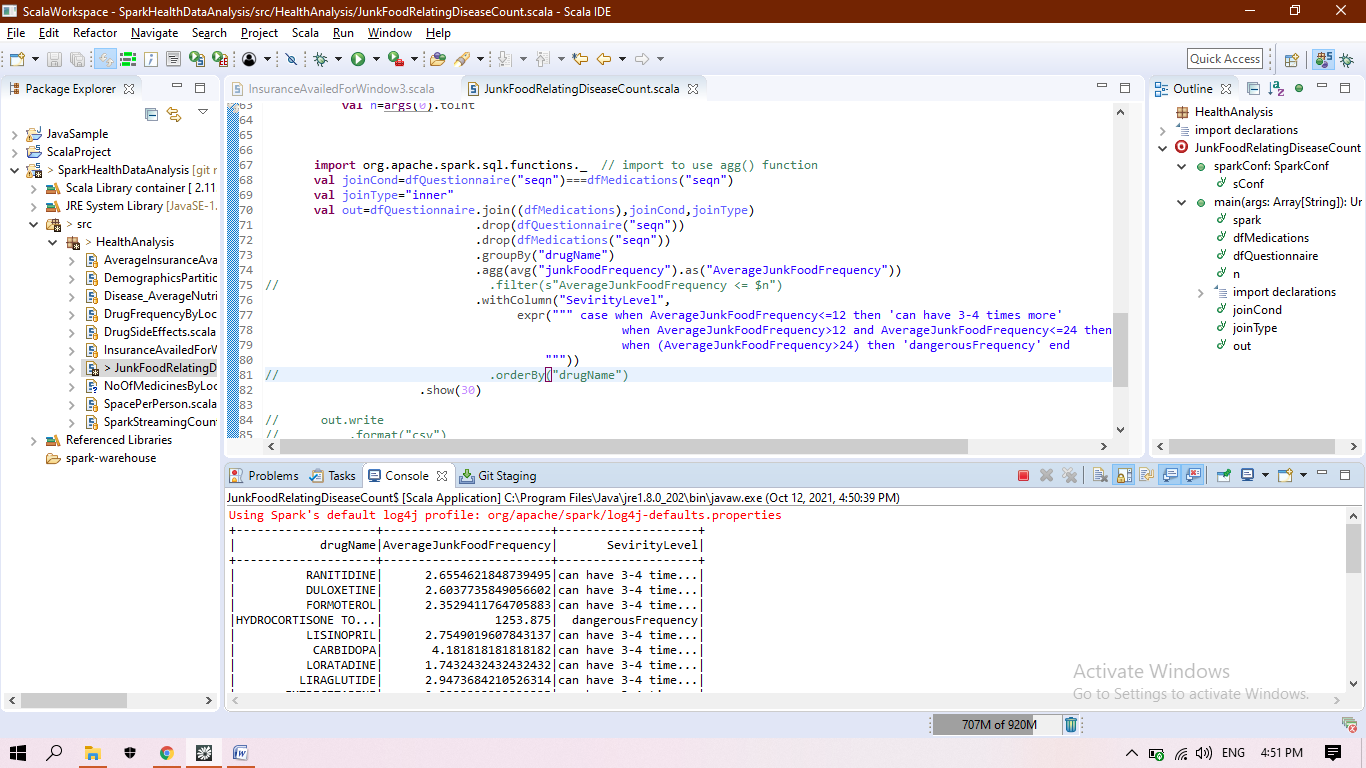
**Input file used:** medications.csv, questionnaire.csv

**Analysis result:** calculating the frequency of eating junk food and the severity of the frequency relating to drug

**Purpose (practice or big data analysis):** big data analysis

**Technique used:** spark, withColumn ( adding new column having severity of the frequency ), user input ( program arguments) , join

**Screenshot:**

****

**Function name:** DiseaseFrequencyInVeganAndNonVegan

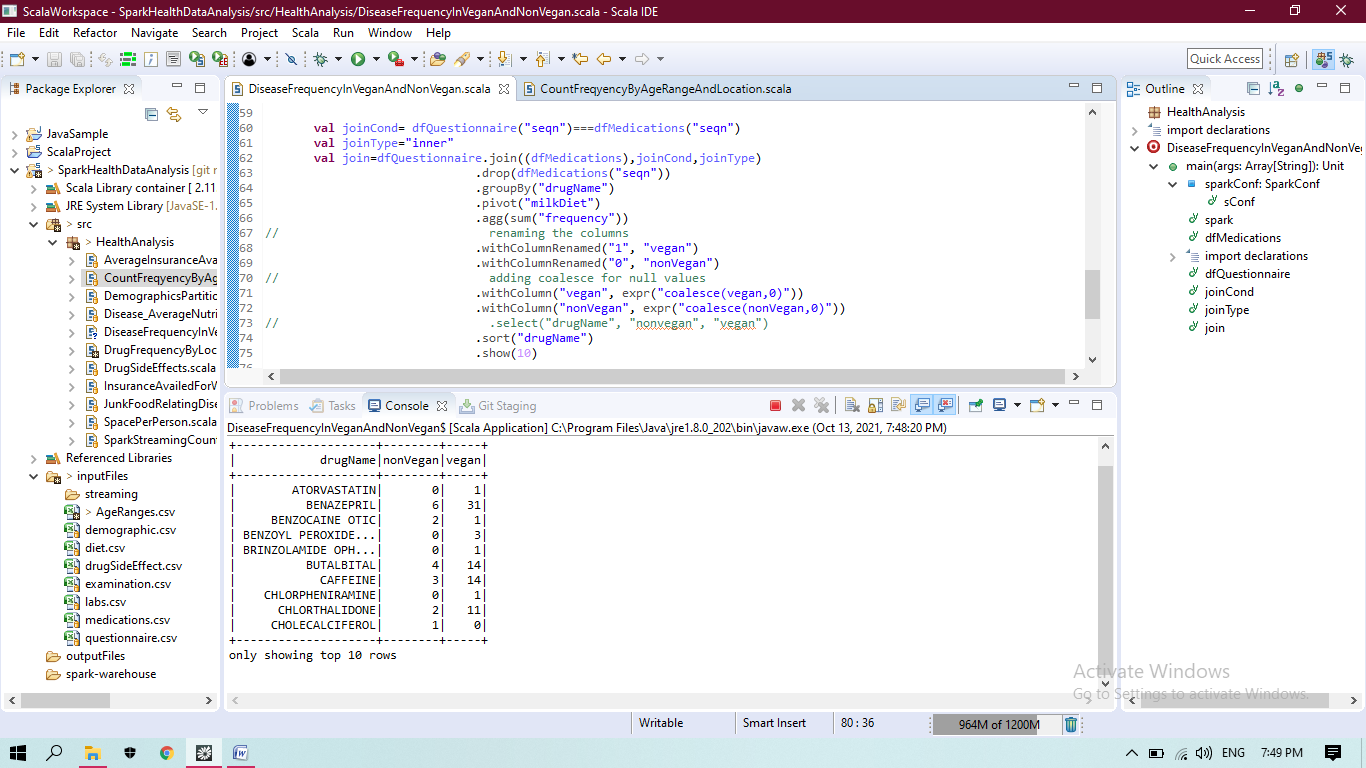
**Input file used:** medications.csv, questionnaire.csv

**Analysis result:** checking the no. of patients taking a particular medicine and relating it to being vegan or not ( taking milk products)

**Purpose (practice or big data analysis):** big data analysis

**Technique used:** spark, withColumn (coalesce to change null values), pivit, join, aggregate

**Screenshot:**

****

**Function name:** CountFreqyencyByAgeRangeAndLocation

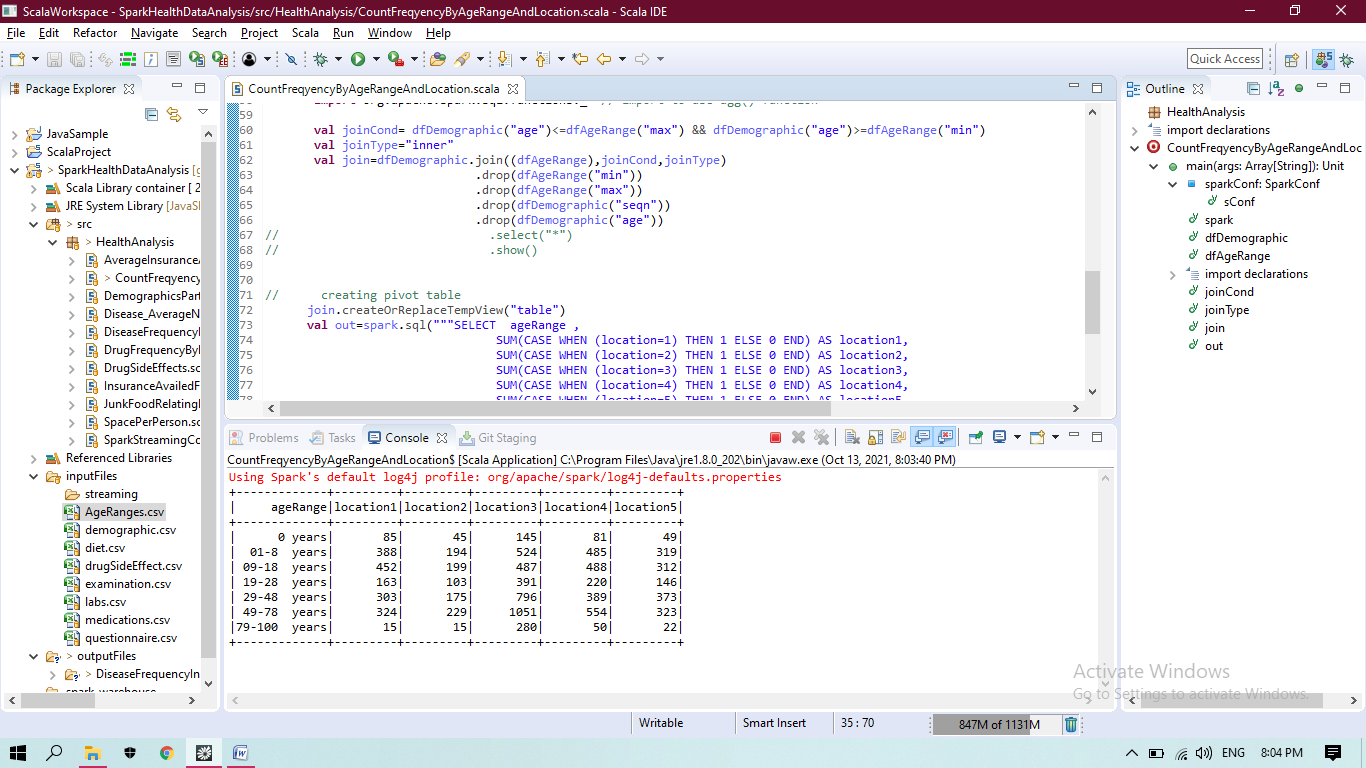
**Input file used:** AgeRanges.csv, demographic.csv

**Analysis result:** created the frequency of patients in particular age range grouping by ageRange and location. Useful to plan for open hospitals catering to particular age related problems,patients.

**Purpose (practice or big data analysis):** big data analysis

**Technique used:** spark, join, spark sql

**Screenshot:**

****

**Function name:** SparkStreamingCountByAgeAndLocation

**Input file used:** demographic.csv

**Analysis result:** counting the frequency of patients by age and location. This analysis can be used by government to plan for opening new hospitals

**Purpose (practice or big data analysis):** big data analysis

**Technique used:** spark, join , withColumn, pivot through sql query, spark streaming