PBDAA GROUP PROJECT
Food Insecurity in NYC
GROUP 9

Group Members

- Krystal Katy Chan (kkc368)
- Nguyen Dang (nd1663)
- Stacy Chao (pyc298)

Data Ingestion

Krystal Katy Chan
Data Source: Retail Food Stores in the State of New York

Description: contains information on location and type of all retail stores licensed by Department of Agriculture and Markets in the State of New York (last updated February 2, 2022)

Link: https://data.ny.gov/Economic-Development/Retail-Food-Stores/9a8c-vfzj

Size of Data: 3.8MB

[Steps to ingest data]

1. Execute commands in FinalCode/etl_code/krystal_katy_chan/ingest_commands.txt
2. NYC retail food stores dataset can be found in /user/kkc368/project/input/retail_food_stores.csv

[Access Method to Data and Output]

a. On HDFS, /user/kkc368/project/input/retail_food_stores.csv
b. FinalCode/etl_code/krystal_katy_chan/01_data_raw/retail_food_stores.csv
c. Screenshot of data ingestion process can be found under FinalCode/screenshots/krystal_katy_chan/data_ingest

Nguyen Dang
Data Source: New York State Median Income

Description: contains information on the median income (in 2020 inflation-adjusted dollars) of households in each zip code in the state of New York.

Link:
https://data.census.gov/table?q=median+income&g=0400000US36$8600000&tid=ACSST5Y2020.S1903

Size of Data: 5.36MB

[Access Method]

a. On HDFS, /user/nd1663/project/datasets/median_income_raw_data.csv
b. FinalCode/etl_code/nd1663/datasets/median_income_raw_data.csv

Stacy Chao
Data Source: DOHMH New York City Restaurant Inspection Results
Description: Contains inspection results for all restaurants in NYC with information on when and the violations occurred in each zipcode, borough and specific address attached.

Size of Data: 102.8 MB

Link: https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/43nn-pn8j

[Access Method]

a. HDFS /user/pyc298/hw9/input
b. FinalCode/etl_code/stacychao/tdel.tsv

ETL

Krystal Katy Chan
[Description of Directories and Files]

- FinalCode/etl_code/krystal_katy_chan: root directory for ETL process
- FinalCode/etl_code/krystal_katy_chan/01_data_raw: contains original dataset
- FinalCode/etl_code/krystal_katy_chan/02_code: contains code and commands needed to clean NYC retail food stores dataset
- FinalCode/etl_code/krystal_katy_chan/03_data_clean: contains cleaned dataset
- FinalCode/screenshots/krystal_katy_chan/etl_code: screenshots of code being executed and output

[Steps to clean code]

1. Execute commands listed in FinalCode/etl_code/krystal_katy_chan/02_code/clean_commands.txt to clean NYC retail food stores dataset

2. Commands will yield cleaned NYC retail food stores data. Output can be found in FinalCode/etl_code/krystal_katy_chan/03_data_clean/retail_food_stores_clean.csv

[Access Method to Data and Output]

1. NYC retail food stores dataset
   a. On HDFS, /user/kkc368/project/input/retail_food_stores.csv
   b. Under FinalCode/etl_code/ krystal_katy_chan/01_data_raw
2. Cleaned NYC retail food stores dataset
   a. On HDFS, /user/kkc368/project/ input/retail_food_stores_clean.csv
   b. Under FinalCode/etl_code/krystal_katy_chan/03_data_clean

Nguyen Dang

I. Median Income Dataset Cleaning

[Description of Directories and Files]

- FinalCode/etl_code/nd1663: root directory for ETL process
- FinalCode/etl_code/nd1663/datasets/median_income_raw_data.csv: original raw median income dataset
- FinalCode/etl_code/nd1663/datasets/median_income_clean.csv: clean dataset with all number of household and income columns
- FinalCode/etl_code/nd1663/datasets/median_income_clean_short.csv: clean dataset with only zip code, number of households, and median income columns
- FinalCode/etl_code/nd1663/1.RunBeforeCleaning.sh: bash script executed before cleaning
- FinalCode/etl_code/nd1663/2.CleanData.scala: Scala script to be executed in Spark shell
- FinalCode/etl_code/nd1663/3.RunAfterCleaning.sh: bash script executed after cleaning
- FinalCode/etl_code/nd1663/headers_long.txt: text file containing all headers of the dataset
- FinalCode/etl_code/nd1663/headers_short.txt: text file containing only the first 3 headers of the dataset
- Screenshots of this step are located in FinalCode/screenshots/nd1663/etl_code

[Steps to clean code]

1. Navigate to FinalCode/etl_code/nd1663/
2. Run the bash script: ./1.RunBeforeCleaning.sh
3. Start Spark shell in the terminal: spark-shell --deploy-mode client
4. Enter the following command: :load 2.CleanData.scala
5. Exit the spark shell: System.exit(0)

6. Run the bash script: ./3.RunAfterCleaning.sh
7. Four CSV files will be generated and put into a new *results* folder:
    a. clean_data.csv - clean data with no headers
    b. clean_data_short.csv - clean data with no headers with only the first 3 columns
    c. clean_data_with_headers.csv - file (i) with headers
    d. clean_data_with_headers_short.csv - file (ii) with headers

[Access Method to Data and Output]

1. NYS median income dataset
    a. On HDFS, /user/nd1663/project/datasets/median_income_raw_data.csv
    b. FinalCode/etl_code/nd1663/datasets/median_income_raw_data.csv
2. Cleaned NYC median income dataset
    a. On HDFS, /user/nd1663/project/datasets/median_income_clean.csv (all columns) and /user/nd1663/project/datasets/median_income_clean_short.csv (3 columns)
    b. In FinalCode/etl_code/nd1663/results after cleaning scripts are executed
    c. In FinalCode/etl_code/nd1663/datasets

II. Secondary Cleaning for Violations Dataset

- Problem: After the violations dataset has been cleaned for the first time by Stacy's ETL code, there were a number of rows in which the violation ID, the inspection date, and the grade columns were identical, while the violation column showed different types of violations. This indicates that these rows referred to the same inspection of the same restaurant on an inspection date, and that these violations were committed by this one restaurant. As such, we needed to merged these rows into one entry, with an additional column violation_count that indicates the number of violations committed by the restaurant.
- Scala script for secondary cleaning

```
val csvFile = sc.textFile("Path to cleaned violations CSV from Stacy")
val headerRow = csvFile.first()
val noHeader = csvFile.filter(row => !row.equals(headerRow))

val pairRDD = noHeader.map(row => (row.split(",")(0) + "," + row.split(",")(1) + "," +
row.split(",")(2) + "," + row.split(",")(3), row.split(",")(4).replace("\"", "")))

val violationGrouped = pairRDD.reduceByKey((x, y) => x + " && " + y)

val violationCleaned = violationGrouped.map(row => row.toString.substring(1,
row.toString.length - 1))

val violationCounted = violationCleaned.map(row => {
        val array = row.split(",")
        if(array.length != 5) {
        row.concat(",0")
        }
        else {
```

```
            val count = (row.split(",")(4)).split("&&").length
            if(count == 0) {
            row.concat(",1")
            }
            else {
            row.concat("," + count)
            }
            }
})

val violationFiltered = violationCounted.filter(row => row.split(",")(5).toInt != 0)

violationFiltered.coalesce(1, true).saveAsTextFile("Output directory")
```

- This clean dataset is then used in all subsequent analysis.

Stacy Chao

[description of directories and files]

- FinalCode/etl_code/stacychao:root directory
    - Tdel.tsv - original data set that clean mapreudcur program takes in
    - Clean.java CleanMapper.java CleanReducer.java: mapreduce java source code
    - .class files and jar files: pulled from peel after successful completion of mapreduce
    - Cleaned.txt: the output of mapreduce program
    - Screenshots: showing compeltiogn of mapreduce

[Steps]

- Go to FinalCode/etl_code/stacychao
    - Use the .java files and the tdel.tsv files
- Run Script:

```
javac -classpath `yarn classpath` -d . Clean.java CleanReducer.java CleanMapper$
jar -cvf Clean.jar *.class
hdfs dfs -mkdir hw9 hdfs dfs -mkdir hw9/input
hdfs dfs -put tdel.tsv hw9/input
hadoop jar Clean.jar Clean hw9/input/tdel.tsv output
```

- Then run hdfs dfs -cat output/part-r-00000

[access method to data and output]

Original Dataset named tdel.tsv

- HDFS: /user/pyc298/hw9/input/tdel.tsv
- FinalCode/etl_code/stacychao/tdel.tsv

Cleaned DataSet

- Run the mapreduce program yourself
- FinalCode/etl_code/stacychao/cleaned.txt

Profiling

Krystal Katy Chan

[Description of Directories and Files]

- FinalCode/profiling_code/krystal_katy_chan: root directory for data profiling
- FinalCode/profiling_code/krystal_katy_chan/01_code: contains code needed to profile original NYC retail food stores dataset and cleaned NYC retail food stores dataset
- FinalCode/screenshots/krystal_katy_chan/profiling_code: screenshots of code being executed and output

[Steps to profile code]

1. Execute commands listed in FinalCode/profiling_code/ krystal_katy_chan/01_code/profile_raw_commands.txt to profile original NYC retail food stores dataset
2. Execute commands listed in FinalCode/profiling_code/krystal_katy_chan/01_code/profile_clean_commands.txt to profile cleaned NYC retail food stores dataset
3. Screenshots of command outputs can be found under FinalCode/screenshots/krystal_katy_chan/profiling_code

[Access Method to Data and Output]

1. NYC retail food stores dataset
   a. On HDFS, /user/kkc368/project/input/retail_food_stores.csv
   b. Under FinalCode/etl_code/krystal_katy_chan/01_data_raw
2. Cleaned NYC retail food stores dataset
   a. On HDFS, /user/kkc368/project/input/retail_food_stores_clean.csv
   b. Under FinalCode/etl_code/03_data_clean
3. Output from profiling NYC retail food stores dataset
   a. On HDFS, /user/kkc368/project/retail_food_stores_count
   b. Under FinalCode/screenshots/krystal_katy_chan/profiling_code
4. Output from profiling cleaned NYC retail food stores dataset
   a. On HDFS, /user/kkc368/project/retail_food_stores_clean_count
   b. Under FinalCode/screenshots/krystal_katy_chan/profiling_code

Nguyen Dang

[Description of Directories and Files]

- FinalCode/profiling_code/nd1663: root directory for data profiling
- FinalCode/profiling_code/nd1663/MedianIncomeProfiling.scala: Scala script for profiling the clean New York City median income dataset
- FinalCode/profiling_code/nd1663/expected_output.txt: The expected output of the Scala script stored in text file form
- Screenshots of this step are located in FinalCode/screenshots/nd1663/profiling_code

[Steps to profile code]

1. Navigate to FinalCode/profiling_code/nd1663/
2. Start spark shell in the terminal: spark-shell --deploy-mode client
3. Enter the following command: :load MedianIncomeProfiling.scala
4. The results of the profiling of the median income dataset will be printed in the terminal

Stacy Chao

[Description of Directories and Files]

- FinalCode/profiling_code/stacychao: root dir for data profiling
  - Screenshots showing results of program
  - .java files for mapreduce program source code
  - .class and .jar files
  - Data.csv: original data used to count in MR

[steps for profiling]

1. Enter peel, run mapreduce program CountRecs.java along with CountRecMapper.java and CountRecsReducer.java with Data.csv (in the same directory)
2. Results of the profiling are shown in the output folder of mapreduce program run command: hdfs dfs -cat output/part-r-00000

[Access Method to Data]

1. Data.csv is in the part1 folder shared already in homework7, along with mapreduce program
2. Enter following script:
   a. javac -classpath `yarn classpath` -d . CountRecs.java CountRecsReducer.java /CountRecsMapper$
   b. jar -cvf CountRecs.jar *.class
   c. //put the data.csv file wherever it needs to be for your system
   d. hadoop jar CountRecs.jar CountRecs data-location/data.csv output
3. Output results are shown in screenshots in root directory mentioned above, or you can use hdfs dfs -cat output/part-r-00000

Data Analytics

Krystal Katy Chan

[Goal of Analytics]

Observe relationship between NYC retail food stores dataset and NYC median household income dataset.

Specifically, answer the following questions:

1. What is the number of retail food stores for the top 10 zipcodes with the highest median income?
2. What is the number of retail food stores for the top 10 zipcodes with the lowest median income?
3. What is the number of each establishment type of retail food stores for the top 10 zipcodes with the highest median income?
4. What is the number of each establishment type of retail food stores for the top 10 zipcodes with the lowest median income?
5. What is the percentage makeup of establishment types of retail food stores for the top 10 zipcodes with the highest median income?
6. What is the percentage makeup of establishment types of retail food stores for the top 10 zipcodes with the lowest median income?

[Description of Directories and Files]

- FinalCode/ana_code/krystal_katy_chan: root directory for data analytics
- FinalCode/ana_code/krystal_katy_chan/01_code: contains code and commands necessary for the analytics
- FinalCode/ana_code/krystal_katy_chan/02_output: contains output from analytics
- FinalCode/screenshots/krystal_katy_chan/ana_code: contains screenshots of code being executed and output
    - FinalCode/screenshots/krystal_katy_chan/01_spark
    - FinalCode/screenshots/krystal_katy_chan/02_impala
    - FinalCode/screenshots/krystal_katy_chan/03_impala_to_csv

[Steps to conduct data analytics]

1. Execute commands from FinalCode/ana_code/krystal_katy_chan/01_code/spark_analytics_commands.txt for steps and descriptions of each query
2. Execute commands from FinalCode/ana_code/krystal_katy_chan/01_code/impala_analytics_commands.txt for steps and descriptions of each query
3. Execute commands from FinalCode/ana_code/krystal_katy_chan/01_code/impala_analytics_to_csv.txt for steps and descriptions of each query

[Access Method to Data and Output]

1. Cleaned NYC retail food stores dataset
    a. On HDFS, /user/kkc368/project/input/retail_food_stores_clean.csv
    b. Under FinalCode/etl_code/03_data_clean
2. Median Income dataset with common zipcodes
    a. On HDFS, /user/kkc368/project/input/median_income_3_cols.csv
    b. On HDFS, /user/nd1663/project/datasets/median_income_clean_short.csv
    c. Under FinalCode/etl_code/nd1663/datasets/median_income_clean_short.csv
3. Output
    a. On HDFS:
        - /user/kkc368/project/ highest_income_num_retail.csv
        - /user/kkc368/project/ lowest_income_num_retail.csv
        - /user/kkc368/project/ highest_income_num_type.csv
        - /user/kkc368/project/ lowest_income_num_type.csv
        - /user/kkc368/project/ highest_income_percen_type.csv
        - /user/kkc368/project/ lowest_income_percen_type.csv
    b. Under FinalCode/ana_code/krystal_katy_chan/02_output:
        - highest_income_num_retail.csv
        - lowest_income_num_retail.csv
        - highest_income_num_type.csv
        - lowest_income_num_type.csv
        - highest_income_percen_type.csv
        - lowest_income_percen_type.csv
4. Screenshots that show execution and output of analytics for every step
    a. Under FinalCode/screenshots/krystal_katy_chan/ana_code/01_spark
    b. Under FinalCode/screenshots/krystal_katy_chan/ana_code/02_impala
    c. Under FinalCode/screenshots/krystal_katy_chan/ana_code/03_impala_to_csv

<u>Nguyen Dang</u>

1. Analysis of Median Income Dataset
    a. Navigate to ana_code/nd1663/
    b. Start spark shell in the terminal: spark-shell --deploy-mode client
    c. Enter the following command: :load MedianIncomeAnalysis.scala
    d. The results of the analysis of the median income dataset will be printed in the terminal
    e. Screenshots of this analysis is available in FinalCode/screenshots/nd1663/median_income, and the expected output is stored as a text file in FinalCode/ana_code/nd1663/income_analysis_output.txt

2. Analysis of all datasets in Impala
    a. Start the impala shell in the terminal: impala-shell
    b. Connect to the cluster: connect hc08.nyu.cluster;
    c. Select database: use ND1663;
    d. Create tables from clean datasets using the following queries:

```
--Median Income
CREATE EXTERNAL TABLE income (zipcode INT, household_count INT, median_income INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE
LOCATION '/user/nd1663/project/datasets/impala_data/income'
TBLPROPERTIES ("skip.header.line.count"="1");

--Retail Food Stores
CREATE EXTERNAL TABLE restaurants (county STRING, license INT, type VARCHAR, name STRING, dba STRING, city STRING, zipcode INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE
LOCATION '/user/nd1663/project/datasets/impala_data/retail'
TBLPROPERTIES ("skip.header.line.count"="1");
--Restaurant Inspections
CREATE EXTERNAL TABLE violations (camisid INT, zipcode INT, inspection_date STRING, grade STRING, violation STRING, violation_count INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE
LOCATION '/user/nd1663/project/datasets/impala_data/violation'
TBLPROPERTIES ("skip.header.line.count"="1");
```

    e. Number of retail food stores in each zip code along with their number of households, median income, and the ratio between the number of households and the number of stores

```
--Top 10 zip codes by median income
WITH restaurant_count AS (SELECT zipcode, count(*) AS count FROM restaurants GROUP BY zipcode)
SELECT income.zipcode, income.household_count, income.median_income, restaurant_count.count, CAST(income.household_count / restaurant_count.count AS INT) AS households_per_store FROM income
INNER JOIN restaurant_count ON income.zipcode = restaurant_count.zipcode
ORDER BY income.median_income DESC LIMIT 10;

--Bottom 10 zip codes by median income
WITH restaurant_count AS (SELECT zipcode, count(*) AS count FROM restaurants GROUP BY zipcode)
SELECT income.zipcode, income.household_count, income.median_income, restaurant_count.count, CAST(income.household_count / restaurant_count.count AS INT) AS households_per_store FROM income
INNER JOIN restaurant_count ON income.zipcode = restaurant_count.zipcode
```

```
WHERE income.median_income > -1 ORDER BY income.median_income ASC LIMIT 10;
```

f. Percentage makeup of establishment types

```
--Total number of establishments (13393)
SELECT count(*) AS total FROM restaurants;

--Breakdown of establishment types, their counts and percentage of total establishments
WITH temp AS (SELECT type, count(*) AS count FROM restaurants GROUP BY type)
SELECT type, temp.count, CAST(temp.count / 13393 AS DECIMAL(10,5)) AS percentage
FROM temp ORDER BY percentage DESC;

--Types of establishment found in the top 10 median income zip codes
SELECT zipcode, type, count(*) AS count FROM restaurants
WHERE zipcode IN (10007, 10282, 10004, 10006, 10005, 10280, 10065, 11215, 11201,
10010)
GROUP BY zipcode, type ORDER BY zipcode, type;

--The above but the top 10 is aggregated as a whole
WITH temp AS (SELECT zipcode, type, count(*) AS count FROM restaurants
WHERE zipcode IN (10007, 10282, 10004, 10006, 10005, 10280, 10065, 11215, 11201,
10010)
GROUP BY zipcode, type ORDER BY zipcode, type)
SELECT type, sum(temp.count) AS count FROM temp
GROUP BY type ORDER BY count DESC;

--Types of establishment found in the bottom 10 median income zip codes
SELECT zipcode, type, count(*) AS count FROM restaurants
WHERE zipcode IN (10454, 10035, 10474, 10460, 11212, 10455, 11239, 10452, 10456,
10453)
GROUP BY zipcode, type ORDER BY zipcode, type;

--The above but the bottom 10 is aggregated as a whole
WITH temp AS (SELECT zipcode, type, count(*) AS count FROM restaurants
WHERE zipcode IN (10454, 10035, 10474, 10460, 11212, 10455, 11239, 10452, 10456,
10453)
GROUP BY zipcode, type ORDER BY zipcode, type)
SELECT type, sum(temp.count) AS count FROM temp
GROUP BY type ORDER BY count DESC;
```

g. Screenshots for the output of these Impala queries can be found in FinalCode/screenshots/nd1663/ana_code/impala

Stacy Chao

Analysis of the makeup of violation grades in top 10 median income zipcodes vs bottom 10 median income zipcodes.

[Description of files and directories]

- FinalCode/ana_code/stacychao: root directory
    - Contains:
    - Screenshots of each of the commands run in impala
    - finalcodedrop.rtf : contains the commands run in the screenshots
    - Violations.csv: data used to create table in impala
    - Analysis_output.txt: analytics from team member Ngyugen that provides information on which zipcodes will be used.

[Steps]

1. Execute commands from finalcodedrop.rtf in impala shell

[Access to data and output]

1. Violations.csv
- FinalCode/ana_code/stacychao/violations.csv
- Hdfs: /user/pyc298/finalcodedrop/violations.csv
2. Output
    a. Refer to the screen shots which will showcase the analytics and results of each of the commands run.