Nizan Howard, Aviva Munshi

Data 516

11/20/23

**Project Milestone**

**Issues faced**

We first began using Postgre SQL, as stated in our project proposal. However, the platform software was incompatible with our MacBooks, and in the interest of time, we started to create clusters and databases in Amazon Redshift on AWS Learning Academy. However, we ran into trouble when we tried to load the data from an S3 onto our CSV file. Aviva’s account threw errors at her when she tried to create a cluster, and Nizan's access to create a Bucket in S3 was denied. Ultimately, both of our accounts were deactivated, and we finally landed on Snowflake, which offered a 30-day free trial account and started querying our data there.

**Data Preprocessing**

When loading the CSV extracted from the World Data Bank to Jupyter Notebook for Data Preprocessing, the CSV would not load due to an encoding error. We later realized this was because of the presence of commas in the rows of our data. There was a reading error since Pandas reads commas in CSV files as delimiters. In order to avoid this and get on with our data processing, we converted the CSV to an Excel file and read that in our Python notebook. Some more changes we made after loading the Excel are:

* We stored the debt for each unique country and indicator group from 1960 to 2022 in a " Debt " column.
* Cleaned the 'Indicator Name' and 'Country Name' columns by replacing commas with semicolons.
* Exported the modified DataFrame df to a CSV file to load into Snowflake.

Further, we made an extra dataset with an additional column for the Continent of each Country. This helped us during Stratified Sampling. Also, the resultant datasets contain 392882 rows.

**Data Files**

The first Snowflake CSV file contains five columns of interest:

* Country Code
* Country Name
* Indicator Code
* Indicator Name
* Debt

The second Snowflake CSV file contains five columns of interest:

* Country Code
* Country Name
* Indicator Code
* Indicator Name
* Debt
* Continent

**Sampling Approaches**

Our process is to run each different query using a different sampling technique as listed above. We test the runtime efficiency on a set of group-by-aggregate queries and collect the runtime data for visualization comparison. These are the sampling techniques we’ve used for each of these queries:

* Simple Random Sampling: This involves randomly selecting a subset of rows from the table to execute a query.
* Stratified Sampling: This involves dividing the data into strata (in our case, continents), and then performing random sampling within each stratum.
* Systematic Sampling: This involves selecting every 393th row from the table. The idea is to take a smaller sample size systematically.

**OUTPUT: Result\_data\_output.csv file**

|  |  | **Runtime for Query 1 (ms)** | | **Runtime for Query 2 (ms)** | | **Runtime for Query 3 (ms)** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Without Sampling** | **Computer 1** | 4500 | 4931 | 954 | 1654 | 150 | 1000 |
| **Computer 2** | 431 | 1400 | 1700 |
| **Simple Random Sampling** | **Computer 1** | 1300 | 1408 | 669 | 823.5 | 119 | 181.5 |
| **Computer 2** | 216 | 309 | 125 |
| **Stratified Sampling** | **Computer 1** | 59 | 89.5 | 370 | 531 | 246 | 273 |
| **Computer 2** | 120 | 161 | 300 |
| **Systematic Sampling** | **Computer 1** | 521 | 722.5 | 1000 | 1309 | 321 | 399.5 |
| **Computer 2** | 403 | 618 | 478 |

**Query 1: Country with the highest debt**

/\* Country with the highest debt \*/

SELECT

Country, SUM(DebtAmount) AS Debt

FROM WORLDDEBT

GROUP BY Country

ORDER BY Debt DESC

LIMIT 1;

/\* Country with the highest debt with simple random sampling \*/

SELECT

Country, SUM(DebtAmount) AS Debt

FROM (

SELECT

Country, DebtAmount

FROM

WORLDDEBT

ORDER BY RANDOM()

LIMIT 39290

) AS SampledData

GROUP BY

Country

ORDER BY

Debt DESC

LIMIT 1;

/\* Country with the highest debt with Stratified Sampling \*/

WITH RankedCountries AS (

SELECT

CountryName,

Continent,

SUM(DebtAmount) AS TotalDebt,

ROW\_NUMBER() OVER(PARTITION BY Continent ORDER BY SUM(DebtAmount) DESC) AS RowNum

FROM WORLDDEBT

GROUP BY CountryName, Continent

)

SELECT

CountryName,

TotalDebt AS Debt,

Continent

FROM RankedCountries

WHERE RowNum = 1;

/\* Country with the highest debt with Systematic Sampling \*/

WITH RankedDebts AS (

SELECT

Country,

SUM(DebtAmount) AS Debt,

ROW\_NUMBER() OVER (ORDER BY SUM(DebtAmount) DESC) AS RowNum

FROM WORLDDEBT

GROUP BY Country

)

SELECT

Country,

Debt

FROM RankedDebts

WHERE RowNum % 393 = 0;

**Query 2: Average debt across all indicators**

/\* Average amount of debt across indicators \*/

SELECT

IndicatorCode,

IndicatorName,

AVG(DebtAmount) AS AverageDebt

FROM WORLDDEBT

GROUP BY IndicatorCode, IndicatorName

ORDER BY AverageDebt DESC

LIMIT 10;

/\* Average amount of debt across indicators with Simple Random Sampling \*/

SELECT

IndicatorCode,

IndicatorName,

AVG(DebtAmount) AS AverageDebt

FROM WORLDDEBT

GROUP BY IndicatorCode, IndicatorName

ORDER BY RANDOM()

LIMIT 10;

/\* Average amount of debt across indicators with Stratified Sampling \*/

WITH StratifiedSample AS (

SELECT

CountryName,

DebtAmount,

Continent,

AVG(DebtAmount) OVER(PARTITION BY Continent) AS AverageDebt,

ROW\_NUMBER() OVER(PARTITION BY Continent ORDER BY RANDOM()) AS RowNum

FROM CONTINENTDEBT

)

SELECT

CountryName,

DebtAmount,

Continent,

AverageDebt

FROM StratifiedSample

WHERE RowNum = 1;

/\* Average amount of debt across indicators with Systematic Sampling \*/

WITH SampledData AS (

SELECT

\*,

ROW\_NUMBER() OVER (ORDER BY IndicatorCode, IndicatorName) AS RowNum

FROM WORLDDEBT

)

SELECT

IndicatorCode,

IndicatorName,

AVG(DebtAmount) AS AverageDebt

FROM SampledData

WHERE MOD(RowNum - 1, 393) = 0

GROUP BY IndicatorCode, IndicatorName

ORDER BY IndicatorCode, IndicatorName;

**Query 3: Most common debt indicator**

/\* The most common debt indicator \*/

SELECT

IndicatorName,

COUNT(\*) AS IndicatorCount

FROM

WORLDDEBT

GROUP BY

IndicatorName

ORDER BY

IndicatorCount DESC

LIMIT 1;

/\* The most common debt indicator with Simple Random Sampling \*/

SELECT

IndicatorName,

COUNT(\*) AS IndicatorCount

FROM

WORLDDEBT

GROUP BY

IndicatorName

ORDER BY

RANDOM()

LIMIT 1;

/\* The most common debt indicator with Stratified Sampling\*/

WITH IndicatorCounts AS (

SELECT

IndicatorName,

Continent,

COUNT(\*) AS IndicatorCount

FROM CONTINENTDEBT

GROUP BY IndicatorName, Continent

)

SELECT

IndicatorName,

Continent,

IndicatorCount

FROM (

SELECT

IndicatorName,

Continent,

IndicatorCount,

ROW\_NUMBER() OVER(PARTITION BY Continent ORDER BY IndicatorCount DESC) AS RowNum

FROM IndicatorCounts

)

WHERE RowNum = 1;

/\* The most common debt indicator with Systematic Sampling\*/

WITH SampledData AS (

SELECT

IndicatorName,

COUNT(\*) AS IndicatorCount,

ROW\_NUMBER() OVER (ORDER BY IndicatorName) AS RowNum

FROM WORLDDEBT

GROUP BY IndicatorName

)

SELECT

IndicatorName,

IndicatorCount

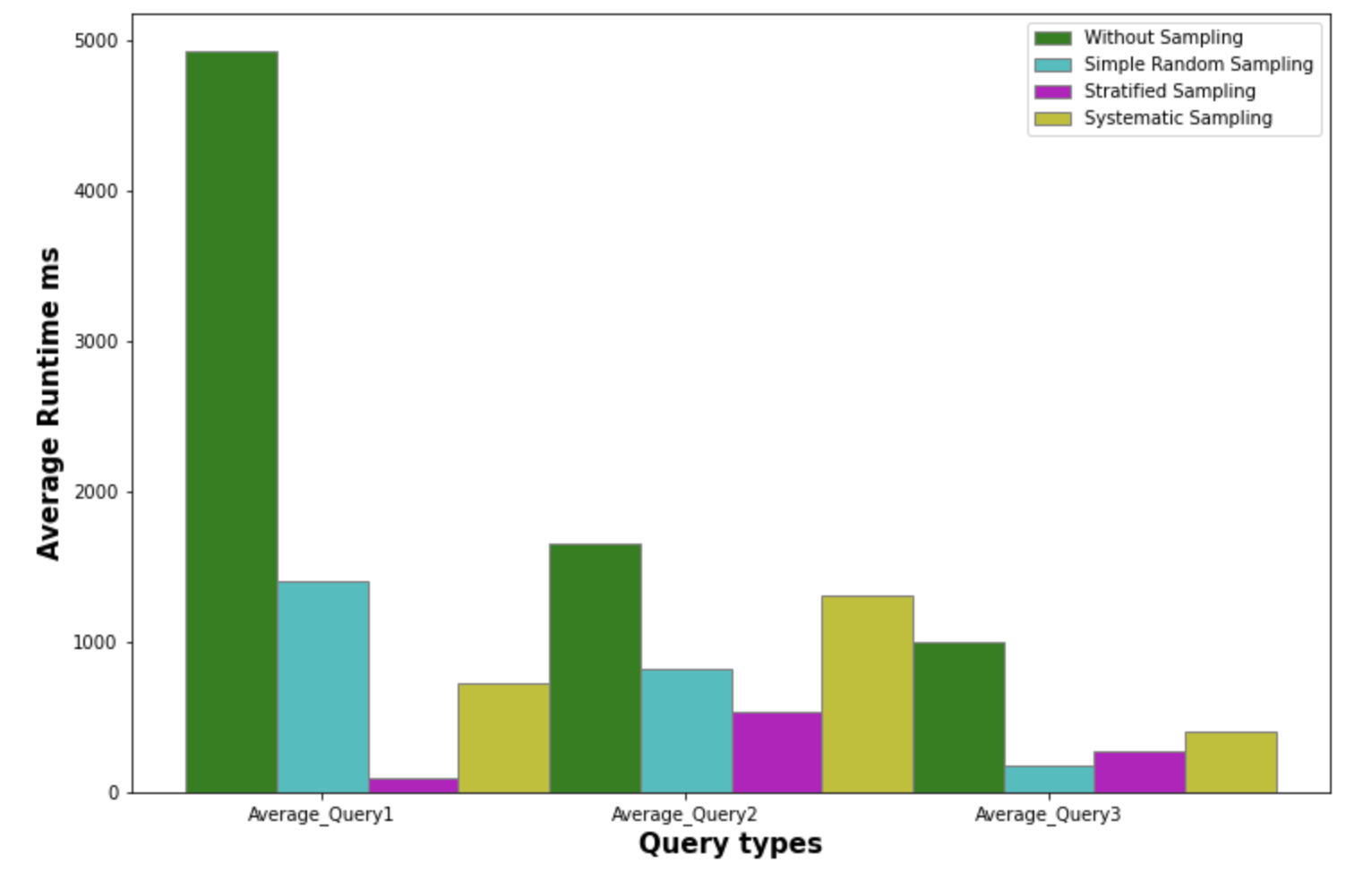
FROM SampledData

WHERE MOD(RowNum - 1, 393) = 0;

**Visualizations:**

**Visual 1:**

**Sampling Type by Average Runtime in Snowflake SQL**

****

**Further investigation**

As a group, we have begun the group-by-aggregations on the different sampling techniques. We must repeat the same approach on joins to further progress on this project and explore further insights on the effect of sampling types on query efficiency.

**Joins**

when has the average data been collected

which currency has the most debt

self join: (Find the Employee ID and Name of the managers with a minimum of 7 employees directly reporting to them) - link https://theanalyticsgeek.com/sql-interview-question-on-self-join/

Find the currency type and country code of countries that utilize the Euro?

**Joins**

**Which currency unit has the highest debt?**

**Currency with the highest debt without sampling**

SELECT CD.CURRENCYUNIT, SUM(WD.DEBTAMOUNT) AS TotalDebt

FROM CURRENCYDATA CD

JOIN WORLDDEBT WD ON CD.COUNTRYCODE = WD.COUNTRYCODE

GROUP BY CD.CURRENCYUNIT

ORDER BY TotalDebt DESC

LIMIT 1;

Query time on Computer 1: 16ms

Query time on Computer 2: 23 ms

**Currency with the highest debt with Simple Random Sampling:**

SELECT CD.CURRENCYUNIT, SUM(WD.DEBTAMOUNT) AS TotalDebt

FROM CURRENCYDATA CD

JOIN (SELECT \* FROM WORLDDEBT TABLESAMPLE SYSTEM (10)) WD

ON CD.COUNTRYCODE = WD.COUNTRYCODE

GROUP BY CD.CURRENCYUNIT

ORDER BY TotalDebt DESC

LIMIT 1;

Query time on Computer 1: 673ms

Query time on Computer 2: 593 ms

**Currency with the highest debt with Stratified Sampling:**

WITH HighestDebtCountryCodes AS (

SELECT CD.COUNTRYCODE

FROM CURRENCYDATA CD

JOIN WORLDDEBT WD ON CD.COUNTRYCODE = WD.COUNTRYCODE

GROUP BY CD.COUNTRYCODE

ORDER BY SUM(WD.DEBTAMOUNT) DESC

LIMIT 1

),

StratifiedSample AS (

SELECT CD.COUNTRYCODE, CD.COUNTRYNAME, CD.INDICATORCODE, CD.INDICATORNAME, CD.DEBTAMOUNT, CD.CONTINENT,

ROW\_NUMBER() OVER (PARTITION BY CD.CONTINENT ORDER BY RANDOM()) AS row\_num

FROM CONTINENTDEBT CD

JOIN HighestDebtCountryCodes HDC ON CD.COUNTRYCODE = HDC.COUNTRYCODE

)

SELECT \*

FROM StratifiedSample

WHERE row\_num <= 10;

Query time on Computer 1: 467ms

Query time on Computer 2: 401 ms

**Currency with the highest debt with Systematic Sampling:**

WITH systematic\_sample AS (

SELECT

CD.CURRENCYUNIT,

SUM(WD.DEBTAMOUNT) AS TotalDebt,

NTILE(10) OVER (ORDER BY CD.CURRENCYUNIT) AS tile

FROM

CURRENCYDATA CD

JOIN

WORLDDEBT WD ON CD.COUNTRYCODE = WD.COUNTRYCODE

GROUP BY

CD.CURRENCYUNIT

)

SELECT \*

FROM systematic\_sample

WHERE tile = 1;

Query time on Computer 1: 31 ms

Query time on Computer 2: 38 ms

## **When has the average data been collected?**

**When has the most data been collected without Sampling**

WITH JoinedData AS (

SELECT WD.COUNTRYCODE, CD.LATESTTRADEDATA

FROM WORLDDEBT WD

JOIN CENSUSDATA CD ON WD.COUNTRYCODE = CD.COUNTRYCODE

)

SELECT LATESTTRADEDATA, COUNT(\*) AS Frequency

FROM JoinedData

GROUP BY LATESTTRADEDATA

ORDER BY Frequency DESC

LIMIT 1;

Query time on Computer 1: 19ms

Query time on Computer 2: 32 ms

**When has the most data been collected with Simple Random Sampling**

WITH SampledWorldDebt AS (

SELECT \*

FROM WORLDDEBT

SAMPLE (10) -- Adjust the number as needed for your sample size

),

SampledCensusData AS (

SELECT \*

FROM CENSUSDATA

SAMPLE (10)

),

JoinedData AS (

SELECT WD.COUNTRYCODE, CD.LATESTTRADEDATA

FROM SampledWorldDebt WD

JOIN SampledCensusData CD ON WD.COUNTRYCODE = CD.COUNTRYCODE

)

SELECT LATESTTRADEDATA, COUNT(\*) AS Frequency

FROM JoinedData

GROUP BY LATESTTRADEDATA

ORDER BY Frequency DESC

LIMIT 1;

Query time on Computer 1: 249ms

Query time on Computer 2: 236 ms

**When has the most data been collected with Stratified Sampling**

WITH JoinedData AS (

SELECT WD.COUNTRYCODE, CD.LATESTTRADEDATA, CD.COUNTRYCODE AS ContinentCode

FROM WORLDDEBT WD

JOIN CENSUSDATA CD ON WD.COUNTRYCODE = CD.COUNTRYCODE

),

StratifiedSample AS (

SELECT JD.\*, CD.CURRENCYUNIT,

ROW\_NUMBER() OVER(PARTITION BY JD.ContinentCode ORDER BY RANDOM()) AS RowNum

FROM JoinedData JD

JOIN CURRENCYDATA CD ON JD.ContinentCode = CD.COUNTRYCODE

)

SELECT CURRENCYUNIT, LATESTTRADEDATA, COUNT(\*) AS Frequency

FROM StratifiedSample

WHERE RowNum <= 10

GROUP BY CURRENCYUNIT, LATESTTRADEDATA

QUALIFY ROW\_NUMBER() OVER(PARTITION BY CURRENCYUNIT ORDER BY COUNT(\*) DESC) = 1;

Query time on Computer 1: 389ms

Query time on Computer 2: 428 ms

**When has the most data been collected with Systematic Sampling**

WITH RankedData AS (

SELECT \*,

ROW\_NUMBER() OVER (PARTITION BY WD.COUNTRYCODE ORDER BY WD.COUNTRYCODE) AS RowNum

FROM WORLDDEBT WD

JOIN CENSUSDATA CD ON WD.COUNTRYCODE = CD.COUNTRYCODE

)

SELECT LATESTTRADEDATA, COUNT(\*) AS Frequency

FROM RankedData

WHERE MOD(RowNum, 10) = 0

GROUP BY LATESTTRADEDATA

ORDER BY Frequency DESC

LIMIT 1;

Query time on Computer 1: 38ms

Query time on Computer 2: 24 ms

**Find the Countries that use the Euro.**

**Find the Countries that use the Euro without sampling**

WITH EuroCountries AS (

SELECT CD.COUNTRYCODE

FROM CURRENCYDATA CD

JOIN CURRENCYDATA CD2 ON CD.COUNTRYCODE = CD2.COUNTRYCODE

WHERE CD.CURRENCYUNIT = 'Euro'

)

SELECT DISTINCT EC.COUNTRYCODE

FROM EuroCountries EC;

Query time on Computer 1:

Query time on Computer 2: 30ms

**Find the Countries that use the Euro with Simple Random Sampling**

WITH EuroSubset AS (

SELECT COUNTRYCODE

FROM CURRENCYDATA

WHERE CURRENCYUNIT = 'Euro'

)

SELECT DISTINCT EC.COUNTRYCODE

FROM (

SELECT COUNTRYCODE

FROM EuroSubset

SAMPLE (10)

) EC

JOIN CURRENCYDATA EC2 ON EC.COUNTRYCODE = EC2.COUNTRYCODE;

Query time on Computer 1:

Query time on Computer 2: 49 ms

**Find the Countries that use the Euro with Stratified Sampling**

-- -- Identify countries that use the Euro by continent

CREATE TABLE EuroCountriesByContinent AS

SELECT CN.COUNTRYCODE, CN.CONTINENT

FROM CURRENCYDATA CD

JOIN CONTINENTDEBT CN ON CN.COUNTRYCODE = CN.COUNTRYCODE

WHERE CD.CURRENCYUNIT = 'Euro';

-- Loop through each continent and perform limited sampling (up to 10 countries) for Euro-currency countries

CREATE TABLE StratifiedSample AS

SELECT DISTINCT EC.COUNTRYCODE, EC.CONTINENT

FROM EuroCountriesByContinent EC

JOIN (

SELECT EC2.CONTINENT, EC2.COUNTRYCODE,

ROW\_NUMBER() OVER (PARTITION BY EC2.CONTINENT ORDER BY RANDOM()) AS RowNum

FROM EuroCountriesByContinent EC2

) AS GroupedEuroCountries

ON EC.CONTINENT = GroupedEuroCountries.CONTINENT

WHERE GroupedEuroCountries.RowNum <= 10;

-- Display or use the sampled countries in StratifiedSample table

SELECT \* FROM StratifiedSample;

Query time on Computer 1: 11606ms

Query time on Computer 2: 12867

**Find the Countries that use the Euro with Systematic Sampling**

WITH IndexedEuroCountries AS (

SELECT COUNTRYCODE, ROW\_NUMBER() OVER (ORDER BY COUNTRYCODE) AS RowNum

FROM CURRENCYDATA

WHERE CURRENCYUNIT = 'Euro'

)

SELECT DISTINCT EC.COUNTRYCODE

FROM IndexedEuroCountries EC

JOIN IndexedEuroCountries EC2 ON EC.RowNum = EC2.RowNum \* 10

Query time on Computer 1: 71

Query time on Computer 2: 65

Join Runtime Table

|  |  | **Runtime for Query 1 (ms)** | | **Runtim for Query 2 (ms)** | | **Runtim for Query 3 (ms)** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Without Sampling** | **Computer 1** | 16 | 19.5 | 19 | 25.2 | 32 | 31 |
| **Computer 2** | 23 | 32 | 30 |
| **Simple Random Sampling** | **Computer 1** | 678 | 635.5 | 249 | 242.5 | 270 | 260.5 |
| **Computer 2** | 593 | 236 | 251 |
| **Stratified Sampling** | **Computer 1** | 467 | 434 | 389 | 408.5 | 11606 | 12236.5 |
| **Computer 2** | 401 | 428 | 12867 |
| **Systematic Sampling** | **Computer 1** | 31 | 34.5 | 38 | 31 | 71 | 68 |
| **Computer 2** | 38 | 24 | 65 |