Slide 2

Group 1 chose the Data Engineering track and decided to use the Bureau of Labor Statistics as our data source. In the initial discovery phase, we discussed the impacts of unemployment on the four largest cities in NC for a time period. We decided on 2015-2024.

Initial challenges encountered were parsing the Series ID’s. We struggled to parse the data through looping through all Series ID’s as a group, but were unsuccessful. We then pivoted to parsing through each Series ID individually to glean unemployment data from 2015-2024.

Discuss dependencies and how we arrived at using them.

Slide 3

We discovered the Series ID’s represented unique unemployment values/rates for each city in North Carolina.

We decided to capture 9 years of data (2015-2024).

The “for loop” generated a JSON for each unique Series ID which included monthly unemployment values per city.

Slide 4

Checking for errors by printing the response for JSON requests.

Established the file names for unemployment rate and value in each of the four cities.

Created variables to store the JSON data in each

Slide 5

Encapsulates the print statements for all Series ID’s.

Slide 6

Created an empty list to store individual Pandas Dataframes.

Load the JSON data from each file and combine into one DataFrame.

Flatten the data structure into a list of dictionaries and create a DataFrame from this list

Combine all DataFrames into one single Pandas DataFrame

New Library Usage: Create a Dask DataFrame from the combined concatenated DataFrame

Slide 7

Data integrity check of DataFrame through .head function

Saving the Dask DataFrame as CSV

Slide 8

Extract data.

Define RegEx Pattern to match decimal numbers

Search for decimals in value

Load data into Dask DataFrame

Extract decimals, create new column and remove decimals from “value” column

Slide 9

Data integrity check of DataFrame through .head function

Clean DataFrame through Drop Columns and Rename Columns

Check data integrity with .head function

Slide 10

Extract and Transform Charlotte Data

Output cleaned data to CSV

New DataFrame with cleaned data

Data check through .tail function

Slide 11

Extract and Transform Raleigh Data

Output cleaned data to CSV

New DataFrame with cleaned data

Data check through .tail function

Slide 12

Extract and Transform Greensboro Data

Output cleaned data to CSV

New DataFrame with cleaned data

Data check through .tail function

Slide 13

Extract and Transform Winston-Salem Data

Output cleaned data to CSV

New DataFrame with cleaned data

Data check through .tail function

Slide 14

Load Raleigh SQL Table

Utilized cleaned DataFrame for columns:

Year (varchar)

Month (varchar)

Value (varchar represents unemployment value)

Rate (varchar) represents unemployment rate

Slide 15

Load Greensboro SQL Table

Utilized cleaned DataFrame for columns:

Year (varchar)

Month (varchar)

Value (varchar represents unemployment value)

Rate (varchar) represents unemployment rate

Slide 16

Load Winston-Salem SQL Table

Utilized cleaned DataFrame for columns:

Year (varchar)

Month (varchar)

Value (varchar represents unemployment value)

Rate (varchar) represents unemployment rate

Slide 17

Load Charlotte SQL Table

Utilized cleaned DataFrame for columns:

Year (varchar)

Month (varchar)

Value (varchar represents unemployment value)

Rate (varchar) represents unemployment rate

Slide 18

Load Pandas DataFrame and convert to SQLite for each city

Define Raleigh ID’s as two unique Series ID’s

Raleigh Transform Data with Drop Columns, compute, FillNA, GroupBy, Aggregate

Data integrity check with .head function

Output to CSV format

Repeat for 3 remaining cities: Greensboro, Winston-Salem, Charlotte