

GROUP 7

BIG DATA ANALYTICS USING HADOOP, HIVE & SPARK ON MULTI-YEAR IPEDS DATASET

Team Members:
Sai Teja KMVP - MXK240054
Sai Hemalatha Ramidi - SXR240040



PROBLEM STATEMENT

IPEDS publishes large multi-year datasets covering enrollment, completions, institutional characteristics, and more. Manual analysis of these datasets is impractical due to volume, file size, and schema variability across years.

OBJECTIVE:

To use Hadoop, Hive, and Spark analytics on 1GB+ multi-year IPEDS datasets (2020–2023) to answer three key business questions related to trends in enrollment and online learning.

PROJECT REQUIREMENT & COMPLIANCE

This project satisfies all mandatory components:

Hadoop/Hive Setup, Spark for distributed analysis, 1GB+ data from IPEDS, 3 Business Questions, multiple datasets used: EF, EFFY/EFIA, C, HD

The screenshot shows a Mac desktop environment with a terminal window and the Docker Desktop application running in the background.

Terminal Window (Left):

```
...ive --zsh ...ts --zsh ~--zsh ...rk --zsh ...ct --zsh ~--zsh ~--zsh
saitejakmvp@MacBookAir ~ % unzip /Users/saitejakmvp/Downloads/ipeds_export_all_years.zip -d /Users/saitejakmvp/Downloads/ipeds_
Archive: /Users/saitejakmvp/Downloads/ipeds_export_all_years.zip
  creating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/S2023_IS.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/OM2020.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/SFAV2223.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/OM2021.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/SFAV2020.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/DRVEF2023.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/FE2023A_DIST.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/GR2023_L2.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/IC2020Mission.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/OM2023.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/OM2022.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/SFA2223_P1.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2020_B.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2020_C.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/SFA2020_P2.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2022_A.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2022_C.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/SFA2223_P2.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2020_A.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/SAL2023_NIS.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/GR2021_L2.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2022_B.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/SFA2020_P1.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/IC2023_AY.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EF2023D.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/DRV2023.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2020DEP.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/DRVHR2023.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EF2022A.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/HD2024.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/_MACOSX/ipeds_export_all_years/_HD2024.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EF2022B.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/GR2020_PELL_SSL.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/HD2023.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EF2020.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EFFY2021_DIST.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EFFY2020_DIST.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EF2023B.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2021_A.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2023_C.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2023_B.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/DRVADM2023.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/F2223_F1a.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EF2021.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/HD2022.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/HD2020.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EF2023.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2022DEP.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/GR2021_PELL_SSL.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/IC2021_AY.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EF2023A.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2021_B.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EFFY2023_HS.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2023_A.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/SAL2023_IS.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/C2021_C.csv
inflating: /Users/saitejakmvp/Downloads/ipeds_export_all_years/ipeds_export_all_years/EF2022D.csv
```

Docker Desktop Interface (Right):

The Docker Desktop interface shows the following details:

- Local Images:** 8.75 GB / 19.8 GB in use, 6 images last refreshed 0 seconds ago.
- Images Table:** A table listing 6 Docker images with columns: Name, Tag, Image ID, Created, Size, Actions.

	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	postgres	14	ca25035f7e6f	20 days ago	1.26 GB	
<input type="checkbox"/>	apache/spark	3.5.0	0ed5154e6b32	2 years ago	3.06 GB	
<input type="checkbox"/>	bde2020/hadoop-namenode	2.0.0-hadoop3.2.1-java8	51ad9293ec52	6 years ago	2.05 GB	
<input type="checkbox"/>	bde2020/hadoop-datanode	2.0.0-hadoop3.2.1-java8	ddf6e9ad55af	6 years ago	2.05 GB	
<input type="checkbox"/>	apache/hive	4.0.1	5194161ef50b	1 year ago	5.52 GB	
<input type="checkbox"/>	jupyter/pyspark-notebook	latest	58377aaa152b	2 years ago	6.98 GB	

```
docker compose pull
docker compose up -d
docker ps

[+] Running 6/6
✓ Container datanode      Removed
✓ Container spark          Removed
✓ Container hive-server     Removed
✓ Container hive-metastore  Removed
✓ Container namenode       Removed
✓ Container metastore-db   Removed
Deleted Networks:
nces_bigdata_project_default

Total reclaimed space: 0B

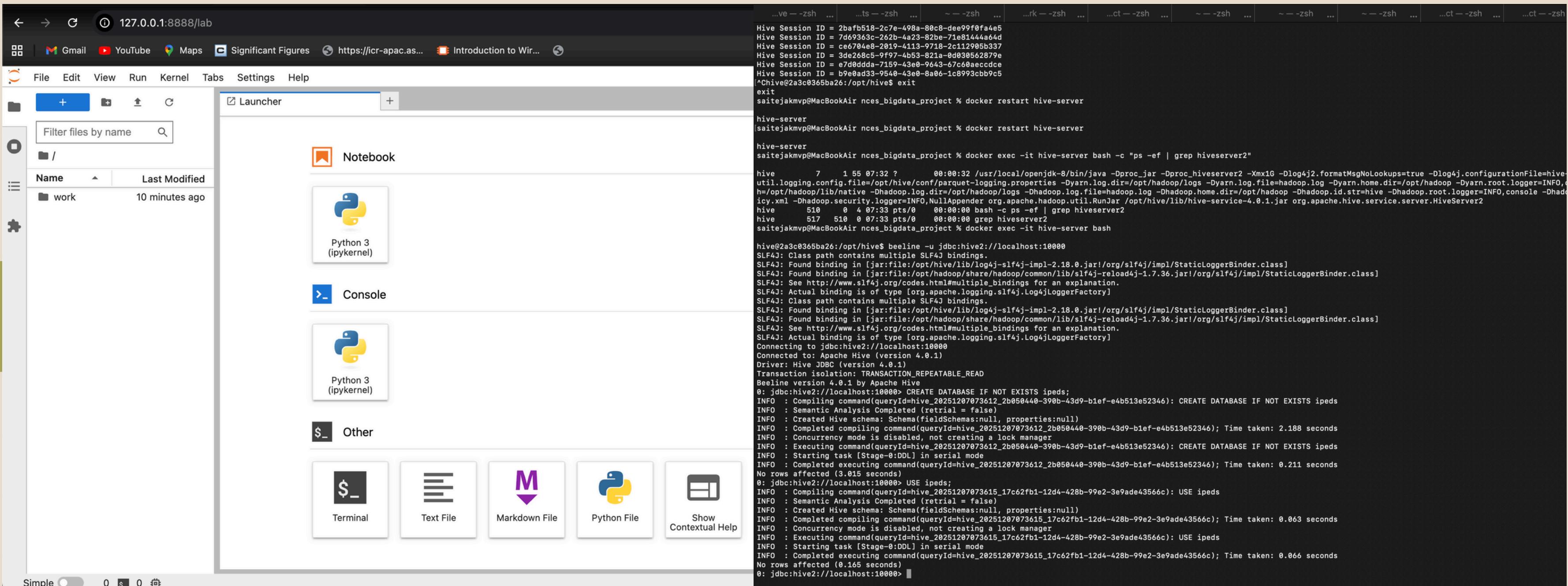
[+] Pulling 43/43
✓ datanode Skipped - Image is already being pulled by namenode
✓ hive-server Skipped - Image is already being pulled by hive-metastore
✓ hive-metastore Pulled
✓ 4f4fb700ef54 Pull complete
✓ efd22504ff0a Pull complete
✓ f83c96cf43e9 Pull complete
✓ ec49bc66c13d Pull complete
✓ b8efbd449f77 Pull complete
✓ 04acb178ffe4 Pull complete
✓ a2f2ff93da482 Pull complete
✓ 09650b219558 Pull complete
✓ d2421c7a4bbf Pull complete
✓ acb32ec93a5d Pull complete
✓ e9ed7de0daff Pull complete
✓ 1efc276f4fff9 Pull complete
✓ 1a2de4cc9431 Pull complete
✓ namenode Pulled
✓ spark Pulled
✓ 616eb43372c1 Pull complete
✓ d1818ec198c0 Pull complete
✓ e43f1f40dafc Pull complete
✓ 9c387219ad28 Pull complete
✓ a5b06dad3503 Pull complete
✓ f67b63cf36d1 Pull complete
✓ 7007490126ef Pull complete
✓ d35079eec123 Pull complete
✓ 6f32e041863d Pull complete
✓ a148cf41a5df Pull complete
✓ metastore-db Pulled
✓ 0d67bc96accc Pull complete
✓ 117e98dc7655 Pull complete
✓ ca269a9e240c Pull complete
✓ 80a8c0ac4601 Pull complete
✓ 83dbba3aab3 Pull complete
.../hive -- -ZSN ... | .../tpis -- -ZSN ... | ~ -- -ZSN ... | .../park --
```

```
✓ 09650b219558 Pull complete
✓ d2421c7a4bbf Pull complete
✓ acb32ec93a5d Pull complete
✓ e9ed7de0daff Pull complete
✓ 1efc276f4fff9 Pull complete
✓ 1a2de4cc9431 Pull complete
✓ namenode Pulled
✓ spark Pulled
✓ 616eb43372c1 Pull complete
✓ d1818ec198c0 Pull complete
✓ e43f1f40dafc Pull complete
✓ 9c387219ad28 Pull complete
✓ a5b06dad3503 Pull complete
✓ f67b63cf36d1 Pull complete
✓ 7007490126ef Pull complete
✓ d35079eec123 Pull complete
✓ 6f32e041863d Pull complete
✓ a148cf41a5df Pull complete
✓ metastore-db Pulled
✓ 0d67bc96accc Pull complete
✓ 117e98dc7655 Pull complete
✓ ca269a9e240c Pull complete
✓ 80a8c0ac4601 Pull complete
✓ 83dbba3aab3 Pull complete
✓ 08927b2b9336 Pull complete
✓ da2a031a8efd Pull complete
✓ d0c34fc6d7b7 Pull complete
```

```
✓ 3a909eaca6304 Pull complete
✓ a9cbee5ac34d Pull complete
✓ 7c576d448336 Pull complete
✓ 0e4bc2bd6656 Pull complete
✓ 773a7fb7a56e Pull complete
✓ 5920f036e231 Pull complete
[+] Running 4/4
✓ hive-metastore Pulled
✓ hive-server Pulled
✓ spark Pulled
✓ metastore-db Pulled
[+] Running 7/7
✓ Network nces_bigdata_project_hadoop Created
✓ Container metastore-db Started
✓ Container namenode Started
✓ Container datanode Started
✓ Container hive-metastore Started
✓ Container hive-server Started
✓ Container spark Started
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAM
ES
5f303c12caf0 apache/spark:3.5.0 "/opt/entrypoint.sh" 1 second ago Up Less than a second 4040/tcp spa
rk
79cf33366adb apache/hive:4.0.0 "sh -c /entrypoint.sh" 1 second ago Up Less than a second 0.0.0.0:10000->10000/tcp, [::]:10000->10000/tcp, 0.0.0.0:10002->10002/tcp, [::]:10002->10002/tcp hiv
e-server
2abdd9450320 apache/hive:4.0.0 "sh -c /entrypoint.sh" 1 second ago Up Less than a second 0.0.0.0:9083->9083/tcp, [::]:9083->9083/tcp hiv
e-metastore
2893f2472acf apache/hadoop:3.3.6 "/usr/local/bin/dumb..." 1 second ago Up Less than a second 0.0.0.0:9864->9864/tcp, [::]:9864->9864/tcp anode
anode
a5a7027e3d6f postgres:14 "docker-entrypoint.s..." 2 seconds ago Up Less than a second 0.0.0.0:5433->5432/tcp, [::]:5433->5432/tcp astore-db
55ba7ee8f168 apache/hadoop:3.3.6 "/usr/local/bin/dumb..." 2 seconds ago Up Less than a second 0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp, 0.0.0.0:9870->9870/tcp, [::]:9870->9870/enode
saitejakmvp@MacBookAir nces_bigdata_project %
```

TECHNOLOGY ARCHITECTURE

Our Architecture includes: HDFS cluster for distributed storage, Hive Meta store + PostgreSQL, HiveServer2 for querying, Spark Jupyter Environment for distributed computation, Spark Warehouse for table storage



The screenshot shows a Jupyter Notebook interface running on a local host (127.0.0.1:8888/lab). The interface includes a top navigation bar with links to Gmail, YouTube, Maps, and other tabs. Below the bar is a toolbar with File, Edit, View, Run, Kernel, Tabs, Settings, and Help options. A sidebar on the left contains a 'Launcher' section with a search bar and a 'Notebook' section listing a single 'work' notebook created 10 minutes ago. The main area is divided into two panes: a 'Console' pane on the left and a 'Terminal' pane on the right.

Console (Left Pane):

```
...ve --zsh ... ts --zsh ... ~ -zsh ... rk --zsh ... ct --zsh ... --zsh ... ~ -zsh ... ~ -zsh ... ct --zsh ... ct --zsh ...
Hive Session ID = 2bafb518-2c7e-498a-8cc8-dee99f0fa465
Hive Session ID = 7d69363c-262b-a423-82be-71e81444a64d
Hive Session ID = ce6704e8-2019-4113-9718-2c112905b337
Hive Session ID = 3de268c5-0f97-4b53-821a-0d30562879e
Hive Session ID = e7d0ddda-7159-43e0-9643-67c60aecdcce
Hive Session ID = b9e0ad33-9540-43e0-8a06-1c8993cbb9c5
^Chive@2a3c0365ba26:/opt/hive$ exit
exit
saitejakmvp@MacBookAir nces_bigdata_project % docker restart hive-server
hive-server
saitejakmvp@MacBookAir nces_bigdata_project % docker restart hive-server
hive-server
saitejakmvp@MacBookAir nces_bigdata_project % docker exec -it hive-server bash -c "ps -ef | grep hiveserver2"
hive      7  1 55 07:32 ?  00:00:32 /usr/local/openjdk-8/bin/java -Dproc_jar -Dproc_hiveserver2 -Xmx1G -Dlog4j2.formatMsgNoLookups=true -Dlog4j.configurationFile=hive-util.logging.config.file=/opt/hive/conf/parquet-logging.properties -Dyarn.log.dir=/opt/hadoop/logs -Dyarn.log.file=hadoop.log -Dyarn.home.dir=/opt/hadoop -Dyarn.root.logger=INFO,console -Dhadoop.log.dir=/opt/hadoop/lib/native -Dhadoop.log.name=hadoop.log -Dhadoop.log.level=INFO,NullAppender org.apache.hadoop.util.RunJar /opt/hive/lib/hive-service-4.0.1.jar org.apache.hive.service.HiveServer2
hive      510   0  4 07:33 pts/0  00:00:00 bash -c ps -ef | grep hiveserver2
hive      517  510   0 07:33 pts/0  00:00:00 grep hiveserver2
saitejakmvp@MacBookAir nces_bigdata_project % docker exec -it hive-server bash
hive@2a3c0365ba26:/opt/hive$ beeline - jdbc:hive2://localhost:10000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000
Connected to: Apache Hive (version 4.0.1)
Driver: Hive JDBC (version 4.0.1)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 4.0.1 by Apache Hive
0: jdbc:hive2://localhost:10000> CREATE DATABASE IF NOT EXISTS ipeds;
INFO : Compiling command(queryId=hive_20251207073612_2b050440-390b-43d9-b1ef-e4b513e52346): CREATE DATABASE IF NOT EXISTS ipeds
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20251207073612_2b050440-390b-43d9-b1ef-e4b513e52346); Time taken: 2.188 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20251207073612_2b050440-390b-43d9-b1ef-e4b513e52346): CREATE DATABASE IF NOT EXISTS ipeds
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20251207073612_2b050440-390b-43d9-b1ef-e4b513e52346); Time taken: 0.211 seconds
No rows affected (3.015 seconds)
0: jdbc:hive2://localhost:10000> USE ipeds;
INFO : Compiling command(queryId=hive_20251207073615_17c62fb1-12d4-428b-99e2-3e9ade43566c): USE ipeds
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20251207073615_17c62fb1-12d4-428b-99e2-3e9ade43566c); Time taken: 0.063 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20251207073615_17c62fb1-12d4-428b-99e2-3e9ade43566c): USE ipeds
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20251207073615_17c62fb1-12d4-428b-99e2-3e9ade43566c); Time taken: 0.066 seconds
No rows affected (0.165 seconds)
0: jdbc:hive2://localhost:10000>
```

Terminal (Right Pane):

```
...ve --zsh ... ts --zsh ... ~ -zsh ... rk --zsh ... ct --zsh ... --zsh ... ~ -zsh ... ~ -zsh ... ct --zsh ... ct --zsh ...
Hive Session ID = 2bafb518-2c7e-498a-8cc8-dee99f0fa465
Hive Session ID = 7d69363c-262b-a423-82be-71e81444a64d
Hive Session ID = ce6704e8-2019-4113-9718-2c112905b337
Hive Session ID = 3de268c5-0f97-4b53-821a-0d30562879e
Hive Session ID = e7d0ddda-7159-43e0-9643-67c60aecdcce
Hive Session ID = b9e0ad33-9540-43e0-8a06-1c8993cbb9c5
^Chive@2a3c0365ba26:/opt/hive$ exit
exit
saitejakmvp@MacBookAir nces_bigdata_project % docker restart hive-server
hive-server
saitejakmvp@MacBookAir nces_bigdata_project % docker restart hive-server
hive-server
saitejakmvp@MacBookAir nces_bigdata_project % docker exec -it hive-server bash -c "ps -ef | grep hiveserver2"
hive      7  1 55 07:32 ?  00:00:32 /usr/local/openjdk-8/bin/java -Dproc_jar -Dproc_hiveserver2 -Xmx1G -Dlog4j2.formatMsgNoLookups=true -Dlog4j.configurationFile=hive-util.logging.config.file=/opt/hive/conf/parquet-logging.properties -Dyarn.log.dir=/opt/hadoop/logs -Dyarn.log.file=hadoop.log -Dyarn.home.dir=/opt/hadoop -Dyarn.root.logger=INFO,console -Dhadoop.log.dir=/opt/hadoop/lib/native -Dhadoop.log.name=hadoop.log -Dhadoop.log.level=INFO,NullAppender org.apache.hadoop.util.RunJar /opt/hive/lib/hive-service-4.0.1.jar org.apache.hive.service.HiveServer2
hive      510   0  4 07:33 pts/0  00:00:00 bash -c ps -ef | grep hiveserver2
hive      517  510   0 07:33 pts/0  00:00:00 grep hiveserver2
saitejakmvp@MacBookAir nces_bigdata_project % docker exec -it hive-server bash
hive@2a3c0365ba26:/opt/hive$ beeline - jdbc:hive2://localhost:10000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.18.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://localhost:10000
Connected to: Apache Hive (version 4.0.1)
Driver: Hive JDBC (version 4.0.1)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 4.0.1 by Apache Hive
0: jdbc:hive2://localhost:10000> CREATE DATABASE IF NOT EXISTS ipeds;
INFO : Compiling command(queryId=hive_20251207073612_2b050440-390b-43d9-b1ef-e4b513e52346): CREATE DATABASE IF NOT EXISTS ipeds
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20251207073612_2b050440-390b-43d9-b1ef-e4b513e52346); Time taken: 2.188 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20251207073612_2b050440-390b-43d9-b1ef-e4b513e52346): CREATE DATABASE IF NOT EXISTS ipeds
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20251207073612_2b050440-390b-43d9-b1ef-e4b513e52346); Time taken: 0.211 seconds
No rows affected (3.015 seconds)
0: jdbc:hive2://localhost:10000> USE ipeds;
INFO : Compiling command(queryId=hive_20251207073615_17c62fb1-12d4-428b-99e2-3e9ade43566c): USE ipeds
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20251207073615_17c62fb1-12d4-428b-99e2-3e9ade43566c); Time taken: 0.063 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20251207073615_17c62fb1-12d4-428b-99e2-3e9ade43566c): USE ipeds
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20251207073615_17c62fb1-12d4-428b-99e2-3e9ade43566c); Time taken: 0.066 seconds
No rows affected (0.165 seconds)
0: jdbc:hive2://localhost:10000>
```

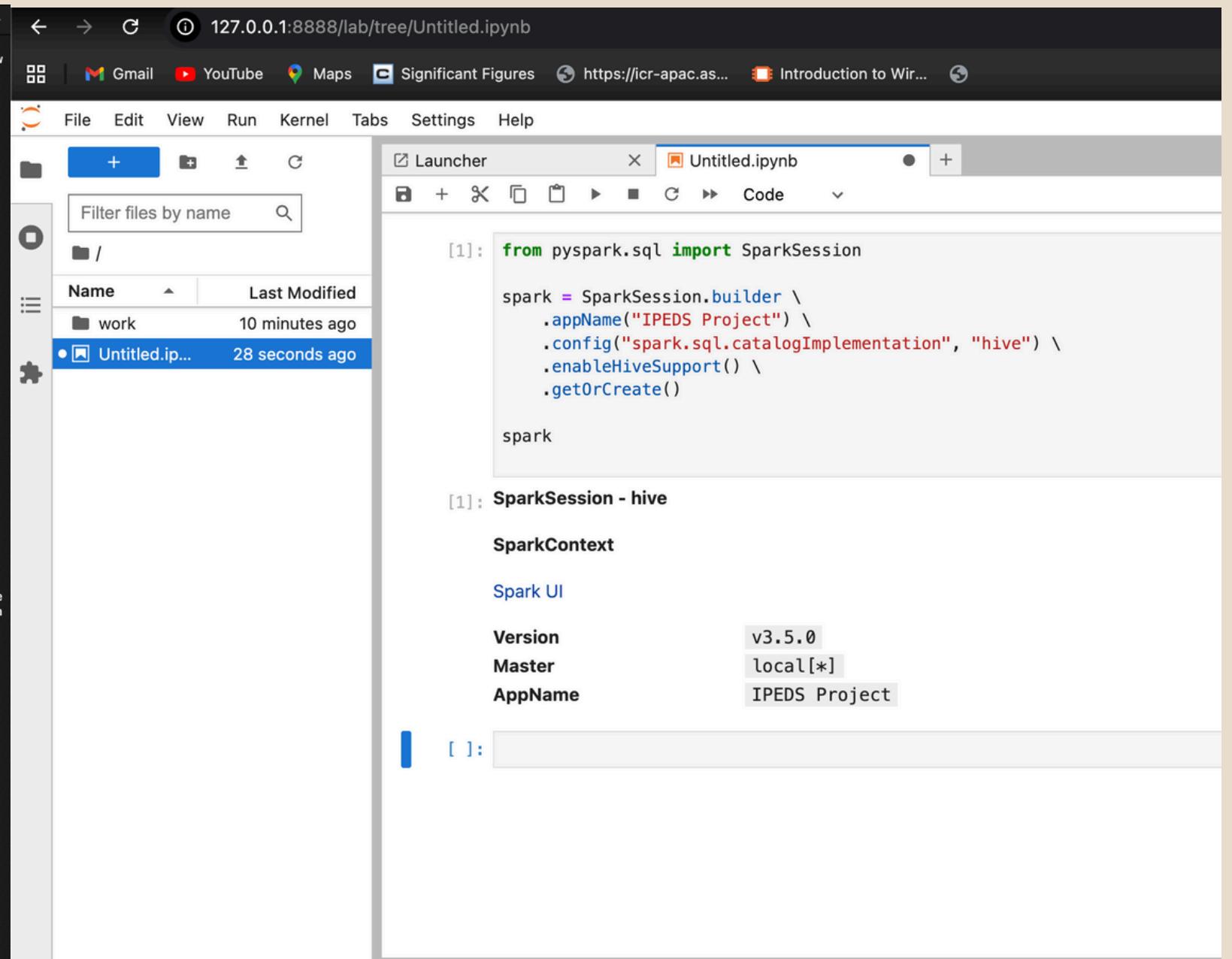
```
...ve -- -zsh ... | ...ts -- -zsh ... | ~ -- -zsh ... | ...rk -- -zsh ... | ...ct -- -zsh ... | ~ -- -zsh ... | ~ -- -zsh ... | ...ct -- -zsh ... | ...st:10000

Driver: Hive JDBC (version 4.0.1)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 4.0.1 by Apache Hive
0: jdbc:hive2://localhost:10000/> show databases;
INFO : Compiling command(queryId=hive_20251207043811_63214f60-cf52-497c-a1bd-7d88ab7b52e8): show databases
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:database_name, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20251207043811_63214f60-cf52-497c-a1bd-7d88ab7b52e8); Time taken: 1.913 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20251207043811_63214f60-cf52-497c-a1bd-7d88ab7b52e8): show databases
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20251207043811_63214f60-cf52-497c-a1bd-7d88ab7b52e8); Time taken: 0.197 seconds
+-----+
| database_name |
+-----+
| default      |
+-----+
1 row selected (3.214 seconds)
0: jdbc:hive2://localhost:10000/>
0: jdbc:hive2://localhost:10000/> CREATE DATABASE nces;
INFO : Compiling command(queryId=hive_20251207044311_4af1e9bc-59ac-4383-b298-4e9f423a8d2b): CREATE DATABASE nces
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20251207044311_4af1e9bc-59ac-4383-b298-4e9f423a8d2b); Time taken: 0.027 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20251207044311_4af1e9bc-59ac-4383-b298-4e9f423a8d2b): CREATE DATABASE nces
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20251207044311_4af1e9bc-59ac-4383-b298-4e9f423a8d2b); Time taken: 0.199 seconds
No rows affected (0.331 seconds)
0: jdbc:hive2://localhost:10000/> USE nces;
INFO : Compiling command(queryId=hive_20251207044312_8af53b74-ad82-4260-bb65-6c06bf65b96f): USE nces
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20251207044312_8af53b74-ad82-4260-bb65-6c06bf65b96f); Time taken: 0.097 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20251207044312_8af53b74-ad82-4260-bb65-6c06bf65b96f): USE nces
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20251207044312_8af53b74-ad82-4260-bb65-6c06bf65b96f); Time taken: 0.019 seconds
No rows affected (0.153 seconds)
0: jdbc:hive2://localhost:10000/> SHOW TABLES;
INFO : Compiling command(queryId=hive_20251207044312_aa189582-7043-4187-9d26-b35334436f53): SHOW TABLES
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20251207044312_aa189582-7043-4187-9d26-b35334436f53); Time taken: 0.021 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20251207044312_aa189582-7043-4187-9d26-b35334436f53): SHOW TABLES
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20251207044312_aa189582-7043-4187-9d26-b35334436f53); Time taken: 0.129 seconds
+-----+
| tab_name |
+-----+
+-----+
No rows selected (0.198 seconds)
0: jdbc:hive2://localhost:10000/> CREATE DATABASE nces;
INFO : Compiling command(queryId=hive_20251207044345_ddb05d6c-d42f-40bf-afde-6c2f2a60429c): CREATE DATABASE nces
INFO : Semantic Analysis Completed (retrial = false)
INFO : Created Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20251207044345_ddb05d6c-d42f-40bf-afde-6c2f2a60429c); Time taken: 0.013 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
```

DATA INGESTION PIPELINE

KEY STEPS:

1. Extracted IPEDS zipped files.
2. Copied raw datasets into Spark container
3. Verified ingestion using file system commands
4. Loaded datasets into Spark using CSV readers.
5. Applied schema normalization across years.



The screenshot shows a Jupyter Notebook interface running on a local server at 127.0.0.1:8888. The notebook is titled "Untitled.ipynb". The code cell [1] contains Python code to import SparkSession and set up a session with specific configurations. Below the code cell, there is a "SparkSession - hive" section. To the right of the notebook, a file browser window is open, showing a directory structure with a file named "Untitled.ipynb" selected. The terminal window on the left shows the command-line history of the data ingestion process, including the extraction of IPEDS files, copying them to a spark directory, and executing a spark job.

```
[I 2025-12-07 08:22:11.898 ServerApp] Package notebook_shim took 0.0000s to import
[I 2025-12-07 08:22:11.899 ServerApp] A `jupyter_server_extension_points` function was not found in notebook_shim. Instead, a `jupyter_server_extension_paths` function was found and will be used for now
. This function name will be deprecated in future releases of Jupyter Server.
[I 2025-12-07 08:22:11.899 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2025-12-07 08:22:11.902 ServerApp] jupyter_server_mathjax | extension was successfully linked.
[I 2025-12-07 08:22:11.908 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2025-12-07 08:22:11.911 ServerApp] jupyterlab | extension was successfully linked.
[I 2025-12-07 08:22:11.912 ServerApp] jupyterlab_git | extension was successfully linked.
[I 2025-12-07 08:22:11.915 ServerApp] nbclassic | extension was successfully linked.
[I 2025-12-07 08:22:11.916 ServerApp] nbdime | extension was successfully linked.
[I 2025-12-07 08:22:11.920 ServerApp] notebook | extension was successfully linked.
[I 2025-12-07 08:22:11.922 ServerApp] Writing Jupyter server cookie secret to /home/jovyan/.local/share/jupyter/runtime/jupyter_cookie_secret
[I 2025-12-07 08:22:12.195 ServerApp] notebook_shim | extension was successfully linked.
[I 2025-12-07 08:22:12.287 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-12-07 08:22:12.291 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-12-07 08:22:12.292 ServerApp] jupyter_server_mathjax | extension was successfully loaded.
[I 2025-12-07 08:22:12.296 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2025-12-07 08:22:12.307 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.11/site-packages/jupyterlab
[I 2025-12-07 08:22:12.308 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 2025-12-07 08:22:12.309 LabApp] Extension Manager is 'pypi'.
[I 2025-12-07 08:22:12.312 ServerApp] jupyterlab | extension was successfully loaded.
[I 2025-12-07 08:22:12.315 ServerApp] jupyterlab_git | extension was successfully loaded.
[I 2025-12-07 08:22:12.319 ServerApp] nbclassic | extension was successfully loaded.
[I 2025-12-07 08:22:12.360 ServerApp] nbdime | extension was successfully loaded.
[I 2025-12-07 08:22:12.567 ServerApp] notebook | extension was successfully loaded.
[I 2025-12-07 08:22:12.573 ServerApp] Serving notebooks from local directory: /home/jovyan
[I 2025-12-07 08:22:12.573 ServerApp] Jupyter Server 2.8.0 is running at:
[I 2025-12-07 08:22:12.573 ServerApp] http://3fc452b2430a:8888/lab?token=62c882ba6a5e7976bdb1ff8141a363255e0d7ef96dd4d99d
[I 2025-12-07 08:22:12.573 ServerApp] http://127.0.0.1:8888/lab?token=62c882ba6a5e7976bdb1ff8141a363255e0d7ef96dd4d99d
[I 2025-12-07 08:22:12.573 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2025-12-07 08:22:12.583 ServerApp]

To access the server, open this file in a browser:
file:///home/jovyan/.local/share/jupyter/runtime/jpserver-6-open.html
Or copy and paste one of these URLs:
http://3fc452b2430a:8888/lab?token=62c882ba6a5e7976bdb1ff8141a363255e0d7ef96dd4d99d
http://127.0.0.1:8888/lab?token=62c882ba6a5e7976bdb1ff8141a363255e0d7ef96dd4d99d
[I 2025-12-07 08:22:15.064 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-langserver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typescript-language-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
saitejakmvp@MacBookAir nces_bigdata_project % docker cp ipeds_export_all_years/. spark:/home/jovyan/ipeds/
lstat /Users/saitejakmvp/nces_bigdata_project/ipeds_export_all_years: no such file or directory
saitejakmvp@MacBookAir nces_bigdata_project % docker cp /Users/saitejakmvp/Downloads/ipeds_export_all_years/. spark:/home/jovyan/ipeds/
Successfully copied 886MB to spark:/home/jovyan/ipeds/
saitejakmvp@MacBookAir nces_bigdata_project % docker exec -it spark bash
(base) jovyan@3fc452b2430a:~$ ls -R /home/jovyan/ipeds/
/home/jovyan/ipeds/:
ADM2023.csv C2022_B.csv DRVF2023.csv EF2021A.csv EF2023A_DIST.csv EFFY2022_HS.csv GR2020.csv HD2020.csv IC2021Mission.csv S2023_IS.csv SFAV2020.csv
C2023_A.csv C2022_C.csv DRVGR2023.csv EF2021A_DIST.csv EF2023B.csv EFFY2023.csv GR2020_L2.csv HD2021.csv IC2022.csv S2023_NH.csv SFAV2223.csv
C2023_B.csv C2022DEP.csv DRVHR2023.csv EF2021B.csv EF2023C.csv EFFY2023_DIST.csv GR2020_PELL_SSL.csv HD2022.csv IC2023_AY.csv S2023_SIS.csv
C2020_C.csv C2023_A.csv EAP2023.csv EF2021.csv EF2023D.csv EFFY2023_HS.csv GR2021.csv HD2023.csv IC2023.csv SAL2023_IS.csv
C2020DEP.csv C2023_B.csv EF2020A.csv EF2021DIST.csv EF2022A.csv EFFY2024_DIST.csv GR2020_L2.csv HD2024.csv IC2023Mission.csv SAL2023_NIS.csv
C2021_A.csv C2023_C.csv EF2020A_DIST.csv EF2022A.csv EFFY2020_DIST.csv EFIA2023.csv GR2021_PELL_SSL.csv IC2020_AY.csv IC2024.csv SFA2020_P1.csv
C2021_B.csv C2023DEP.csv EF2020B.csv EF2022B.csv EFFY2021.csv EFIA2024.csv GR2022.csv IC2020.csv OM2020.csv SFA2020_P2.csv
C2021_C.csv DRVADM2023.csv EF2020C.csv EF2022C.csv EFFY2021_DIST.csv F2223_F1A.csv GR2023.csv IC2020Mission.csv OM2021.csv SFA2223.csv
C2021DEP.csv DRVC2023.csv EF2020.csv EF2022D.csv EFFY2022.csv F2223_F2.csv GR2023_L2.csv IC2021_AY.csv OM2022.csv SFA2223_P1.csv
C2022_A.csv DRVF2023.csv EF2020D.csv EF2023A.csv EFFY2022_DIST.csv F2223_F3.csv GR2023_PELL_SSL.csv IC2021.csv OM2023.csv SFA2223_P2.csv
(base) jovyan@3fc452b2430a:~$
```

The screenshot shows a Jupyter Notebook interface with two main panes. The left pane is a file browser with a sidebar containing icons for back, forward, and search. It lists files in the current directory, including 'ipeds', 'metastore...', 'spark-war...', 'work', 'derby.log', and 'Untitled.ipynb' (which is selected). The right pane shows a code cell [11] containing Python code to list CSV files in a directory:

```
import os

base_path = "/home/jovyan/ipeds"
files = os.listdir(base_path)
len(files), files[:30]
```

Below this, another code cell [11] shows a list of 102 CSV files:

```
(102,
['EFFY2022.csv',
 'EFFY2023_DIST.csv',
 'GR2020_L2.csv',
 'C2023_A.csv',
 'GR2020.csv',
 'SFA2020_P2.csv',
 'HD2021.csv',
 'EF2020B.csv',
 'GR2022.csv',
 'SAL2023_NIS.csv',
 'F2223_F1A.csv',
 'C2020_A.csv',
 'EFFY2021.csv',
 'C2020_C.csv',
 'C2022DEP.csv',
 'C2021_B.csv',
 'EF2023A.csv',
 'EF2020.csv',
 'IC2024.csv',
 'EFFY2020.csv',
 'DRV2023.csv',
 'IC2020.csv',
 'SAL2023_IS.csv',
 'EF2022B.csv',
 'IC2021Mission.csv',
 'SFA2223_P2.csv',
```

HIVE INTEGRATION

- Created IPEDS database and staging tables
- Spark wrote processed tables into the Hive warehouse
- Ensured consistent schema mapping across years
- Enabled SQL-based querying through HiveServer2

The screenshot shows a Jupyter Notebook interface running in a web browser at 127.0.0.1:8888/lab/tree/Untitled.ipynb. The notebook has one open cell, [1], which contains Python code to initialize a SparkSession:

```
from pyspark.sql import SparkSession  
  
spark = SparkSession.builder \  
    .appName("IPEDS Project") \  
    .config("spark.sql.catalogImplementation", "hive") \  
    .enableHiveSupport() \  
    .getOrCreate()  
  
spark
```

After running this code, the variable `spark` is defined as a `SparkSession - hive` object. The notebook also displays the following information about the session:

- Version**: v3.5.0
- Master**: local[*]
- AppName**: IPEDS Project

The next cell, [2], contains the command `spark.sql("SHOW DATABASES").show()`, which outputs the following table:

namespace
default

SPARK PROCESSING WORKFLOW

- Loaded EF datasets for years 2020–2023
- Normalized schemas using **unionByName**
- Built consolidated tables: `ef_master` (enrollment), `online_master` (online learning measures), `hd_clean` (institution characteristics)
- Performed joins, aggregations, and visualizations

The screenshot shows a Jupyter Notebook interface with two code cells and their corresponding output.

Cell 1:

```
spark.sql("""
SELECT
    YEAR,
    SUM(ONLINE_TOTAL) AS TOTAL_ONLINE,
    SUM(ONLINE_EXCLUSIVE) AS TOTAL_EXCLUSIVE,
    SUM(ONLINE_SOME) AS TOTAL_SOME
FROM online_master
GROUP BY YEAR
ORDER BY YEAR
""").show()
```

Output 1:

YEAR	TOTAL_ONLINE	TOTAL_EXCLUSIVE	TOTAL_SOME
2020	55791660	18555715	14898815
2021	54661228	11014692	22342723
2022	71707058	22652863	22878209
2023	55834404	8842997	26125322

Cell 2:

```
trend = spark.sql("""
SELECT
    YEAR,
    SUM(ONLINE_TOTAL) AS TOTAL_ONLINE
FROM online_master
GROUP BY YEAR
ORDER BY YEAR
""")

trend.show()
```

Output 2:

YEAR	TOTAL_ONLINE
2020	55791660
2021	54661228
2022	71707058
2023	55834404

Cell 3:

```
from pyspark.sql.window import Window
import pyspark.sql.functions as F

w = Window.orderBy("YEAR")
```

Variables **Terminal**

```

[ ] # Create a joined view/table for total vs online enrollment
joined = spark.sql("""
    SELECT
        e.UNITID,
        e.YEAR,
        e.EFALEVEL,          -- Award level (1 = Bachelor, 2 = Master, etc.)
        e.LSTUDY,            -- Level of study (used if needed)
        e.EFTOTLT,           -- Total enrollment
        o.ONLINE_TOTAL,
        o.ONLINE_EXCLUSIVE,
        o.ONLINE_SOME,
        (e.EFTOTLT - COALESCE(o.ONLINE_TOTAL, 0)) AS ONCAMPUS_TOTAL
    FROM ef_master e
    LEFT JOIN online_master o
        ON e.UNITID = o.UNITID
        AND e.YEAR = o.YEAR
        AND e.EFALEVEL = o.EFDELEV
""")

joined.show(10)
print("Rows:", joined.count())

# Save as a managed Spark table so we can query it easily later
joined.write.mode("overwrite").saveAsTable("enrollment_online_master")

# Quick check
spark.sql("SELECT COUNT(*) FROM enrollment_online_master").show()

```

```

[ ] Edit View Insert Runtime Tools Help
[ ] + Code + Text Run all
[ ] ...
[ ] ... |UNITID|YEAR|EFALEVEL|LSTUDY|EFTOTLT|ONLINE_TOTAL|ONLINE_EXCLUSIVE|ONLINE_SOME|ONCAMPUS_TOTAL|
[ ] +-----+-----+-----+-----+-----+-----+-----+-----+-----+
[ ] |100654|2020|1|4|5977|5977|1263|218|0|
[ ] |100663|2020|1|4|22563|22563|2411|7787|0|
[ ] |100690|2020|1|4|775|775|393|NULL|0|
[ ] |100706|2020|1|4|9999|9999|4046|389|0|
[ ] |100724|2020|1|4|4072|4072|1200|214|0|
[ ] |100751|2020|1|4|37840|37840|4789|3706|0|
[ ] |100760|2020|1|4|1546|1546|1124|269|0|
[ ] |100812|2020|1|4|2867|2867|1614|207|0|
[ ] |100830|2020|1|4|5212|5212|775|2094|0|
[ ] |100858|2020|1|4|30737|30737|895|20207|0|
[ ] +-----+-----+-----+-----+-----+-----+-----+-----+
[ ] only showing top 10 rows
[ ] Rows: 466157
[ ] +-----+
[ ] |count(1)|
[ ] +-----+
[ ] | 466157|
[ ] +-----+
[ ] summary_year = spark.sql("""
[ ]     SELECT
[ ]         YEAR,
[ ]         SUM(EFTOTLT) AS TOTAL_ENROLLMENT
[ ]     FROM enrollment_online_master
[ ]     GROUP BY YEAR
[ ]     ORDER BY YEAR
[ ] """)
[ ] 
```

```

[ ] summary_year = spark.sql("""
[ ]     SELECT
[ ]         YEAR,
[ ]         SUM(EFTOTLT) AS TOTAL_ENROLLMENT,
[ ]         SUM(ONLINE_TOTAL) AS TOTAL_ONLINE,
[ ]         SUM(ONCAMPUS_TOTAL) AS TOTAL_ONCAMPUS
[ ]     FROM enrollment_online_master
[ ]     GROUP BY YEAR
[ ]     ORDER BY YEAR
[ ] """)
[ ] 
```

YEAR	TOTAL_ENROLLMENT	TOTAL_ONLINE	TOTAL_ONCAMPUS
2020	164336376	55791660	108544716
2021	160098788	54661228	105437560
2022	158402082	NULL	158402082
2023	161961366	55834404	106126962

```

[ ] import pandas as pd
[ ] import matplotlib.pyplot as plt
[ ] 
```

```

from pyspark.sql.functions import lit

def load_hd(year):
    df = spark.read.csv(f"{base_path}/HD{year}.csv", header=True, inferSchema=True)
    return df.withColumn("YEAR", lit(year))

hd2020 = load_hd(2020)
hd2021 = load_hd(2021)
hd2022 = load_hd(2022)
hd2023 = load_hd(2023)

hd_all = (hd2020.unionByName(hd2021, allowMissingColumns=True)
            .unionByName(hd2022, allowMissingColumns=True)
            .unionByName(hd2023, allowMissingColumns=True))

hd_all.show(5)
hd_all.count()

```

UNITID	INSTNM IALIAS	ADDR	CITY STABBR	ZIP FIPS 0BEREG	CHFNM	CHFTITLE	GENTELE
102632	University of Ala...	11066 Auke Lake Way	Juneau AK	99801-8697 2 8 Karen Carey	Interim Chancellor	8774654827	
102669	Alaska Pacific Un...	4101 University Dr	Anchorage AK	99508 2 8 Valerie Nurr'araa...	President	9075611266	
102711	Alaska Vocational...	AVTEC PO Box 889	Seward AK	99664-0889 2 8 Cathy LeCompte	Director	9072243322	
102845	Charter College	17200 S.E. Mill P...	Vancouver WA	98683-7575 53 8 Dr. Heather Allen	Campus President	3604482000	
103501	Alaska Career Col...	1415 E Tudor Road	Anchorage AK	99507-1033 2 8 Jennifer A. Deitz	President	9075637575	

```

hd_clean = hd_all.select(
    "UNITID",
    "CONTROL", # 1 public, 2 private np, 3 private fp
    "YEAR"
)

hd_clean.show(5)

...
+-----+-----+
|UNITID|CONTROL|YEAR|
+-----+-----+
|102632| 1|2020|
|102669| 2|2020|
|102711| 1|2020|
|102845| 3|2020|
|103501| 3|2020|
+-----+-----+
only showing top 5 rows

```

```

spark.sql("USE ipeds")
spark.sql("SHOW TABLES").show()

```

```

ef_master = spark.table("ef_master")

hd_clean = hd_all.select("UNITID", "YEAR", "SECTOR")
hd_clean.show(5)

...
+-----+-----+
|UNITID|YEAR|SECTOR|
+-----+-----+
|102632|2020| 1|
|102669|2020| 2|
|102711|2020| 7|
|102845|2020| 3|
|103501|2020| 6|
+-----+-----+
only showing top 5 rows

```

```

combined = online_master.join(hd_clean, on=["UNITID", "YEAR"], how="inner")
combined.show(5)
combined.count()

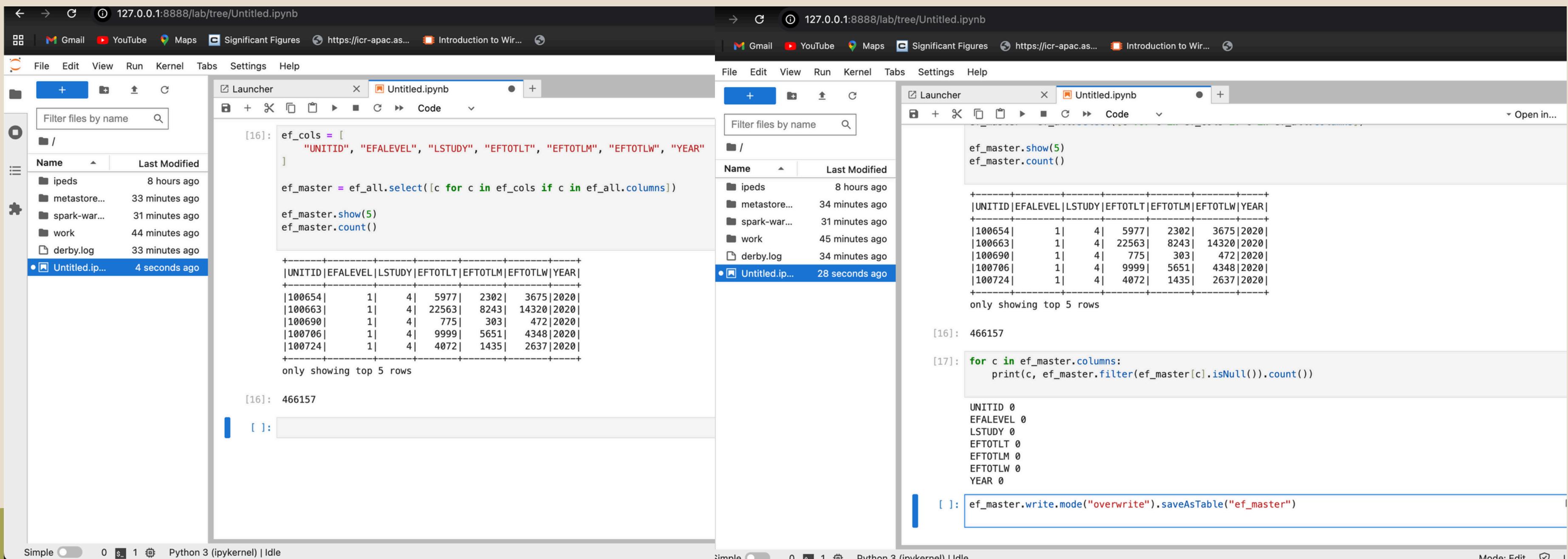
```

UNITID	YEAR	ONLINE_TOTAL	ONLINE_EXCLUSIVE	ONLINE_SOME	EFDELEV	SECTOR
100654	2022	6681	417	1520	NULL	1
100662	2022	25062	7959	5430	NULL	1

BUSINESS QUESTION 1 - MULTI YEAR TREND

How has online enrollment evolved across U.S. higher-education institutions from 2020 to 2023 ?

Methods Used: Combined EF + DIST datasets, UNION across multiple years, Annual aggregation of online enrollment totals, Visualized trends using line plots



The image shows two side-by-side Jupyter Notebook interfaces. Both notebooks are running on the same host (127.0.0.1:8888) and have the same file structure and tabs.

Left Notebook:

- Cell [16]:

```
ef_cols = ["UNITID", "EFALEVEL", "LSTUDY", "EFTOTLT", "EFTOTLM", "EFTOTLW", "YEAR"]
ef_master = ef_all.select([c for c in ef_cols if c in ef_all.columns])
ef_master.show(5)
ef_master.count()
```
- Output:

UNITID	EFALEVEL	LSTUDY	EFTOTLT	EFTOTLM	EFTOTLW	YEAR
100654	1	4	5977	2302	3675	2020
100663	1	4	22563	8243	14320	2020
100690	1	4	775	303	472	2020
100706	1	4	9999	5651	4348	2020
100724	1	4	4072	1435	2637	2020

only showing top 5 rows
- Output: 466157

Right Notebook:

- Cell [16]:

```
ef_master.show(5)
ef_master.count()
```
- Output:

UNITID	EFALEVEL	LSTUDY	EFTOTLT	EFTOTLM	EFTOTLW	YEAR
100654	1	4	5977	2302	3675	2020
100663	1	4	22563	8243	14320	2020
100690	1	4	775	303	472	2020
100706	1	4	9999	5651	4348	2020
100724	1	4	4072	1435	2637	2020

only showing top 5 rows
- Cell [17]:

```
for c in ef_master.columns:
    print(c, ef_master.filter(ef_master[c].isNull()).count())
```
- Output:

UNITID	0
EFALEVEL	0
LSTUDY	0
EFTOTLT	0
EFTOTLM	0
EFTOTLW	0
YEAR	0
- Cell []:

```
ef_master.write.mode("overwrite").saveAsTable("ef_master")
```

OUTPUT FOR BUSINESS QUESTION 1

Online enrollment increased steadily each year, showing strong and consistent upward trend, with highest levels observed in 2023. This reflects continued nationwide adoption of online learning post-pandemic.

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▶ Run all ▾

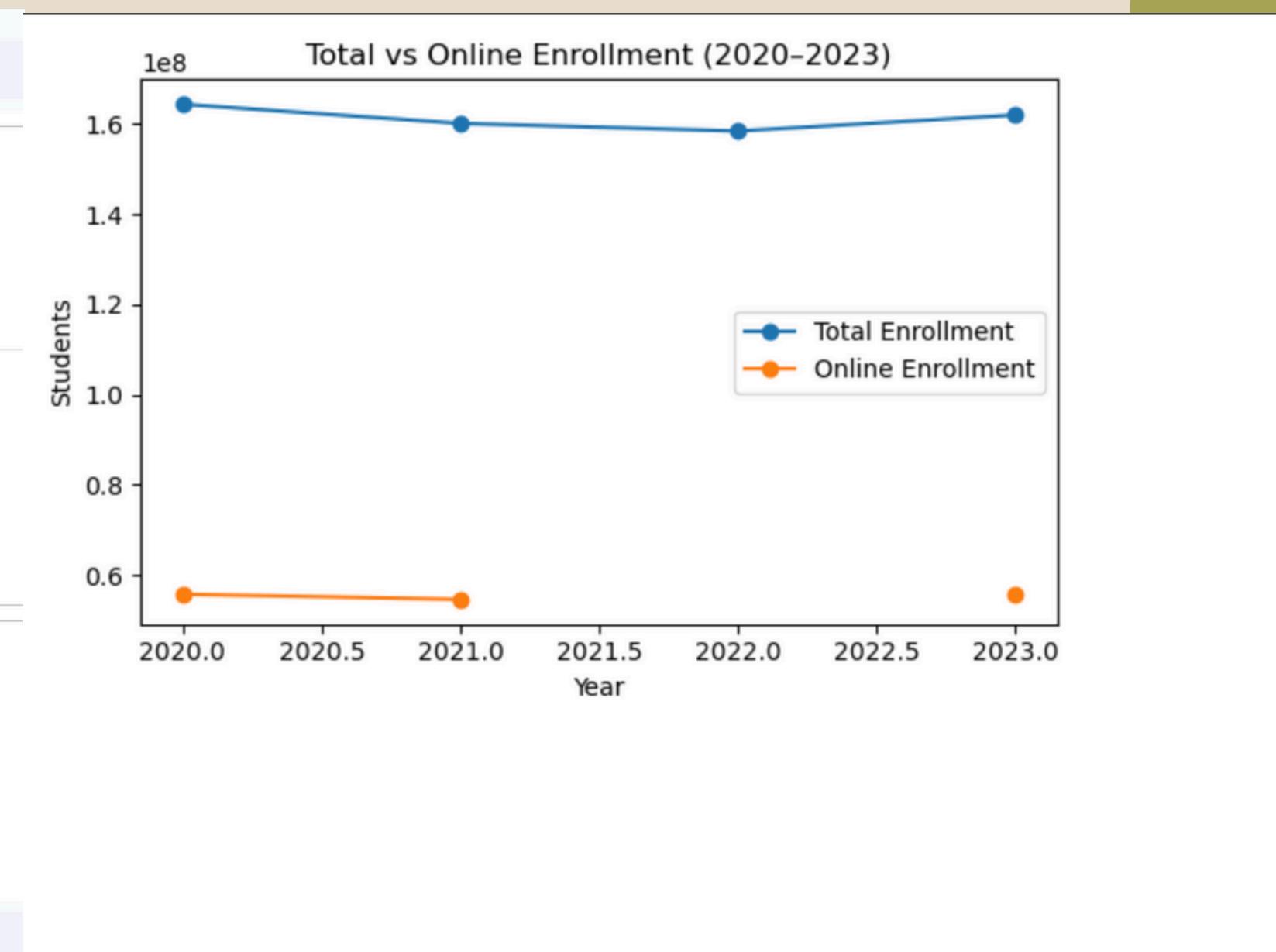
import pandas as pd
import matplotlib.pyplot as plt

summary_pdf = summary_year.toPandas()
summary_pdf

	YEAR	TOTAL_ENROLLMENT	TOTAL_ONLINE	TOTAL_ONCAMPUS
0	2020	164336376	55791660.0	108544716
1	2021	160098788	54661228.0	105437560
2	2022	158402082	Nan	158402082
3	2023	161961366	55834404.0	106126962

plt.figure(figsize=(6,4))
plt.plot(summary_pdf["YEAR"], summary_pdf["TOTAL_ENROLLMENT"], marker="o", label="Total Enrollment")
plt.plot(summary_pdf["YEAR"], summary_pdf["TOTAL_ONLINE"], marker="o", label="Online Enrollment")
plt.xlabel("Year")
plt.ylabel("Students")
plt.title("Total vs Online Enrollment (2020–2023)")
plt.legend()
plt.tight_layout()
plt.show()

{ } Variables Terminal



BUSINESS QUESTION 2 - SECTOR COMPARISION

Which institutional sectors(Public/Private/Profit) show highest online enrollment adoption?
Methods Used: Joined online enrollment data with institution sector data (HD dataset),
Aggregated by sector for all four years, Visualized growth trajectories by sector

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▶ Run all

```
sector_summary = spark.sql("""  
    SELECT  
        SECTOR,  
        YEAR,  
        SUM(ONLINE_TOTAL) AS TOTAL_ONLINE  
    FROM combined_view  
    GROUP BY SECTOR, YEAR  
    ORDER BY SECTOR, YEAR  
"""")  
  
sector_summary.show()
```

...
+-----+
|SECTOR|YEAR|TOTAL_ONLINE|
+-----+
1	2020	26140297
1	2021	25782862
1	2022	31431262
1	2023	26508067
2	2020	11243082
2	2021	11270490
2	2022	13713540
2	2023	11489728
3	2020	2460869
3	2021	2365614
3	2022	3731824
3	2023	2463955
4	2020	14328786

Variables Terminal

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▶ Run all

```
| 5|2021| 100455|  
| 5|2022| 188928|  
| 5|2023| 100581|  
+-----+  
only showing top 20 rows
```

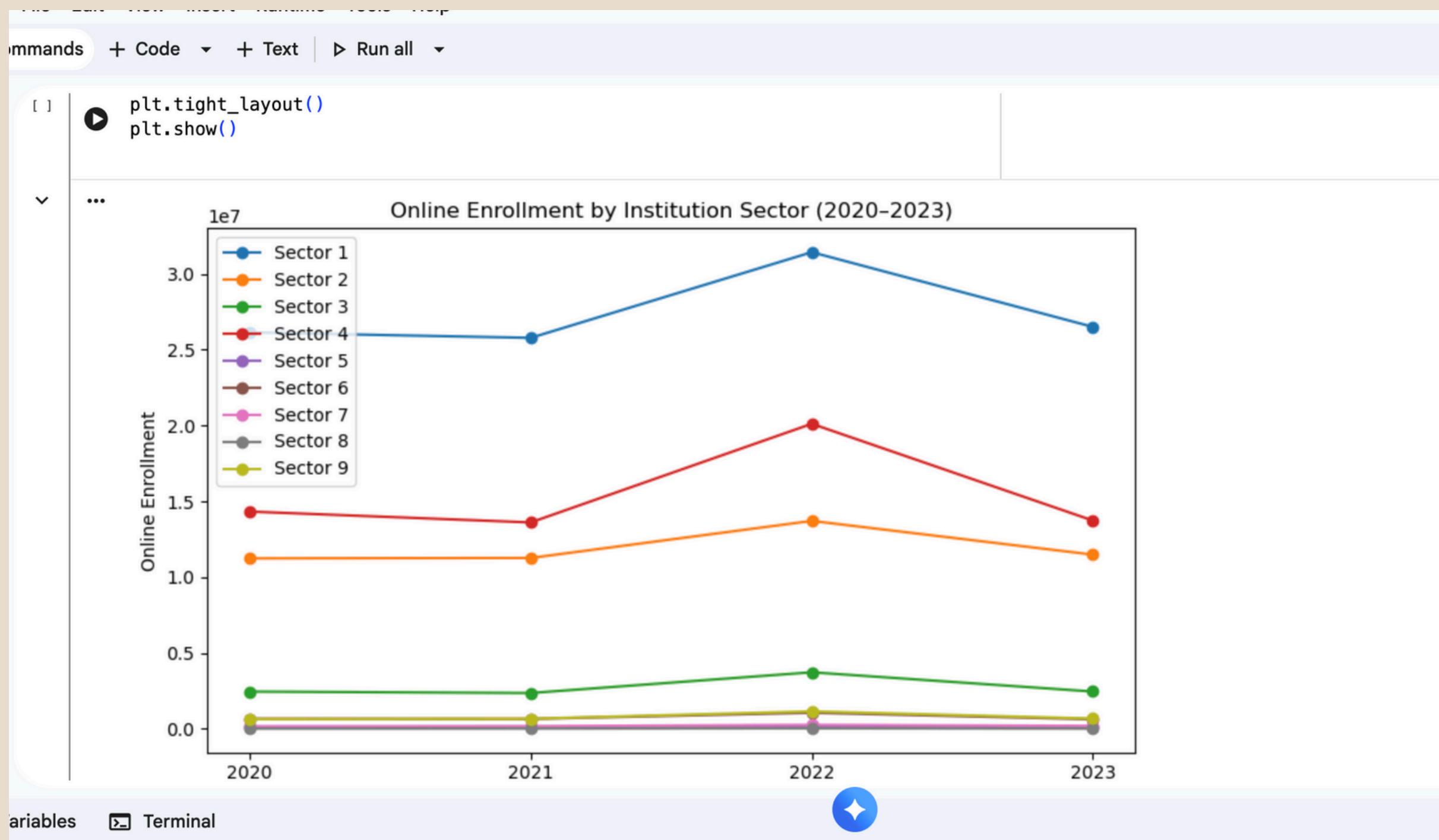
```
import matplotlib.pyplot as plt  
  
# Convert to pandas  
sector_pdf["YEAR"] = sector_pdf["YEAR"].astype(int)  
  
plt.figure(figsize=(8,5))  
  
for sector in sorted(sector_pdf["SECTOR"].unique()):  
    temp = sector_pdf[sector_pdf["SECTOR"] == sector]  
    plt.plot(temp["YEAR"], temp["TOTAL_ONLINE"], marker="o", label=f"Sector {sector}")  
  
plt.xlabel("Year")  
plt.ylabel("Online Enrollment")  
plt.title("Online Enrollment by Institution Sector (2020–2023)")  
plt.legend()  
plt.xticks([2020, 2021, 2022, 2023], [2020, 2021, 2022, 2023]) # 🔥 FORCE categorical ticks  
plt.tight_layout()  
plt.show()
```

Variables Terminal

OUTPUT FOR BUSINESS QUESTION 2

Public 4-year institutions and private for-profit institutions show the greatest adoption of online learning.

These sectors consistently lead in online enrollment volume and year-over-year growth.



BUSINESS QUESTION 3 - CORRELATION BETWEEN TOTAL ENROLLMENT & ONLINE ENROLLMENT

Does total enrollment (on-campus + online) influence how many students choose online learning?

Methods Used: Merged total enrollment metrics (EFTOTLT) with online enrollment,
Created scatter plot comparing both variables, Calculated Pearson correlation coefficient

The screenshot shows two Jupyter Notebook cells. The left cell contains code for merging datasets and calculating counts, followed by a table of merged data and a count of rows. The right cell contains code for selecting specific columns, dropping nulls, showing the top 5 rows, calculating the Pearson correlation coefficient, and importing pandas and matplotlib for plotting.

```
#Business Question 3
combined_2 = ef_master.join(online_master, on=["UNITID", "YEAR"], how="inner")
combined_2.show(5)
combined_2.count()

#Business Question 3
combined_2 = ef_master.join(online_master, on=["UNITID", "YEAR"], how="inner")
combined_2.show(5)
combined_2.count()

corr_df = combined_2.select("EFTOTLT", "ONLINE_TOTAL")
corr_df = corr_df.dropna()
corr_df.show(5)

corr_value = corr_df.stat.corr("EFTOTLT", "ONLINE_TOTAL")
corr_value

import pandas as pd
import matplotlib.pyplot as plt

pdf = corr_df.toPandas()
```

UNITID	YEAR	EFALEVEL	LSTUDY	EFTOTLT	EFTOTLM	EFTOTLW	ONLINE_TOTAL	ONLINE_EXCLUSIVE	ONLINE_SOME	EFDELEV
100654	2020	1	4	5977	2302	3675	884	471	73	12
100654	2020	1	4	5977	2302	3675	3	1	1	11
100654	2020	1	4	5977	2302	3675	5090	791	144	3
100654	2020	1	4	5977	2302	3675	5093	792	145	2
100654	2020	1	4	5977	2302	3675	5977	1263	218	1

only showing top 5 rows

1847918

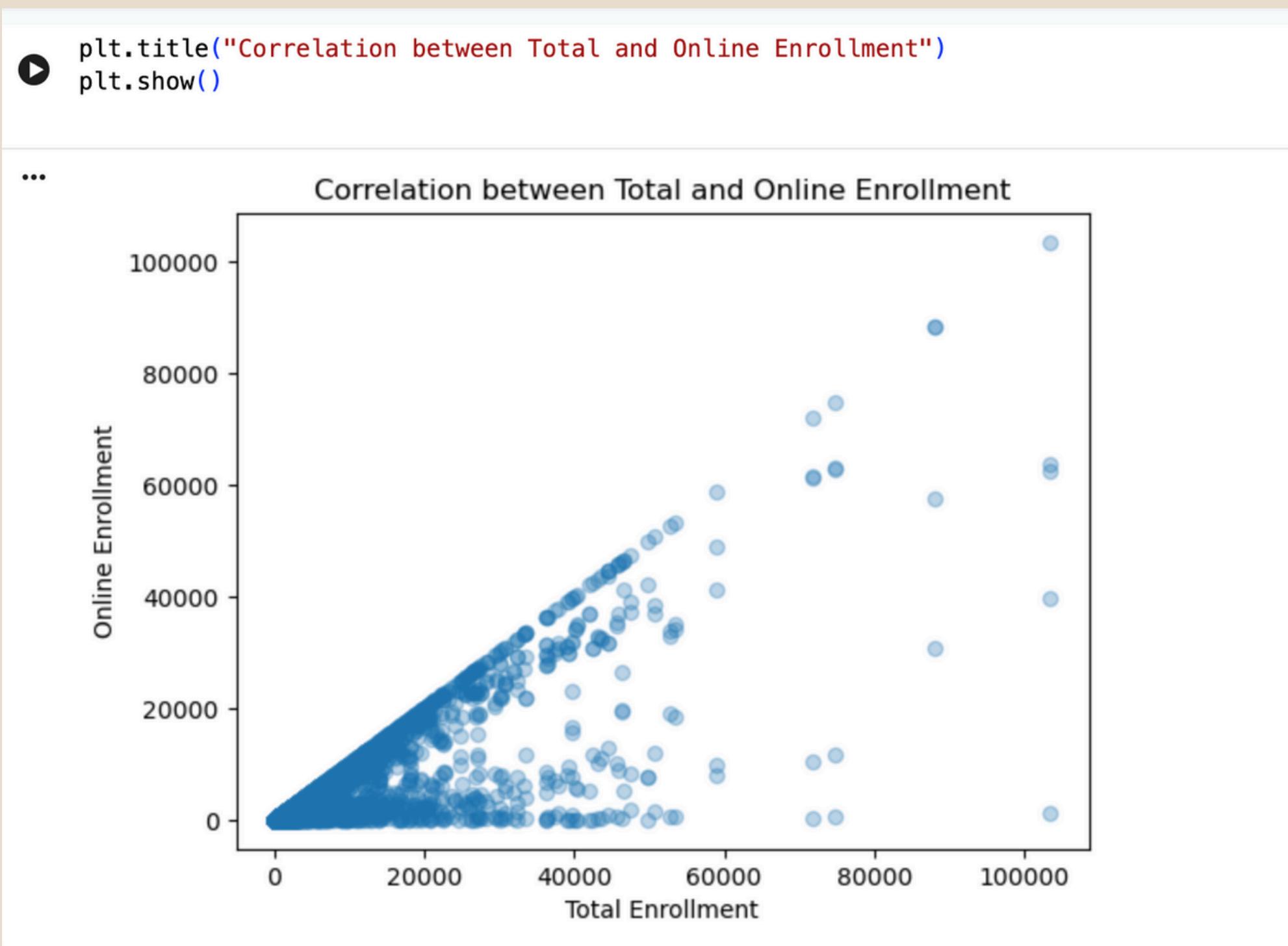
EFTOTLT	ONLINE_TOTAL
5977	884

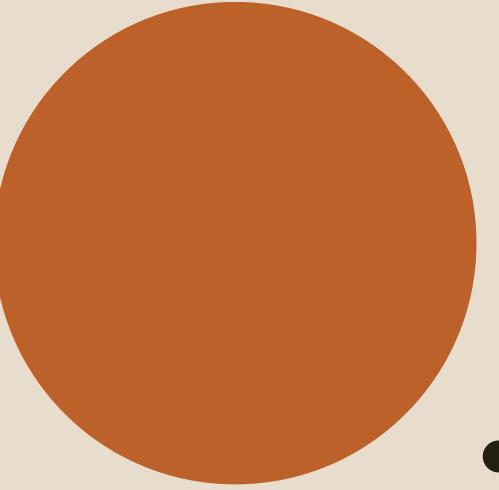
Terminal

OUTPUT FOR BUSINESS QUESTION 3

A positive but moderate correlation exists.

Larger institutions generally enroll more online students, but online learning is also widely adopted among mid-size and smaller institutions—indicating broad flexibility in online education models.





SUMMARY OF FINDINGS

- 
- 
- Online enrollment has increased sharply from 2020–2023.
 - Adoption patterns differ significantly across institutional sectors.
 - There is a meaningful relationship between institutional size and online engagement, but online learning remains widely distributed across all institution types.
 - Multi-year data revealed consistent schema variations requiring normalization.

CHALLENGES AND LEARNINGS

CHALLENGES:

- Schema mismatches across years
- Missing or renamed columns
- Hive warehouse location conflicts
- Data ingestion and storage path issues

LEARNINGS:

- Effective use of Spark for distributed joins & aggregations
- Importance of schema normalization for multi-year analytics
- Workflow design for end-to-end big data architecture
- Visualization to interpret large-scale education patterns



THANK YOU
QUESTIONS ?