# [https://avatars2.githubusercontent.com/u/4156894?v=3&s=100](http://www.calstatela.edu/centers/hipic) CIS4560 Term Project Tutorial

#### Authors: Jonas Sanchez

#### Instructor: [Jongwook Woo](https://www.linkedin.com/in/jongwook-woo-7081a85)

#### Date: 11/22/2024

**Lab Tutorial**

Jonas Sanchez (jsanc479@calstatela.edu)

**Car Accident Analysis using Big Data**

**Objectives**

In this hands-on lab, you will learn how to:

* Use LFS to upload Big data Sets to Github
* Import Data into a Hadoop Cluster
* Parse the Necessary Data into another Table
* SQL commands to perform the analysis.
* Visualization

Step 1: Use LFS to Store Large Amounts of Data into a GitHub Repository

**Git hub is a useful repository for developers to store, create, and work together on various projects. One Limitation that this service has is the file size restrictions. Only files less than 25MB can be hosted in a repository. Well, what if you have files that exceed this limit? You will have to use LFS(Large File System) to store the data locally, but can still be accessed via the repository for users to access.**

1. Download LFS and Git for your OS system
2. Open a git shell and perform the following commands where “” is for you to fillout with your filename extension.

git lfs track "\*.csv" ##Tracks the filename

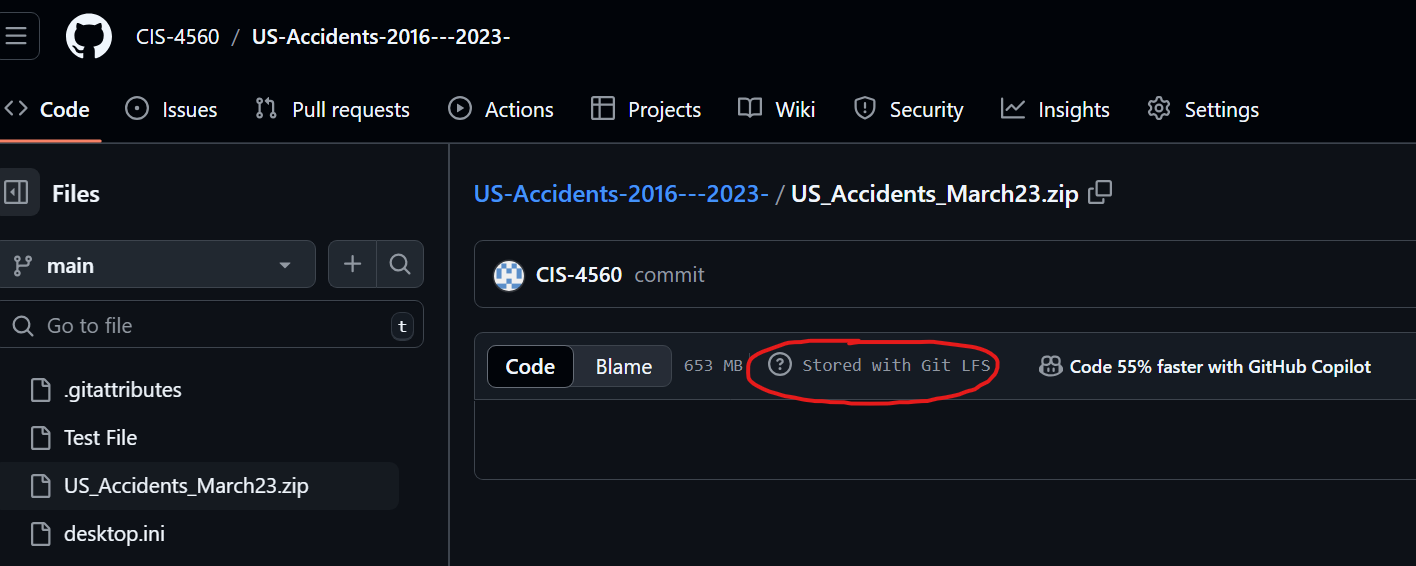
git add file.psd ##Adds the file to the repository

git commit ##Commits to the repo

git push origin main ##Pushes changes to the repo

1. Check your repo and ensure you see the file with an LFS tag

Example:



1. Now you can download the file from the repository. Ensure that the device hosting the source file is accessible and has access to the repository.

Step 2: Import Data into the Hadoop Cluster

**Now that we are able to push and pull large datasets from the repo, we can pull the file into our Oracle Linux Server to and parse the data using the Hadoop Cluster :**

1.Use wget to pull from the github repository and place the files in the directory of your Hadoop cluster user

wget https://github.com/CIS-4560/US-Accidents-2016---2023-/raw/refs/heads/main/US\_Accidents\_March23.zip

unzip US\_Accidents\_March23.zip

hdfs dfs -mkdir tmp/US-Accidents\_02\_16-03\_23

hdfs dfs -mkdir tmp/new\_US-Accidents\_02\_16-03\_23

hdfs dfs -put US\_Accidents\_March23.csv tmp/US-Accidents\_02\_16-03\_23/

2. Using Beeline, create a table for the imported data set. Because the file is a CSV format, we need to use “,” as a delimiter.

DROP TABLE IF EXISTS US\_Car\_Accidents;

CREATE EXTERNAL TABLE IF NOT EXISTS US\_Car\_Accidents ( ID string, Source string, Severity INT, Start\_time TIMESTAMP, End\_time TIMESTAMP, Start\_Lat string, Start\_Long string, End\_Lat string, End\_Long string, Distance\_miles float,

Description string,

Street string,

City string,

County string,

state string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ","

STORED AS TEXTFILE

LOCATION "/user/jsanc479/tmp/US-Accidents\_02\_16-03\_23/"

TBLPROPERTIES ('skip.header.line.count' = '1');

3. Because we want to filter for unwanted data in this dataset, we need to create a separate table that only includes the information we will be using in our data analysis. The new table will be removing **End\_Lat**, **End\_Long**,**Source, Description, Street, and County**. We will also be reducing the max time for the new dataset. We will filter for accidents that occurred in and after 2022.

DROP TABLE IF EXISTS new\_US\_Car\_Accidents;

CREATE TABLE new\_US\_Car\_Accidents

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION "/user/jsanc479/tmp/new\_US-Accidents\_02\_16-03\_23/"

AS

SELECT ID, Severity, Start\_time, End\_time, Start\_Lat, Start\_Long, Distance\_miles

FROM US\_Car\_Accidents

WHERE Start\_time > '2021-12-31 23:59:59'

ORDER BY Start\_time DESC;

4. Once this new table is created, we can go ahead and copy this new dataset from the Oracle Linux Server onto our host machines, where we will be performing Data Analysis.

#Copy new table to linux machine#

hdfs dfs -get /user/\*YOUR\_USERNAME\*/tmp/new\_US-Accidents\_02\_16-03\_23/000000\_0

cat 000000\_0 > US-Car-Accidents\_02\_16-03\_23.txt

#Copy new table to local machine for Data Analysis#

scp \*YOUR\_USERNAME\*@129.146.230.230:/home/\*YOUR\_USERNAME\*/US-Car-Accidents\_02\_16-03\_23.txt

Step 3: Analyze the data using various Data Analysis Tools

**Tableau**

A map of the united states

Description automatically generated

Average Accident time in US January-October 2022

Using tableau, you are able to use Geo-Mapping features to your data to get a visual picture of what your data means. In this chart, we can see the Average Accident time reflected by the different States in the US. OK (Oklahoma) shows as the highest Avg time for an accident to clear from the point of impact, to when traffic is no longer affected. We are able to see that the highest times are reflected more in rural areas rather than cities. We can assume this may be due to the large amounts of area without a heavy population. This can cause accidents to take longer to resolve, because the authorities who need to assist with clearing the roads, may not be as close as in heavily populated areas.

**Excel Power Maps**

A map of the united states

Description automatically generated

 Using Excel Power Maps, we can generate and visualize data over a period of time. This helps determine continuous data as shown above. The visual above shows the number of crashes and their severity by time over two years in the top ten lvl 4 severity cities. As seen in the visual, the number tends to fluctuate throughout the years in all cities. This shows that the top severity levels in cities have periods where accidents are higher and lower, rather than a consistent all-around trend. Seasonal weather may be factor in this data as level 4 severity spikes during the winter months.

 References

1.       URL of Data Source, https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents/data

2.       URL of your Github

https://github.com/CIS-4560/US-Accidents-2016---2023-