**PYSPARK**

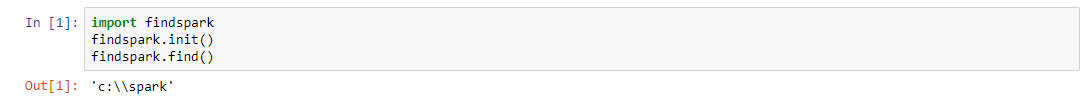
**CASE STUDY**

**UBER ANALYSIS**

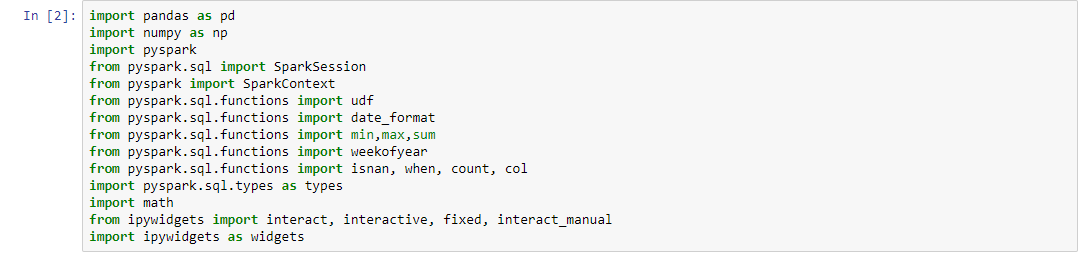
**PALLAV GUPTA**

**46022374**

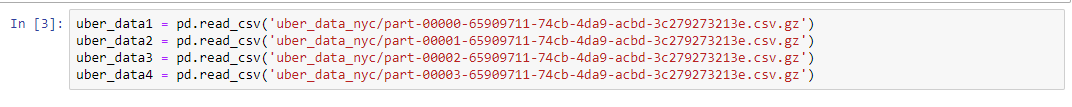
**CONNECTING JUPYTER TO SPARK**



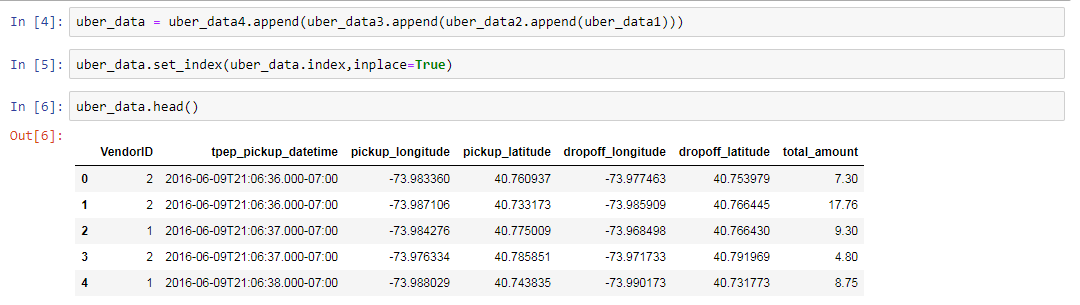
## IMPORT NECESSARY LIBRARIES

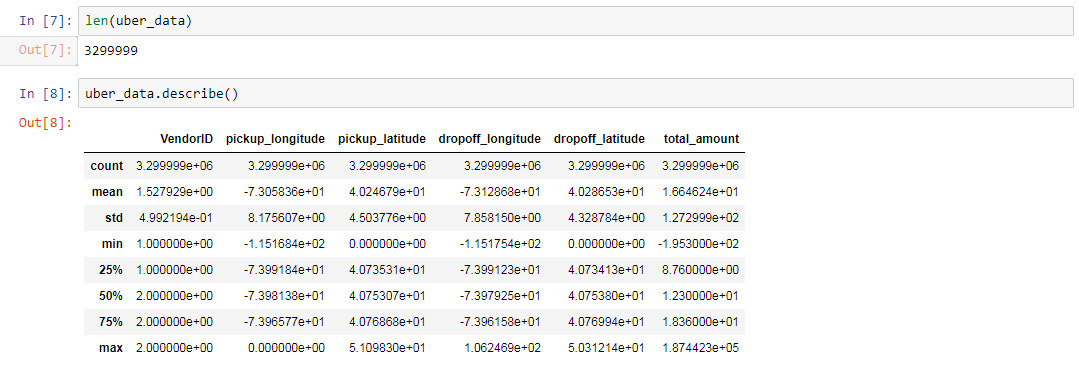


**IMPORTING DATA**

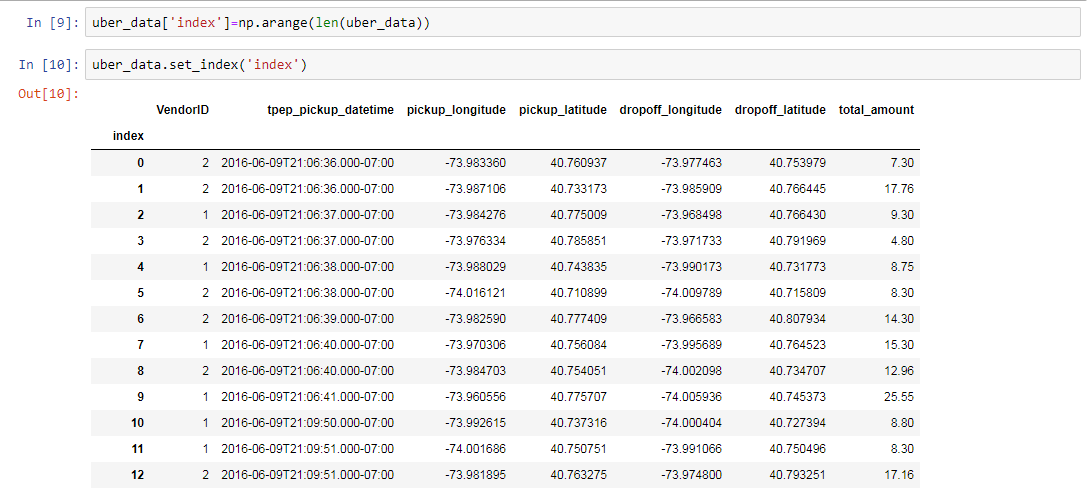


**APPENDING DATA**

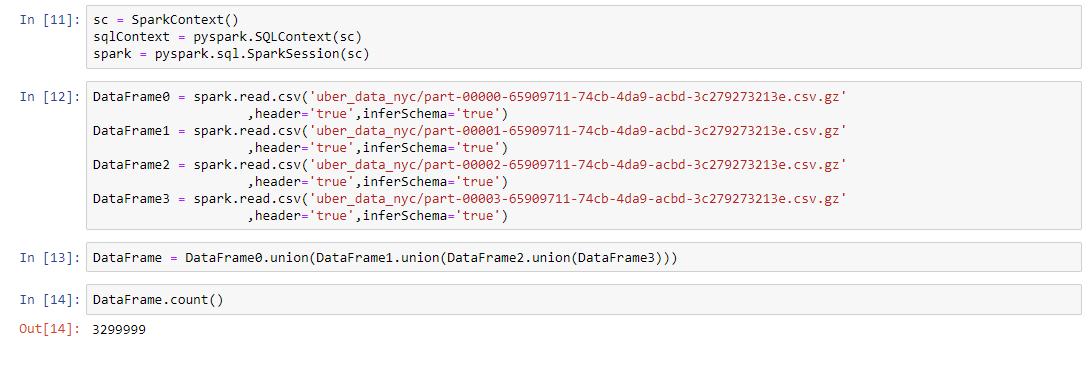




**CREATE INDEX**

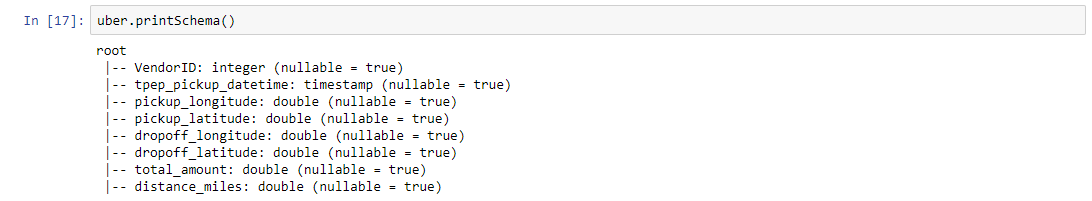


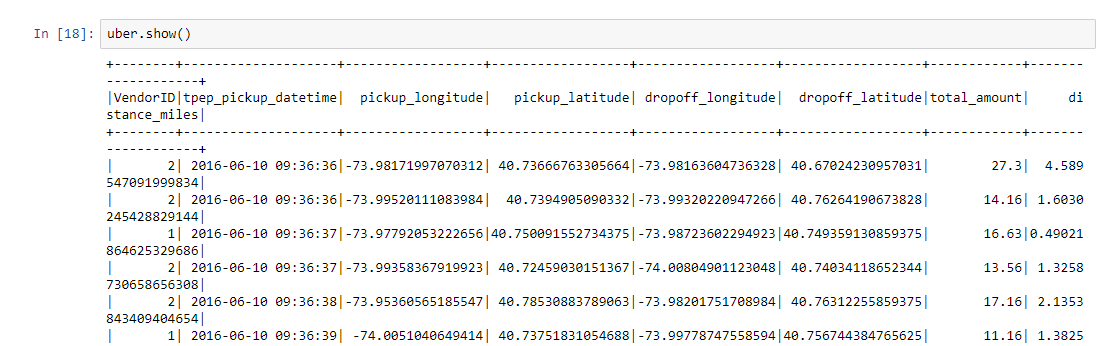
**IMPORT DATA INO SPARK**

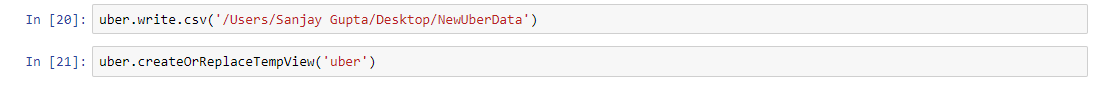


1. **USE THE HAVERSINE FORMULA TO CALCULATE THE DISTANCE FROM THE LATITUDES OF THE PICKUP AND DROP POINTS. INCLUDE THE FORMULA IN A PYTHON PACKAGE AND CREATE THE NECESSARY USER DEFINED FUNCTIONS IN SPARK. STORE THE DATA AS A CSV FILE.**

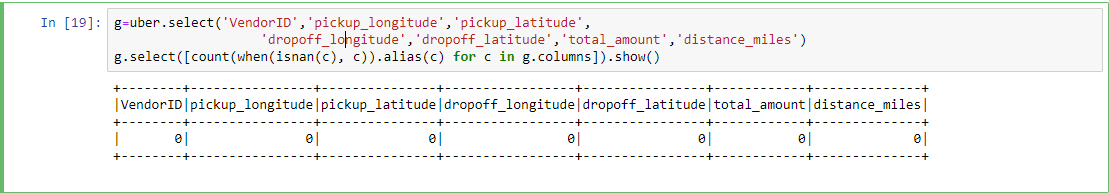








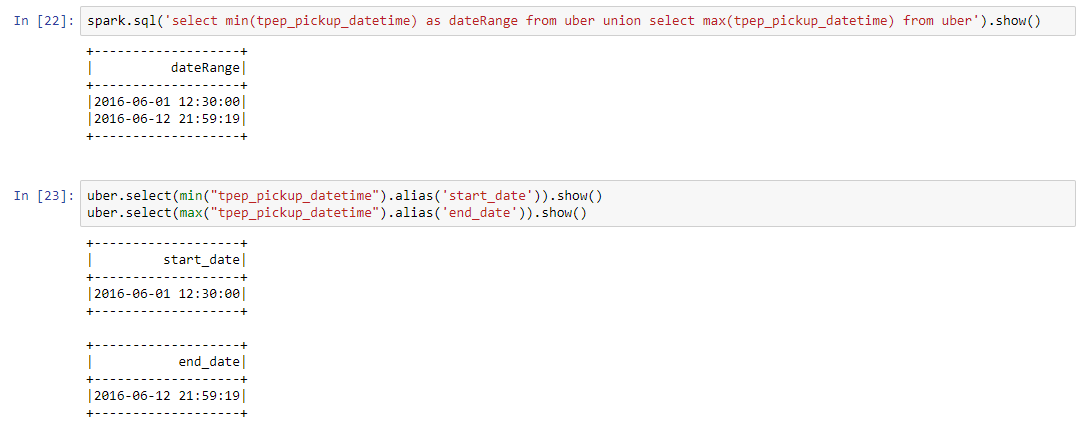
**CLEANING DATA BEFORE STORING**



## FIND THE DATE RANGE FROM THE NEWLY CREATED CSV FILE

**USING SQL QUERY:**

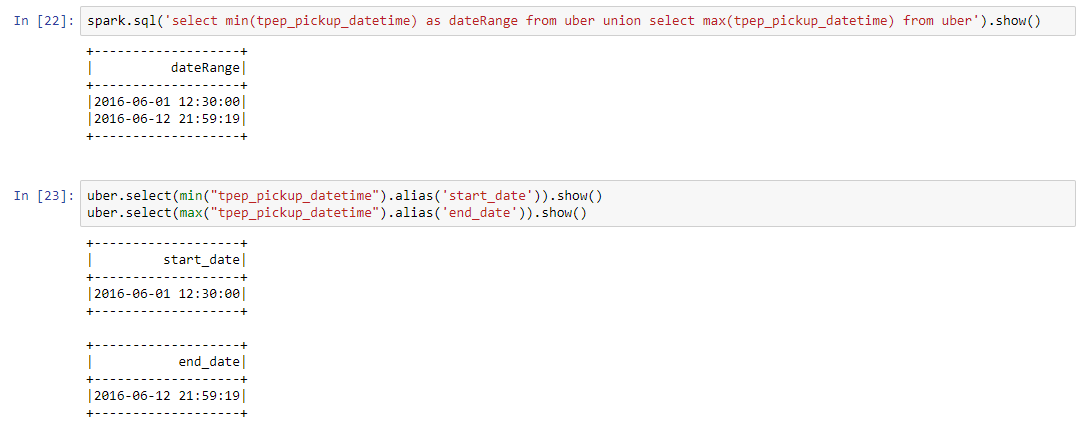
spark.sql('select min(tpep\_pickup\_datetime) as dateRange from uber union select max(tpep\_pickup\_datetime) from uber').show()



**USING DATAFRAME SQL FUNCTIONS:**

uber.select(min("tpep\_pickup\_datetime").alias('start\_date')).show()

uber.select(max("tpep\_pickup\_datetime").alias('end\_date')).show()



## AGGREGATE DATA BASED ON WEEK TO DEPICT WEEKLY DISTANCE OF TRAVEL

**USING SQL QUERY:**

spark.sql("select weekofyear(tpep\_pickup\_datetime) as week,sum(distance\_miles) as total\_distance from uber group by weekofyear(tpep\_pickup\_datetime)").show()

## 

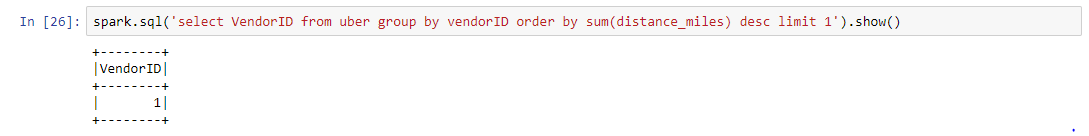
**USING DATAFRAME SQL FUNCTIONS:**

uber.groupBy(weekofyear("tpep\_pickup\_datetime").alias("date\_by\_week")).agg(sum("distance\_miles")).orderBy("date\_by\_week").show()

1. **IDENTIFY THE VENDOR COVERING THE HIGHEST MILES.**

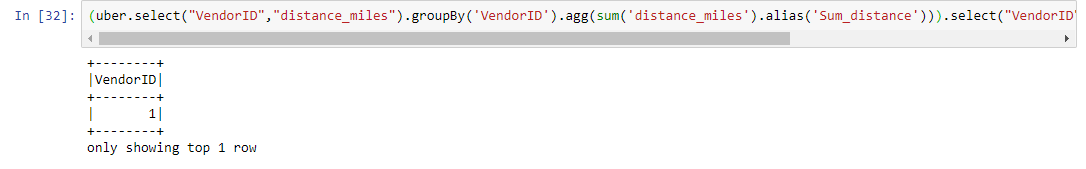
**USING SQL QUERY:**

spark.sql('select VendorID from uber group by vendorID order by sum(distance\_miles) desc limit 1').show()



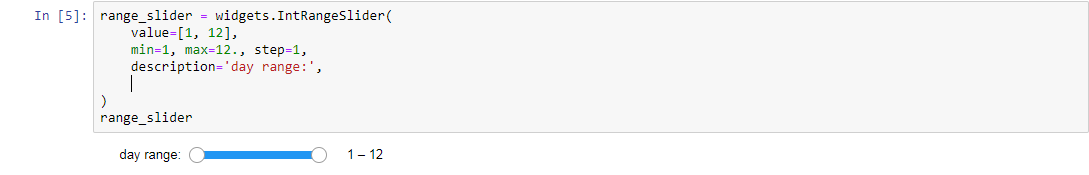
**USING DATAFRAME SQL FUNTCIONS:**

(uber.select("VendorID","distance\_miles").groupBy('VendorID').agg(sum('distance\_miles').alias('Sum\_distance'))).select("VendorID").orderBy("Sum\_distance",ascending=False).show(1)



1. **CALCULATE THE TOTAL REVENUE MADE OVER A PERIOD OF CHOICE**

**CREATING A SLIDER:**

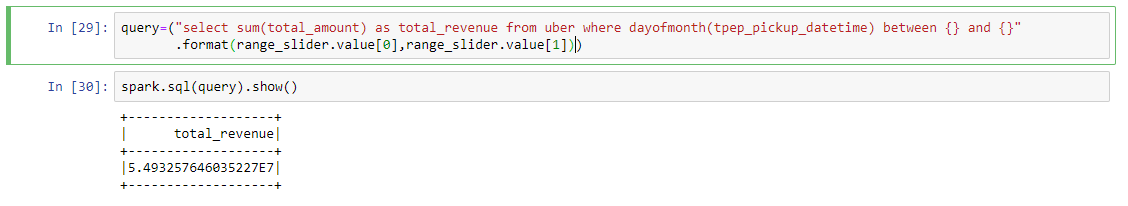


**USING SQL QUERY:**

query=("select sum(total\_amount) as total\_revenue from uber where dayofmonth(tpep\_pickup\_datetime) between {} and {}"

.format(range\_slider.value[0],range\_slider.value[1]))

spark.sql(query).show()



**USING DATAFRAME SQL FUNCTIONS:**

(uber.filter((uber["tpep\_pickup\_datetime"]>="2016-06-10 04:06:40") & (uber["tpep\_pickup\_datetime"]<="2016-06-12 16:29:14"))).agg(sum('total\_amount').alias('total revenue between 2016-06-10 04:06:40 & 2016-06-12 16:29:14')).show()

