**Screenshots and code**

**Databricks**

crimeDataNewYorkDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-New-York-2016.parquet")

crimeDataLosAngelesDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Los-Angeles-2016.parquet")

crimeDataPhiladelphiaDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Philadelphia-2016.parquet")

crimeDataDallasDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Dallas-2016.parquet")

crimeDataChicagoDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Chicago-2016.parquet")

crimeDataBostonDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Boston-2016.parquet")

crimeDataBostonDF.createOrReplaceTempView("boston")

crimeDataLosAngelesDF.createOrReplaceTempView("losangeles")

crimeDataPhiladelphiaDF.createOrReplaceTempView("philadelphia")

crimeDataChicagoDF.createOrReplaceTempView("chicago")

crimeDataDallasDF.createOrReplaceTempView("dallas")

crimeDataNewYorkDF.createOrReplaceTempView("newyork")

1.



**Code:**

%sql

select bos.rows + nyc.rows + la.rows + phil.rows + chi.rows + dal.rows

as total\_crimes, bos.rows as Boston, nyc.rows as NewYork, la.rows as LosAngeles, phil.rows as Philadelphia, chi.rows as Chicago, dal.rows as Dallas

from (

select count(distinct(INCIDENT\_NUMBER,OFFENSE\_CODE)) as rows

from boston

) as bos

cross join (

select count(distinct(complaintNumber)) as rows

from newyork

) as nyc

cross join (

select count(distinct(id)) as rows

from losangeles

) as la

cross join (

select count(distinct(unique\_id)) as rows

from philadelphia

) as phil

cross join (

select count(distinct(id)) as rows

from chicago

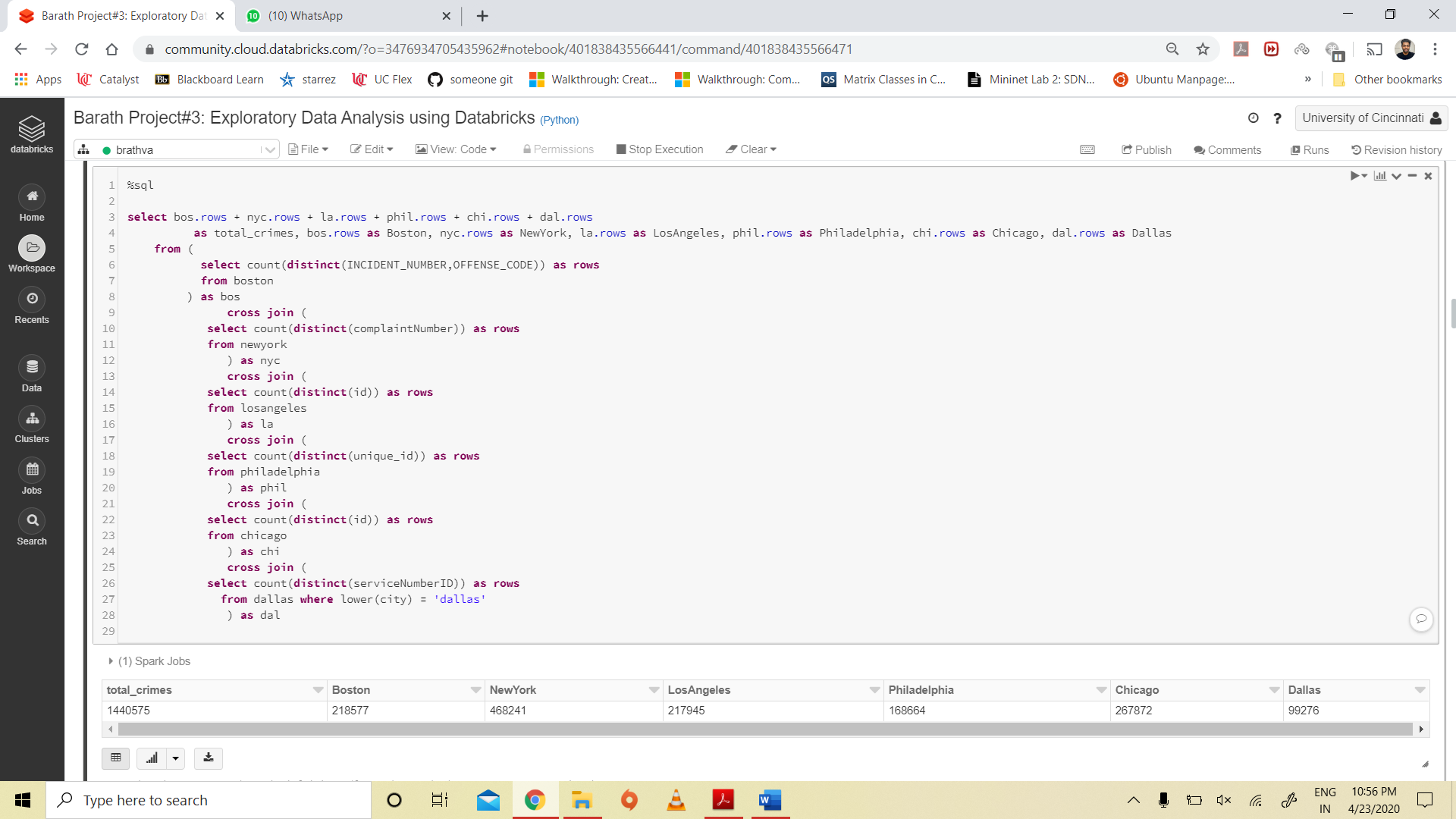
) as chi

cross join (

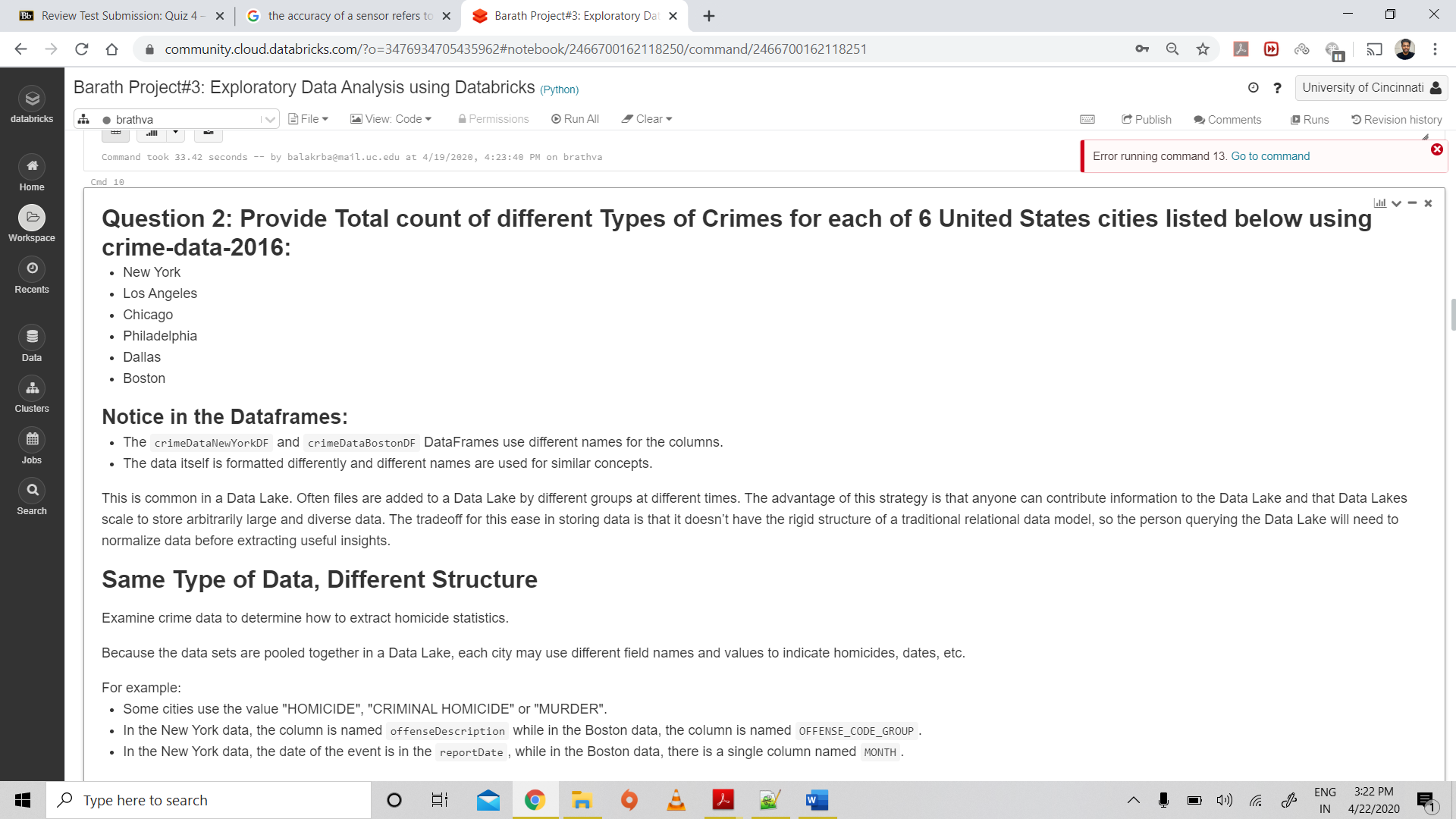
select count(distinct(serviceNumberID)) as rows

from dallas where lower(city) = 'dallas'

) as dal



**2.**



**Code:**

%sql

(select 'BOSTON' as CITY, count(distinct(OFFENSE\_CODE\_GROUP)) as total\_count from boston)

union

(select 'NEWYORK' as CITY, count(distinct(offenseDescription)) as total\_count from newyork)

union

(select 'PHILADELPHIA' as CITY, count(distinct(text\_general\_code)) as total\_count from philadelphia)

union

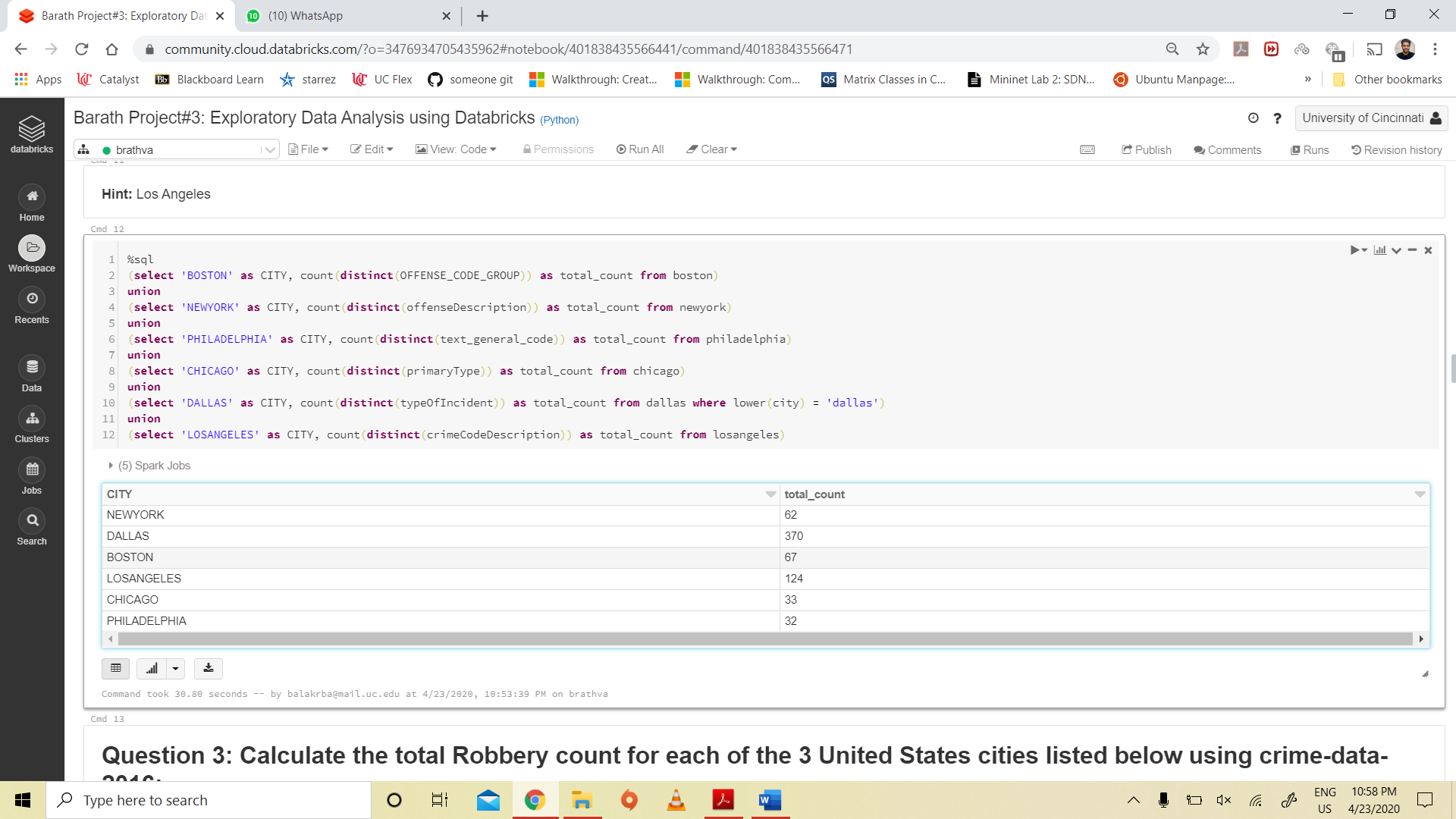
(select 'CHICAGO' as CITY, count(distinct(primaryType)) as total\_count from chicago)

union

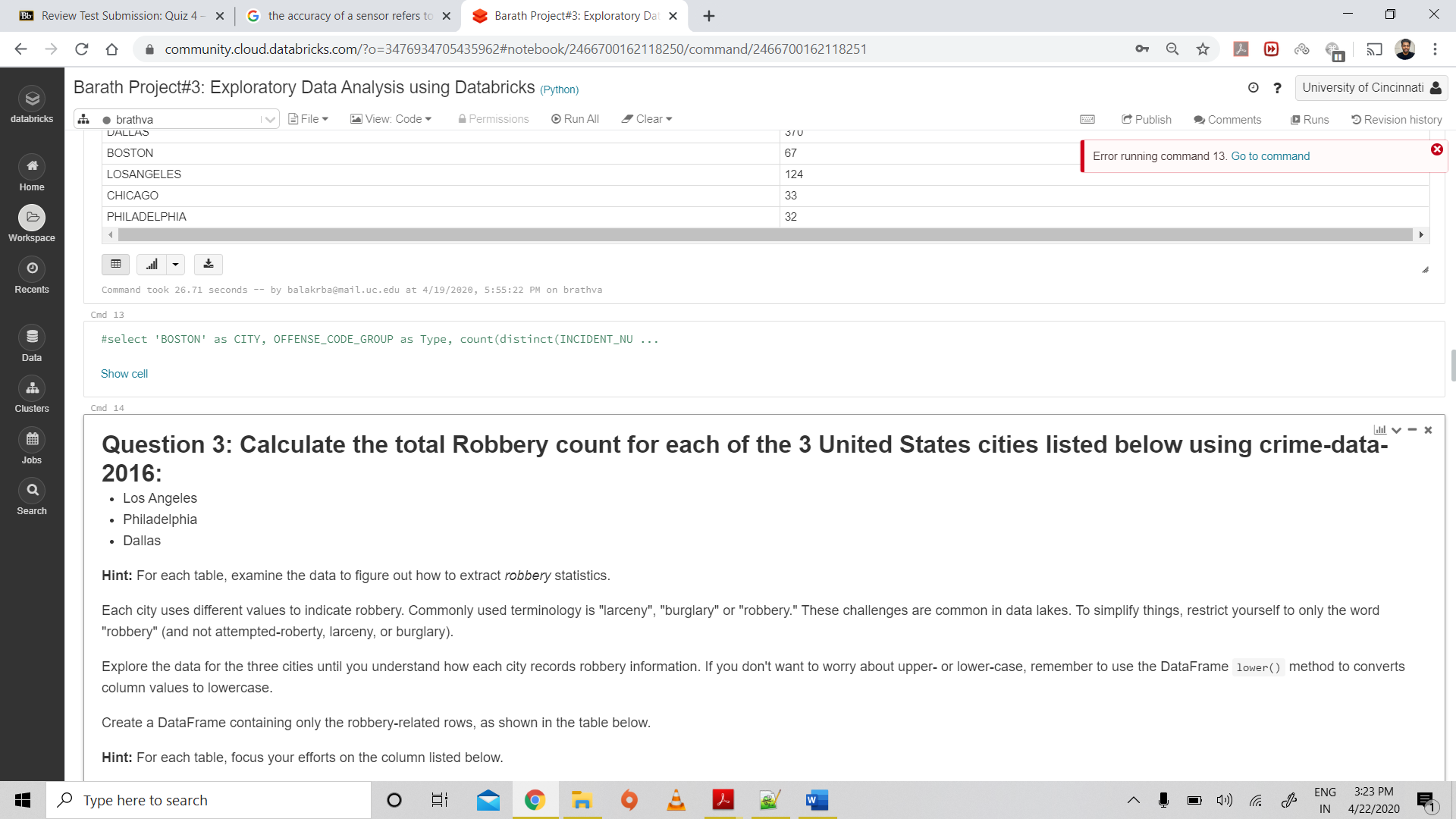
(select 'DALLAS' as CITY, count(distinct(typeOfIncident)) as total\_count from dallas where lower(city) = 'dallas')

union

(select 'LOSANGELES' as CITY, count(distinct(crimeCodeDescription)) as total\_count from losangeles)



**3.**



**Code:**

%sql

select la.rows + phil.rows + dal.rows

as total\_robbery, la.rows as LosAngeles, phil.rows as Philadelphia, dal.rows as Dallas

from ( select count(distinct(id)) as rows

from losangeles where lower(crimeCodeDescription) = 'robbery'

) as la

cross join (

select count(distinct(unique\_id)) as rows

from philadelphia where lower(ucr\_general\_description) = 'robbery'

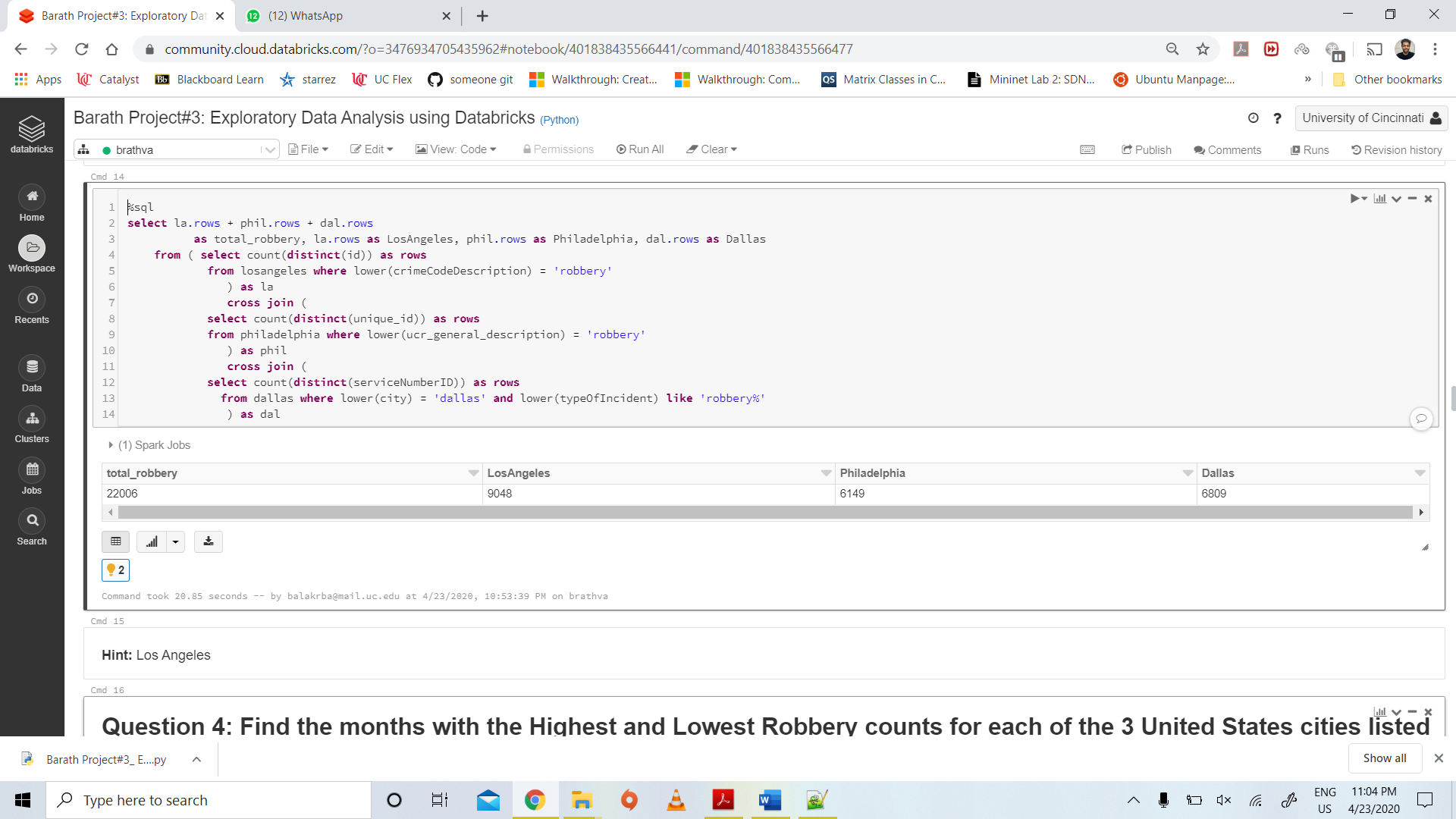
) as phil

cross join (

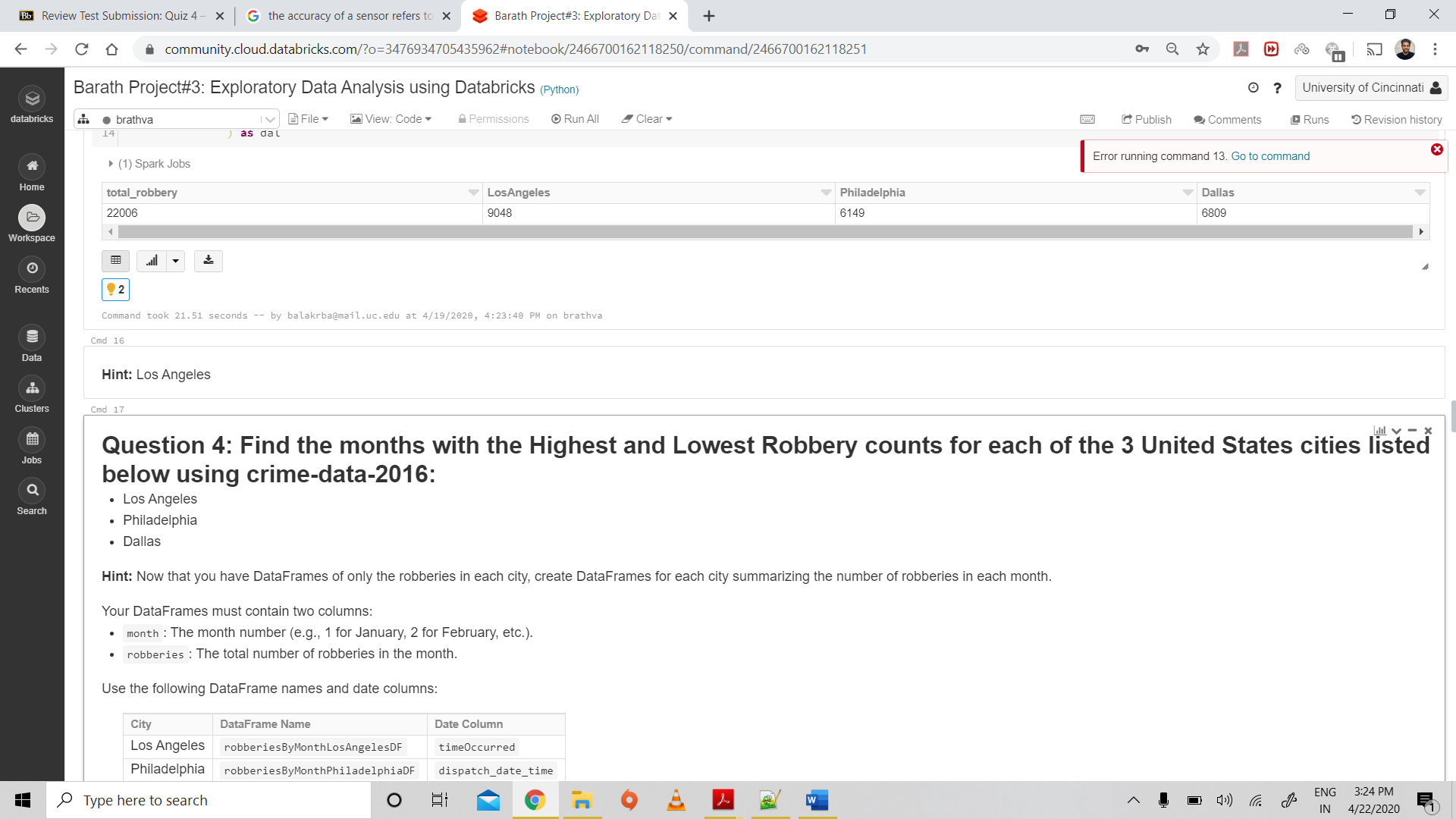
select count(distinct(serviceNumberID)) as rows

from dallas where lower(city) = 'dallas' and lower(typeOfIncident) like 'robbery%'

) as dal



**4.**



**Code:**

%sql

(select 'LOSANGELES\_MAX' as CITY, right(left(timeOccurred,7),2) as month, count(distinct(id)) as count from losangeles

where lower(crimeCodeDescription) = 'robbery' group by month order by count desc limit 1 )

union (

select 'LOSANGELES\_MIN' as CITY, right(left(timeOccurred,7),2) as month, count(distinct(id)) as count from losangeles

where lower(crimeCodeDescription) ='robbery' group by month order by count asc limit 1 )

union (

select 'PHILADELPHIA\_MAX' as CITY, right(left(dispatch\_date\_time,7),2) as month, count(distinct(unique\_id)) as count from philadelphia

where lower(ucr\_general\_description) = 'robbery' group by month order by count desc limit 1 )

union (

select 'PHILADELPHIA\_MIN' as CITY, right(left(dispatch\_date\_time,7),2) as month, count(distinct(unique\_id)) as count from philadelphia

where lower(ucr\_general\_description) = 'robbery'group by month order by count asc limit 1 )

union (

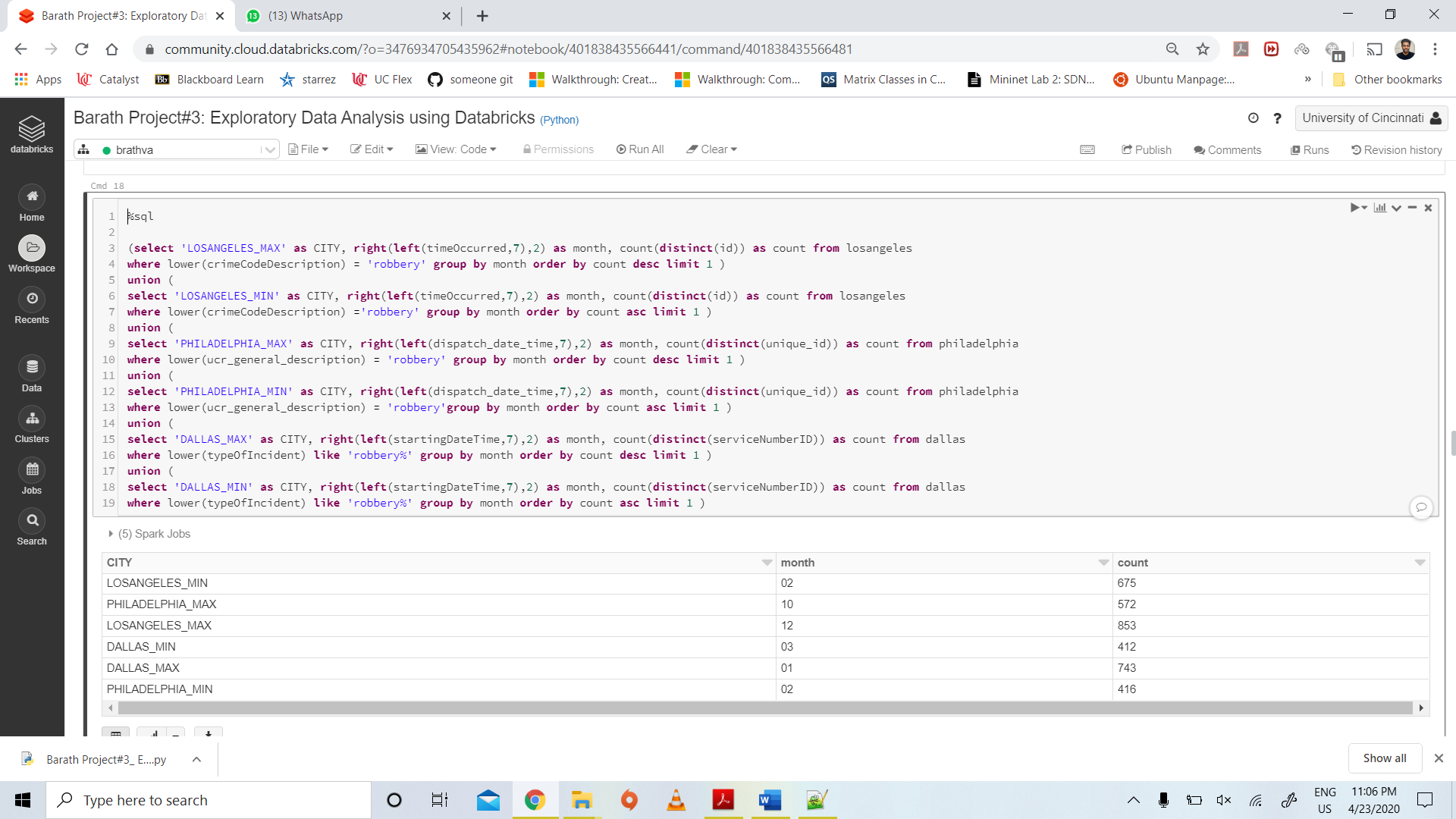
select 'DALLAS\_MAX' as CITY, right(left(startingDateTime,7),2) as month, count(distinct(serviceNumberID)) as count from dallas

where lower(typeOfIncident) like 'robbery%' group by month order by count desc limit 1 )

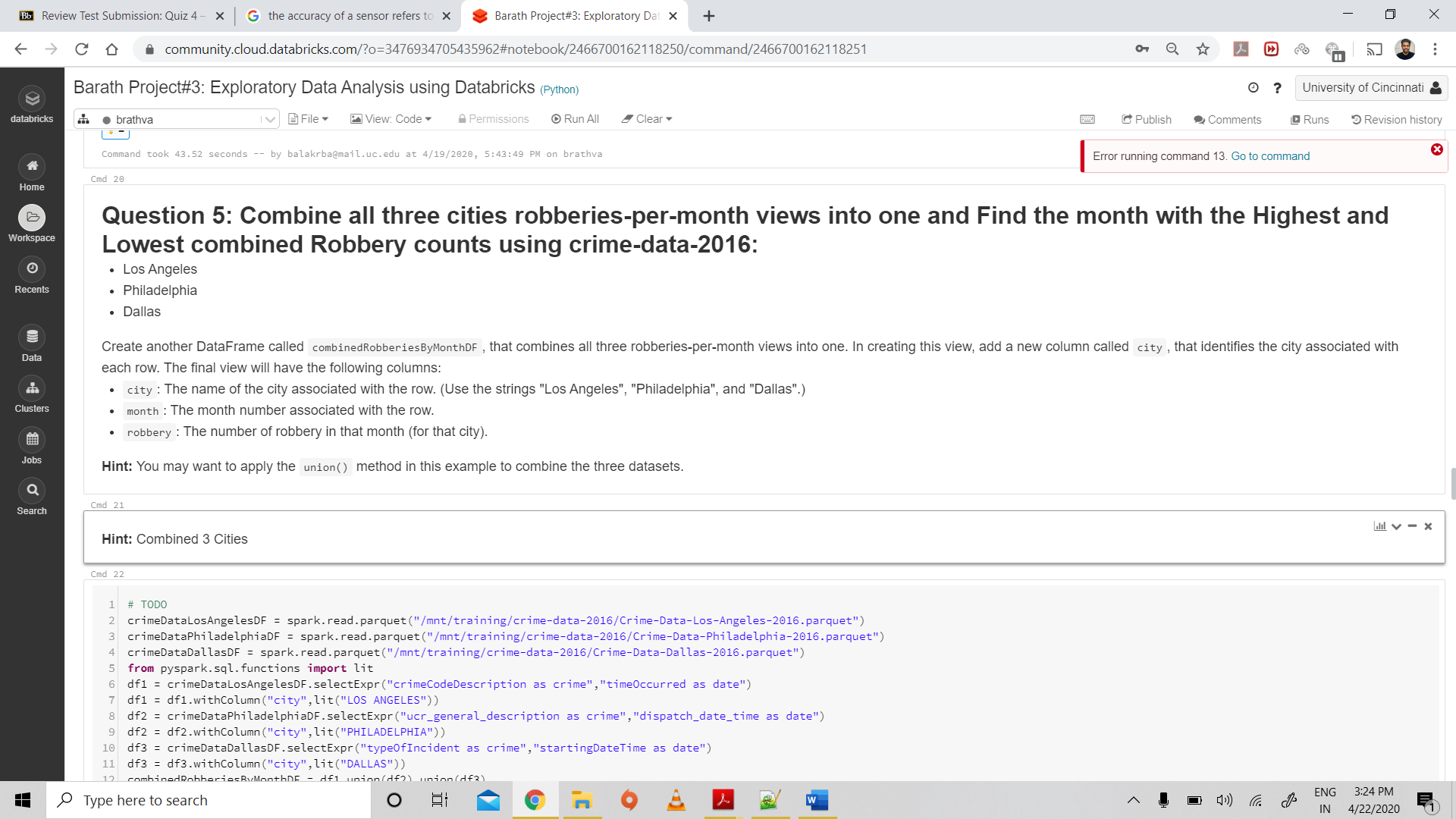
union (

select 'DALLAS\_MIN' as CITY, right(left(startingDateTime,7),2) as month, count(distinct(serviceNumberID)) as count from dallas

where lower(typeOfIncident) like 'robbery%' group by month order by count asc limit 1 )



**5.**



**Code:**

# TODO

crimeDataLosAngelesDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Los-Angeles-2016.parquet")

crimeDataPhiladelphiaDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Philadelphia-2016.parquet")

crimeDataDallasDF = spark.read.parquet("/mnt/training/crime-data-2016/Crime-Data-Dallas-2016.parquet")

from pyspark.sql.functions import lit

df1 = crimeDataLosAngelesDF.selectExpr("crimeCodeDescription as crime","timeOccurred as date")

df1 = df1.withColumn("city",lit("LOS ANGELES"))

df2 = crimeDataPhiladelphiaDF.selectExpr("ucr\_general\_description as crime","dispatch\_date\_time as date")

df2 = df2.withColumn("city",lit("PHILADELPHIA"))

df3 = crimeDataDallasDF.selectExpr("typeOfIncident as crime","startingDateTime as date")

df3 = df3.withColumn("city",lit("DALLAS"))

combinedRobberiesByMonthDF = df1.union(df2).union(df3)

combinedRobberiesByMonthDF.createOrReplaceTempView("combinedRobberies")

%sql

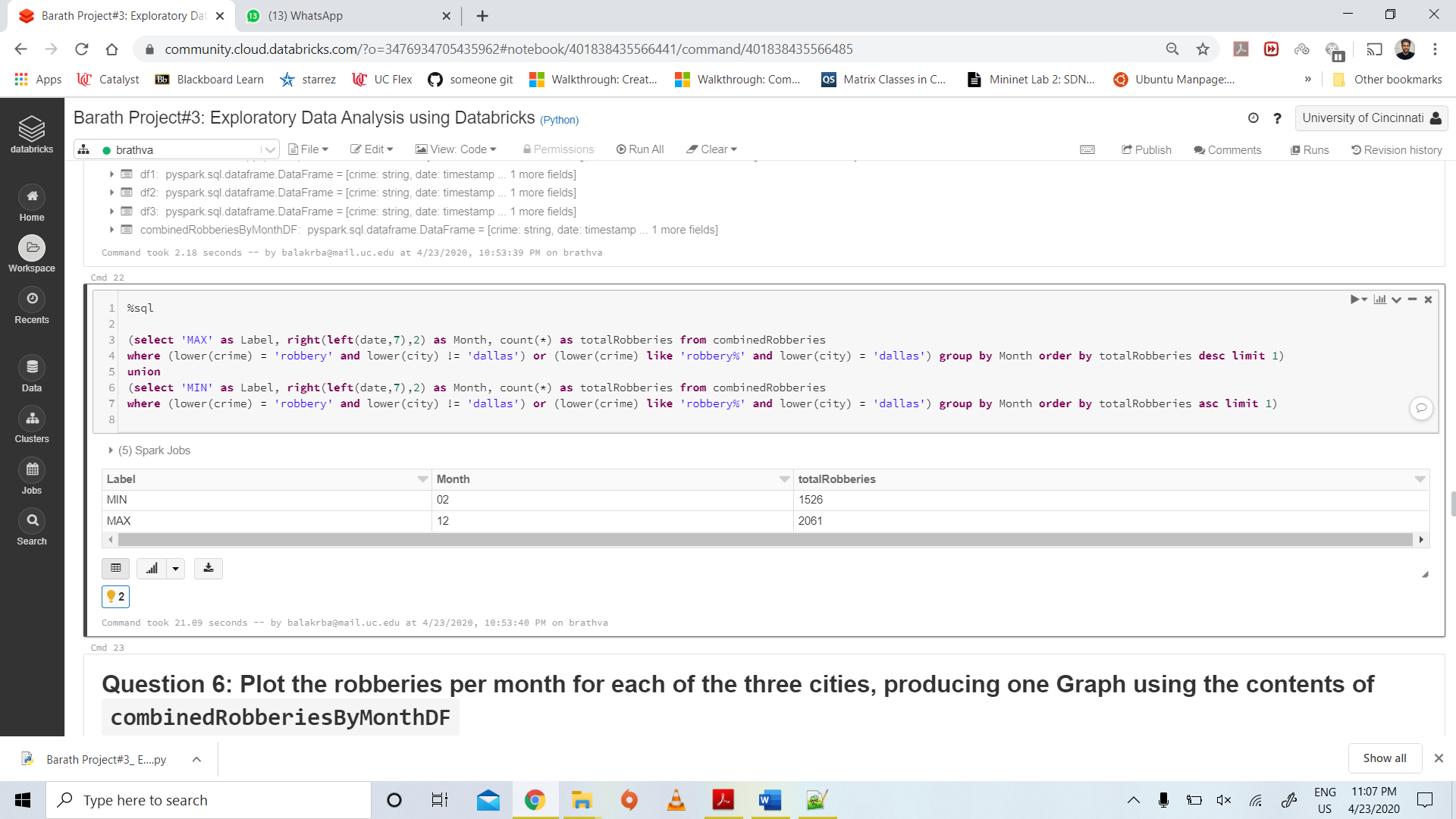
(select 'MAX' as Label, right(left(date,7),2) as Month, count(\*) as totalRobberies from combinedRobberies

where (lower(crime) = 'robbery' and lower(city) != 'dallas') or (lower(crime) like 'robbery%' and lower(city) = 'dallas') group by Month order by totalRobberies desc limit 1)

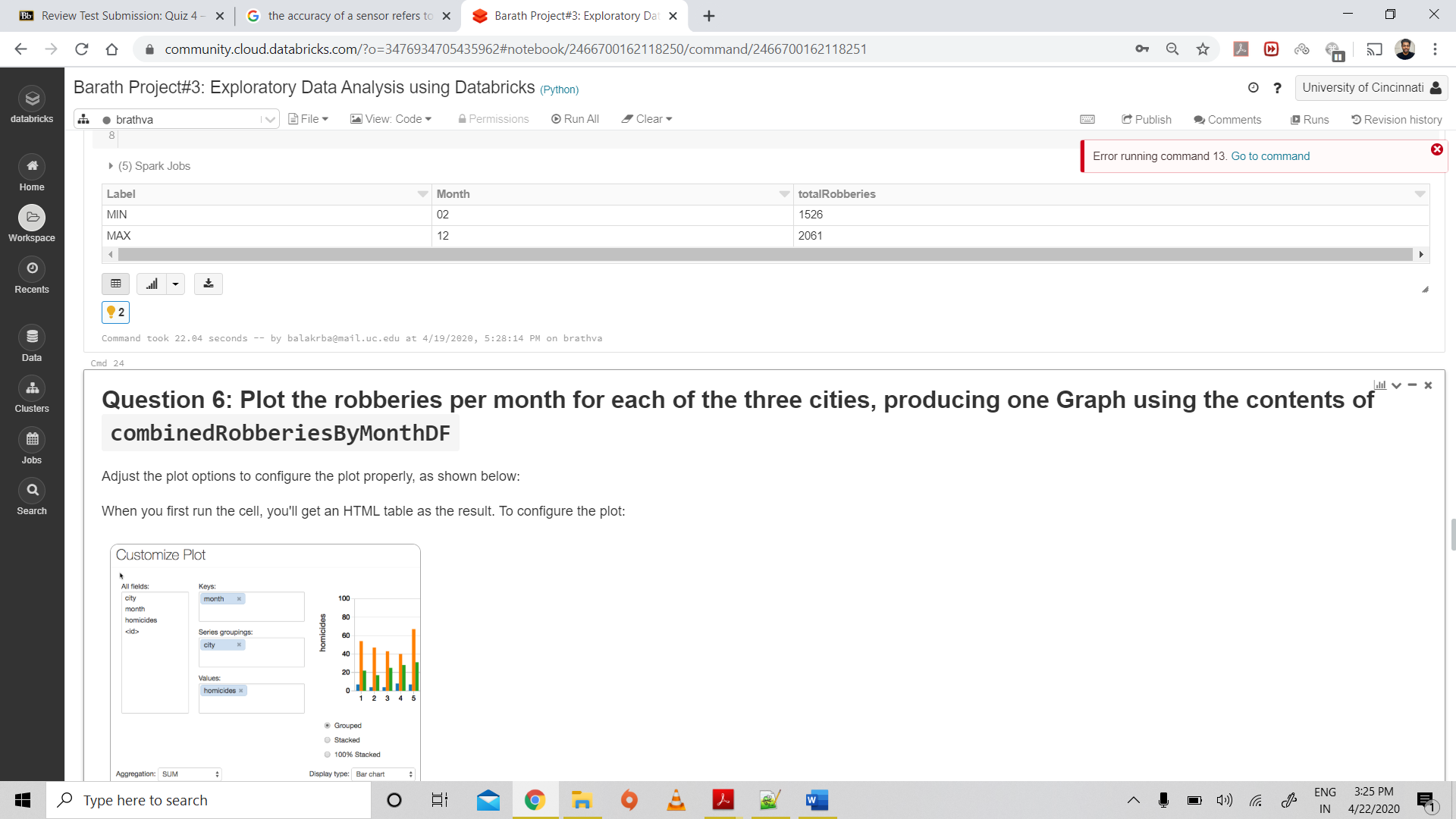
union

(select 'MIN' as Label, right(left(date,7),2) as Month, count(\*) as totalRobberies from combinedRobberies

where (lower(crime) = 'robbery' and lower(city) != 'dallas') or (lower(crime) like 'robbery%' and lower(city) = 'dallas') group by Month order by totalRobberies asc limit 1)



**6.**

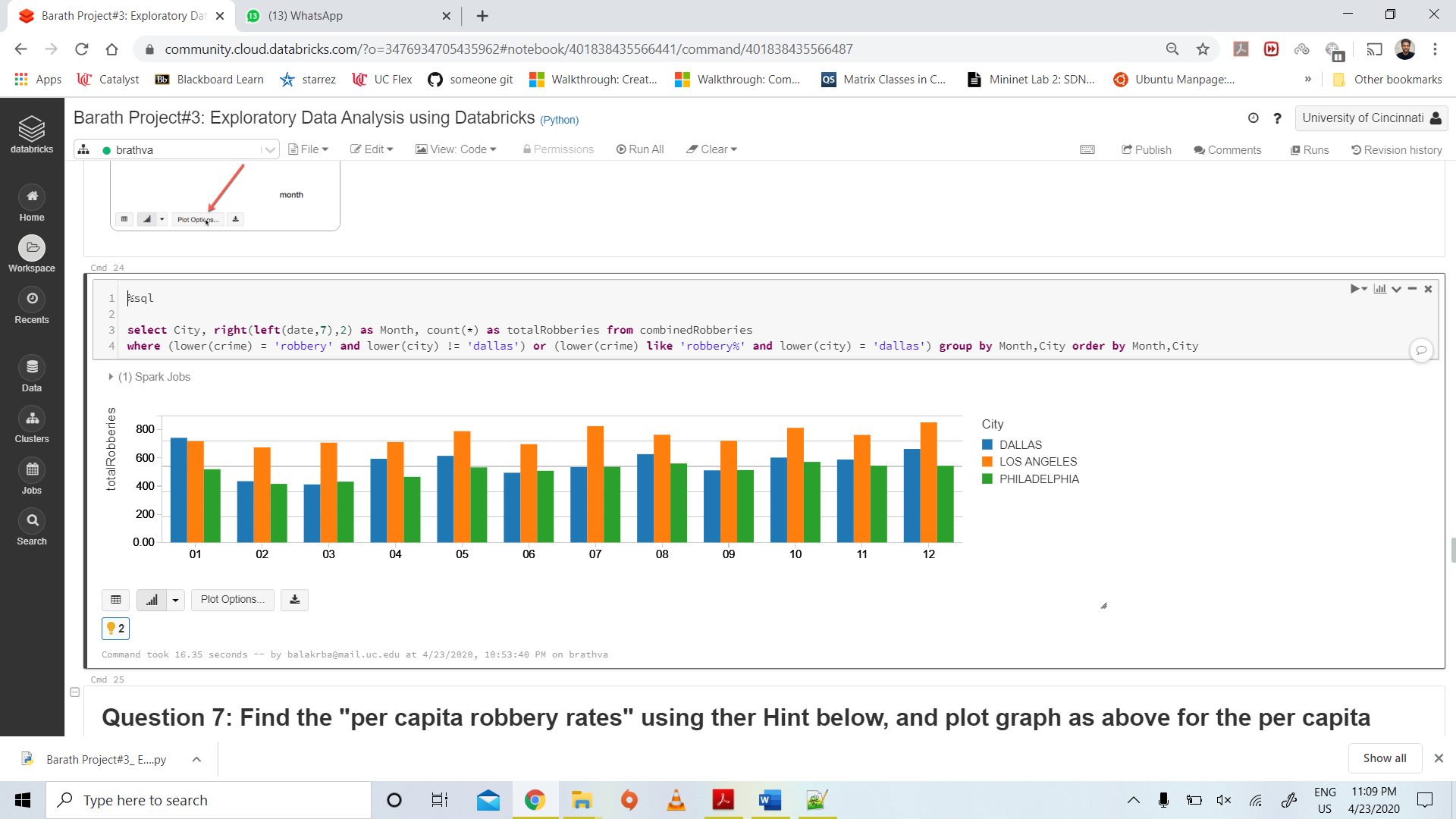


**Code:**

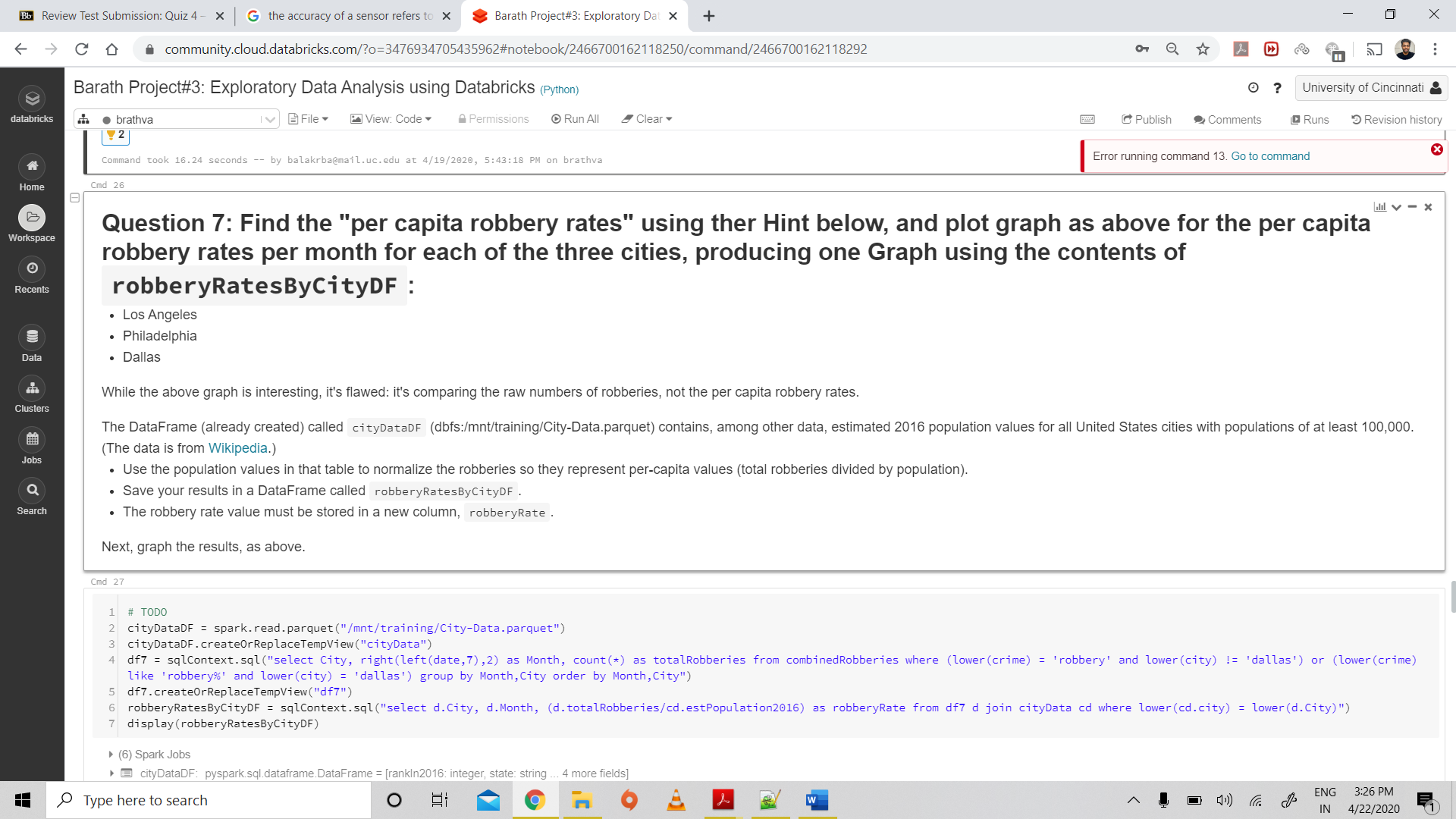
%sql

select City, right(left(date,7),2) as Month, count(\*) as totalRobberies from combinedRobberies

where (lower(crime) = 'robbery' and lower(city) != 'dallas') or (lower(crime) like 'robbery%' and lower(city) = 'dallas') group by Month,City order by Month,City



**7.**



**Code:**

# TODO

cityDataDF = spark.read.parquet("/mnt/training/City-Data.parquet")

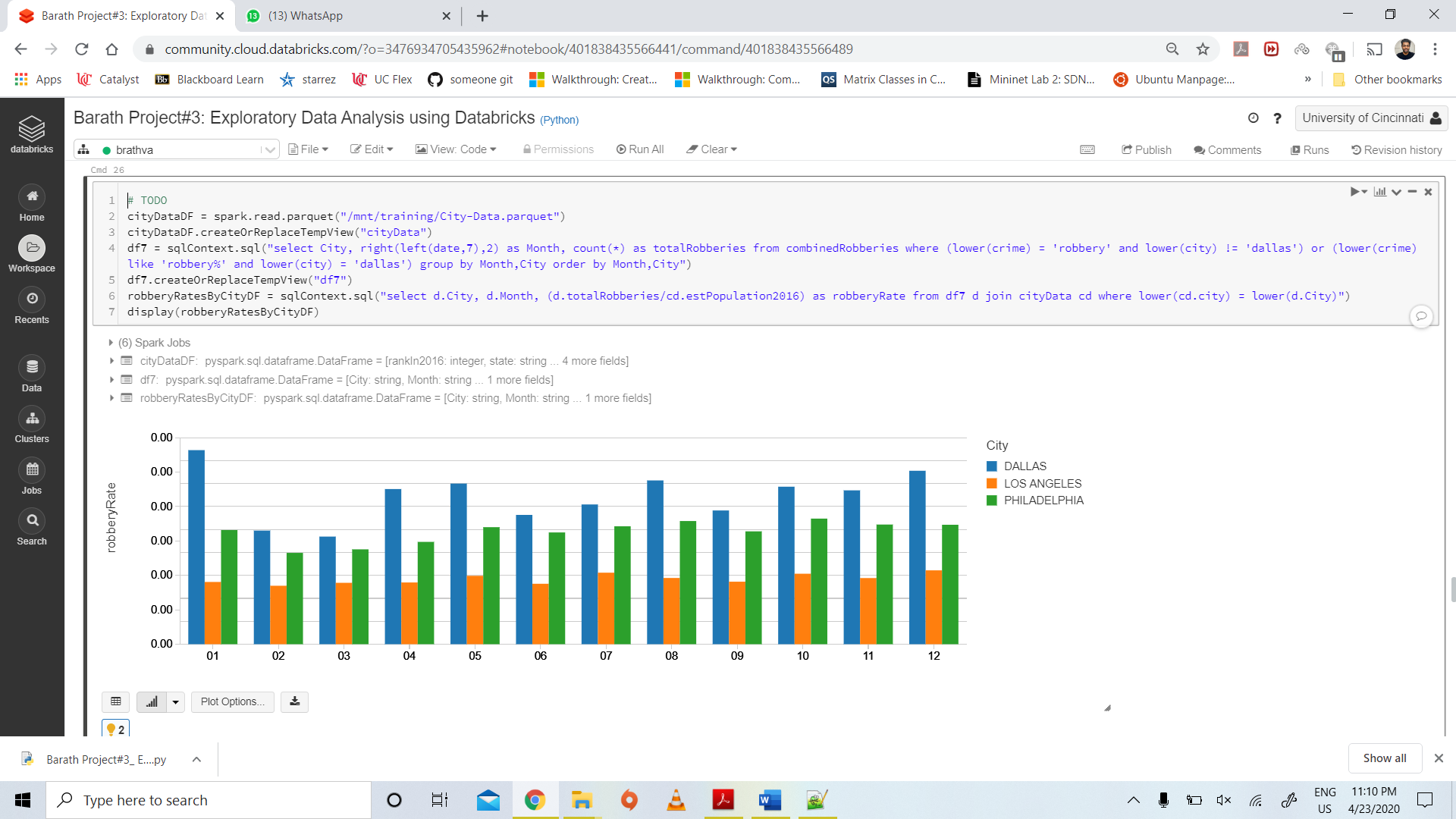
cityDataDF.createOrReplaceTempView("cityData")

df7 = sqlContext.sql("select City, right(left(date,7),2) as Month, count(\*) as totalRobberies from combinedRobberies where (lower(crime) = 'robbery' and lower(city) != 'dallas') or (lower(crime) like 'robbery%' and lower(city) = 'dallas') group by Month,City order by Month,City")

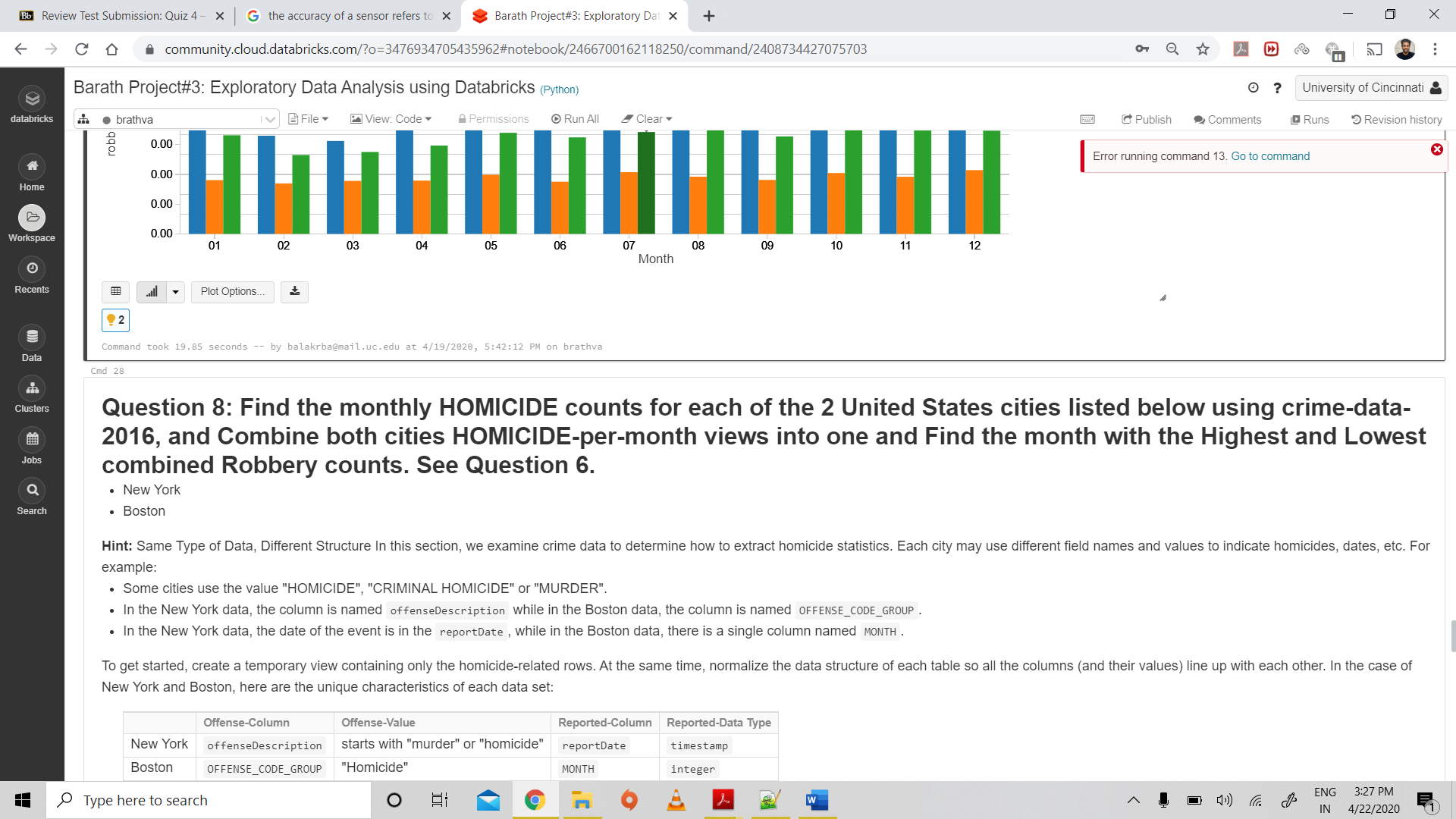
df7.createOrReplaceTempView("df7")

robberyRatesByCityDF = sqlContext.sql("select d.City, d.Month, (d.totalRobberies/cd.estPopulation2016) as robberyRate from df7 d join cityData cd where lower(cd.city) = lower(d.City)")

display(robberyRatesByCityDF)



**8.**



**Code (Monthly Homicide Counts):**

from pyspark.sql.types import IntegerType

from pyspark.sql.functions import col, split

from pyspark.sql.functions import lit

df8 = crimeDataNewYorkDF.selectExpr("offenseDescription as crime","reportDate as date")

df8 = df8.withColumn("month1", split(col("date"), "-").getItem(1))

df8 = df8.withColumn("month", df8["month1"].cast("integer"))

df8 = df8.drop("date","month1")

df8 = df8.withColumn("city",lit("NEW YORK"))

df9 = crimeDataBostonDF.selectExpr("OFFENSE\_CODE\_GROUP as crime","MONTH as month")

df9 = df9.withColumn("city",lit("BOSTON"))

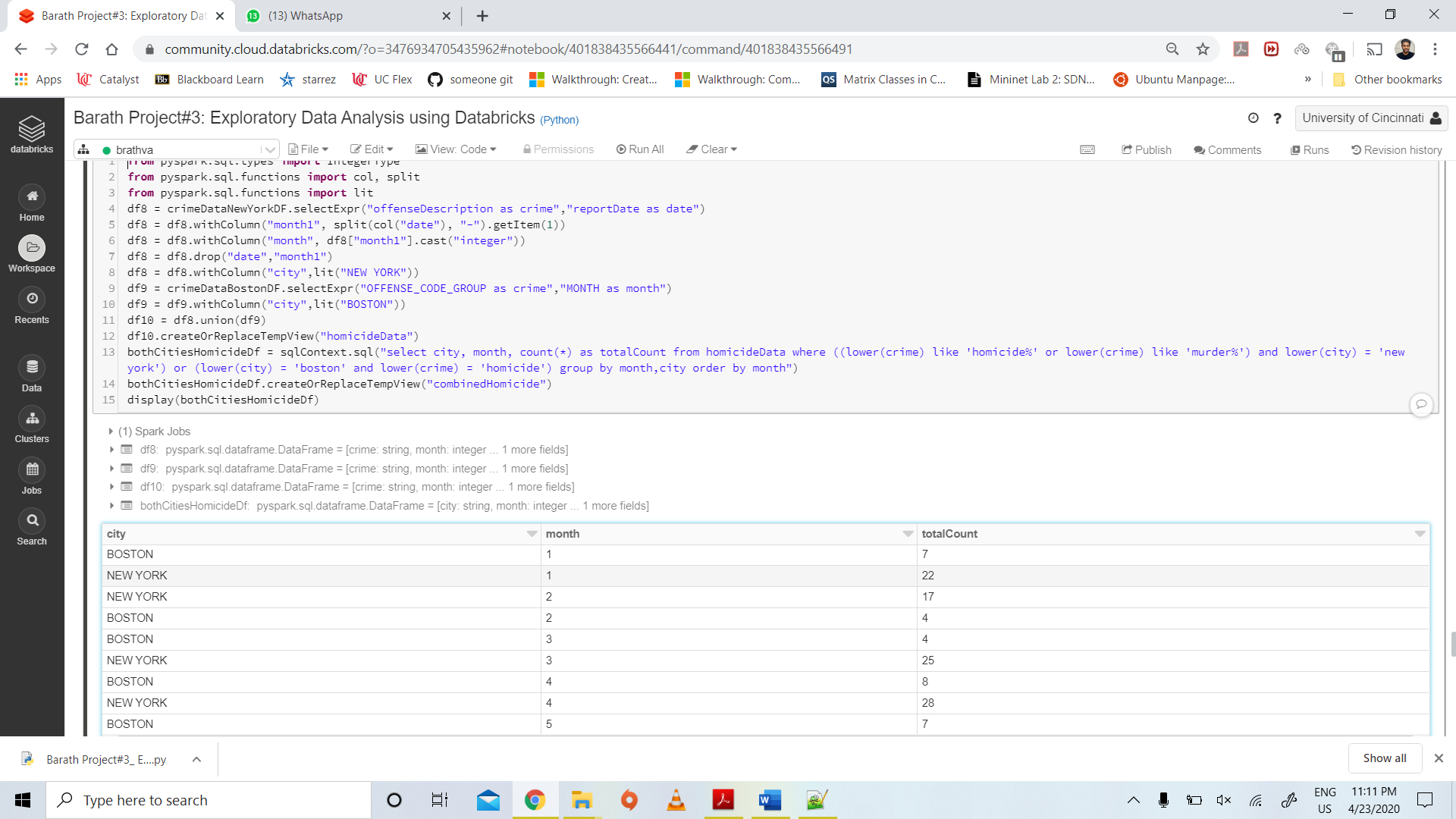
df10 = df8.union(df9)

df10.createOrReplaceTempView("homicideData")

bothCitiesHomicideDf = sqlContext.sql("select city, month, count(\*) as totalCount from homicideData where ((lower(crime) like 'homicide%' or lower(crime) like 'murder%') and lower(city) = 'new york') or (lower(city) = 'boston' and lower(crime) = 'homicide') group by month,city order by month")

bothCitiesHomicideDf.createOrReplaceTempView("combinedHomicide")

display(bothCitiesHomicideDf)



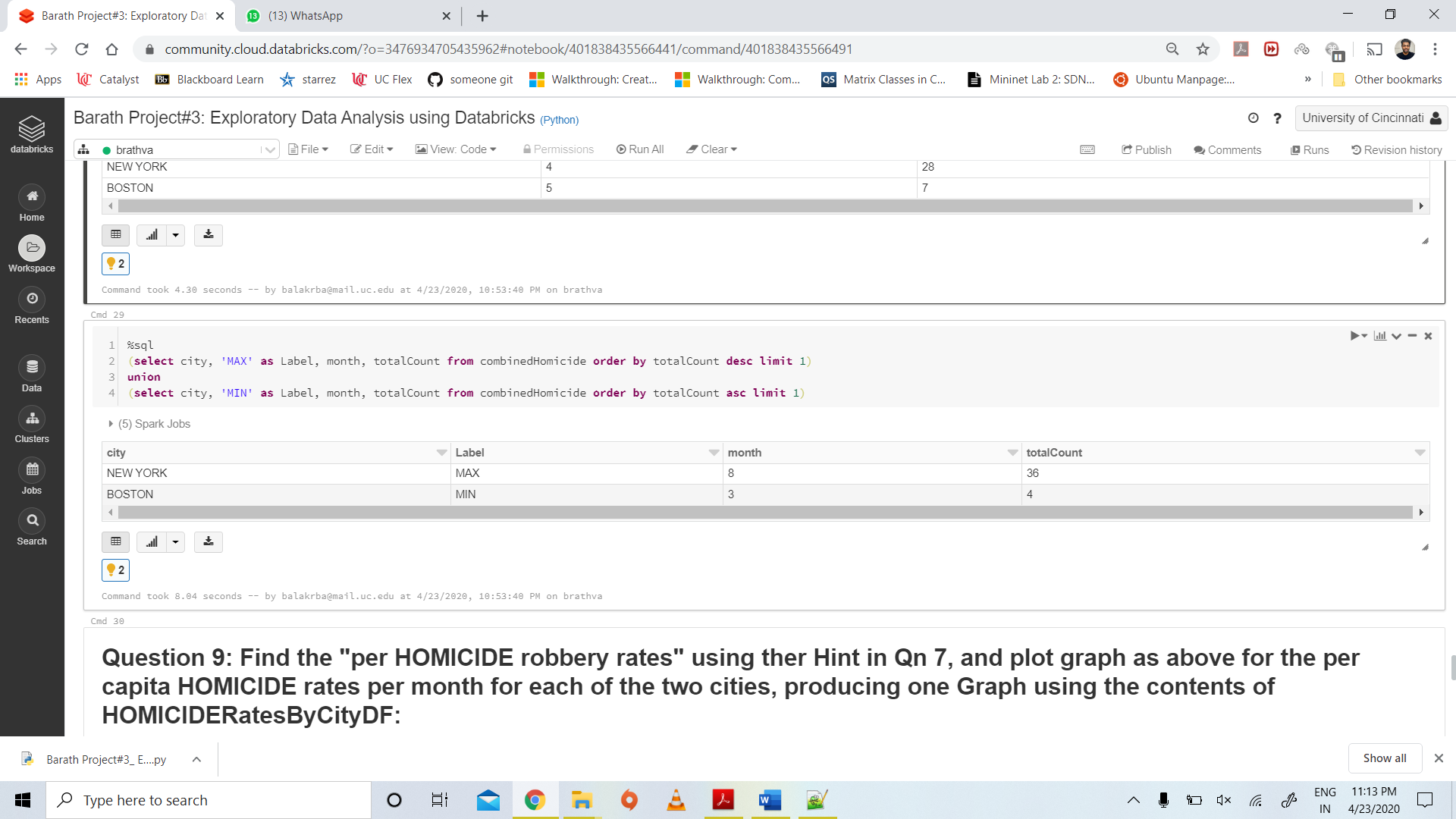
**Code (Highest and Lowest counts):**

%sql

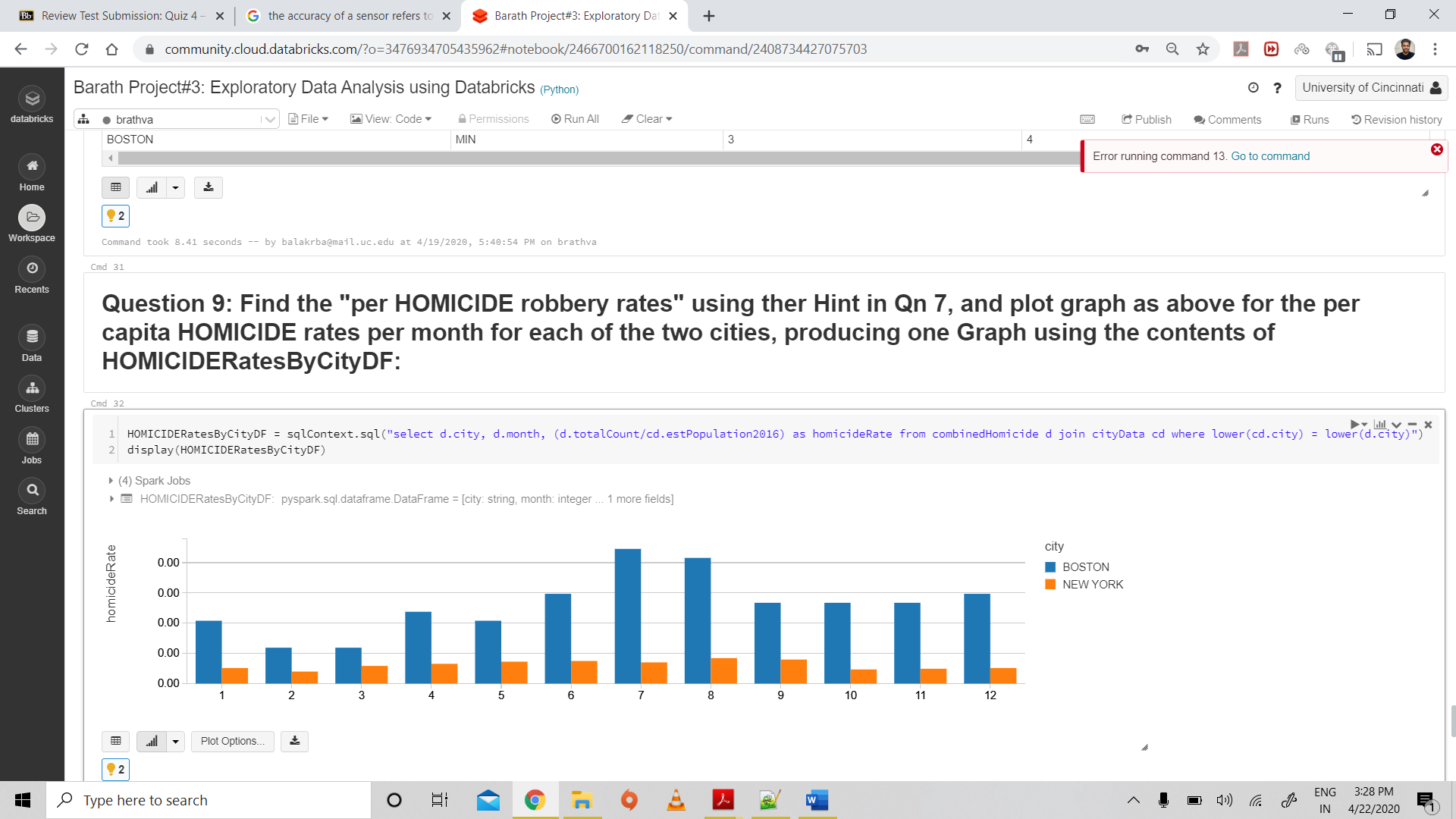
(select city, 'MAX' as Label, month, totalCount from combinedHomicide order by totalCount desc limit 1)

union

(select city, 'MIN' as Label, month, totalCount from combinedHomicide order by totalCount asc limit 1)



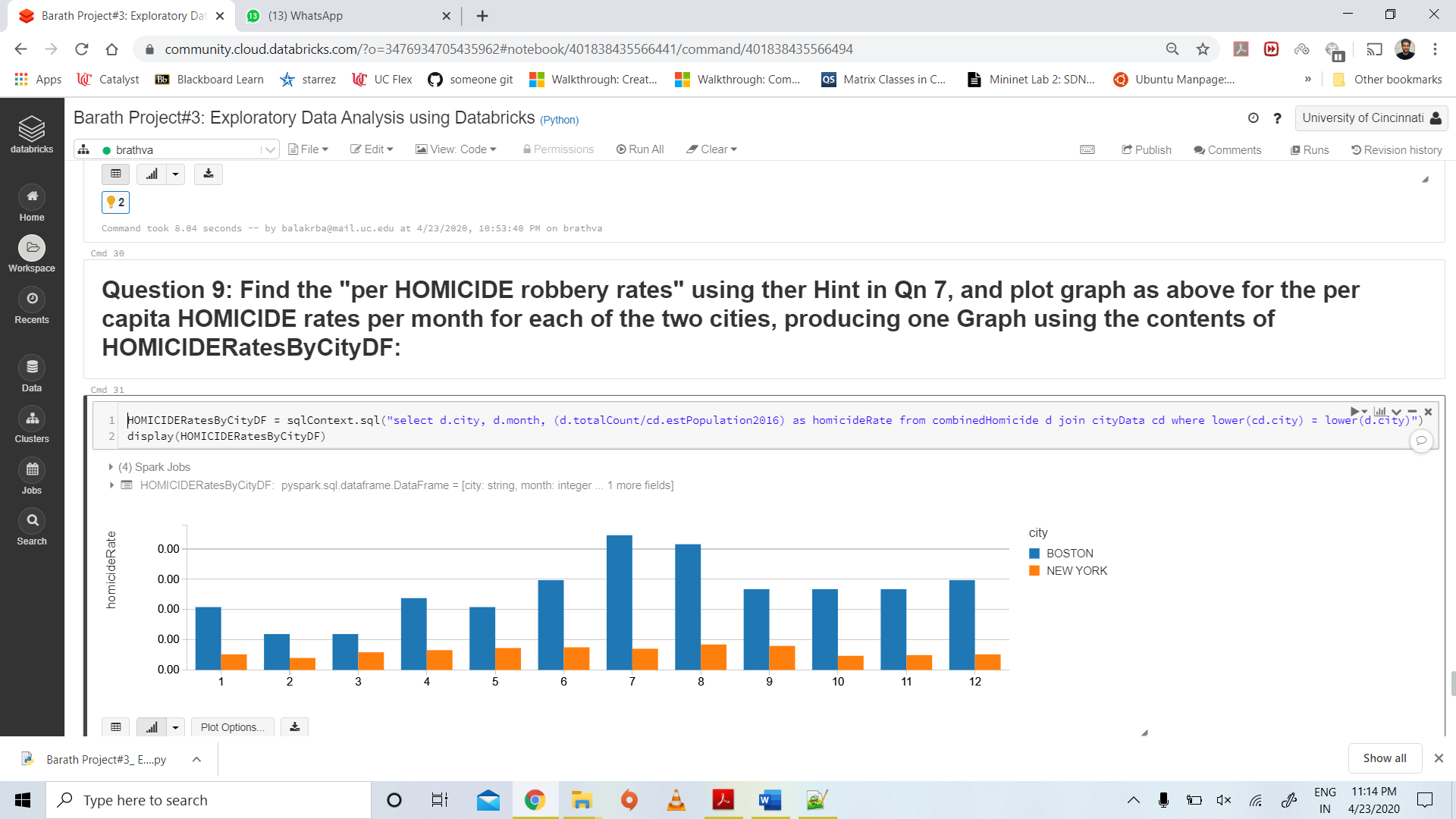
**9.**



**Code:**

HOMICIDERatesByCityDF = sqlContext.sql("select d.city, d.month, (d.totalCount/cd.estPopulation2016) as homicideRate from combinedHomicide d join cityData cd where lower(cd.city) = lower(d.city)")

display(HOMICIDERatesByCityDF)



10.

Code:

