Section 2.3

%pyspark

from decimal import \*

df = spark.read.csv("/data/listings.csv",header = "true")

total=df.select(df.host\_name,df.calculated\_host\_listings\_count).distinct().orderBy('host\_name').count()

d\_hn=df.select(df.host\_name,df.calculated\_host\_listings\_count).filter(df.calculated\_host\_listings\_count > 1).distinct().orderBy('host\_name')

d\_hn\_c = d\_hn.count()

p = round((Decimal(d\_hn\_c) / Decimal(total))\*100,2)

print(d\_hn\_c)

print(total)

print(p)

====================================================================================================

%pyspark

df.createOrReplaceTempView("listings")

sqlDF = spark.sql("SELECT \* FROM listings")

%sql

SELECT CONCAT(YEAR(last\_review), "-" ,MONTH(last\_review)) AS YearMonth, SUM(number\_of\_reviews )

FROM listings

Where last\_review >= '2013-10-21'

GROUP BY YearMonth

Order BY YearMonth

===============================================================================================================

%pyspark

import shutil

shutil.rmtree('TopFiveNeighbourhoods.parquet', ignore\_errors=True)

df = spark.sql("select neighbourhood,avg(price) AS Price from listings WHERE neighbourhood !regexp '^[1-9]+' Group by neighbourhood order by price Limit 5")

df.write.save("TopFiveNeighbourhoods.parquet")

parquetFile = spark.read.load("TopFiveNeighbourhoods.parquet")

df = spark.sql("SELECT count(name) As NumberofListings FROM listings where availability\_365 = '365' AND neighbourhood in (select neighbourhood from parquet.`TopFiveNeighbourhoods.parquet`)")

df.show()