databricksAssociationRules

```
import org.apache.spark
```

```
!pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.8.1-cp38-cp38-manylinux1_x86_64.whl (371 kB)
     |████████████████████████████████| 371 kB 5.4 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.6.1 in /databricks/python3/lib/python3.
8/site-packages (from wordcloud) (1.19.2)
Requirement already satisfied: pillow in /databricks/python3/lib/python3.8/site
-packages (from wordcloud) (8.2.0)
Requirement already satisfied: matplotlib in /databricks/python3/lib/python3.8/
site-packages (from wordcloud) (3.4.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /databricks/python3/lib/pyt
hon3.8/site-packages (from matplotlib->wordcloud) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /databricks/python3/lib/python3.
8/site-packages (from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: pyparsing>=2.2.1 in /databricks/python3/lib/pyth
on3.8/site-packages (from matplotlib->wordcloud) (2.4.7)
Requirement already satisfied: python-dateutil>=2.7 in /databricks/python3/lib/
python3.8/site-packages (from matplotlib->wordcloud) (2.8.1)
Requirement already satisfied: six in /databricks/python3/lib/python3.8/site-pa
ckages (from cycler>=0.10->matplotlib->wordcloud) (1.15.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.8.1
```

```
from pyspark.sql import SparkSession
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from pyspark.sql.functions import collect_set, col, count
from pyspark.ml.fpm import FPGrowth
```

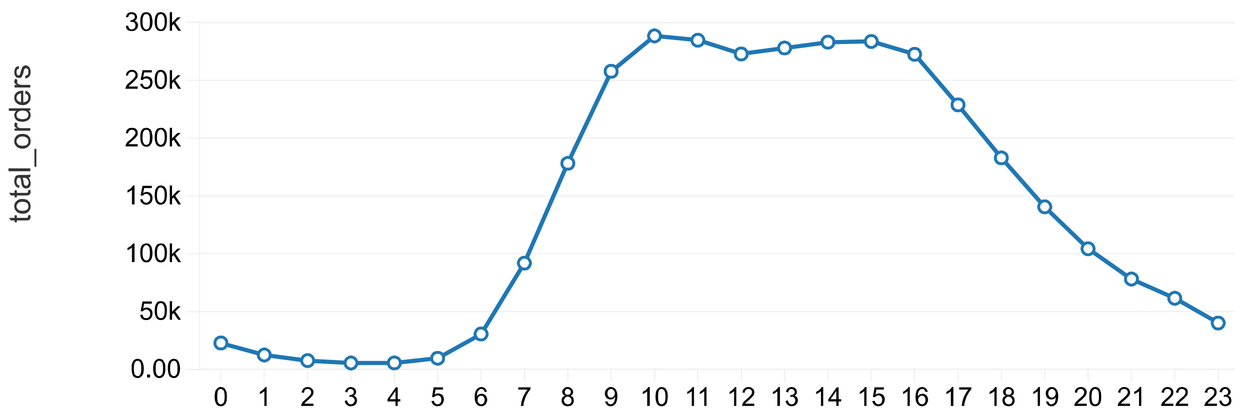
```
orders = spark.read.csv("/FileStore/tables/orders.csv", header="true",
inferSchema="true")
```

```
order_products_prior =
spark.read.csv("/FileStore/tables/order_products__prior.csv", header="true",
inferSchema="true")
```

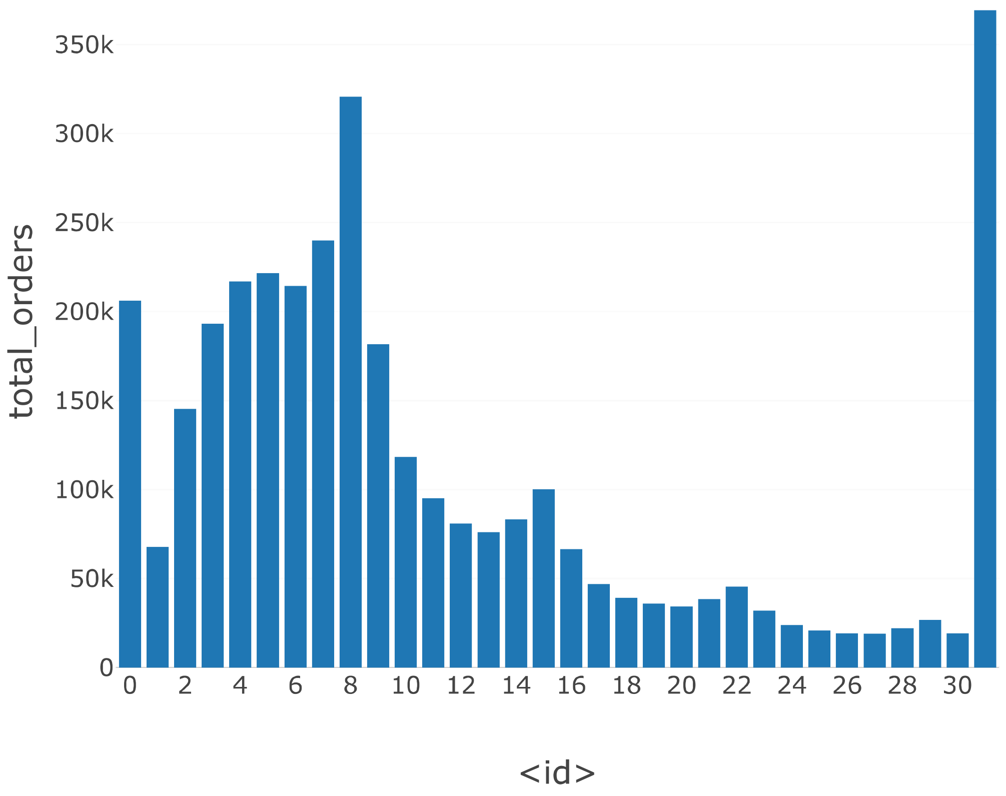
```
urlA = 'https://drive.google.com/file/d/1W8bNivEj7H0WXqZx1X83fQEYz4A3XadY/view?
usp=sharing'
urlA2 = 'https://drive.google.com/uc?id=' + urlA.split('/')[-2]
aislesPD = pd.read_csv(urlA2)
aisles = spark.createDataFrame(aislesPD)


urlD = 'https://drive.google.com/file/d/1unatDL4jGx5CCHYN2Q9YnDjnq43AgtJp/view?
usp=sharing'
urlD2 = 'https://drive.google.com/uc?id=' + urlD.split('/')[-2]
departmentsPD = pd.read_csv(urlD2)
departments = spark.createDataFrame(departmentsPD)



urlOPT =
'https://drive.google.com/file/d/1IyZbHlrD8zXB8zhgx2XKxt812THThGRu/view?
usp=sharing'
urlOPT2 = 'https://drive.google.com/uc?id=' + urlOPT.split('/')[-2]
order_products_trainPD = pd.read_csv(urlOPT2)
order_products_train = spark.createDataFrame(order_products_trainPD)


urlP = 'https://drive.google.com/file/d/1Gkwkg56XgLzX_hyZDjEyHyRbcSjuWKp3/view?
usp=sharing'
urlP2 = 'https://drive.google.com/uc?id=' + urlP.split('/')[-2]
productsPD = pd.read_csv(urlP2)
products = spark.createDataFrame(productsPD)


#putting dataframes in database
aisles.createOrReplaceTempView("aisles")
departments.createOrReplaceTempView("departments")
order_products_prior.createOrReplaceTempView("order_products_prior")
order_products_train.createOrReplaceTempView("order_products_train")
orders.createOrReplaceTempView("orders")
products.createOrReplaceTempView("products")


df = sqlContext.sql("select count(order_id) as total_orders, order_hour_of_day
as hour from orders group by order_hour_of_day order by order_hour_of_day")
df.show()
```

```
+------------+----+
|total_orders|hour|
+------------+----+
|       22758|   0|
|       12398|   1|
|        7539|   2|
|        5474|   3|
|        5527|   4|
|        9569|   5|
|       30529|   6|
|       91868|   7|
|      178201|   8|
|      257812|   9|
|      288418|  10|
|      284728|  11|
|      272841|  12|
|      277999|  13|
|      283042|  14|
|      283639|  15|
|      272553|  16|
```

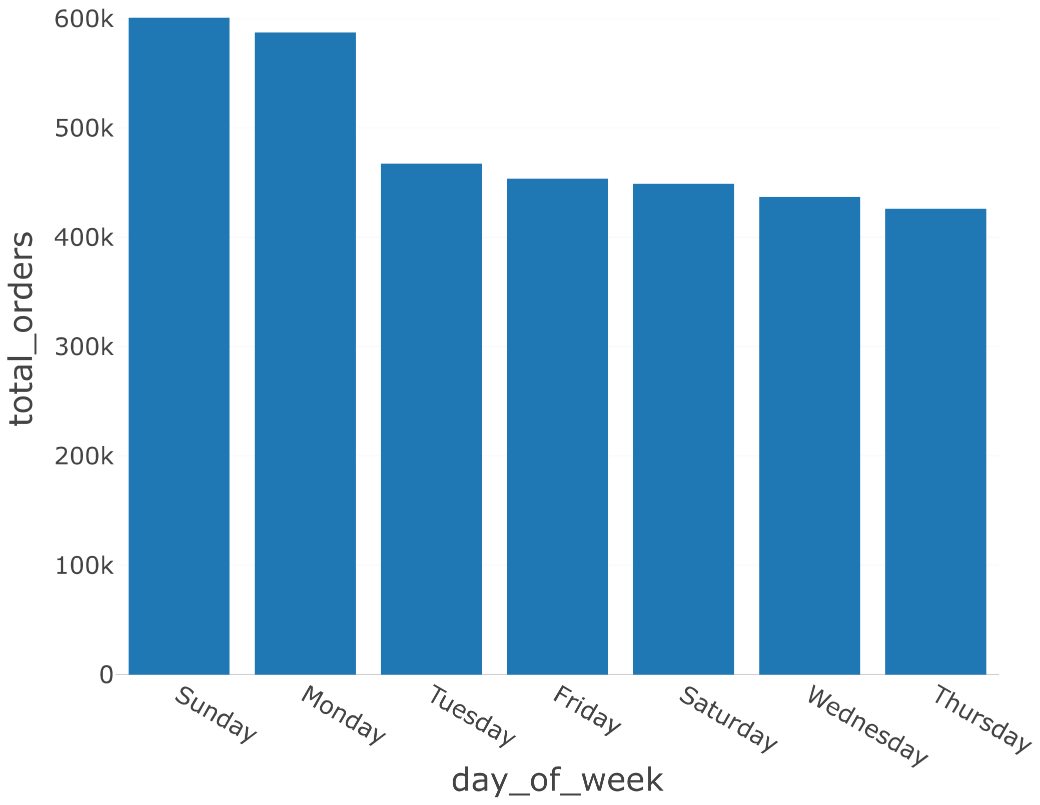
```
%sql
select count(order_id) as total_orders, order_hour_of_day as hour
from orders
group by order_hour_of_day
order by order_hour_of_day
```



```
%sql
select days_since_prior_order, count(order_id) as total_orders
from orders
group by days_since_prior_order
order by days_since_prior_order
```



```
%sql
select count(order_id) as total_orders,
  (case
     when order_dow = '0' then 'Sunday'
     when order_dow = '1' then 'Monday'
     when order_dow = '2' then 'Tuesday'
     when order_dow = '3' then 'Wednesday'
     when order_dow = '4' then 'Thursday'
     when order_dow = '5' then 'Friday'
     when order_dow = '6' then 'Saturday'
   end) as day_of_week
  from orders
 group by order_dow
 order by total_orders desc
```



```
%fs rm -r dbfs:/user/hive/warehouse/data
```

```
res0: Boolean = true
```

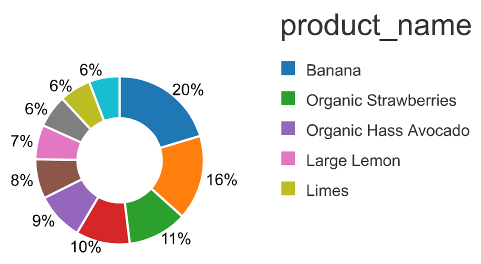
```
%sql
create table data as (select op.*, p.product_name, p.aisle_id, p.department_id,
d.department from (select * from order_products_train union
select * from order_products_prior) as op inner join products as p on
op.product_id = p.product_id inner join departments as d on p.department_id =
d.department_id)
```

Query returned no results

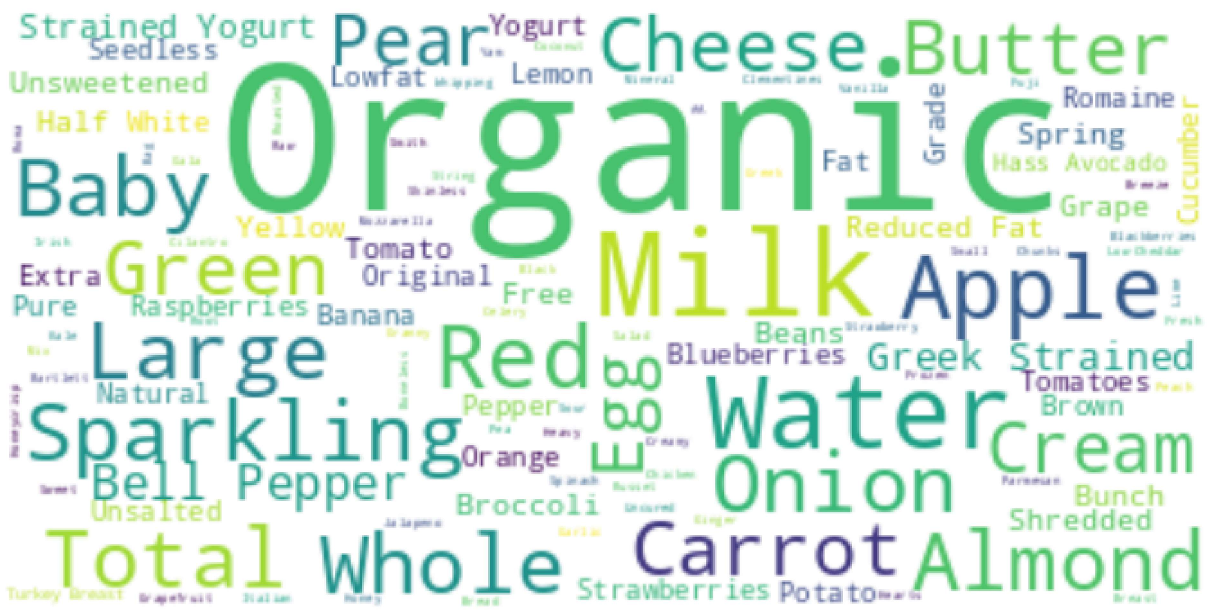
```
%sql
select order_id,count(product_id) as number_of_items
from data
group by order_id
```

Showing sample based on the first 1000 rows.

```
%sql
select product_name, count(*) as orders_count from data
group by product_name
order by orders_count desc
limit 10
```



```
#fetching data from database to dataset
data = sqlContext.sql("SELECT product_name FROM (select product_name, count(*)
as orders_count from data group by product_name order by orders_count desc
limit 200)")
#converting to RDD
dataRDD = data.rdd.flatMap(lambda x: x).collect()
strs = ' '.join(dataRDD)
#plot a wordcloud diagram
wc = WordCloud(background_color="white").generate(strs)
plt.figure(figsize=(12, 8))
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
display()
```



```
sparkData = spark.sql("select p.product_name, o.order_id from products p inner
join order_products_train o where o.product_id = p.product_id")
#get baskets
bskts =
sparkData.groupBy('order_id').agg(collect_set('product_name').alias('items'))
bskts.createOrReplaceTempView('baskets')
print()
sparkData.show(5)
bskts.show(5)
```

```
+--------------------+--------+
|        product_name|order_id|
+--------------------+--------+
|    Bulgarian Yogurt|       1|
|Organic 4% Milk F...|       1|
|Organic Celery He...|       1|
|      Cucumber Kirby|       1|
|Lightly Smoked Sa...|       1|
+--------------------+--------+
only showing top 5 rows

+--------+--------------------+
|order_id|               items|
+--------+--------------------+
|     112|[Umcka Elderberry...|
|     844|[Organic Shredded...|
|    1042|[Pure Irish Butte...|
|    1077|[Sparkling Water,...|
|    1119|[Broccoli Crown, ...|
+--------+--------------------+
```

```
print('The baskets of orders are displayed below.')
display(bskts)
```

```
The baskets of orders are displayed below.
```

	order_id	items
1	112	▸ ["Umcka Elderberry Intensive Cold + Flu Berry Flavor", "Fresh Cauliflower", "I Baked Potato Chips", "Sea Salt Baked Potato Chips", "Premium Epsom Salt", "B "Organic Hass Avocado", "Organic Lemon", "Marinara Pasta Sauce", "Coconut V
2	844	▸ ["Organic Red Radish, Bunch", "Baby Spinach", "Organic Shredded Carrots", "Cheese Pizza Snacks", "Garlic Couscous"]
3	1042	▸ ["Pure Irish Butter", "Organic Oat Cakes", "Organic Lentil Soup", "Applewood Brown Eggs", "Organic Whole Cashews", "Michigan Organic Kale"]
4	1077	▸ ["Sparkling Water", "Organic Strawberries", "Celery Sticks", "Bag of Organic B
5	1119	▸ ["Shallot", "Large Lemon", "Fresh Cauliflower", "Boneless Skinless Chicken Br Brown Eggs", "Broccoli Crown"]
6	1145	▸ ["Mexican Casserole Bowl", "Light Mozzarella String Cheese", "2% Low Fat C Rice", "Classic Stir-fry Sauce", "Nacho Cheese & Bean Snacks", "Everything Bag Raspberry Yogurt", "Milano Milk Chocolate Cookies", "Natural Uncured Turkey H Mild Cheddar Cheese", "Little Bites Blueberry Muffin Pouches", "Strawberries", " Strawberry Kiwi Smoothies", "Spinach Pizza", "Roma Tomato", "Hash Brown Pot Chips", "Harvest Best in 100% Fruit Juice Mandarin Oranges", "Original French "Signature Recipes Vodka Sauce Pasta Sauce", "Eggs, Cheese & Turkey Sausa Bread", "Goldfish Parmesan Baked Snack Crackers", "Strawberry Frozen Greek English Muffins"]

Truncated results, showing first 1000 rows.

```
#traing Frequent pattern mining model
fpg = FPGrowth(itemsCol="items", minSupport=0.001, minConfidence=0)
fpModel = fpg.fit(bskts)
#The frequently bought items are:
fpModel.freqItemsets.show()
```

```
+--------------------+----+
|               items|freq|
+--------------------+----+
|[Extra Ginger Bre...| 259|
|[Whole Strawberries]| 365|
|      [Organic Lime]| 140|
|  [Chopped Tomatoes]| 159|
|[Low Fat Plain Yo...| 200|
|[Organic Tomato B...| 772|
|[Organic Tomato B...| 175|
|[Organic Tomato B...| 144|
|[Organic Tomato B...| 179|
|[Thin & Light Tor...| 301|
|           [Ice Bag]| 135|
|   [Bunched Carrots]| 167|
|[Kids Organic Cho...| 451|
|[Scoops! Tortilla...| 398|
|[Natural Free & C...| 277|
|[Organic Large Br...|1137|
|[Organic Large Br...| 201|
```

```
#The associated rules are:
fpModel.associationRules.show()
```

```
+--------------------+--------------------+--------------------+---------------
---+--------------------+
|          antecedent|          consequent|          confidence|              l
ift|             support|
+--------------------+--------------------+--------------------+---------------
---+--------------------+
|[Trilogy Kombucha...|[Bag of Organic B...|0.21739130434782608|1.8426159982024
493|0.001066999977135...|
|[Organic Shredded...|[Organic Strawber...|  0.163855421686747|1.9734997268309
513|0.001036514263503...|
|[Organic Shredded...|[Bag of Organic B...| 0.1891566265060241|1.6032979203636
253|0.001196564260073623|
|[Organic Shredded...|[Organic Baby Spi...|0.24819277108433735| 3.328406101921
997|0.001570014252071123|
|[Organic Shredded...|            [Banana]|0.22409638554216868| 1.570194523689
117|0.001417585683908...|
| [Green Bell Pepper]|[Orange Bell Pepper]|0.10035700119000397|7.0642391465339
225|0.001928221387252399|
| [Green Bell Pepper]|[Organic Red Bell...|0.10234034113447045| 5.611355545304
109| 0.00196632852929296|
| [Green Bell Pepper]|[Yellow Bell Pepper]| 0.0646568821896073| 7.275784609962
```

```
+--------+--------------------+--------------------+
|order_id|               items|          prediction|
+--------+--------------------+--------------------+
|     112|[Umcka Elderberry...|[Organic Raspberr...|
|     844|[Organic Shredded...|[Organic Strawber...|
|    1042|[Pure Irish Butte...|[Organic Strawber...|
|    1077|[Sparkling Water,...|[Organic Large Br...|
|    1119|[Broccoli Crown, ...|[Organic Large Ex...|
|    1145|[Light Mozzarella...|[Organic Strawber...|
|    1280|[Vanilla Soy Milk...|[Organic Large Ex...|
|     1571|[Clementines, Bag...|[Organic Strawber...|
|     1591|[Honey Graham Sna...|[Blueberry Yoghur...|
|     1983|[Honey Nut Cheeri...|[Banana, Organic ...|
|     2021|[Organic Yellow O...|[Organic Large Ex...|
|     2530|[Total 2% with St...|[Total 2% Lowfat ...|
|     3091|[Lemon Ginger Tea...|[Organic Tomato B...|
|     3200|[Pineapple Spears...|[Packaged Grape T...|
|     3243|[Vanilla Almond M...|[Organic Strawber...|
|     3327|[Coconut Flakes, ...|[Organic Large Ex...|
|     3368|[Organic Yellow O...|[Organic Large Ex...|
|     3484|[Globe Eggplant     |[Organic Large Br   |
```