Facebook Data Analysis

1.Sanity check(using spark 2):

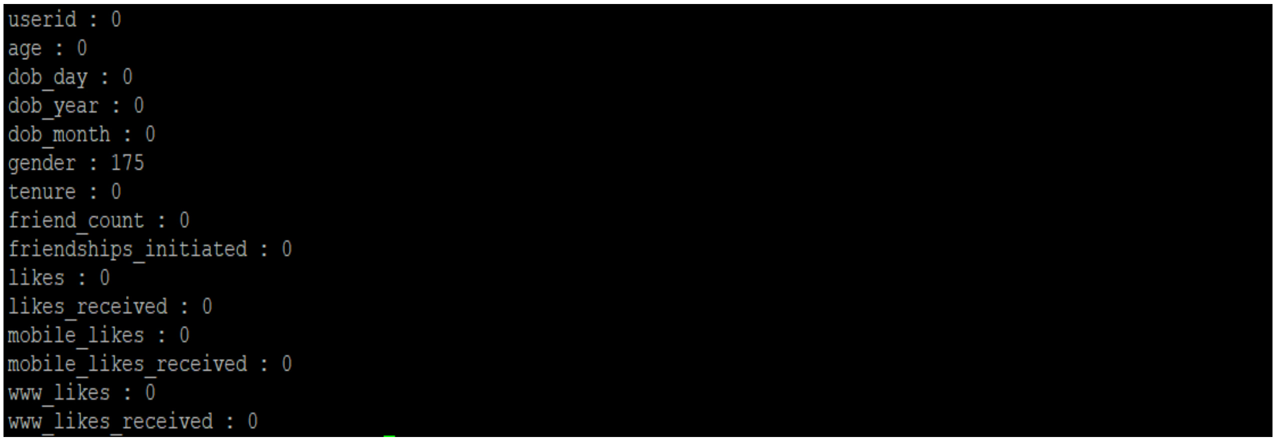
Code:

```
from pyspark.sql import SparkSession
from pyspark.sql import Row
from pyspark.sql import functions

def parseInput(line):
    fields = line.split(',')
    return Row(value = str(fields[i]))

if __name__ == "__main__":
    # Create a SparkSession (the config bit is only for Windows!)
    spark = SparkSession.builder.appName("SanityCheck").getOrCreate()

    # Get the raw data
    lines = spark.sparkContext.textFile("hdfs:///tmp/facebook_data/pseudo_facebook.csv")


a=["userid","age","dob_day","dob_year","dob_month","gender","tenure","friend_count","friendships_initiated","likes","likes_received","mobile_likes","mobile_likes_received","www_likes",$
    for i in range(15):
        # Convert it to a RDD of Row objects with (value)
        x = lines.map(parseInput)
        # Convert that to a DataFrame
        xDF = spark.createDataFrame(x)

        # Compute count of Null Values
        counts = xDF.filter(xDF["value"]=="NA").count()

        # Print them out
        print ("%s : %d"%(a[i],counts))

    # Stop the session
    spark.stop()
```

Command:

```
export SPARK_MAJOR_VERSION=2
spark-submit SanityCheck.py
```

Output:



```
userid : 0
age : 0
dob_day : 0
dob_year : 0
dob_month : 0
gender : 175
tenure : 0
friend_count : 0
friendships_initiated : 0
likes : 0
likes_received : 0
mobile_likes : 0
mobile_likes_received : 0
www_likes : 0
www_likes_received : 0
```

Observation: Gender has null values, we should not delete these as users might have kept it blank .

2: Facebook popularity based on ages(Using Mapreduce (python language))

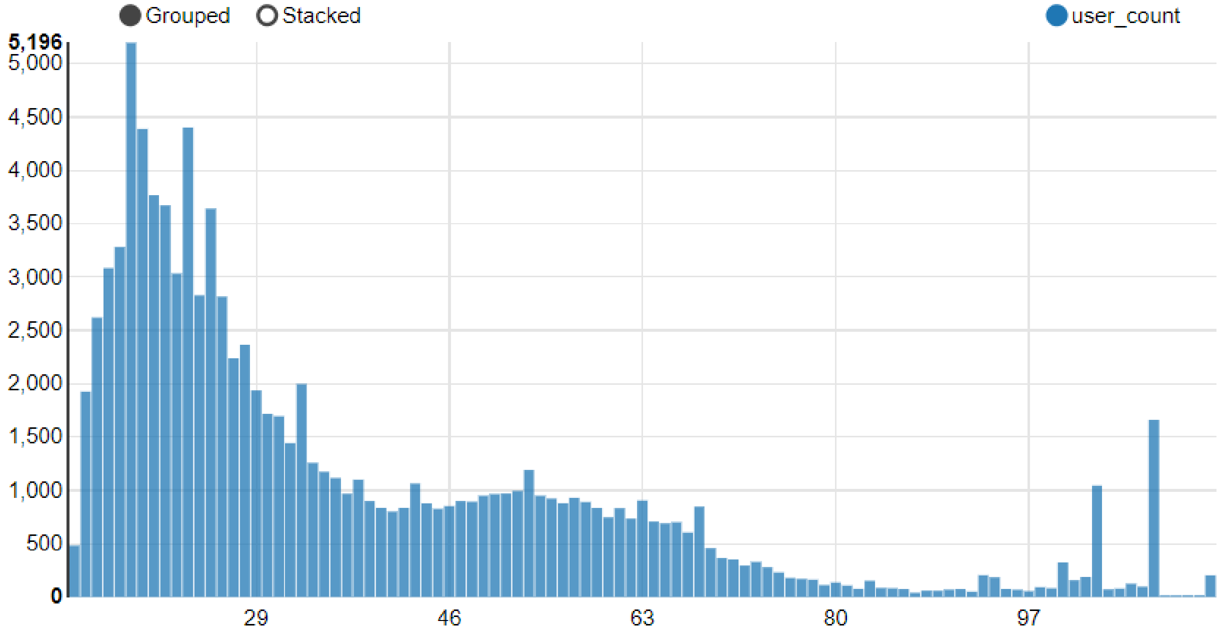
Code:

```
from mrjob.job import MRJob
from mrjob.step import MRStep

class WhatAgeUsesFacebook(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_ages,
                   reducer=self.reducer_count_ages),
            MRStep(reducer=self.reducer_sorted_output)
        ]
```

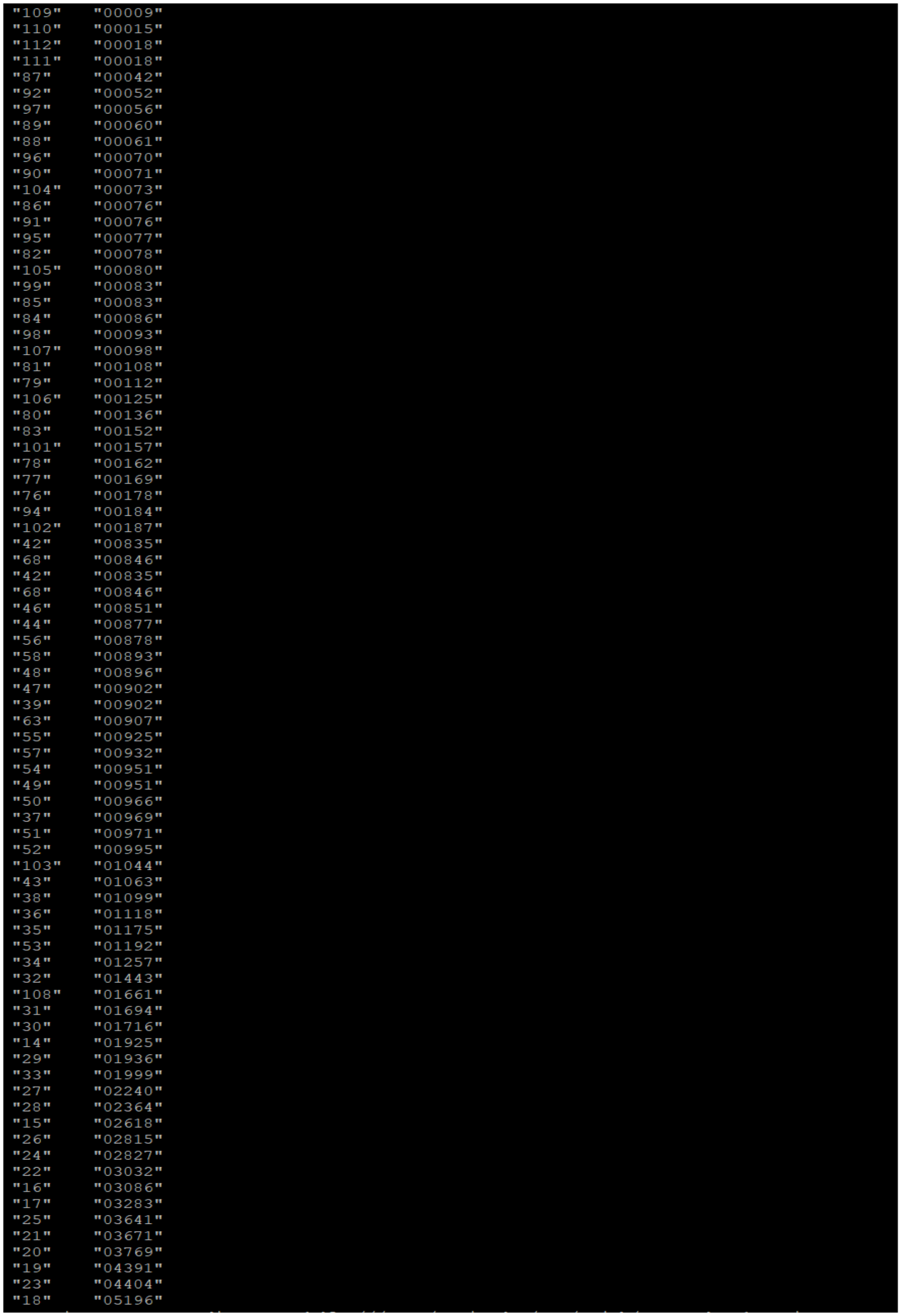
```
  def mapper_get_ages(self, _, line):
    (userid, age, dob_day, dob_year, dob_month, gender, tenure, friend_count,
friendships_initiated, likes, likes_received, mobile_likes, mobile_likes_received, www_likes,
www_likes_receved) = line.split(',')
    yield age, 1

  def reducer_count_ages(self, age, ones):
    yield str(sum(ones)).zfill(5), age

  def reducer_sorted_output(self, count, ages):
    for age in ages:
      yield age, count

if __name__ == '__main__':
  WhatAgeUsesFacebook.run()
```

Command: python map_reduce1.py -r hadoop --hadoop-streaming-jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar hdfs:///tmp/facebook_data/pseudo_facebook.csv

Age wise distribution of users:



Output: (Age,Count)



```
"109"   "00009"
"110"   "00015"
"112"   "00018"
"111"   "00018"
"87"    "00042"
"92"    "00052"
"97"    "00056"
"89"    "00060"
"88"    "00061"
"96"    "00070"
"90"    "00071"
"104"   "00073"
"86"    "00076"
"91"    "00076"
"95"    "00077"
"82"    "00078"
"105"   "00080"
"99"    "00083"
"85"    "00083"
"84"    "00086"
"98"    "00093"
"107"   "00098"
"81"    "00108"
"79"    "00112"
"106"   "00125"
"80"    "00136"
"83"    "00152"
"101"   "00157"
"78"    "00162"
"77"    "00169"
"76"    "00178"
"94"    "00184"
"102"   "00187"
"42"    "00835"
"68"    "00846"
"42"    "00835"
"68"    "00846"
"46"    "00851"
"44"    "00877"
"56"    "00878"
"58"    "00893"
"48"    "00896"
"47"    "00902"
"39"    "00902"
"63"    "00907"
"55"    "00925"
"57"    "00932"
"54"    "00951"
"49"    "00951"
"50"    "00966"
"37"    "00969"
"51"    "00971"
"52"    "00995"
"103"   "01044"
"43"    "01063"
"38"    "01099"
"36"    "01118"
"35"    "01175"
"53"    "01192"
"34"    "01257"
"32"    "01443"
"108"   "01661"
"31"    "01694"
"30"    "01716"
"14"    "01925"
"29"    "01936"
"33"    "01999"
"27"    "02240"
"28"    "02364"
"15"    "02618"
"26"    "02815"
"24"    "02827"
"22"    "03032"
"16"    "03086"
"17"    "03283"
"25"    "03641"
"21"    "03671"
"20"    "03769"
"19"    "04391"
"23"    "04404"
"18"    "05196"
```

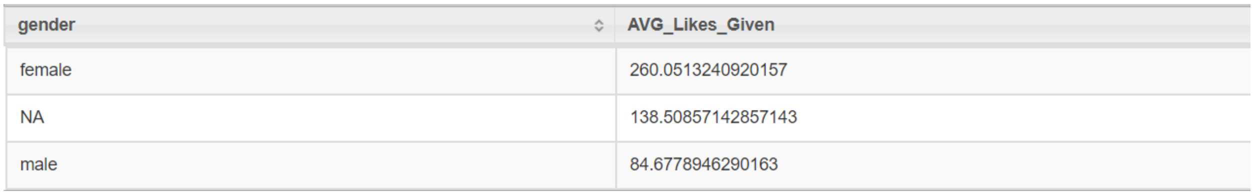
Observation : Facebook is most popular between age groups 16 and 26.

3. Likes Given (Using Drill)

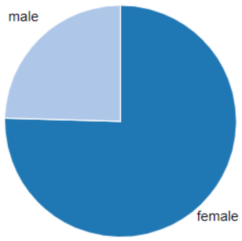
CMD: apache-drill-1.12.0/bin/drillbit.sh start -Ddrill.exec.http.port=8765

Query 1: SELECT gender,avg(likes) AS AVG_Likes_Given
FROM hive.facebook_db.facebook
GROUP BY gender
ORDER BY AVG_Likes_Given DESC

Output: gender vs likes given :

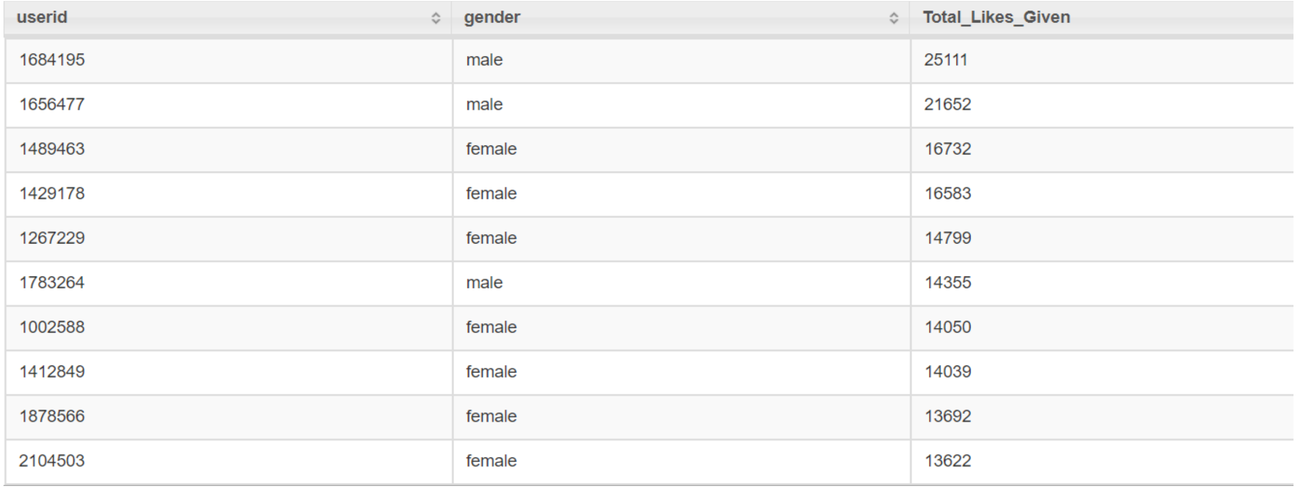


gender	AVG_Likes_Given
female	260.0513240920157
NA	138.50857142857143
male	84.6778946290163



Query 2: SELECT userid, gender, likes AS Total_Likes_Given
FROM hive.facebook_db.facebook
ORDER BY Total_likes_Given DESC LIMIT 10

Output : Top 10 users with most likes given



userid	gender	Total_Likes_Given
1684195	male	25111
1656477	male	21652
1489463	female	16732
1429178	female	16583
1267229	female	14799
1783264	male	14355
1002588	female	14050
1412849	female	14039
1878566	female	13692
2104503	female	13622

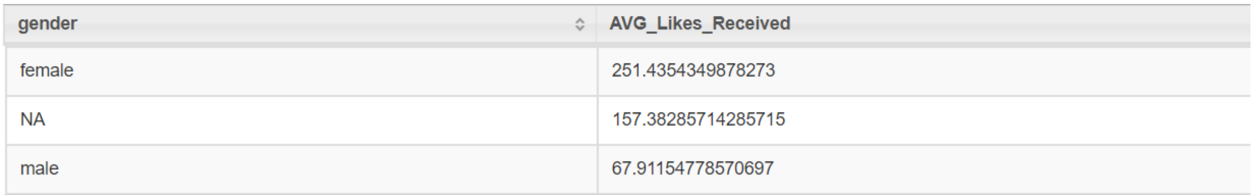
Analysis Result: Females give more likes then men

4. Likes Received (Using Drill)

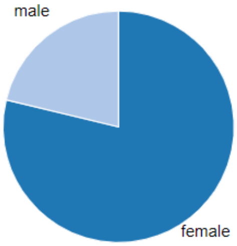
CMD: apache-drill-1.12.0/bin/drillbit.sh start -Ddrill.exec.http.port=8765

Query 1: SELECT gender,avg(likes_received) AS AVG_Likes_Received
FROM hive.facebook_db.facebook
GROUP BY gender
ORDER BY AVG_Likes_Received DESC

Output: gender vs total likes received :

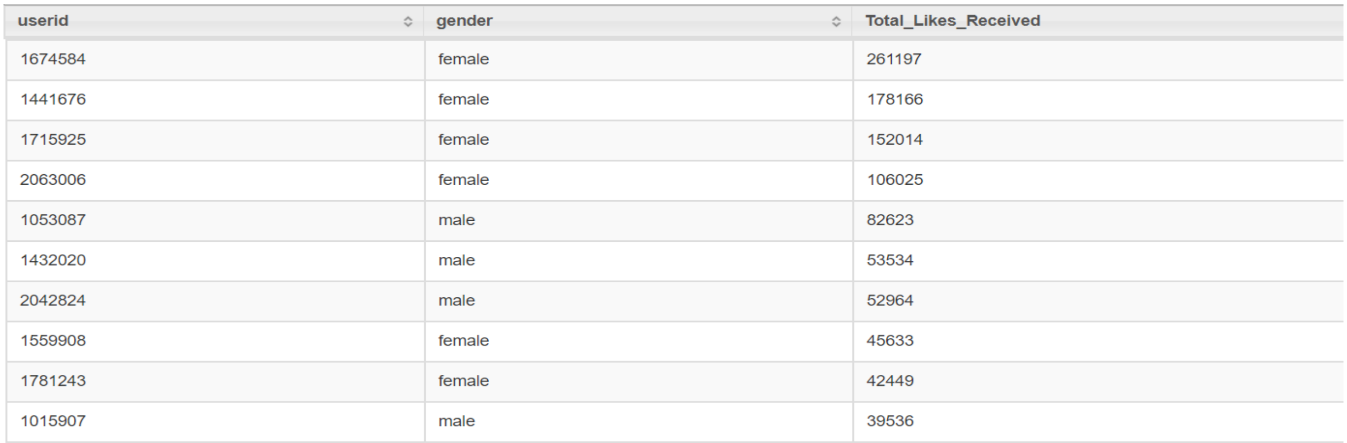


gender	AVG_Likes_Received
female	251.4354349878273
NA	157.38285714285715
male	67.91154778570697



Query 2: SELECT userid, gender, likes_received AS Total_Likes_Received
FROM hive.facebook_db.facebook
ORDER BY likes_received DESC
LIMIT 10

Output : Top 10 users with most likes received



userid	gender	Total_Likes_Received
1674584	female	261197
1441676	female	178166
1715925	female	152014
2063006	female	106025
1053087	male	82623
1432020	male	53534
2042824	male	52964
1559908	female	45633
1781243	female	42449
1015907	male	39536

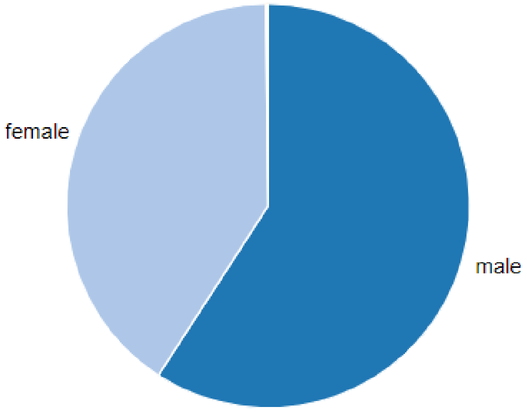
Analysis Result: Females receive more likes then men

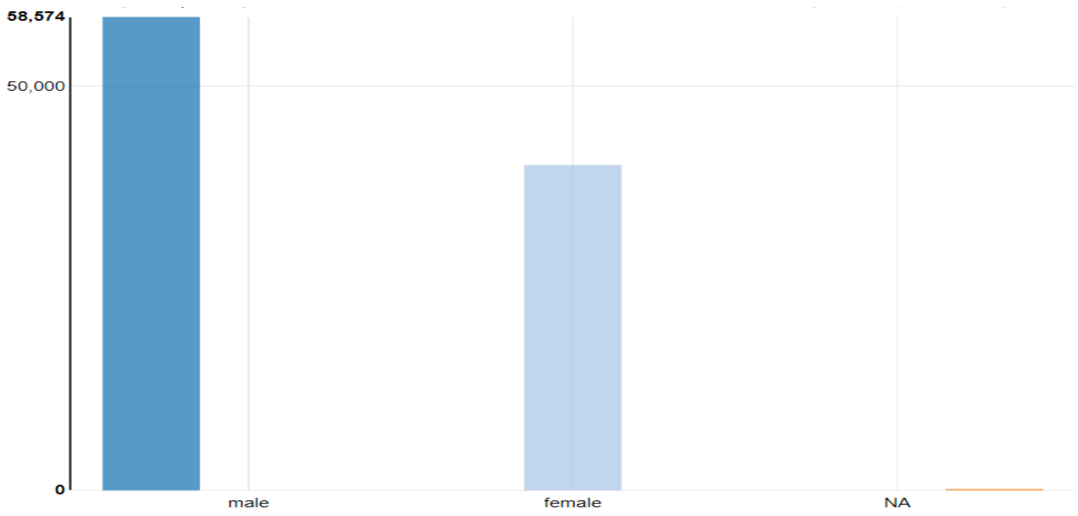
5.Gender Count (Using Zeppelin(Spark code)):

```
val x = fbDF.groupBy("gender").count().orderBy(desc("count")).cache()

x.show()
```

Output:

```
+------+-----+
|gender|count|
+------+-----+
| male |58574|
|female|40254|
| NA   | 175 |
+------+-----+
```





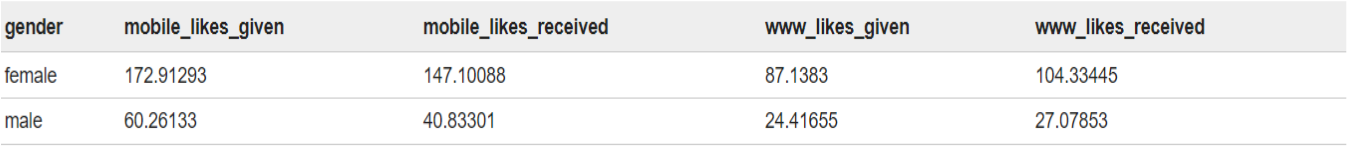
Analysis : There are more male users than female .

6.Likes Split Up (using Zeppelin-sql code)

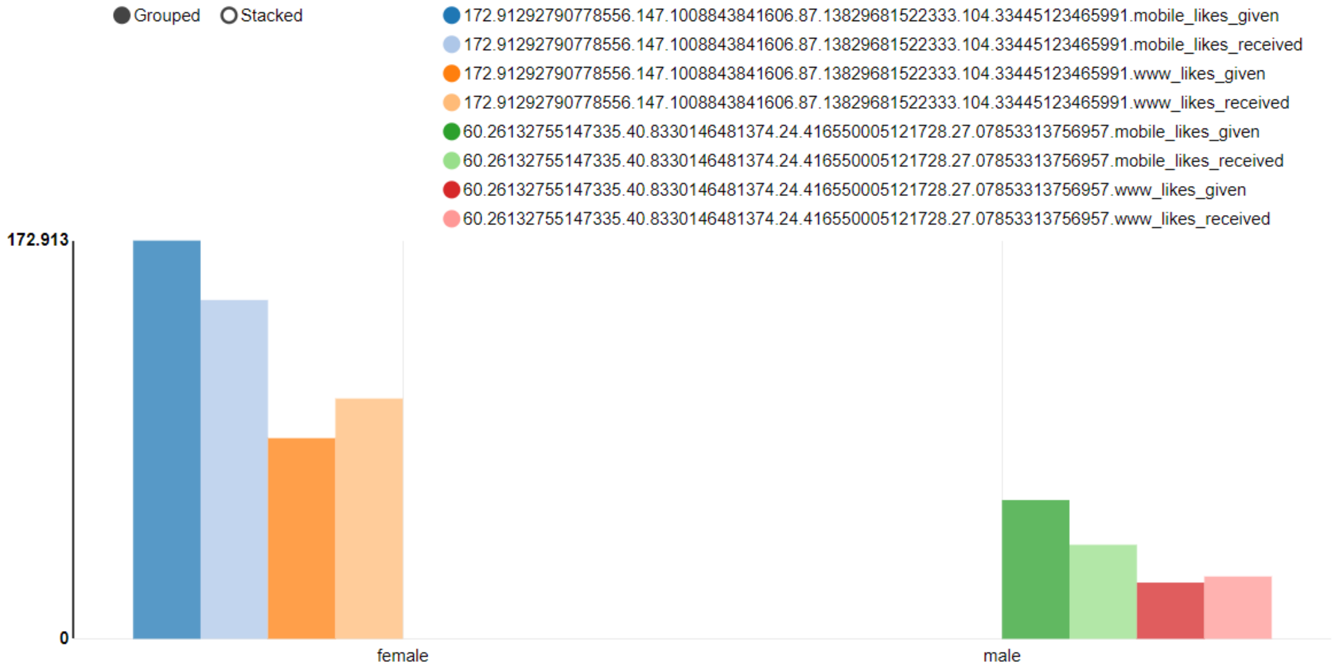
Query 1:

```
SELECT gender,avg(mobile_likes) AS mobile_likes_given,
avg(mobile_likes_received) AS mobile_likes_received, avg(www_likes) AS
www_likes_given, avg(www_likes_received) AS www_likes_received
FROM fb
WHERE gender <> "NA"
GROUP BY gender
```

Output:



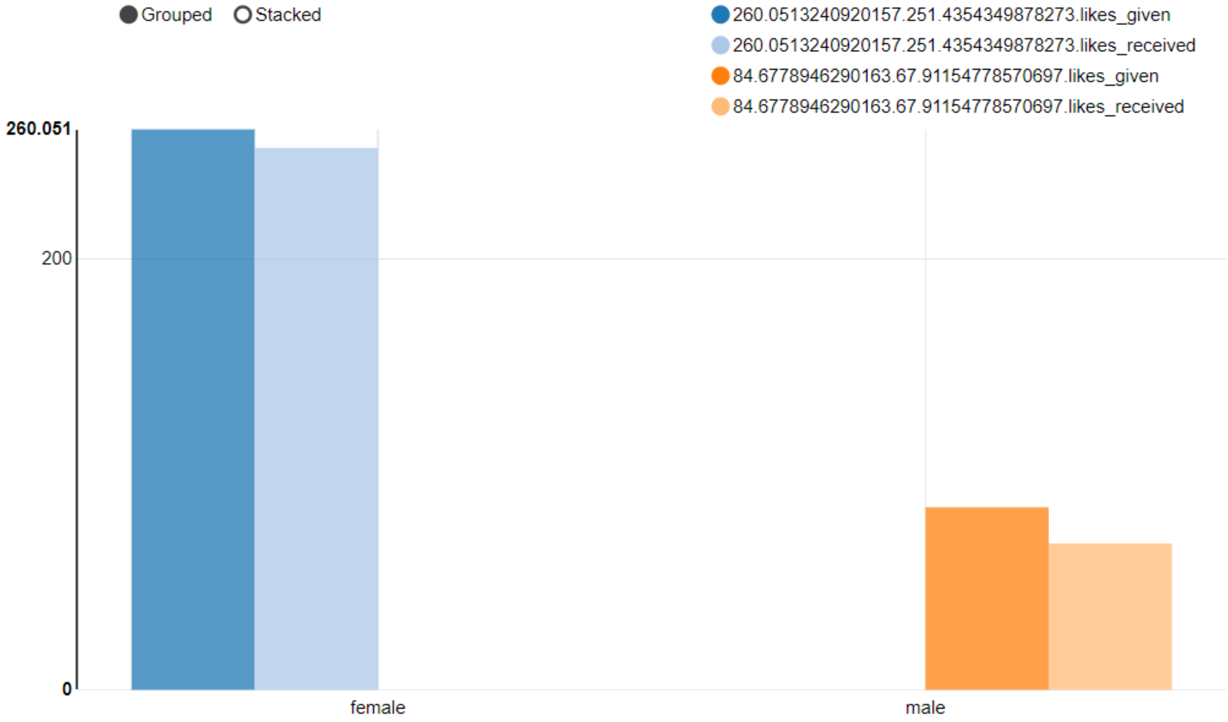
gender	mobile_likes_given	mobile_likes_received	www_likes_given	www_likes_received
female	172.91293	147.10088	87.1383	104.33445
male	60.26133	40.83301	24.41655	27.07853



Query2:

```
%sql
SELECT gender,avg(likes) AS likes_given ,avg(likes_received) AS likes_received
FROM fb
WHERE gender <> "NA"
GROUP BY gender
```

Output(Likes vs Likes Recived by gender):



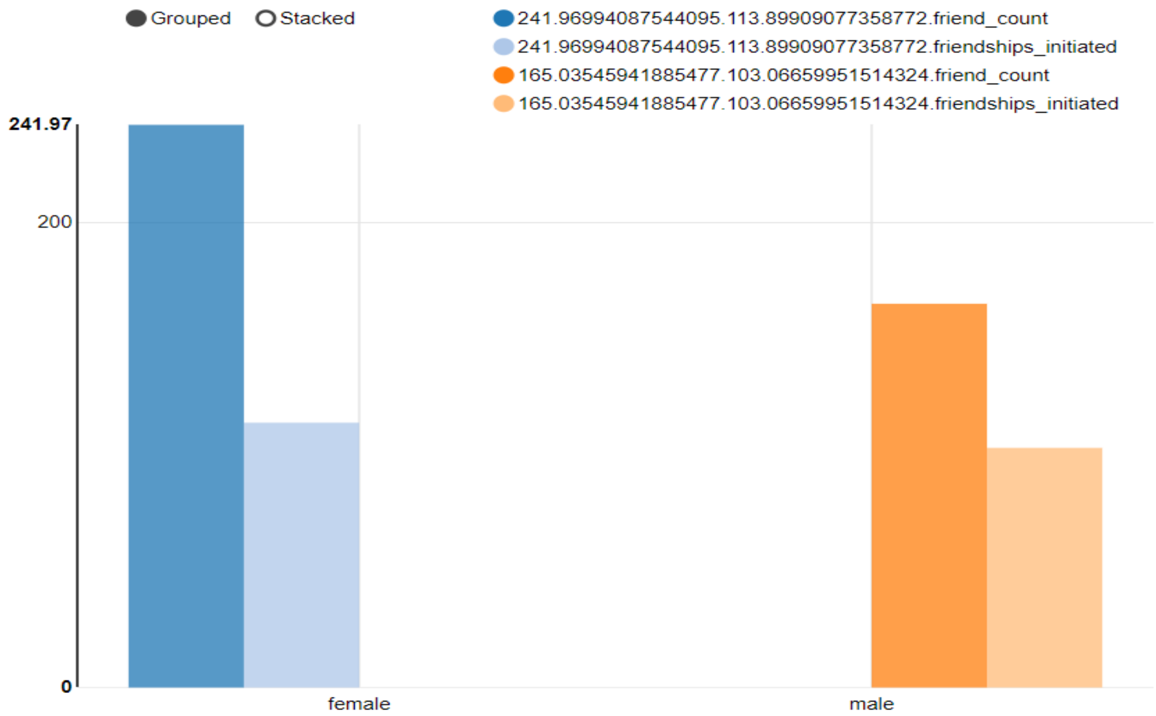
Analysis: Interesting obsservation for gender specific interaction with facebook: women like as well as are liked a lot more than men (nearly 2.5 as much).

7.Friends Counts & Friendships initiated (using Zepplin -sql code)

Query :

```
SELECT gender,avg(friend_count) AS friend_count ,avg(friendships_initiated) AS friendships_initiated
FROM fb
WHERE gender <> "NA"
GROUP BY gender
```

Output : (Friends Count vs Friendships Initiated)

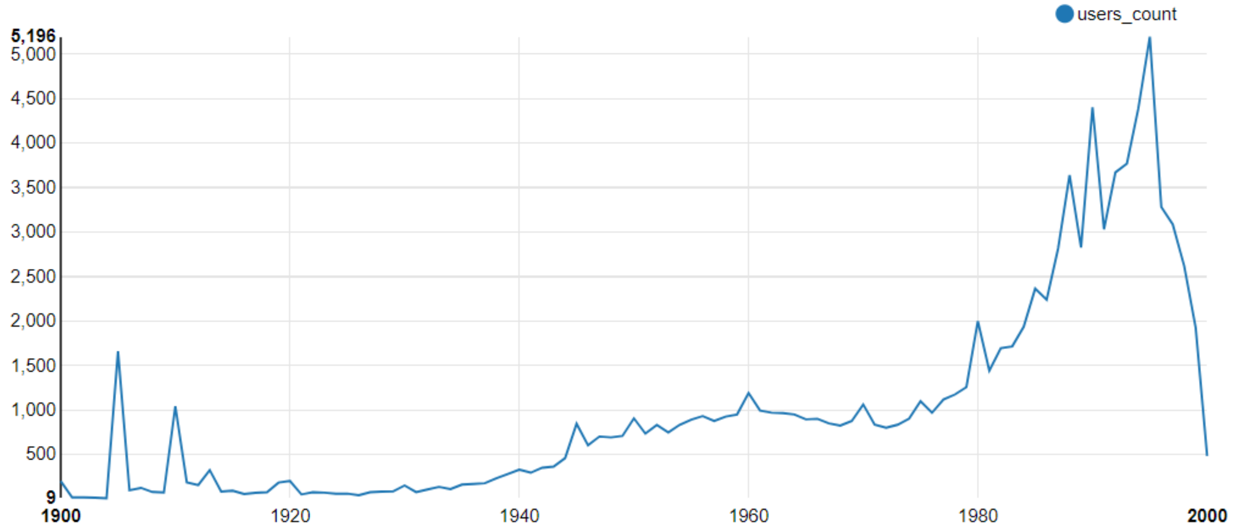


Analysis: Women have more friends than men on facebook, the friendships initiated in proportion to friend count are more in case of men than women.

8. Users w.r.t birth year(using Zepplin -sql code)

Query: SELECT dob_year,count(userid) AS users_count
FROM fb
GROUP BY dob_year

Output:



Analysis:

We see bumps between 1940 to 1980. After 1980 the no. users rocket. Since the data is till 2000 (we see miniscule value in 2000)