**Boston Marathon Data Analytics**

1. **The dataset and brief instructions to reproduce the results.**

This dataset is downloaded from Boston Marathon official website <http://registration.baa.org/2015/cf/Public/iframe_ResultsSearch.cfm>

It includes all the finishers (total 26,596) of the Boston Marathon in 2015, and it contains participants’ name, age, gender, country, city, and state (if available), times at different checkpoints, finished time and pace, overall place, gender place and division place.

To reproduce the following analysis, the data cleaning must be conducted, before we upload them into HBase:

* Remove “Citizen” column from the \*.csv files, as there is not any valid input data.
* Remove “Project Time” column from the \*.csv files, as the values are duplicated with the values of “Official Time” field.
* Change the value of “Official Time” field of index “2378” to “2:59:59”, as the original value is not correct.

1. **The scripts used for building the HBase/Phoenix tables and views.**
2. Copy CSV to HDP sandbox and remove the first row.

scp -P 2222 bm\_2015.csv root@127.0.0.1:/root/

ssh -p 2222 root@127.0.0.1

sed -i '1d' bm\_2015.csv

1. Upload CSV to HDFS

hdfs dfs -put bm\_2015.csv /tmp

1. Create a HBase table “bm\_2015” with three column families: profile, time, and place.

su hbase

hbase shell

create 'bm\_2015', 'profile', 'time', 'place'

exit

1. Import CSV to the HBase table “bm\_2015”.

hbase org.apache.hadoop.hbase.mapreduce.ImportTsv \

-Dimporttsv.separator=, \

-Dimporttsv.columns="HBASE\_ROW\_KEY, \

profile:bib, \

profile:lname, \

profile:fname, \

profile:age, \

profile:gender, \

profile:city, \

profile:state, \

profile:country, \

time:5k, \

time:10k, \

time:15k, \

time:20k, \

time:half, \

time:25k, \

time:30k, \

time:35k, \

time:40k, \

time:pace, \

time:total, \

place:overall, \

place:gender, \

place:division" \

bm\_2015 /tmp/bm\_2015.csv

1. **The queries used for achieving the analysis objectives.**
2. Start Phoenix Sqlline.

cd /usr/hdp/current/phoenix-client/bin

./sqlline.py 127.0.0.1

1. Create a Phoenix view for the HBase table “bm\_2015”.

create view "bm\_2015" (

"index" VARCHAR PRIMARY KEY,

"profile"."bib" VARCHAR,

"profile"."lname" VARCHAR,

"profile"."fname" VARCHAR,

"profile"."age" VARCHAR,

"profile"."gender" VARCHAR,

"profile"."city" VARCHAR,

"profile"."state" VARCHAR,

"profile"."country" VARCHAR,

"time"."5k" VARCHAR,

"time"."10k" VARCHAR,

"time"."15k" VARCHAR,

"time"."20k" VARCHAR,

"time"."half" VARCHAR,

"time"."25k" VARCHAR,

"time"."30k" VARCHAR,

"time"."35k" VARCHAR,

"time"."40k" VARCHAR,

"time"."pace" VARCHAR,

"time"."total" VARCHAR,

"place"."overall" VARCHAR,

"place"."gender" VARCHAR,

"place"."division" VARCHAR);

1. Query how many finishers in each age group (10-19, 20-29, …) from different genders.

SELECT (CAST(TO\_NUMBER("profile"."age") AS INTEGER) / 10) AS age\_group,

"profile"."gender" AS gender,

COUNT(\*) AS amount

FROM "bm\_2015"

GROUP BY age\_group, "profile"."gender";

1. Query average finished time in each age group (10-19, 20-29, …) from different genders.

SELECT (CAST(TO\_NUMBER("profile"."age") AS INTEGER) / 10) AS age\_group,

"profile"."gender" AS gender,

AVG(TO\_NUMBER(TO\_TIME("time"."total",'HH:mm:ss')) / 1000) AS avg\_time

FROM "bm\_2015"

GROUP BY age\_group, "profile"."gender";

1. Query top 10 countries with most finishers.

SELECT "profile"."country", COUNT(\*) as amount

FROM "bm\_2015"

GROUP BY "profile"."country"

ORDER BY amount DESC

LIMIT 10;

1. Query top 10 pace groups with most finishers from different genders.

SELECT “time”.”pace” AS pace\_group,

"profile"."gender" as gender,

COUNT(\*) AS amount

FROM "bm\_2015"

GROUP BY pace\_group, gender

HAVING "profile"."gender" = 'M'

ORDER BY COUNT(\*) DESC

LIMIT 10;

SELECT “time”.”pace” AS pace\_group,

"profile"."gender" as gender,

COUNT(\*) AS amount

FROM "bm\_2015"

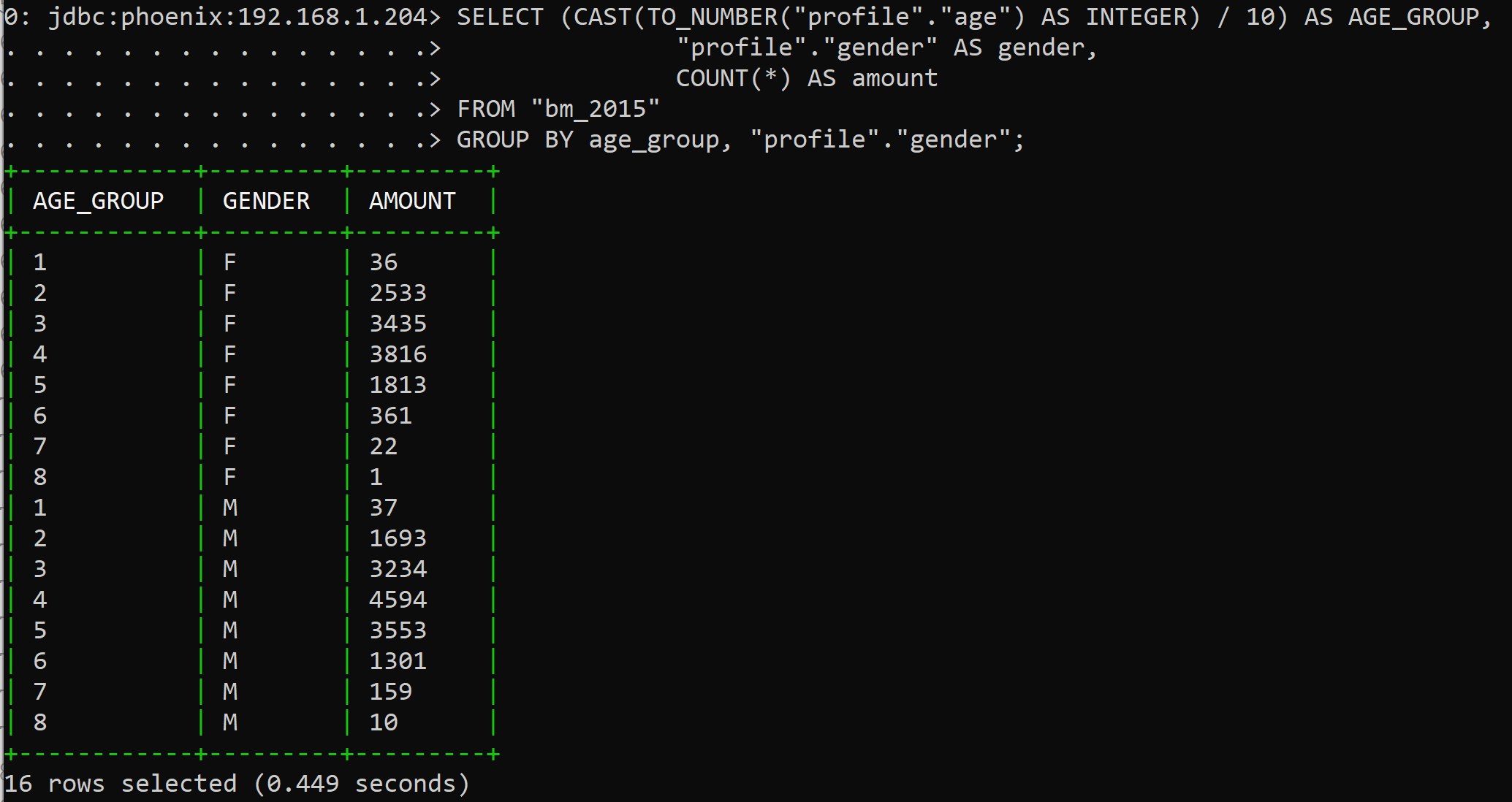
GROUP BY pace\_group, gender

HAVING "profile"."gender" = 'F'

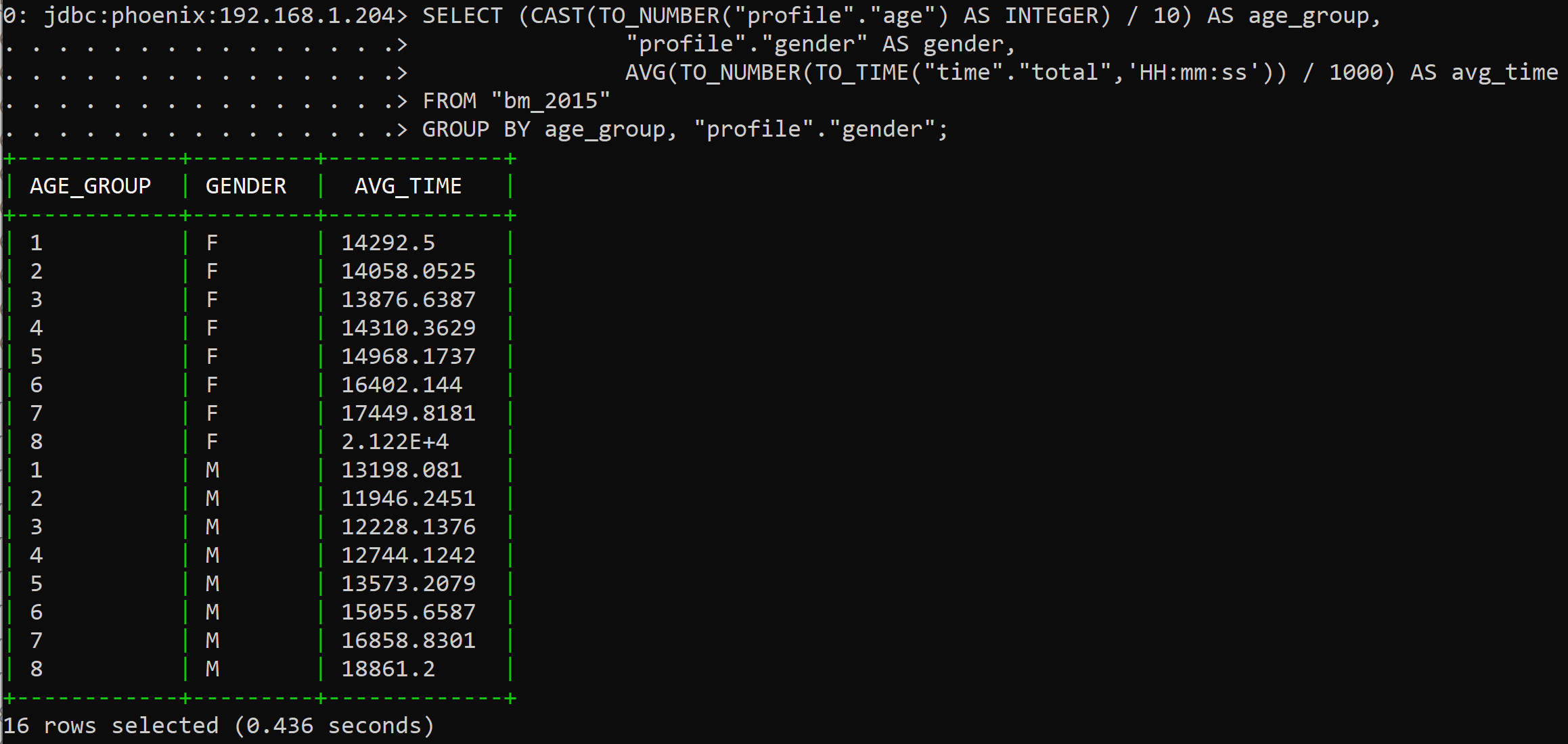
ORDER BY COUNT(\*) DESC

LIMIT 10;

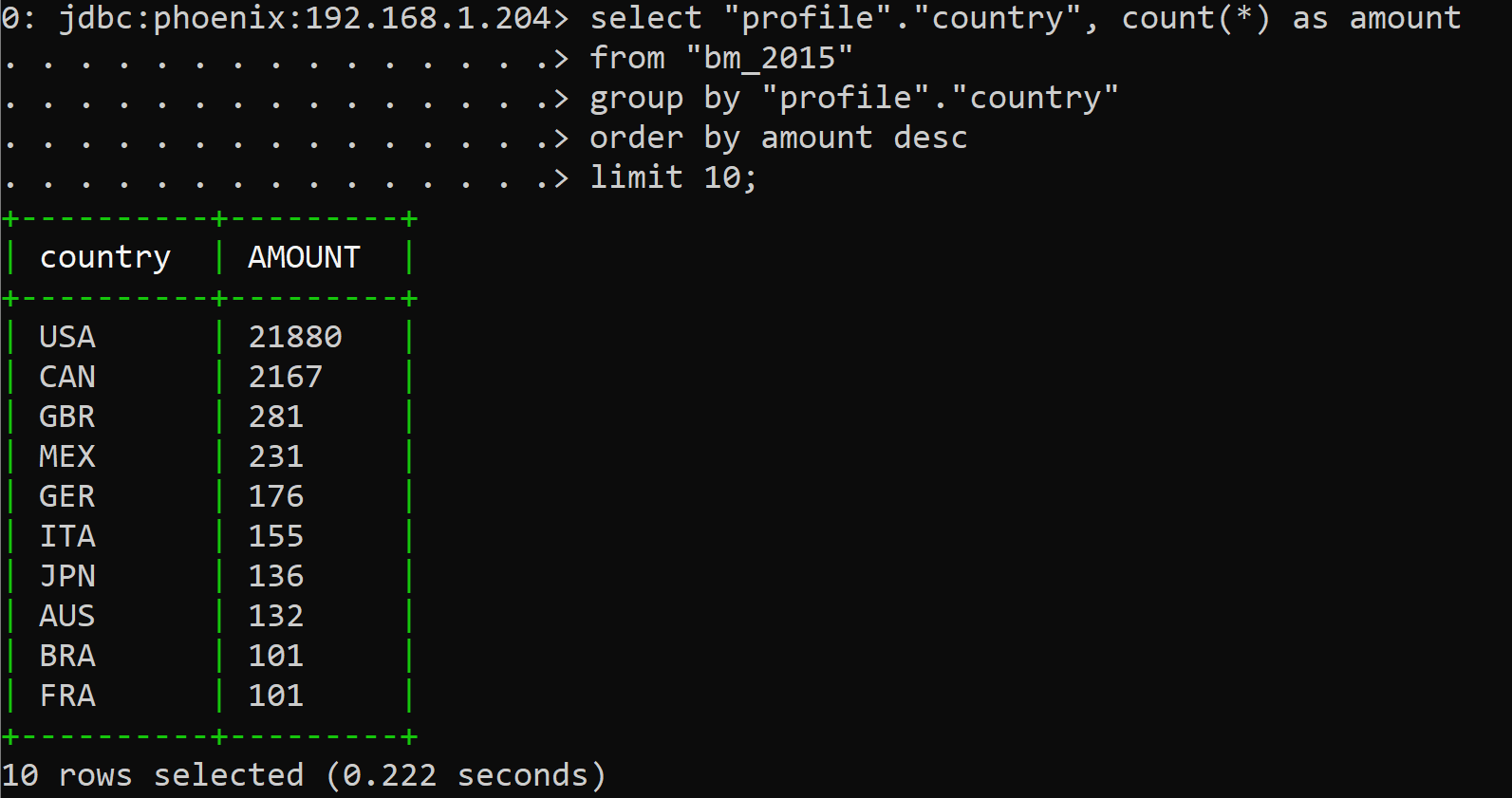
1. **The analysis outcome (any visualization of results: tables or figures).**
2. Query how many finishers in each age group (10-19, 20-29, …) from different genders.



1. Query average finished time in each age group (10-19, 20-29, …) from different genders.



1. Query top 10 countries with most finishers.



1. Query top 10 pace groups with most finishers from different genders.





