Boston Marathon Data Analytics II

By Bibo Gao

Scopes

1. Runner and Pace distribution per age group
   Place distribution per finishing time

2. Finishing time of each year, top countries, and top loyal runners

3. How to win ? (analyze gap between elite and average runners)

Data Identification

1. Download 2015, 2016, and 2017 data directly from Kaggle

2. Scrape 2018 and 2019 data from Boston Marathon official sites

   http://registration.baa.org/2018/cf/Public/iframe_ResultsSearch.cfm
   http://registration.baa.org/2019/cf/Public/iframe_ResultsSearch.cfm

```
try:
    response = requests.post(
        'http://registration.baa.org/2019/cf/Public/iframe_ResultsSearch.cfm',
        headers=headers,
        params=params,
        data={'start': start, 'next': 'Next 25 Records'},
    )
    soup = BeautifulSoup(response.text, 'lxml')
    table = soup.find("table", attrs={"class": "tablegrid_table"})
    if table == None:
        print("table is none")
        continue
    rows = table.findAll("tr")
    for row in rows:
        a = [t.text.strip() for t in row.findAll("td")][0:]
        #Don't store lines without data
        if len(a) > 0 and a != [''] and a !=['',''] and a != ['', '', '']:
            #print(a)
            results.append(a)
```

Data Filtering (132,126 records)

1. Remove empty columns (citizen and projected time) and extra comma in the city column

2. Split the name column into first name and last name columns

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1		Bib	LastName	FirstName	Age	M/F	City	State	Country	5K	10K	15K	20K	Half	25K	30K	35K	40K	Pace	Official Time	Overall	Gender	Division
2	0	3	Desisa	Lelisa	25	M	Ambo		ETH	0:14:43	0:29:43	0:44:57	1:00:29	1:04:02	1:16:07	1:32:00	1:47:59	2:02:39	0:04:56	2:09:17	1	1	1
3	1	4	Tsegay	Yemane Adhane	30	M	Addis Ababa		ETH	0:14:43	0:29:43	0:44:58	1:00:28	1:04:01	1:16:07	1:31:59	1:47:59	2:02:42	0:04:58	2:09:48	2	2	2
4	2	8	Chebet	Wilson	29	M	Marakwet		KEN	0:14:43	0:29:43	0:44:57	1:00:29	1:04:02	1:16:07	1:32:00	1:47:59	2:03:01	0:04:59	2:10:22	3	3	3
5	3	11	Kipyego	Bernard	28	M	Eldoret		KEN	0:14:43	0:29:44	0:45:01	1:00:29	1:04:02	1:16:07	1:32:00	1:48:03	2:03:47	0:05:00	2:10:47	4	4	4
6	4	10	Korir	Wesley	32	M	Kitale		KEN	0:14:43	0:29:44	0:44:58	1:00:28	1:04:01	1:16:07	1:32:00	1:47:59	2:03:27	0:05:00	2:10:49	5	5	5
7	5	9	Chepkwony	Frankline	30	M	Koibatek		KEN	0:14:44	0:29:45	0:44:59	1:00:29	1:04:02	1:16:07	1:32:00	1:47:59	2:03:18	0:05:00	2:10:52	6	6	6
8	6	14	Ritzenhein	Dathan	32	M	Rockford	MI	USA	0:14:45	0:29:45	0:45:20	1:00:43	1:04:03	1:16:05	1:31:59	1:48:06	2:04:05	0:05:01	2:11:20	7	7	7
9	7	1	Keflezighi	Meb	39	M	San Diego	CA	USA	0:14:44	0:29:44	0:44:59	1:00:30	1:04:02	1:16:07	1:31:59	1:47:59	2:04:58	0:05:04	2:12:42	8	8	8
10	8	5	Tola	Tadese	27	M	Addis Ababa		ETH	0:14:43	0:29:43	0:44:58	1:00:28	1:04:02	1:16:07	1:32:00	1:48:00	2:04:39	0:05:06	2:13:35	9	9	9
11	9	16	Shafar	Vitaliy	33	M	Lutsk		UKR	0:15:14	0:30:34	0:46:05	1:01:43	1:05:07	1:17:18	1:33:11	1:49:43	2:06:16	0:05:07	2:13:52	10	10	10
12	10	22	Tegenkamp	Matt	33	M	Portland	OR	USA	0:14:46	0:29:50	0:45:33	1:01:20	1:04:48	1:17:08	1:33:12	1:49:52	2:06:55	0:05:07	2:13:52	11	11	11
13	11	19	Eggleston	Jeffrey	30	M	Boulder	CO	USA	0:15:14	0:30:34	0:46:06	1:01:42	1:05:07	1:17:19	1:33:30	1:50:12	2:06:47	0:05:08	2:14:17	12	12	12
14	12	15	April	Lusapho	32	M	Uitenhage		RSA	0:14:44	0:29:44	0:44:59	1:00:29	1:04:03	1:16:07	1:32:00	1:48:07	2:06:50	0:05:13	2:16:25	13	13	13
15	13	20	Arciniaga	Nicholas	31	M	Flagstaff	AZ	USA	0:14:44	0:29:44	0:45:14	1:01:07	1:04:35	1:17:50	1:35:43	1:53:33	2:10:40	0:05:16	2:18:02	14	14	14
16	14	76	Goffi	Danilo	42	M	Parabiago - Milan		ITA	0:15:53	0:32:17	0:48:32	1:04:49	1:08:21	1:21:17	1:38:02	1:54:55	2:11:25	0:05:18	2:18:44	15	15	1
17	15	28	Canaday	Sage	29	M	Boulder	CO	USA	0:16:08	0:32:19	0:48:41	1:05:01	1:08:38	1:21:18	1:38:01	1:54:54	2:11:37	0:05:19	2:19:12	16	16	15
18	16	54	Zyryanov	Sergey	30	M	Moscow		RUS	0:15:53	0:32:17	0:48:32	1:04:49	1:08:20	1:21:17	1:38:02	1:54:57	2:11:37	0:05:19	2:19:17	17	17	16
19	17	29	Chavez	Chris	28	M	Burlingame	CA	USA	0:15:57	0:32:21	0:48:43	1:05:21	1:09:00	1:21:50	1:38:46	1:56:05	2:12:58	0:05:21	2:20:04	18	18	17
20	18	27	Macpherson	Scott	28	M	Columbia	MO	USA	0:16:08	0:32:17	0:48:31	1:04:44	1:08:17	1:20:55	1:37:41	1:54:45	2:12:23	0:05:22	2:20:25	19	19	18
21	19	33	Zablocki	Christopher	26	M	Essex	CT	USA	0:16:08	0:32:20	0:48:42	1:05:20	1:08:58	1:21:50	1:38:40	1:55:47	2:13:07	0:05:22	2:20:35	20	20	19
22	20	26	Hasegawa	Kiyokatsu	32	M	Tokyo		JPN	0:14:46	0:29:51	0:45:33	1:01:27	1:05:04	1:18:05	1:35:34	1:54:00	2:12:27	0:05:22	2:20:42	21	21	20

Data Extraction

1. The collected data have same format: same fields and same data type

2. The collected data are divided into 5 partitions, and one partition for each year

```
CREATE TABLE bm_by_year (rowid INT,
    bib             STRING,
    lname           STRING,
    fname           STRING,
    age             SMALLINT,
    gender          STRING,
    city            STRING,
    state           STRING,
    country         STRING,
    time_5k         STRING,
    time_10k        STRING,
    time_15k        STRING,
    time_20k        STRING,
    time_half       STRING,
    time_25k        STRING,
    time_30k        STRING,
    time_35k        STRING,
    time_40k        STRING,
    time_pace       STRING,
    time_total      STRING,
    place_overall   INT,
    place_gender    INT,
    place_division  INT)
PARTITIONED BY (year INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

ALTER TABLE bm_by_year ADD PARTITION (year=2015) LOCATION '2015/';
ALTER TABLE bm_by_year ADD PARTITION (year=2016) LOCATION '2016/';
ALTER TABLE bm_by_year ADD PARTITION (year=2017) LOCATION '2017/';
ALTER TABLE bm_by_year ADD PARTITION (year=2018) LOCATION '2018/';
ALTER TABLE bm_by_year ADD PARTITION (year=2019) LOCATION '2019/';
```

Data Aggregation

1. Aggregate all the 5 partitions of data from 2015, 2016, 2017, 2018, and 2019

2. Aggregate age, gender, place, country, and runner for different analysis purpose

Data Analysis and Visualization

1. Apache Hive and HiveQL are used to manipulate and query the data

2. HUE Web UI is used to visualize the data

Analysis on scope 1

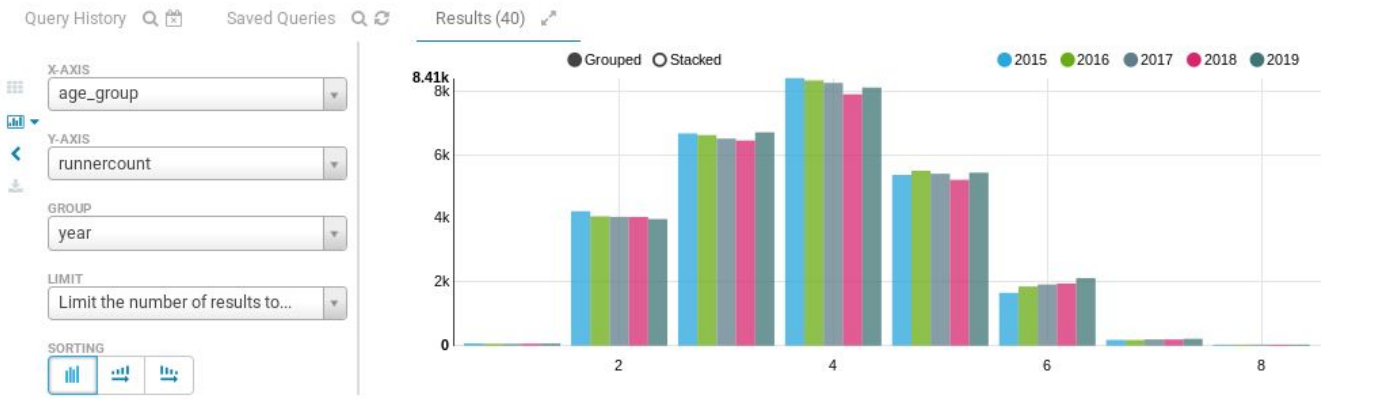
1. Runner and Pace distribution per age group
   Place distribution per finishing time

2. Finishing time of each year, top countries, and top loyal runners

3. How to win ? (analyze gap between elite and average runners)

Runner count of each year (per age group)

```
1 SELECT cast(AGE /10 as int) as age_group, year, COUNT(1) AS runnercount
2 FROM bm_by_year GROUP BY cast(AGE /10 as int), year;
```



Pace of each year (per age group)

18.47s default text

```
1 SELECT cast(AGE /10 as int) as age_group, year,
2 cast( avg(UNIX_TIMESTAMP(time_pace,'HH:mm:ss') -  UNIX_TIMESTAMP('00:00:00','HH:mm:ss')) AS DECIMAL(10,2) ) as avgPace
3 FROM bm_by_year
4 GROUP BY cast(AGE /10 as int), year;
```

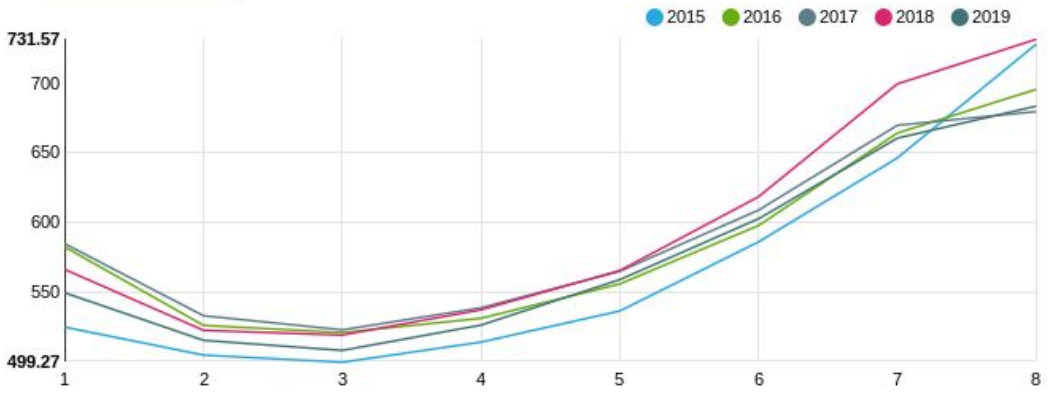
Query History | Saved Queries | Results (40)

X-AXIS
age_group

Y-AXIS
- [ ] age_group
- [ ] year
- [x] avgpace

LIMIT
Limit the number of results to...

SORTING



Place of men and women (per finishing time)

18.14s default text ?

```
1 select cast((place_gender / 100) AS int) as rank , gender,
2 CAST( avg(UNIX_TIMESTAMP(time_total,'HH:mm:ss') -  UNIX_TIMESTAMP('00:00:00','HH:mm:ss'))/3600 AS DECIMAL(10,2)) as avgfinishtime
3 from bm_by_year
4 where year = '2016'
5 group by  cast((place_gender / 100) AS int), gender;
6
```

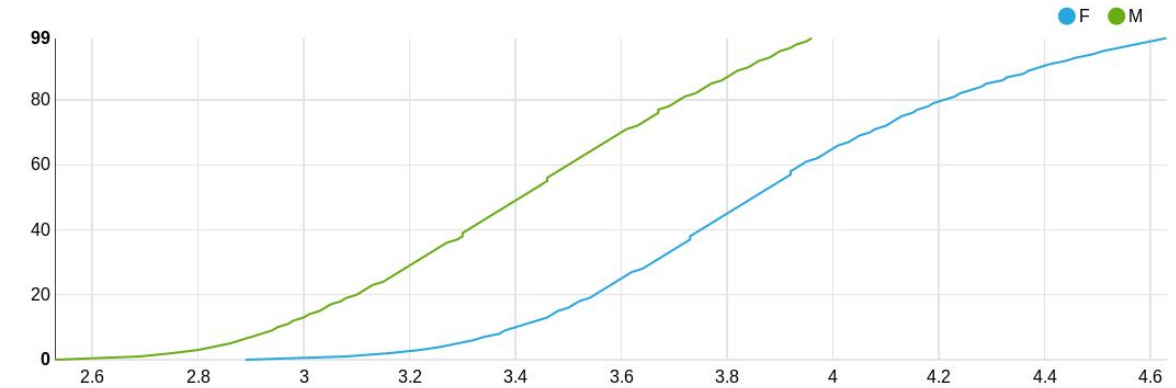
Query History Saved Queries Results (267)

X-AXIS
avgfinishtime

Y-AXIS
☑ rank
☐ avgfinishtime

LIMIT
Limit the number of results to...

SORTING



Analysis on scope 2

1. Runner and Pace distribution per age group
   Place distribution per finishing time

2. Finishing time of each year, top countries, and top loyal runners

3. How to win ? (analyze gap between elite and average runners)

Finishing time of each year

8m, 5s default text ?

```
1 SELECT CAST(avg(hour(time_total) * 3600 + minute(time_total) * 60 + second(time_total)) / 3600 AS DECIMAL(10,2)) as avg_time,
2        year,
3        gender
4 FROM bm_by_year
5 GROUP BY  year, gender;
6
```

Query History Saved Queries Results (10)

X-AXIS: gender

Y-AXIS: avg_time

GROUP: year

LIMIT: Limit the number of results to...

SORTING

Finishing time of top countries

1m, 49s default text ?

```
1  select a.country, a.year,
2      avg(UNIX_TIMESTAMP(a.time_total,'HH:mm:ss') -  UNIX_TIMESTAMP('00:00:00','HH:mm:ss')) / 3600 as avgTime
3  from bm_by_year a
4  where a.country in (
5      select b.country from bm_by_year b
6      where b.year = 2017
7      group by b.country
8      HAVING count(1) > 150
9      limit 10
10 )
11 group by a.country, a.year ;
```

Query History Saved Queries Results (50)

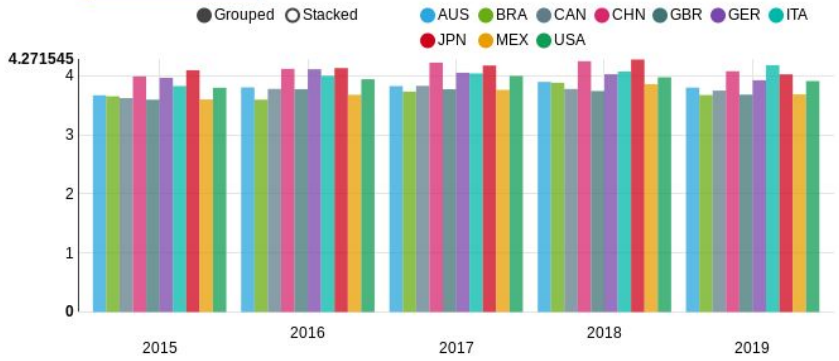
X-AXIS: a.year

Y-AXIS: avgtime

GROUP: a.country

LIMIT: Limit the number of result...

SORTING



Finishing time of runners who have run 5 years

```
select a.fname, a.lname, a.year,  (UNIX_TIMESTAMP(a.time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',  'HH:mm:ss'))/3600 as totaltime
from bm_by_year a
where concat( a.fname," ", a.lname,  a.year - a.age, a.city, a.state, a.city) in (
    select concat( b.fname," ", b.lname, b.year - b.age, b.city, b.state, b.city)
    from bm_by_year b
    where b.place_gender < 1000
    group by concat( b.fname," ", b.lname, b.year - b.age, b.city, b.state, b.city)
    HAVING count(*)  > 4
)
order by a.fname desc, a.year
limit 50;
```

Query History | Saved Queries | Results (40)

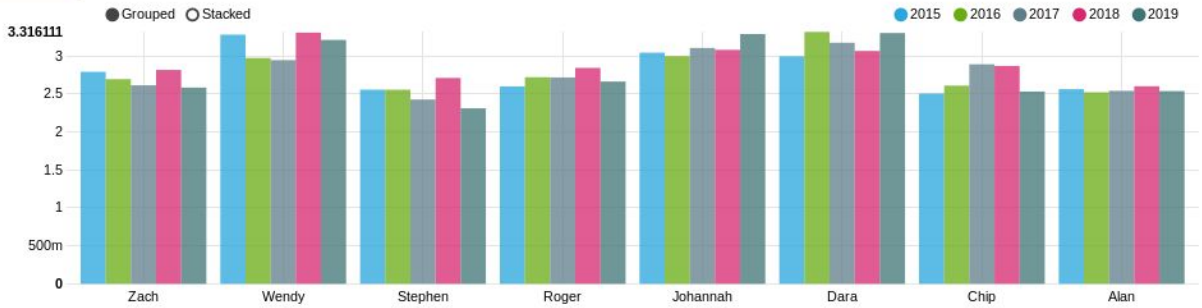
X-AXIS: a.fname

Y-AXIS: totaltime

GROUP: a.year

LIMIT: Limit the number of results to...

SORTING



Analysis on scope 3

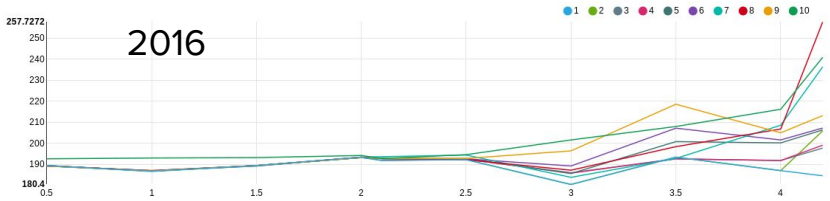
1. Runner and Pace distribution per age group
   Place distribution per finishing time

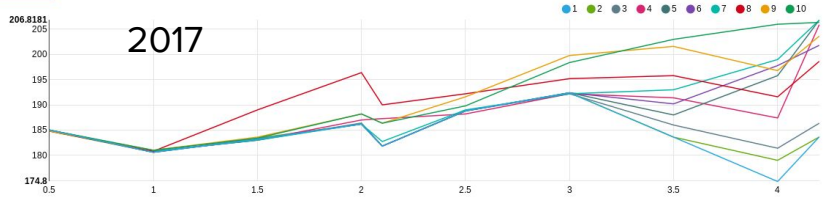
2. Finishing time of each year, top countries, and top loyal runners

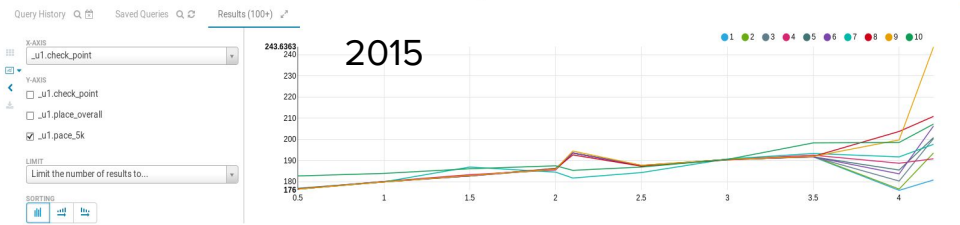
3. How to win ? (analyze gap between elite and average runners)

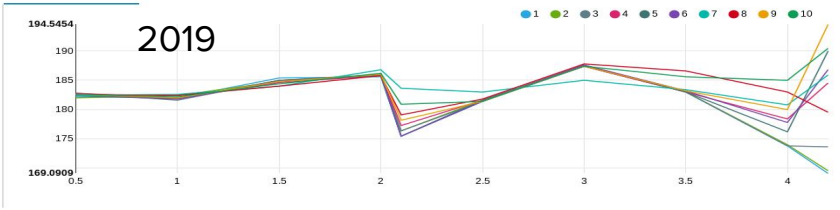
Pace of top 10 runners

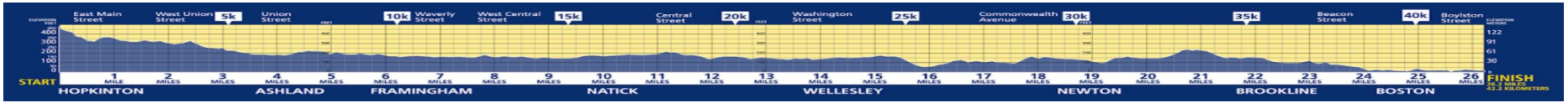
```
select 0.5 as check_point,  place_overall, (UNIX_TIMESTAMP(time_5k,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00','HH:mm:ss')) / 5 as pace_5k from bm_by_year  where place_overall <= 10 and year = 201
UNION ALL
select 1 as check_point,  place_overall,(UNIX_TIMESTAMP(time_10k,'HH:mm:ss') - UNIX_TIMESTAMP(time_5k,'HH:mm:ss'))/5 as pace_5k from bm_by_year where place_overall <= 10 and year = 2015 sort b
UNION ALL
select 1.5 as check_point,  place_overall, (UNIX_TIMESTAMP(time_15k,'HH:mm:ss') - UNIX_TIMESTAMP(time_10k,'HH:mm:ss'))/5 as pace_5k from bm_by_year where place_overall <= 10 and year = 2015 so
UNION ALL
select 2 as check_point,  place_overall, (UNIX_TIMESTAMP(time_20k,'HH:mm:ss') - UNIX_TIMESTAMP(time_15k,'HH:mm:ss'))/5 as pace_5k from bm_by_year where place_overall <= 10 and year = 2015 sort
UNION ALL
select 2.1 as check_point, place_overall, (UNIX_TIMESTAMP(time_half,'HH:mm:ss') - UNIX_TIMESTAMP(time_20k,'HH:mm:ss'))/1.1 as pace_5k from bm_by_year where place_overall <= 10 and year = 2015
UNION ALL
select 2.5 as check_point, place_overall,(UNIX_TIMESTAMP(time_25k,'HH:mm:ss') - UNIX_TIMESTAMP(time_20k,'HH:mm:ss'))/5 as pace_5k from bm_by_year where place_overall <= 10 and year = 2015 sort
UNION ALL
select 3 as check_point, place_overall, (UNIX_TIMESTAMP(time_30k,'HH:mm:ss') - UNIX_TIMESTAMP(time_25k,'HH:mm:ss'))/5 as pace_5k from bm_by_year where place_overall <= 10 and year = 2015 sort
UNION ALL
select 3.5 as check_point, place_overall, (UNIX_TIMESTAMP(time_35k,'HH:mm:ss') - UNIX_TIMESTAMP(time_30k,'HH:mm:ss'))/5 as pace_5k from bm_by_year where place_overall <= 10 and year = 2015 sor
UNION ALL
```







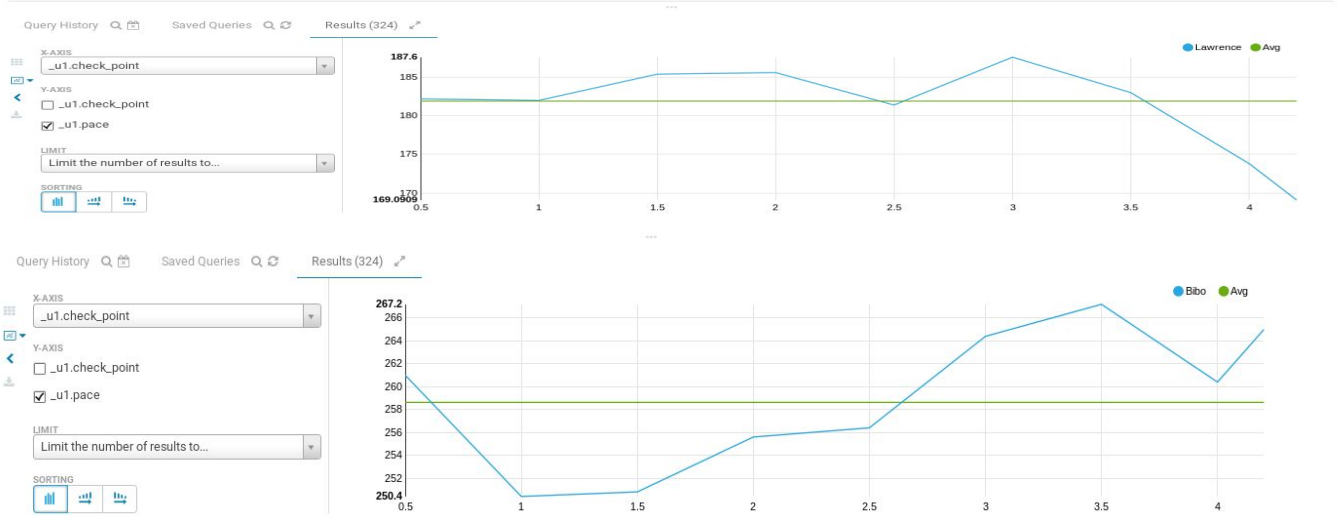




Fast Starts = Slow Finishes

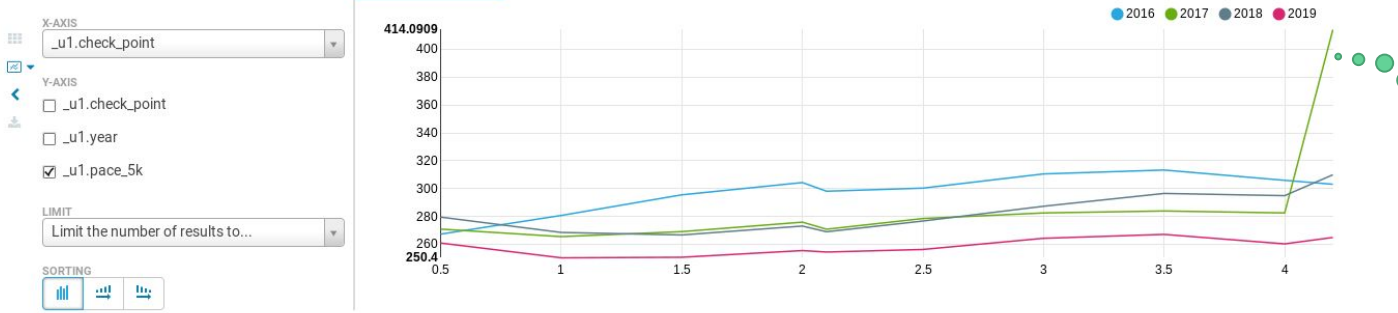
```
select 0.5 as check_point, fname, (UNIX_TIMESTAMP(time_5k,    'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00','HH:mm:ss')) / 5.0 as pace from bm_by_year where pla
UNION ALL select 1.0 as check_point, fname, (UNIX_TIMESTAMP(time_10k,   'HH:mm:ss') - UNIX_TIMESTAMP(time_5k,    'HH:mm:ss')) / 5.0 as pace from bm_by_year where pla
UNION ALL select 1.5 as check_point, fname, (UNIX_TIMESTAMP(time_15k,   'HH:mm:ss') - UNIX_TIMESTAMP(time_10k,   'HH:mm:ss')) / 5.0 as pace from bm_by_year where pla
UNION ALL select 2.0 as check_point, fname, (UNIX_TIMESTAMP(time_20k,   'HH:mm:ss') - UNIX_TIMESTAMP(time_15k,   'HH:mm:ss')) / 5.0 as pace from bm_by_year where pla
UNION ALL select 2.5 as check_point, fname, (UNIX_TIMESTAMP(time_25k,   'HH:mm:ss') - UNIX_TIMESTAMP(time_20k,   'HH:mm:ss')) / 5.0 as pace from bm_by_year where pla
UNION ALL select 3.0 as check_point, fname, (UNIX_TIMESTAMP(time_30k,   'HH:mm:ss') - UNIX_TIMESTAMP(time_25k,   'HH:mm:ss')) / 5.0 as pace from bm_by_year where pla
UNION ALL select 3.5 as check_point, fname, (UNIX_TIMESTAMP(time_35k,   'HH:mm:ss') - UNIX_TIMESTAMP(time_30k,   'HH:mm:ss')) / 5.0 as pace from bm_by_year where pla
UNION ALL select 4.0 as check_point, fname, (UNIX_TIMESTAMP(time_40k,   'HH:mm:ss') - UNIX_TIMESTAMP(time_35k,   'HH:mm:ss')) / 5.0 as pace from bm_by_year where pla
UNION ALL select 4.2 as check_point, fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP(time_40k,   'HH:mm:ss')) / 2.2 as pace from bm_by_year where pla

UNION ALL select 0.5 as check_point, 'Avg' as fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',   'HH:mm:ss')) / 42.2 as pace from bm_by_ye
UNION ALL select 1.0 as check_point, 'Avg' as fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',   'HH:mm:ss')) / 42.2 as pace from bm_by_ye
UNION ALL select 1.5 as check_point, 'Avg' as fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',   'HH:mm:ss')) / 42.2 as pace from bm_by_ye
UNION ALL select 2.0 as check_point, 'Avg' as fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',   'HH:mm:ss')) / 42.2 as pace from bm_by_ye
UNION ALL select 2.5 as check_point, 'Avg' as fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',   'HH:mm:ss')) / 42.2 as pace from bm_by_ye
UNION ALL select 3.0 as check_point, 'Avg' as fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',   'HH:mm:ss')) / 42.2 as pace from bm_by_ye
UNION ALL select 3.5 as check_point, 'Avg' as fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',   'HH:mm:ss')) / 42.2 as pace from bm_by_ye
UNION ALL select 4.0 as check_point, 'Avg' as fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',   'HH:mm:ss')) / 42.2 as pace from bm_by_ye
UNION ALL select 4.2 as check_point, 'Avg' as fname, (UNIX_TIMESTAMP(time_total,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00',   'HH:mm:ss')) / 42.2 as pace from bm_by_ye
```



My pace of each year

```
select 0.5 as check_point,  year, (UNIX_TIMESTAMP(time_5k,'HH:mm:ss') - UNIX_TIMESTAMP('00:00:00','HH:mm:ss')) / 5 as pace_5k from bm_by_year  where fname = 'Bi
UNION ALL
select 1 as check_point,  year,(UNIX_TIMESTAMP(time_10k,'HH:mm:ss') - UNIX_TIMESTAMP(time_5k,'HH:mm:ss'))/5 as pace_5k from bm_by_year where fname = 'Bibo'
UNION ALL
select 1.5 as check_point,  year, (UNIX_TIMESTAMP(time_15k,'HH:mm:ss') - UNIX_TIMESTAMP(time_10k,'HH:mm:ss'))/5 as pace_5k from bm_by_year where fname = 'Bibo'
UNION ALL
select 2 as check_point,  year, (UNIX_TIMESTAMP(time_20k,'HH:mm:ss') - UNIX_TIMESTAMP(time_15k,'HH:mm:ss'))/5 as pace_5k from bm_by_year where fname = 'Bibo'
```



Something different in 2017 ?

Analysis Outcome

1. Most runner age 40-49 and fast runner age 20-30
   Finishing time gap between men and women with same place 30 min

2. The overall average finishing time 3:50
   Top ten countries finishing time 3:45
   Top loyal runners finishing time 2:45

3. Finishing strong - maintaining your pace over the last third of a race while those around you are slowing