商户统计及分析模块Spark

```
%pyspark

# 编写者:凌晨
# 功能:找出美国最常见的前20的商家
# 时间:2024.1.3

from pyspark.sql import HiveContext

hc = HiveContext(sc)
result = hc.sql("SELECT name, COUNT(name) as name_count FROM business GROUP BY name ORDER BY name_count DESC LIMIT 20")

z.show(result)
```

name	name_count
Starbucks	724
McDonald's	703
Dunkin'	510
Subway	459
Taco Bell	365
CVS Pharmacy	345
Walgreens	341
Burger King	338

```
%pyspark

# 编写者:凌晨
# 功能:找出美国最常见的前20的商家,并显示平均得分(对得分进行了数据筛选)
# 时间:2024.1.4

from pyspark.sql import HiveContext

hc = HiveContext(sc)
result = hc.sql("SELECT name, COUNT(name) as name_count,SUM(stars)/COUNT(name) AS avr_stars FROM business WHERE stars IS NOT NULL GROUP BY name ORDER BY name_count DESC LIMIT 20")

z.show(result)
```

Area Chart

name	name_count	avr_stars
Starbucks	724	3.126381215469613
McDonald's	703	1.8634423897581793
Dunkin'	510	2.302941176470588
Subway	459	2.5860566448801743
Taco Bell	365	2.154794520547945
CVS Pharmacy	345	2.4565217391304346
Walgreens	341	2.624633431085044
Burger King	338	2.0281065088757395

```pyspark
%pyspark

# 编写者:凌晨
# 功能:找出商户最多的前5个城市
# 时间:2024.1.3

from pyspark.sql import HiveContext

hc = HiveContext(sc)
result = hc.sql("SELECT city, COUNT(city) as city_count FROM business WHERE city IS NOT NULL GROUP BY city ORDER BY city_count DESC LIMIT 5")

z.show(result)
```

city	city_count
Philadelphia	14569
Tucson	9250
Tampa	9050
Indianapolis	7540
Nashville	6971

```pyspark
%pyspark

# 编写者:凌晨
# 功能:找出商户最多的前10个州
# 时间:2024.1.3

from pyspark.sql import HiveContext

hc = HiveContext(sc)
result = hc.sql("SELECT state, COUNT(state) as state_count FROM business WHERE state IS NOT NULL GROUP BY state ORDER BY state_count DESC LIMIT 10")

```

```
12 z.show(result)
```

state	state_count
PA	34039
FL	26330
TN	12056
IN	11247
MO	10913
LA	9924
AZ	9912
NJ	8536

```
1  %pyspark
2
3  # 编写者:凌晨
4  # 功能:找出商户最多的前10个邮政编码
5  # 时间:2024.1.3
6
7  from pyspark.sql import HiveContext
8
9  hc = HiveContext(sc)
10 result = hc.sql("SELECT postal_code, COUNT(postal_code) AS postal_code_count
   FROM business WHERE postal_code IS NOT NULL GROUP BY postal_code ORDER BY
   postal_code_count DESC LIMIT 10")
11
12 z.show(result)
```

postal_code	postal_code_count
93101	1866
89502	1804
70130	1512
19103	1362
19107	1353
19147	1255
37203	1179
85705	1069

```
1  %pyspark
2
3  # 编写者:凌晨
4  # 功能:找出评分最高的前10个州,评分次数大于20
5  # 时间:2024.1.4
6
7  from pyspark.sql import HiveContext
```

```
8
9  hc = HiveContext(sc)
10 result = hc.sql("SELECT state, SUM(stars)/COUNT(state) AS
   arv_stars,COUNT(state) AS cot FROM business WHERE state IS NOT NULL AND stars
   IS NOT NULL GROUP BY state HAVING cot>20 ORDER BY arv_stars DESC LIMIT 10")
11
12 z.show(result)
```

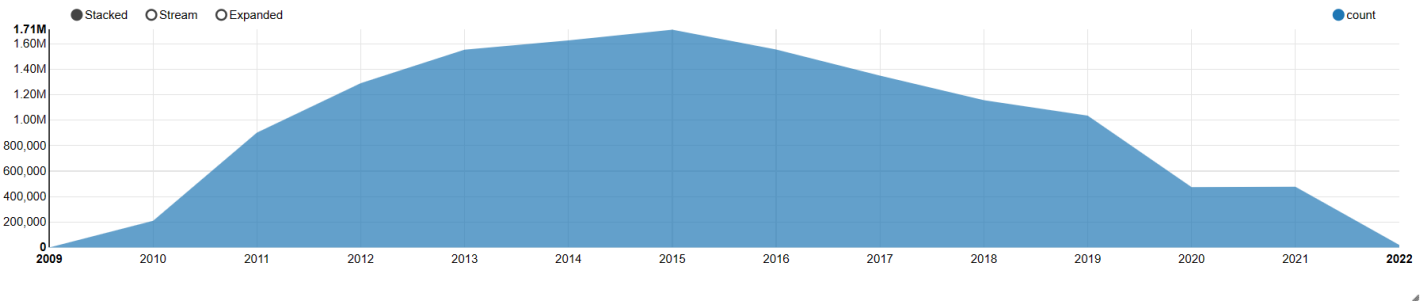
state	arv_stars	cot
CA	3.9967326542379396	5203
NV	3.7368762151652626	7715
ID	3.7076337586747257	4467
LA	3.679161628375655	9924
FL	3.6109570831750855	26330
AZ	3.5920096852300243	9912
IN	3.5882457544234017	11247
PA	3.5730191838773173	34039

```
1  %pyspark
2
3  # 编写者:凌晨
4  # 功能:找出评分最高的前10个城市，评分次数大于20
5  # 时间:2024.1.3
6
7  from pyspark.sql import HiveContext
8
9  hc = HiveContext(sc)
10 result = hc.sql("SELECT city, SUM(stars)/COUNT(city) AS arv_stars,COUNT(city)
   AS cot FROM business WHERE city IS NOT NULL AND stars IS NOT NULL GROUP BY
   city HAVING cot>20 ORDER BY arv_stars DESC,cot DESC LIMIT 10")
11
12 z.show(result)
```

city	arv_stars	cot
Virginia City	4.328125	32
Montecito	4.155913978494624	93
Washington Crossing	4.136363636363637	22
Wyndmoor	4.12962962962963	27
Safety Harbor	4.129496402877698	139
Tierra Verde	4.074074074074074	27
Garnet Valley	4.065217391304348	23
Santa Barbara	4.05144946461217	3829

```
1  %pyspark
2
```

```
# 编写者:钟鹏
# 功能:统计每年打卡次数
# 时间:2024.1.3

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df = hcx.table('checkin')

result = df.select(col('business_id'), explode(split(col('checkin_dates'), ',')).alias('datetime'))\
    .select('business_id', year(to_timestamp(trim(col('datetime')))).alias('year')) \
    .groupBy('year')\
    .count()\
    .orderBy('year', ascending=False)

z.show(result)
```




```
%pyspark

# 编写者:钟鹏
# 功能:统计每月打卡次数
# 时间:2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df = hcx.table('checkin')

result = df.select(col('business_id'), explode(split(col('checkin_dates'), ',')).alias('datetime'))\
```

```
    .select('business_id',
year(to_timestamp(trim(col('datetime')))).alias('year'),
month(to_timestamp(trim(col('datetime')))).alias('month')) \
    .groupBy('year', 'month')\
    .count()\
    .orderBy('year', 'month', ascending=False)

z.show(result)
```

year	month	count
2022	1	20940
2021	12	39383
2021	11	36960
2021	10	41588
2021	9	36224
2021	8	42800
2021	7	47740
2021	6	43422

```
%pyspark

# 编写者:钟鹏
# 功能:统计每周打卡次数
# 时间:2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df = hcx.table('checkin')

result = df.select(col('business_id'), explode(split(col('checkin_dates'),
','')).alias('datetime'))\
    .select('business_id',
year(to_timestamp(trim(col('datetime')))).alias('year'),
weekofyear(to_timestamp(trim(col('datetime')))).alias('week')) \
    .groupBy('year', 'week')\
    .count()\
    .orderBy('year', 'week', ascending=False)

z.show(result)

```

year	week	count
2022	52	2605
2022	3	2638
2022	2	8105
2022	1	7592
2021	53	3916
2021	52	6980
2021	51	7852
2021	50	8662

```
%pyspark

# 编写者:钟鹏
# 功能:统计每季打卡次数
# 时间:2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df = hcx.table('checkin')

result = df.select(col('business_id'), explode(split(col('checkin_dates'), ',')).alias('datetime'))\
    .select('business_id', year(to_timestamp(trim(col('datetime')))).alias('year'), quarter(to_timestamp(trim(col('datetime')))).alias('quarter')) \
    .groupBy('year', 'quarter')\
    .count()\
    .orderBy('year', 'quarter', ascending=False)

z.show(result)

```

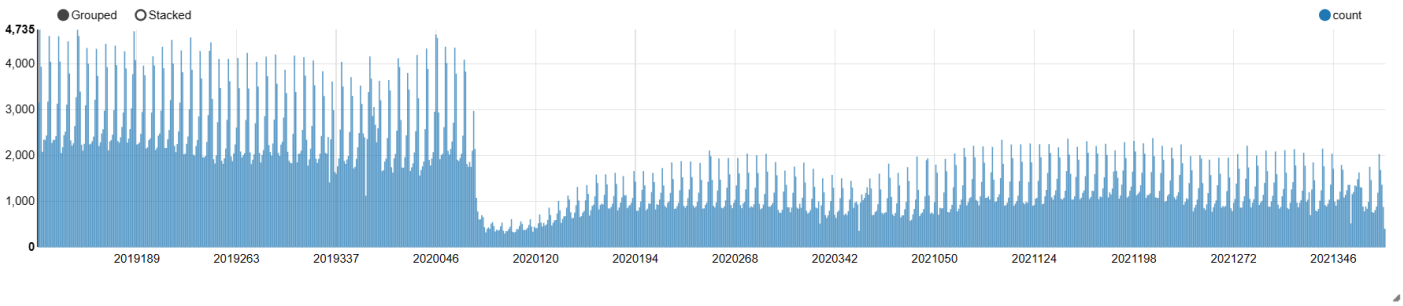
year	quarter	count
2022	1	20940
2021	4	117931
2021	3	126764
2021	2	129905
2021	1	102867
2020	4	99049
2020	3	107778
2020	2	65521

```pyspark
%pyspark

# 编写者：凌晨
# 功能：统计每天打卡次数，year和day为两个属性
# 时间：2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df = hcx.table('checkin')

result = df.select(col('business_id'), explode(split(col('checkin_dates'), ',')).alias('datetime')) \
    .select('business_id', year(to_date(trim(col('datetime')))).alias('year'), dayofyear(to_date(trim(col('datetime')))).alias('day')) \
    .groupBy('year', 'day') \
    .count() \
    .orderBy('year', 'day',ascending=False)

z.show(result)
```

year ▾₂	day	count ▾₁
2021	300	998
2021	82	998
2020	330	997
2020	219	997
2020	198	995
2021	347	994
2021	279	994
2020	358	993

```pyspark
%pyspark

# 编写者：凌晨
# 功能：统计每天打卡次数，year和day合并为一个属性
# 时间：2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

```

```
df = hcx.table('checkin')

result = df.select(col('business_id'), explode(split(col('checkin_dates'), ',')).alias('datetime')) \
    .select('business_id', concat(year(to_date(trim(col('datetime')))), lpad(dayofyear(to_date(trim(col('datetime')))), 3, '0')).alias('year_day')) \
    .groupBy('year_day') \
    .count() \
    .orderBy(desc('year_day'))

z.show(result)
```

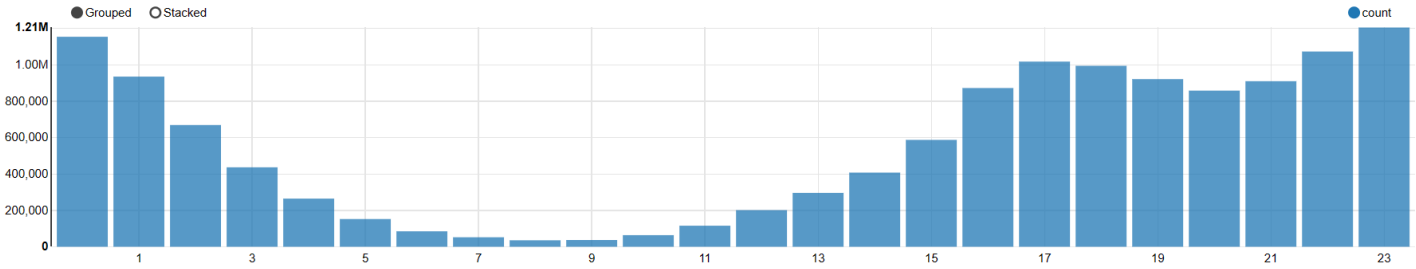



```
%pyspark

# 编写者:钟鹏
# 功能:统计每小时打卡次数
# 时间:2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df = hcx.table('checkin')

result = df.select(col('business_id'), explode(split(col('checkin_dates'), ',')).alias('datetime'))\
    .select('business_id',
year(to_timestamp(trim(col('datetime')))).alias('year'),
month(to_timestamp(trim(col('datetime')))).alias('month'),
dayofmonth(to_timestamp(trim(col('datetime')))).alias('day'),
hour(to_timestamp(trim(col('datetime')))).alias('hour')) \
    .groupBy('year', 'month', 'day', 'hour')\
    .count()\
    .orderBy('year', 'month', 'day', 'hour', ascending=False)
```

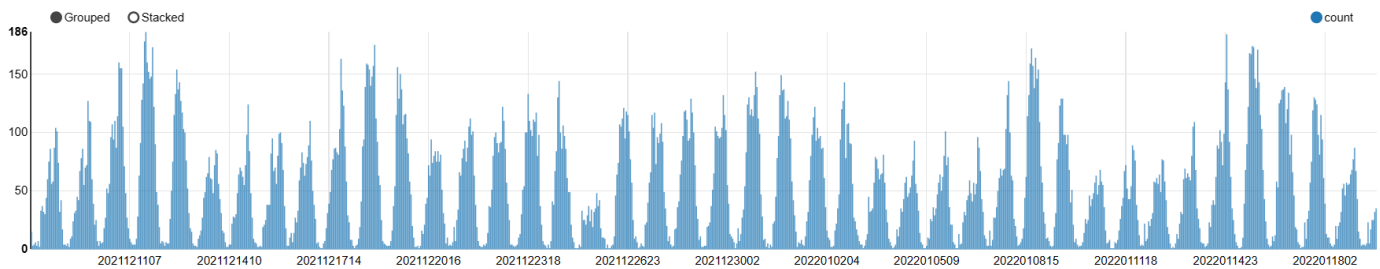
```
19
20 z.show(result)
```

year	month	day	hour	count
2022	1	19	16	35
2022	1	19	15	32
2022	1	19	14	25
2022	1	19	13	25
2022	1	19	12	17
2022	1	19	11	5
2022	1	19	10	23
2022	1	19	9	5

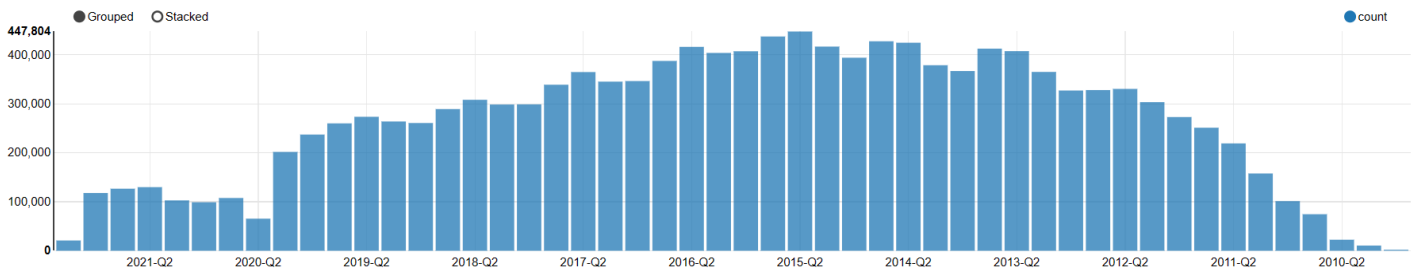
```
1  %pyspark
2  # 编写者:钟鹏
3  # 功能:统计每日打卡高峰期
4  # 时间:2024.1.4
5  from pyspark.sql import HiveContext
6  from pyspark.sql.functions import *
7
8  hcx = HiveContext(sc)
9
10 df = hcx.table('checkin')
11
12 result = df.select(col('business_id'), explode(split(col('checkin_dates'),
   ',')).alias('datetime'))
13
14 result = result.select('business_id', hour(to_timestamp(trim(col('datetime')),
   "yyyy-MM-dd HH:mm:ss")).alias('hour')) \
15
16 result = result.groupBy('hour') \
17
18 result = result.count() \
19
20 result = result.orderBy('hour') \
21
22 z.show(result)
```




```
%pyspark

# 编写者:钟鹏
# 功能:统计每小时打卡次数
# 时间:2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df = hcx.table('checkin')

result = df.select(col('business_id'), explode(split(col('checkin_dates'), ',')).alias('datetime'))\
    .select('business_id', concat(year(to_timestamp(trim(col('datetime')))), lpad(month(to_timestamp(trim(col('datetime')))), 2, '0'), lpad(dayofmonth(to_timestamp(trim(col('datetime')))), 2, '0'), lpad(hour(to_timestamp(trim(col('datetime')))), 2, '0')).alias('datetime_combined')) \
    .groupBy('datetime_combined')\
    .count()\
    .orderBy('datetime_combined', ascending=False)

z.show(result)
```

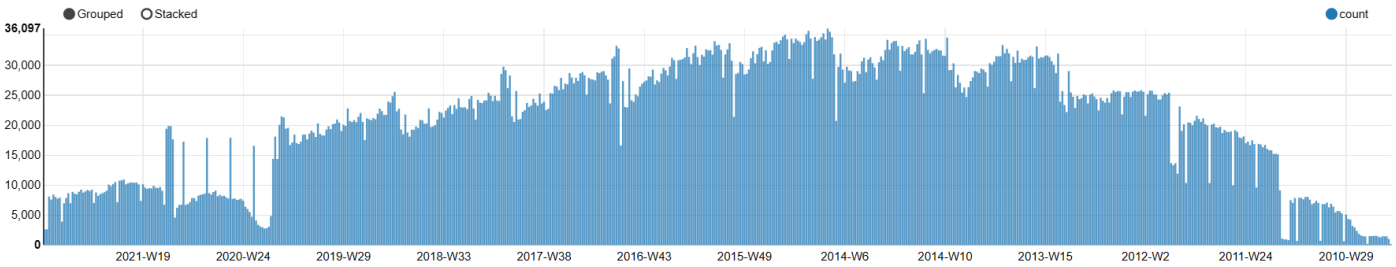


```pyspark
%pyspark

# 编写者:钟鹏
# 功能:统计每季打卡次数
# 时间:2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df = hcx.table('checkin')

result = df.select(col('business_id'), explode(split(col('checkin_dates'), ',')).alias('datetime'))\
    .select('business_id', concat(year(to_timestamp(trim(col('datetime')))), lit('-Q'), quarter(to_timestamp(trim(col('datetime'))))).alias('year_quarter')) \
    .groupBy('year_quarter')\
    .count()\
    .orderBy('year_quarter', ascending=False)

z.show(result)
```

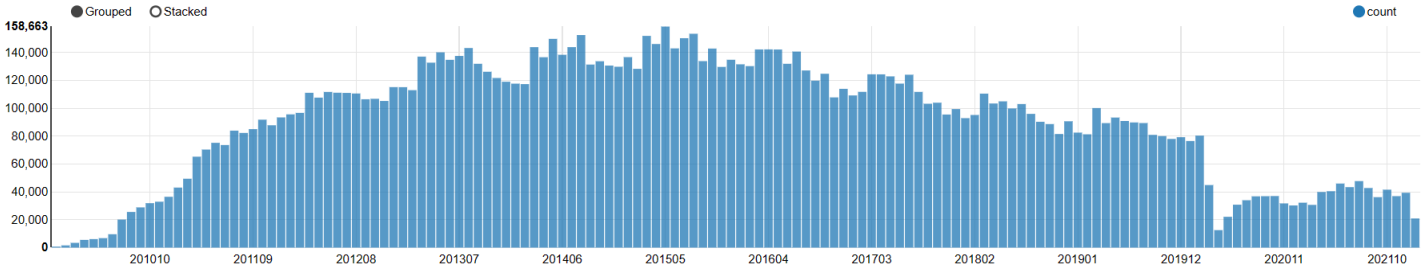



```pyspark
%pyspark

# 编写者:钟鹏
# 功能:统计每周打卡次数
# 时间:2024.1.4

from pyspark.sql import HiveContext
from pyspark.sql.functions import *

hcx = HiveContext(sc)

df = hcx.table('checkin')
```

```
13
14 result = df.select(col('business_id'), explode(split(col('checkin_dates'),
   ',')).alias('datetime'))\
15     .select('business_id', concat(year(to_timestamp(trim(col('datetime')))),
   lit('-W'),
   weekofyear(to_timestamp(trim(col('datetime'))))).alias('year_week')) \
16     .groupBy('year_week')\
17     .count()\
18     .orderBy('year_week', ascending=False)
19
20 z.show(result)
```




```
1 %pyspark
2
3 # 编写者:钟鹏
4 # 功能:统计每月打卡次数
5 # 时间:2024.1.4
6
7 from pyspark.sql import HiveContext
8 from pyspark.sql.functions import *
9
10 hcx = HiveContext(sc)
11
12 df = hcx.table('checkin')
13
14 result = df.select(col('business_id'), explode(split(col('checkin_dates'),
   ',')).alias('datetime'))\
15     .select('business_id', concat(year(to_timestamp(trim(col('datetime')))),
   lpad(month(to_timestamp(trim(col('datetime')))), 2,
   '0')).alias('datetime_combined')) \
16     .groupBy('datetime_combined')\
17     .count()\
18     .orderBy('datetime_combined', ascending=False)
19
20 z.show(result)
```


Grouped
Stacked
count
158,663
140,000
120,000
100,000
80,000
60,000
40,000
20,000
0
201010
201109
201208
201307
201406
201505
201604
201703
201802
201901
201912
202011
202110