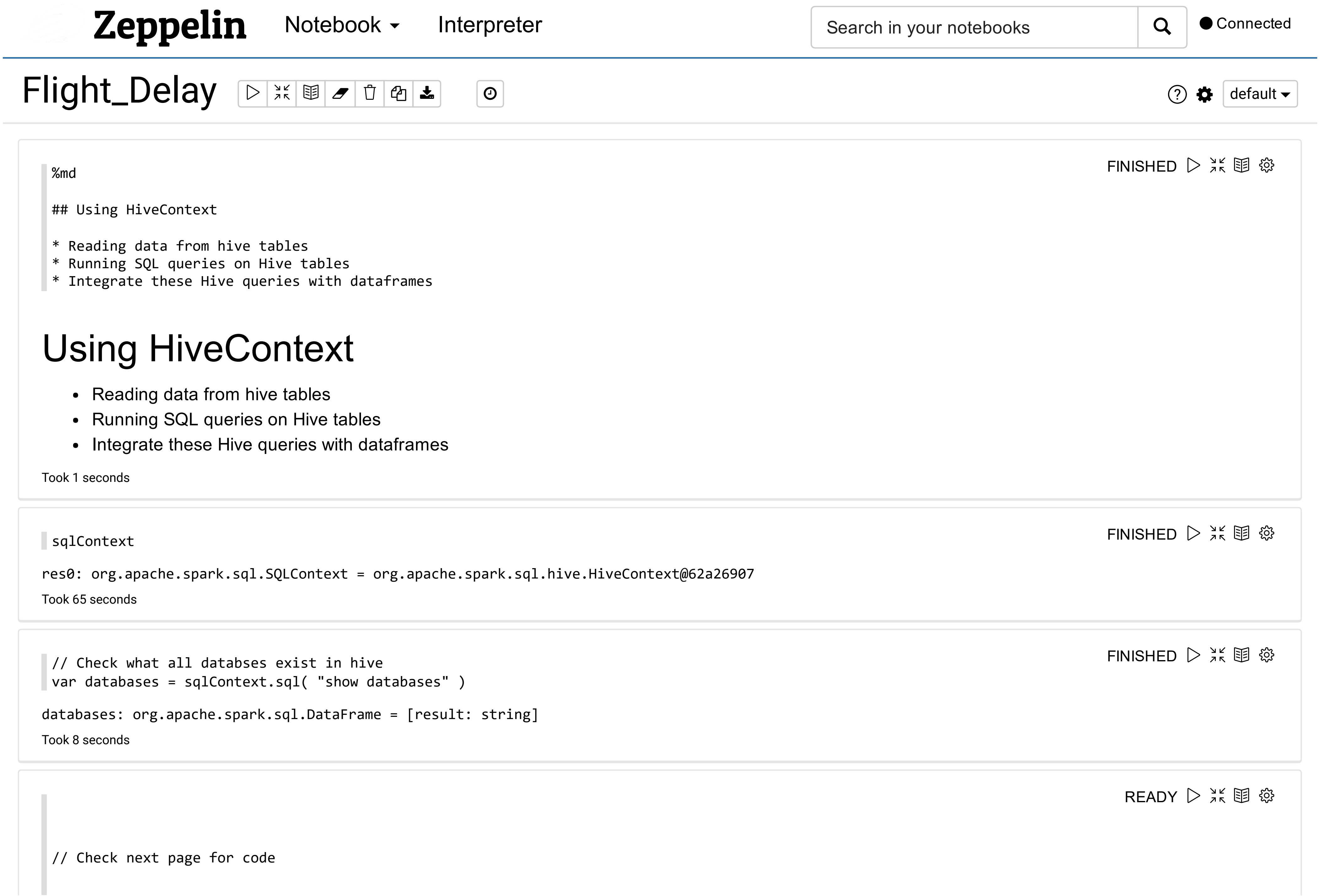


Flight_Delay

```
%md

## Using HiveContext

* Reading data from hive tables
* Running SQL queries on Hive tables
* Integrate these Hive queries with dataframes
```

Using HiveContext

- Reading data from hive tables
- Running SQL queries on Hive tables
- Integrate these Hive queries with dataframes

Took 1 seconds

```
sqlContext
```

```
res0: org.apache.spark.sql.SQLContext = org.apache.spark.sql.hive.HiveContext@62a26907
```

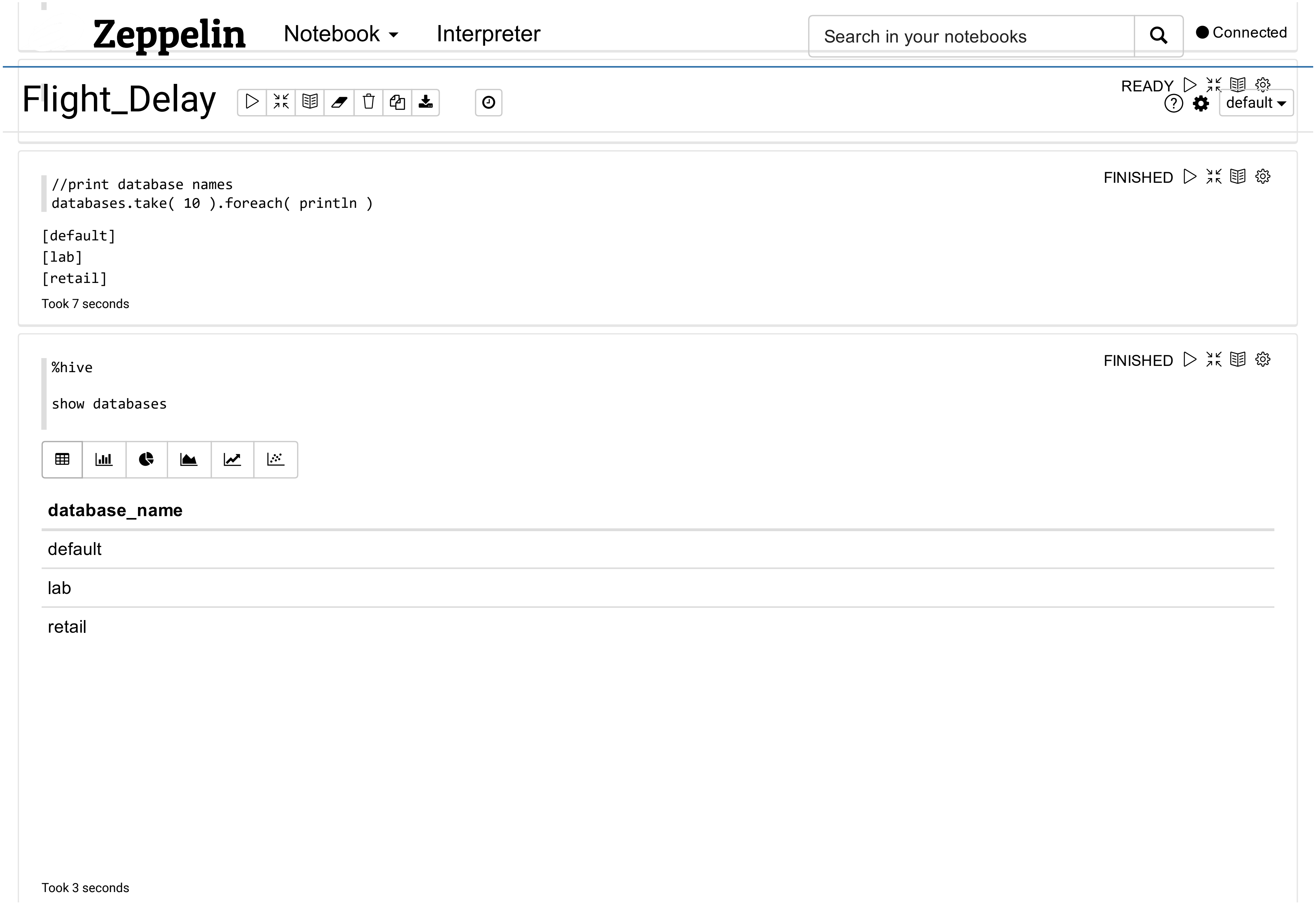
Took 65 seconds

```
// Check what all databses exist in hive
var databases = sqlContext.sql( "show databases" )
```

```
databases: org.apache.spark.sql.DataFrame = [result: string]
```

Took 8 seconds

```
// Check next page for code
```



Flight_Delay

```
//print database names
databases.take( 10 ).foreach( println )
```

```
[default]
[lab]
[retail]
```

Took 7 seconds

```
%hive

show databases
```

database_name
default
lab
retail

Took 3 seconds

READY

READY

FINISHED

```
%hive
use lab
```

Update Count

```
-1
```

Took 0 seconds

FINISHED

```
%hive
select * from flights2008 limit 10
```

flights2008.year	flights2008.month	flights2008.dayofmonth	flights2008.dayofweek	flights2008.deptime	flights2008.crsdeptime	flights2008.arrti
Year	Month	null	null	null	null	null
2,008	1	3	4	2,003	1,955	2,211
2,008	1	3	4	754	735	1,002
2,008	1	3	4	628	620	804
2,008	1	3	4	926	930	1,054
2,008	1	3	4	1,829	1,755	1,959
2,008	1	3	4	1,940	1,915	2,121
2,008	1	3	4	1,937	1,830	2,037
2,008	1	3	4	1,039	1,040	1,132

Took 3 seconds

```
%hive

desc flights2008
```

FINISHED

col_name	data_type
year	string
month	string
dayofmonth	int
dayofweek	int
deptime	int
crsdeptime	int
arrtime	int
crsarrtime	int

crsarrtime	int

Took 0 seconds

FINISHED ▷

```
var delay_by_carriers = sqlContext.sql( "select uniquecarrier, count(*) as num_delays from lab.flights2008 where depdelay > 15 group by uniquecarrie
```

```
delay_by_carriers: org.apache.spark.sql.DataFrame = [uniquecarrier: string, num_delays: bigint]
```

Took 1 seconds

FINISHED ▷

```
delay_by_carriers.cache()

delay_by_carriers.sort( desc( "num_delays" ) ).limit( 10 ).registerTempTable( "delay_by_carriers" )
```

```
res12: org.apache.spark.sql.DataFrame = [uniquecarrier: string, num_delays: bigint]
```

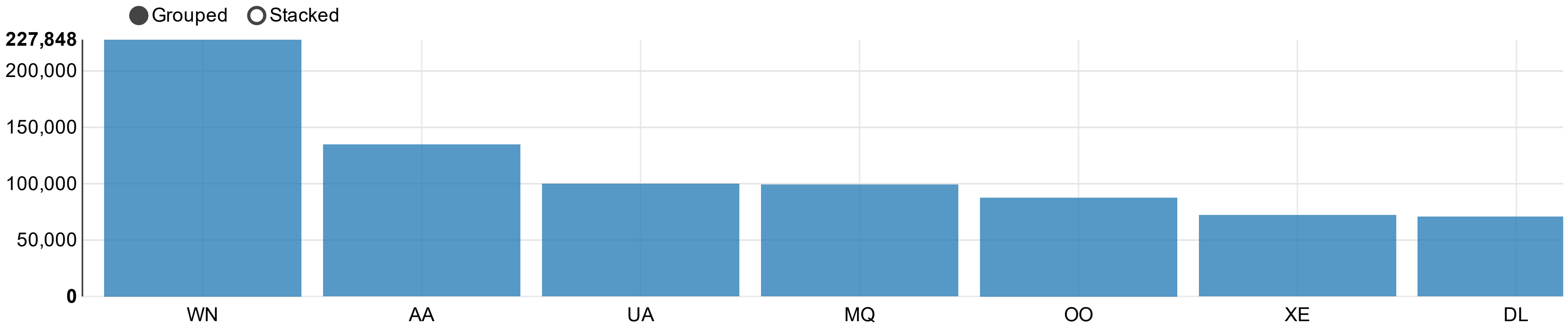
Took 1 seconds

FINISHED ▷

```
%sql

select * from delay_by_carriers limit 10
```

settings ▾



Took 54 seconds (outdated)

FINISHED

```
var delay_by_month = sqlContext.sql( "select Month, AVG( DepDelay ) as DelayInMins from lab.flights2008 group by Month ORDER BY DelayInMins DESC")
delay_by_month.registerTempTable( "delay_by_month" )
```

```
delay_by_month: org.apache.spark.sql.DataFrame = [Month: string, DelayInMins: double]
```

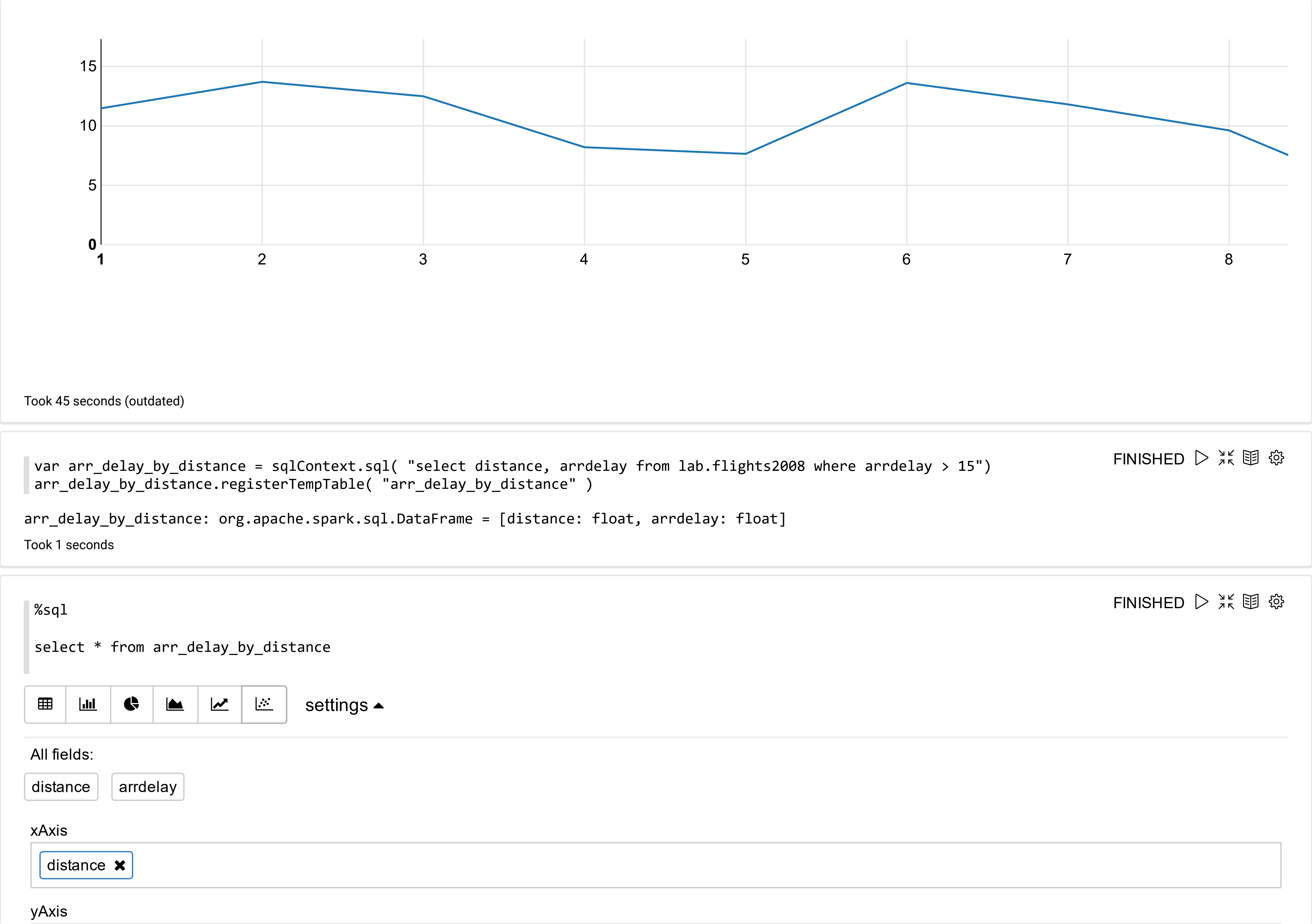
Took 1 seconds

FINISHED

```
%sql

select * from delay_by_month
```

settings ▾



Took 45 seconds (outdated)

```
var arr_delay_by_distance = sqlContext.sql( "select distance, arrdelay from lab.flights2008 where arrdelay > 15")
arr_delay_by_distance.registerTempTable( "arr_delay_by_distance" )
```

```
arr_delay_by_distance: org.apache.spark.sql.DataFrame = [distance: float, arrdelay: float]
```

Took 1 seconds

```
%sql

select * from arr_delay_by_distance
```

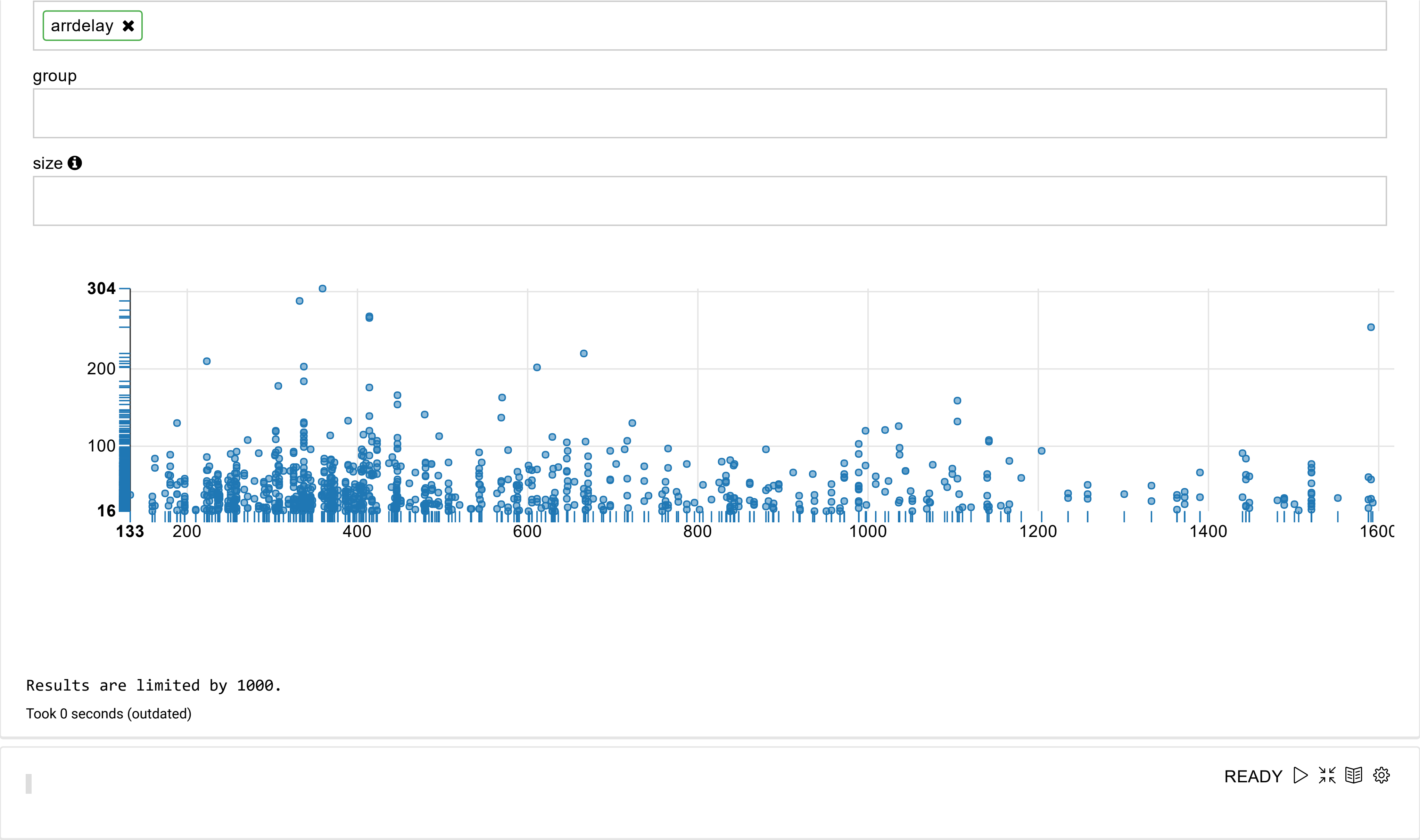
All fields:

distance arrdelay

xAxis

distance ✖

yAxis



```
Results are limited by 1000.
```

Took 0 seconds (outdated)