# into repo2018\_programming\_file\_size\_by\_extension\_99

# run in standard SQL

select

f.repo\_name

, lower(reverse(substr(reverse(path), 0, strpos(reverse(path),'.')))) as extension

, count(distinct lower(reverse(substr(reverse(path), 0, strpos(reverse(path),'.'))))) as extensions\_num

, count(distinct path) as files

, count(distinct f. repo\_name) as repositories

, avg(size) as avg\_size

, avg(case when size > 180000 then 180000 else size end) as capped\_avg\_file

, stddev(size) as std\_size

, stddev(case when size > 180000 then 180000 else size end) as capped\_std\_file

from

`bigquery-public-data.github\_repos.files` as f

join

`bigquery-public-data.github\_repos.contents` as c

on f.id = c.id

Join

`micro-root-221719.git.repos2018` As repos

On

f.repo\_name = repos.repo\_name

Where

lower(reverse(substr(reverse(path), 0, strpos(reverse(path),'.'))))

In

('.bat', '.c', '.cc', '.coffee', '.cpp', '.cs', '.cxx', '.go',

'.groovy', '.hs', '.java', '.js', '.lua', '.m',

'.module', '.php', '.pl', '.pm', '.py', '.rb', '.s', '.scala',

'.sh', '.swift', '.tpl', '.twig')

group by f.repo\_name, extension

order by f.repo\_name, extension

;

# into repos2018\_programming\_files

Select

Repo\_name

, sum(files) as files

From

`micro-root-221719.git.repo2018\_programming\_file\_size\_by\_extension\_99`

Group by

repo\_name

# into repo2018\_programming\_file\_size\_with\_major\_extension99

Select

p.\*

, m.extension as major\_extension

, m.files as major\_files

, m.avg\_size as major\_avg\_size

, m.capped\_avg\_file as major\_capped\_avg\_file

, m.std\_size as major\_std\_size

, m.capped\_std\_file as major\_capped\_std\_file

, 1.0\*m.files/p.files as major\_extension\_ratio

From

`micro-root-221719.git.repos2018\_programming\_files` as p

Join

`micro-root-221719.git.repo2018\_programming\_file\_size\_by\_extension\_99` as m

On

P.repo\_name = m.repo\_name

Where

M.files > p.files/2

;