# Goto Repo properties

# This file contains the query that defines the scope of the research.

# It contains the condition to be included in the research (active enough in two years).

# It also runs the linguistic model in order to analyze later the CCP.

# The model are implemented using regular expression.

# We can count matches by replacing the match twice, once with ‘#’ and then with ‘’ check the # length difference.

# into repos2018

Select

repo\_name

, count(distinct commit) as commits

, count(distinct committer.email) as commiters

, min(committer.date) as start\_time

, max(committer.date) as end\_time

, count(distinct case when (

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

) then commit else null end) as hits

, 1.0\*count(distinct case when

(

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

) then commit else null end) /count(distinct commit) as hit\_ratio

, count(distinct case when year(committer.date) = 2016 then commit else null end) as y2016\_commits

, count(distinct case when

year(committer.date) = 2016 and(

(

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

)

)

then commit else null end) as y2016\_hits

, count(distinct case when year(committer.date) = 2017 then commit else null end) as y2017\_commits

, count(distinct case when

year(committer.date) = 2017 and(

(

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

)

)

then commit else null end) as y2017\_hits

, count(distinct case when year(committer.date) = 2018 then commit else null end) as y2018\_commits

, count(distinct case when

year(committer.date) = 2018 and(

(

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

)

)

then commit else null end) as y2018\_hits

, count(distinct case when

LENGTH(REGEXP\_REPLACE(lower(message)

,r'(\\s|\\.|\\?|\\!|\\[|\\]|\)|\\:|^|$|\\,|\'|")(about|all|also|and|because|but|can|come|could|day|even|find|first|for|from|get|give|have|her|here|him|his|how|into|its|just|know|like|look|make|man|many|more|new|not|now|one|only|other|our|out|people|say|see|she|some|take|tell|than|that|the|their|them|then|there|these|they|thing|think|this|those|time|two|use|very|want|way|well|what|when|which|who|will|with|would|year|you|your)(\\s|\\.|\\?|\\!|\\[|\\]|\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message)

,r'(\\s|\\.|\\?|\\!|\\[|\\]|\)|\\:|^|$|\\,|\'|")(about|all|also|and|because|but|can|come|could|day|even|find|first|for|from|get|give|have|her|here|him|his|how|into|its|just|know|like|look|make|man|many|more|new|not|now|one|only|other|our|out|people|say|see|she|some|take|tell|than|that|the|their|them|then|there|these|they|thing|think|this|those|time|two|use|very|want|way|well|what|when|which|who|will|with|would|year|you|your)(\\s|\\.|\\?|\\!|\\[|\\]|\)|\\:|^|$|\\,|\'|")', '')) > 0

then commit else null end) as English\_hits

from

[bigquery-public-data:github\_repos.commits]

Group by

Repo\_name

Having

count(distinct case when year(committer.date) = 2018 then commit else null end) > 500