In [1]:
```
#!pip install GitPython textblob nltk pandas seaborn matplotlib   --quiet
```

In [2]:
```
# repo = Repo.clone_from("https://github.com/pola-rs/polars", "./_git_polars_/")
```

In [3]:
```
# !rm -rf _git_pandas_
# !rm -rf _git_polars_
```

Loading data and preprocessing

In [15]:
```
import re
from datetime import datetime

import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from git import Repo

sns.set_theme(context="notebook")

pd.options.display.max_colwidth = 300


## clone the repo
# repo = Repo.clone_from("https://github.com/pandas-dev/pandas.git", "./_git_pandas_/")

## with existing repo
repo=Repo("./_git_pandas_/")
```

In [16]:
```
# reading commit messages
commits = list(repo.iter_commits("main"))
commit_details = [
    (commit.committed_datetime.timestamp(), commit.author.email.lower(), commit.message)
    for commit in commits
]
df = pd.DataFrame(commit_details)
df.columns = ["date", "email", "message"]
df["date"] = df["date"].astype("datetime64[s]")

# obfuscating emails
df["email"] = df["email"].apply(
    lambda x: "".join([chr(ord(l) + 1) for l in x.split("@")[0]])[:6]
    + "@"
    + "".join(x.split("@")[1:])
)
```

In [17]:
```
df.head(4)
```

Out[17]:

	date	email	message
0	2024-05-17 16:10:24	uvijot@gmail.com	DOC: Add SA01 ES01 for pandas.Series.dt.components (#58751)\n\n* DOC: add SA01 ES01 for pandas.Series.dt.components\r\n\r\n* DOC: remove SA01 ES01 for pandas.Series.dt.components
1	2024-05-16 21:53:42	217581@users.noreply.github.com	DOC: Sort whatsnew 3.0 entries (#58745)\n\n
2	2024-05-16 21:50:43	njlf/k@gmail.com	DOC: Update orc.py to say orc instead of parquet (#58744)\n\nUpdate orc.py to say orc instead of parquet
3	2024-05-16 21:46:49	217581@users.noreply.github.com	CLN: Require ExtensionArray._reduce(keepdims=) (#58739)\n\n

In [18]:
```
df.shape
```

Out[18]:
```
(35011, 3)
```

In [19]:
```
df.head(3)
```

Out[19]:

	date	email	message
0	2024-05-17 16:10:24	uvijot@gmail.com	DOC: Add SA01 ES01 for pandas.Series.dt.components (#58751)\n\n* DOC: add SA01 ES01 for pandas.Series.dt.components\r\n\r\n* DOC: remove SA01 ES01 for pandas.Series.dt.components
1	2024-05-16 21:53:42	217581@users.noreply.github.com	DOC: Sort whatsnew 3.0 entries (#58745)\n\n
2	2024-05-16 21:50:43	njlf/k@gmail.com	DOC: Update orc.py to say orc instead of parquet (#58744)\n\nUpdate orc.py to say orc instead of parquet

In [20]:
```
# clean messages, extract words in UPPER CASE , extract abbreviations
import re

def clean_message(message):
    # lower case, replace newlines, git-related words
    message = re.sub(
        r"merge pull request #\d+|\s+|\:|git|github|-svn-id|commit|\(\#\d+\)|\[pre\-\.ci\]|(\w+\-){
        " ",
        message,
    ).strip()

    message = re.sub(
        r"—",
        " ",
        message,
    ).strip()

    return message

df["raw"]=df['message']

df["message"] = df["message"].apply(clean_message)
```

In [21]:
```
df.head(4)
```

Out[21]:

	date	email	message	ra
0	2024-05-17 16:10:24	uvijot@gmail.com	DOC Add SA01 ES01 for pandas.Series.dt.components * DOC add SA01 ES01 for pandas.Series.dt.components * DOC remove SA01 ES01 for pandas.Series.dt.components	DOC: Add SA01 ES01 pandas.Series.dt.componer (#58751)\n\n* DOC: add SA01 ES pandas.Series.dt.components\r\n\r DOC: remove SA01 ES01 pandas.Series.dt.componer
1	2024-05-16 21:53:42	217581@users.noreply.github.com	DOC Sort whatsnew 3.0 entries	DOC: Sort whatsnew 3.0 entri (#58745)\r
2	2024-05-16 21:50:43	njlf/k@gmail.com	DOC Update orc.py to say orc instead of parquet Update orc.py to say orc instead of parquet	DOC: Update orc.py to say c instead of parqu (#58744)\n\nUpdate orc.py to s orc instead of parqu
3	2024-05-16 21:46:49	217581@users.noreply.github.com	CLN Require ExtensionArray._reduce(keepdims=)	CLN: Requ ExtensionArray._reduce(keepdims (#58739)\r

In [22]:
```
df["tkns_in_upper_case"] = df["message"].apply(
    lambda x: set(re.findall(" ([A-Z]{3,})", x))
)
df["ntkns_in_upper_case"] = df["tkns_in_upper_case"].str.len()

abbrv = list()
for cell in df[df["ntkns_in_upper_case"] > 0]["tkns_in_upper_case"].to_list():
    for tk in cell:
```

```
        abbrv.append(tk)

print("abbreviations extracted", len(abbrv), ", some examples:", abbrv[:10])
abbrv = pd.Series(abbrv).value_counts().head(50).index.to_list()
abbrv.extend(["AFAIK", "MANIFEST"])

df["abbrv"] = df["tkns_in_upper_case"].apply(
    lambda x: ",".join(x.intersection(set(abbrv)))
)

df["tkns_in_upper_case"] = df["tkns_in_upper_case"].apply(
    lambda x: ",".join(x.difference(set(abbrv)))
)
df["ntkns_in_upper_case"] = df["tkns_in_upper_case"].str.split(",").str.len()
```

```
abbreviations extracted 9601 , some examples: ['DOC', 'CLN', 'COW', 'DOC', 'DOC', 'DOC', 'NDF', 'CL
N', 'DOC', 'DOC']
```

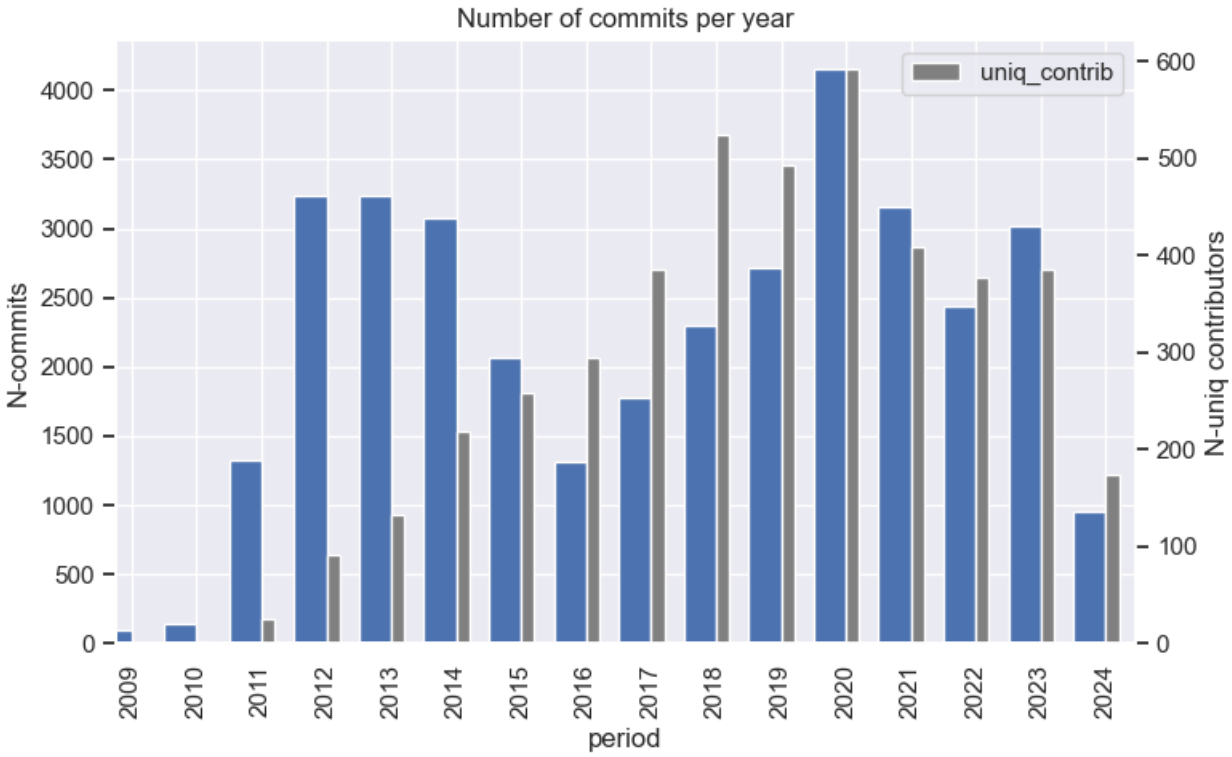
In [23]:
```
df[df["ntkns_in_upper_case"] > 3].head(1)
```

Out[23]:

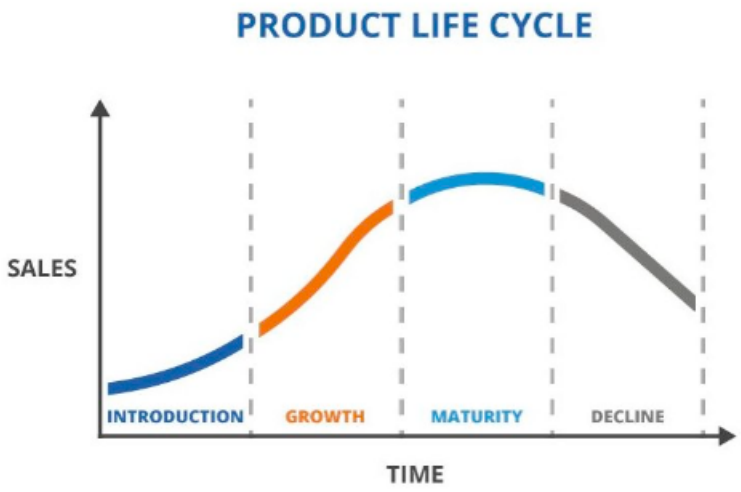
	date	email	message	raw	tkns_in_upper_case	ntkns_in_upp
160	2024-04-30 00:25:50	lp31:2@nyu.edu	DOC fixing RT03 erros for Index duplicated and nunique * DOC fixing RT03 erros for Index duplicated and nunique * deleting it lines from code_checks * fixing EXPECTED TO FAIL, BUT NOT FAILING error * fixing code_checks issue * fixed Expected to fail error	DOC: fixing RT03 erros for Index: duplicated and nunique (#58432)\n\n* DOC: fixing RT03 erros for Index: duplicated and nunique\r\n\r\n* deleting it lines from code_checks\r\n\r\n* fixing EXPECTED TO FAIL, BUT NOT FAILING error\r\n\r\n* fixing code_checks issue\r\n\r\n* fixed Expected to fail error	FAIL,EXPECTED,NOT,FAILING,BUT	

Descriptive stats

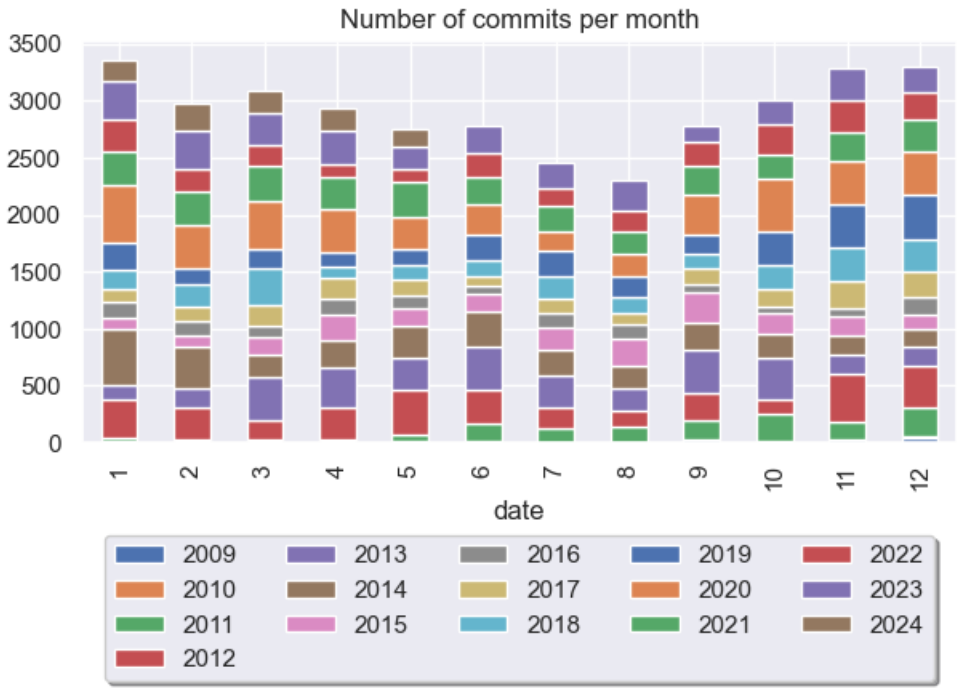
In [24]:
```
fig, ax = plt.subplots(1, figsize=(8, 5), sharex=True)
ax2 = ax.twinx()

df["period"] = df["date"].dt.to_period("1Y")
df.groupby("period").size().plot(
    ax=ax, kind="bar", title="Number of commits per year", position=1
)
df.groupby("period").agg(uniq_contrib=("email", "nunique")).plot(
    ax=ax2, kind="bar", width=0.2, grid=False, position=0, color="grey"
)
ax.set_ylabel("N-commits")
ax2.set_ylabel("N-uniq contributors")
plt.tight_layout()
plt.show()
```



In [25]:
```
## ^ this might reflect the product phase
```

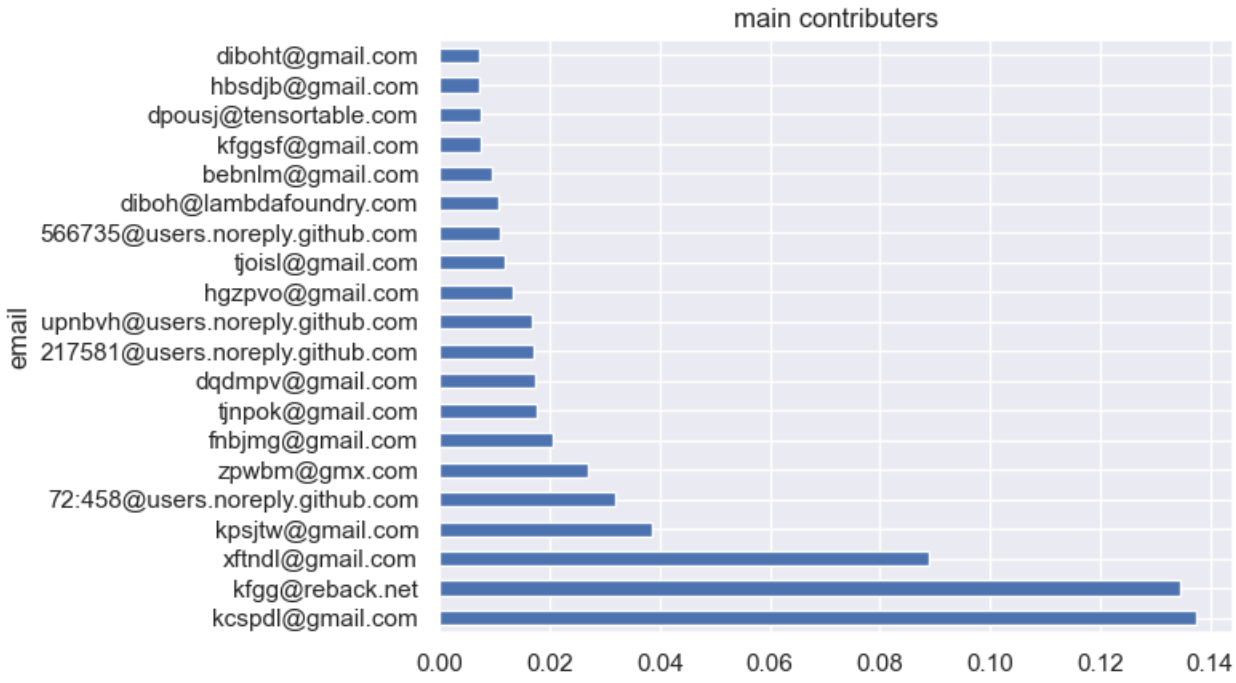


In [26]:
```
# seasonality, when the project is progressing

fig, ax = plt.subplots(1)
df.groupby([df.date.dt.month, df.date.dt.year]).size().unstack().plot(
    ax=ax, kind="bar", stacked=True, title="Number of commits per month"
)
ax.legend(
    loc="upper center", bbox_to_anchor=(0.5, -0.2), fancybox=True, shadow=True, ncol=5
)
plt.tight_layout()
plt.show()
```



In [27]:
```
# main contributers
df["email"].value_counts(normalize=True).head(20).plot(
    kind="barh", title="main contributers"
)
```

Out[27]: <Axes: title={'center': 'main contributers'}, ylabel='email'>



word frequencies

In [28]:
```
from collections import Counter
```

In [29]:
```
all_words = " ".join(df["message"].to_list()).split()
common_words = Counter(all_words).most_common(1000)
```

In [30]:
```
common_words[:20]
```

Out[30]:
```
[('*', 20363),
 ('to', 10367),
 ('in', 10198),
 ('for', 9140),
 ('BUG', 7742),
 ('DOC', 6362),
 ('and', 6017),
 ('from', 5930),
 ('with', 5295),
 ('the', 5183),
 ('of', 4952),
 ('fix', 3767),
 ('TST', 3730),
 ('ENH', 3424),
 ('Merge', 3285),
 ('test', 3280),
 ('a', 3118),
 ('Fix', 3111),
 ('on', 2957),
 ('pull', 2935)]
```

^ not very meaningfull , what we can do ?

In [31]:
```
## Stop word - is a commonly used word, which doesn't add much of additional information
```

In [32]:
```
from nltk.corpus import stopwords
```

In [33]:
```
list(stopwords.words("english"))[:10]
```

Out[33]:
```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're"]
```

In [34]:
```
stopwords.words("english").extend(abbrv)
```

In [35]:
```
[(word, count, word in stopwords.words("english")) for word, count in common_words][:20]
```

Out[35]:
```
[('*', 20363, False),
 ('to', 10367, True),
 ('in', 10198, True),
 ('for', 9140, True),
 ('BUG', 7742, False),
 ('DOC', 6362, False),
 ('and', 6017, True),
 ('from', 5930, True),
 ('with', 5295, True),
 ('the', 5183, True),
 ('of', 4952, True),
 ('fix', 3767, False),
 ('TST', 3730, False),
 ('ENH', 3424, False),
 ('Merge', 3285, False),
 ('test', 3280, False),
 ('a', 3118, True),
 ('Fix', 3111, False),
 ('on', 2957, True),
 ('pull', 2935, False)]
```

In [36]:
```
# extending stop words

my_stopwords = set(stopwords.words("english"))
my_stopwords.update(
    {"fix", "add", "python", "type", "test", "text", "change", "file", "make", "master"}
)
my_stopwords.update(set(abbrv))
```

In [37]:
```
[(word, count) for word, count in common_words if word not in my_stopwords][:20]
```

Out[37]:
```
[('*', 20363),
 ('Merge', 3285),
 ('Fix', 3111),
 ('pull', 2935),
 ('request', 2932),
 ('tests', 2782),
 ('Add', 2594),
 ('remove', 1967),
 ('Update', 1801),
 ('Co', 1791),
 ('authored', 1780),
 ('use', 1406),
 ('Remove', 1391),
 ('>', 1368),
 ('whatsnew', 1304),
 ('Closes', 1224),
 ('CI', 1196),
 ('closes', 1182),
 ('docstring', 1171),
 ('update', 1153)]
```

^ - we have same words here, like 'fixed' and 'fixes' ; TODO: fine tune stop words - as not meaningfull

Tokenization, lemmatization

Token - is the part of the text, i.e. sentence, word. Here we'll split text by word.

Lemma - is the root of the word, i.e. "fixing" "fixed" "to fix" would be transformed into a single word

In [38]:
```
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize


def get_lemmas(text):
    tokens = word_tokenize(text.lower())
    lemmatizer = WordNetLemmatizer()
    stop_words = my_stopwords
    processed_tokens = [
        lemmatizer.lemmatize(word)
        for word in tokens
        if word not in stop_words and word.isalpha()
    ]
    return " ".join(processed_tokens)


def get_tagged_lemmas(text):
    tokens = word_tokenize(text.lower())
    lemmatizer = WordNetLemmatizer()
    stop_words = my_stopwords
    processed_tokens = [
        lemmatizer.lemmatize(word)
        for word in tokens
        if word not in stop_words and word.isalpha()
    ]

    tagged = nltk.tag.pos_tag(processed_tokens)
    return tagged
```

In [39]:
```
df["message_lemmas"] = df["message"].apply(get_lemmas)
```

In [40]:
```
df["message_tagged_lemmas"] = df["message"].apply(get_tagged_lemmas)
```

In [41]:
```
df[["message", "message_lemmas", "message_tagged_lemmas"]].head(3)
```

Out[41]:

	message	message_lemmas	message_tagged_lemmas
0	DOC Add SA01 ES01 for pandas.Series.dt.components * DOC add SA01 ES01 for pandas.Series.dt.components * DOC remove SA01 ES01 for pandas.Series.dt.components	doc doc doc remove	[(doc, NN), (doc, NN), (doc, NN), (remove, VB)]
1	DOC Sort whatsnew 3.0 entries	doc sort whatsnew entry	[(doc, NN), (sort, NN), (whatsnew, VBD), (entry, NN)]
2	DOC Update orc.py to say orc instead of parquet Update orc.py to say orc instead of parquet	doc update say orc instead parquet update say orc instead parquet	[(doc, JJ), (update, JJ), (say, VBP), (orc, JJ), (instead, RB), (parquet, JJ), (update, JJ), (say, VBP), (orc, JJ), (instead, RB), (parquet, NN)]

In [42]:
```
from collections import Counter

all_words = " ".join(df["message_lemmas"].to_list()).split()
common_words = Counter(all_words).most_common(1000)
```

In [43]:
```
top_words=[(word, count) for word, count in common_words if word not in my_stopwords][:50]
top_words
```

Out[43]:
```
[('bug', 8987),
 ('doc', 8540),
 ('tst', 3775),
 ('merge', 3650),
 ('enh', 3468),
 ('remove', 3420),
 ('update', 3141),
 ('close', 3141),
 ('request', 2946),
 ('pull', 2938),
 ('cln', 2937),
 ('ref', 2117),
 ('added', 2073),
 ('use', 2052),
 ('index', 1911),
 ('authored', 1799),
 ('co', 1798),
 ('error', 1755),
 ('ci', 1512),
 ('series', 1486),
 ('method', 1453),
 ('perf', 1402),
 ('whatsnew', 1382),
 ('docstring', 1369),
 ('column', 1330),
 ('issue', 1297),
 ('fixed', 1248),
 ('string', 1212),
 ('gh', 1179),
 ('warning', 1143),
 ('dtype', 1117),
 ('depr', 1088),
 ('dataframe', 1070),
 ('example', 1060),
 ('value', 1031),
 ('check', 1015),
 ('author', 1011),
 ('function', 1003),
 ('note', 987),
 ('name', 981),
 ('non', 938),
 ('groupby', 936),
 ('move', 929),
 ('api', 923),
 ('support', 911),
 ('code', 900),
 ('following', 873),
 ('object', 858),
 ('typ', 819),
 ('return', 814)]
```

In [44]:
```
df[['message_lemmas','message']]
```

Out[44]:

	message_lemmas	message
0	doc doc doc remove	DOC Add SA01 ES01 for pandas.Series.dt.components * DOC add SA01 ES01 for pandas.Series.dt.components * DOC remove SA01 ES01 for pandas.Series.dt.components
1	doc sort whatsnew entry	DOC Sort whatsnew 3.0 entries
2	doc update say orc instead parquet update say orc instead parquet	DOC Update orc.py to say orc instead of parquet Update orc.py to say orc instead of parquet
3	cln require	CLN Require ExtensionArray._reduce(keepdims=)
4	cln misc copy write cleanup cln remove remove unused context remove cow comment arrow remove copy	CLN Misc copy on write cleanups * CLN Remove using_cow * Remove unused contexts * Remove some COW comments in arrow * Remove other copy
...	...	...
35006	first cleaned code	first with cleaned up code
35007	added svn ignore	added svn ignore
35008	oops	oops
35009	adding trunk	adding trunk
35010	initial directory structure	Initial directory structure.

35011 rows × 2 columns

classifying commits by using lemmas

In [45]:
```
df["is_bug"] = df["message_lemmas"].apply(
    lambda x: bool(
        [
            token
            for token in x.split(" ")
            if token in ["bug", "issue", "fix", "correct", "fixes", "fixed", "error"]
        ]
    )
)
```

In [46]:
```
df = df.join(
    df[~df["is_bug"]]["message_lemmas"]
    .apply(
        lambda x: bool(
            [token for token in x.split(" ") if token in ["feature", "whatsnew", "new"]]
        )
    )
    .rename("is_feature")
)
```

In [47]:
```
df["is_feature"] = df["is_feature"].fillna(False)
```

```
/var/folders/7r/4f58k_j13ks_wpyxs010dk8m0000gn/T/ipykernel_18372/581789242.py:1: FutureWarning: Dow
ncasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will change in a future v
ersion. Call result.infer_objects(copy=False) instead. To opt-in to the future behavior, set `pd.se
t_option('future.no_silent_downcasting', True)`
  df["is_feature"] = df["is_feature"].fillna(False)
```

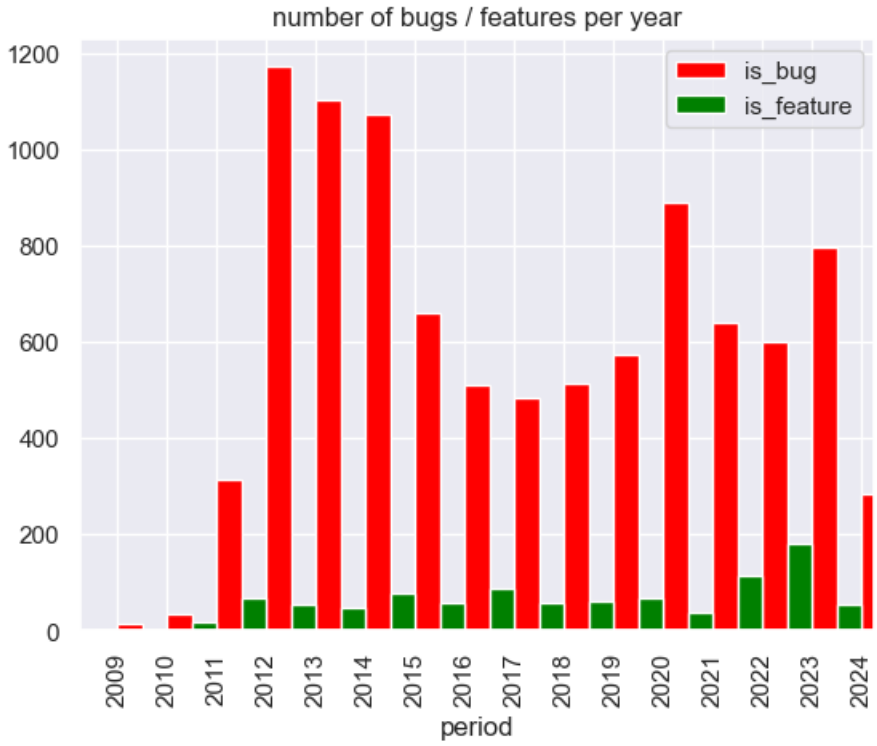
In [48]:
```
# vocabulary
vocabulary = (
    df.set_index(["is_bug", "is_feature"])["message_lemmas"]
    .str.split(" ")
    .explode()
    .drop_duplicates()
    .reset_index()
)
```

In [49]:
```
bug_vocab=list(vocabulary[vocabulary["is_bug"]]['message_lemmas'].unique())
```

In [50]:
```
feature_vocab=list(vocabulary[vocabulary["is_feature"]]['message_lemmas'].unique())
```

In [51]:
```
df.groupby("period")["is_bug"].sum().plot(kind="bar", position=0, color="red")
df.groupby("period")["is_feature"].sum().plot(
    kind="bar", position=1, color="green", title="number of bugs / features per year"
)
plt.legend()
```

Out[51]:
```
<matplotlib.legend.Legend at 0x1436abfe0>
```



Triggering words

In [52]:
```
## lets search for any agressive or bad wourds in messages
bad_words = pd.read_csv("https://www.cs.cmu.edu/~biglou/resources/bad-words.txt")
```

In [53]:
```
bad_words.columns = ["word"]
```

In [54]:
```
bad_words = bad_words["word"].apply(WordNetLemmatizer().lemmatize).to_list()
```

In [55]:
```
my_stopwords.update(
    {
        "failure",
        "black",
        "failed",
        "executed",
        "execute",
        "reject",
        "crash",
        "corruption",
    }
)
```

In [56]:
```
df["bad_words"] = df["message_lemmas"].apply(
    lambda x: ",".join(
        list(
            set(
```

```
                token
                for token in x.split(" ")
                if all([token in bad_words, token not in my_stopwords])
            )
        )
    )
)
```

In [57]:
```
df[df["bad_words"].str.len() > 0]["bad_words"].unique()
```

Out[57]:
```
array(['hook', 'bigger', 'period', 'color', 'white,color', 'dead',
       'white', 'roach,pro', 'pro', 'fire', 'period,hook', 'mad',
       'stupid', 'colored', 'destroy', 'chinese', 'meth', 'fat', 'screw',
       'execution', 'die', 'dong,period,color', 'lynch', 'dong', 'latin',
       'scum', 'illegal', 'crack', 'remains', 'period,shit', 'japanese',
       'dumb', 'premature', 'christian', 'bast', 'cum', 'laid',
       'period,meth', 'swallow', 'abuse', 'israel', 'fu', 'bi',
       'white,conservative', 'explosion', 'harder', 'burn', 'hell', 'dy',
       'barf', 'european', 'nuke', 'killed', 'hole', 'mole'], dtype=object)
```

In [58]:
```
df[df["bad_words"].str.len() > 0]["email"].value_counts().head(5)
```

Out[58]:
```
email
kcspdl@gmail.com    45
xftndl@gmail.com    42
kfgg@reback.net     28
zpwbm@gmx.com       28
tjoisl@gmail.com    26
Name: count, dtype: int64
```

In [59]:
```
df[df["bad_words"] == "dumb"]
```

Out[59]:

	date	email	message	raw
17444	2018-08-07 18:00:45	kcspdl@gmail.com	~Finish collecting m8[ns] tests, start collecting division by zero tests * implement box fixture, move a couple tests from timedelta.test_arithmetic, parametrize more * port floordiv tests * port test_td64arr_rfloordiv_tdlike_scalar * make fixtures, port last of TimedeltaIndex div/mul tests * ...	~Finish collecting m8[ns] tests, start collecting division by zero tests (#22153)\n\n* implement box fixture, move a couple tests from timedelta.test_arithmetic, parametrize more\r\n\r\n* port floordiv tests\r\n\r\n* port test_td64arr_rfloordiv_tdlike_scalar\r\n\r\n* make fixtures, port last of ...
31356	2012-09-08 02:28:40	cfokbn@gmail.com	undo dumb setuptools bug clobbering .pyx sources back to .c setuptools (not distribute) will replace '.pyx' extensions with '.c' if *pyrex* is not importable. This checks if that happened, and reverses it if so. closes #1805	undo dumb setuptools bug clobbering .pyx sources back to .c\n\nsetuptools (not distribute) will replace '.pyx' extensions with '.c'\nif *pyrex* is not importable. This checks if that happened,\nand reverses it if so.\n\ncloses #1805\n
32696	2012-03-21 20:13:08	bebnlm@gmail.com	ENH added dumb snap function to datetimeindex, to get to nearest offset	ENH: added dumb snap function to datetimeindex, to get to nearest offset\n

Sentiment analysis

example: lets search for very negative and very positive sentiment words

In [60]:
```
from nltk.corpus import sentiwordnet as swn
```

In [61]:
```
list(swn.senti_synsets("bug"))
```

Out[61]:
```
[SentiSynset('bug.n.01'),
 SentiSynset('bug.n.02'),
 SentiSynset('bug.n.03'),
 SentiSynset('hemipterous_insect.n.01'),
 SentiSynset('microbe.n.01'),
 SentiSynset('tease.v.01'),
 SentiSynset('wiretap.v.01')]
```

In [62]:
```
list(swn.senti_synsets("feature", "n"))[0].neg_score()
```

Out[62]:
```
0.0
```

In [63]:
```
list(swn.senti_synsets("exciting", "a"))[0].pos_score()
```

Out[63]:
```
0.375
```

In [64]:
```
swn_tagpos_mapping = {"NN": "n", "VBG": "s", "RB": "a", "FW": "n"}
```

In [65]:
```
def get_neg_sentiments_by_word(text, threshold=0.80):
    tokens = set()

    if not text:
        return None

    for token, pos in text:
        try:
            if pos_ := swn_tagpos_mapping.get(pos):
                score = swn.senti_synsets(token, pos_).__next__().neg_score()
                if score > threshold:
                    tokens.add((token, pos, score))
        except Exception as e:
            pass

    if tokens:
        return tokens
    else:
        return None


def get_pos_sentiments_by_word(text, threshold=0.80):
    tokens = set()

    if not text:
        return None

    for token, pos in text:
        try:
            if pos_ := swn_tagpos_mapping.get(pos):
                score = swn.senti_synsets(token, pos_).__next__().pos_score()
                if score > threshold:
                    tokens.add((token, pos, score))
        except Exception as e:
            pass

    if tokens:
        return tokens
    else:
        return None
```

In [66]:
```
df["neg_words"] = df["message_tagged_lemmas"].apply(
    lambda x: get_neg_sentiments_by_word(x, 0.5)
)
```

In [67]:
```
df["pos_words"] = df["message_tagged_lemmas"].apply(
    lambda x: get_pos_sentiments_by_word(x, 0.5)
)
```

In [68]:
```
df[df["neg_words"].notnull()]["neg_words"].value_counts()[:10]
```

Out[68]:
```
neg_words
{(error, NN, 0.625)}                          1182
{(still, RB, 0.625)}                            87
{(problem, NN, 0.625)}                          68
{(difference, NN, 0.625)}                       26
{(yahoo, NN, 0.75)}                             18
{(confusing, VBG, 0.625)}                       17
{(mistake, NN, 0.625)}                          17
{(mismatch, NN, 0.667)}                         14
{(deprecating, VBG, 0.75)}                      14
{(error, NN, 0.625), (problem, NN, 0.625)}       7
Name: count, dtype: int64
```

In [69]:
```
df[df["pos_words"].notnull()]["pos_words"].value_counts()[:10]
```

Out[69]:
```
pos_words
{(well, RB, 0.75)}              63
{(functionality, NN, 0.625)}    53
{(catching, VBG, 0.625)}        39
{(improving, VBG, 0.75)}        19
{(taking, VBG, 0.625)}          18
{(add, NN, 0.625)}              13
{(refinement, NN, 0.625)}       12
{(saving, VBG, 0.625)}          11
{(compliance, NN, 0.625)}        9
{(better, RB, 0.875)}            8
Name: count, dtype: int64
```

In [70]:
```
## ^  - if these words and sentiments are not meaningfull, add these words into stopwords
```

In [71]:
```
df["avg_neg_score"] = df["neg_words"].apply(
    lambda x: sum([w[2] for w in x]) / len(x) if x else 0
)

df["avg_pos_score"] = df["pos_words"].apply(
    lambda x: sum([w[2] for w in x]) / len(x) if x else 0
)

df["one"] = 1
```

In [72]:
```
fig, ax = plt.subplots(nrows=1, sharex=True)
df = df.set_index(df["date"].dt.date).sort_index()
df["avg_pos_score"].cumsum().plot(ax=ax, color="green")
df["avg_neg_score"].cumsum().plot(
    ax=ax, linestyle="--", color="red", title="pos/neg sentiment"
)
plt.xticks(rotation=45)
plt.legend()
plt.show()
```

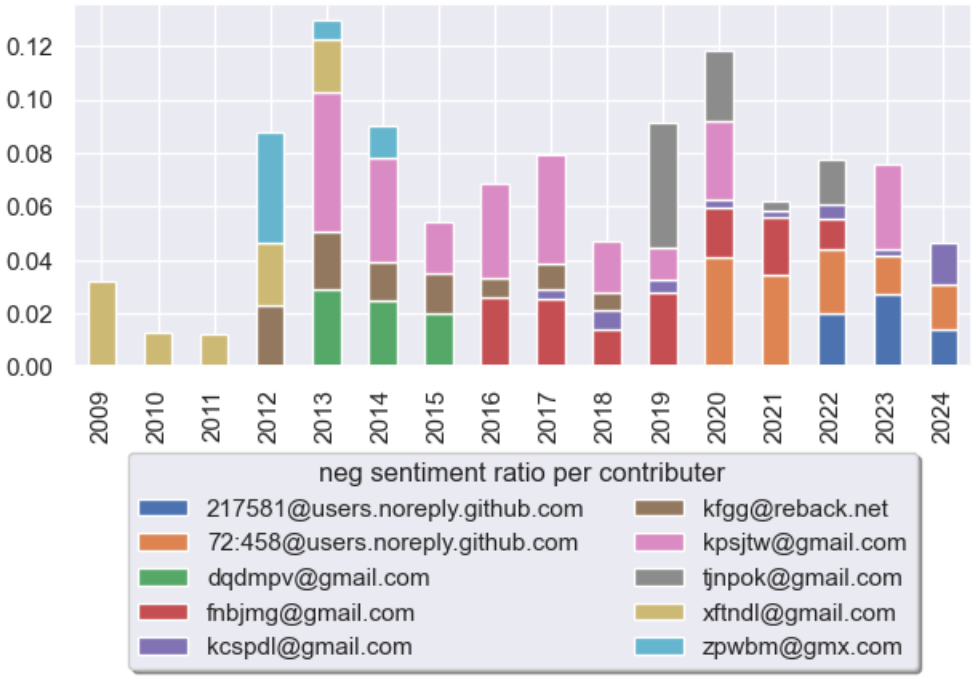
In [73]:
```
n = 10
top_contributors_emails = df.value_counts("email")[:n].index.to_list()

top_contributors = df[df["email"].isin(top_contributors_emails)]

top_contributors = (
    top_contributors.groupby(["email", "period"])
    .aggregate(
        commits=("avg_neg_score", "count"),
        neg_score=("avg_neg_score", "sum"),
        pos_score=("avg_pos_score", "sum"),
    )
    .assign(ratio=lambda x: x["neg_score"] / x["commits"])
    .assign(ratio_pos=lambda x: x["pos_score"] / x["commits"])
)

top_contributors
```

Out[73]:

email	period	commits	neg_score	pos_score	ratio	ratio_pos
217581@users.noreply.github.com	2022	124	2.500	0.750	0.020161	0.006048
	2023	342	9.417	2.250	0.027535	0.006579
	2024	136	1.875	0.750	0.013787	0.005515
72:458@users.noreply.github.com	2020	107	4.375	0.000	0.040888	0.000000
	2021	199	6.875	0.000	0.034548	0.000000
...	...	...	...	...	...	...
xftndl@gmail.com	2012	1526	36.000	6.750	0.023591	0.004423
	2013	138	2.750	0.000	0.019928	0.000000
zpwbm@gmx.com	2012	229	9.500	0.625	0.041485	0.002729
	2013	492	3.375	3.375	0.006860	0.006860
	2014	222	2.625	0.750	0.011824	0.003378

63 rows × 5 columns

In [74]:
```
fig, ax = plt.subplots(1)
top_contributors.swaplevel()["ratio"].unstack().plot(ax=ax, kind="bar", stacked=True)
ax.legend(
    loc="upper center",
    bbox_to_anchor=(0.5, -0.2),
    fancybox=True,
    shadow=True,
    ncol=2,
    title="neg sentiment ratio per contributer",
)
plt.tight_layout()
plt.show()
```



In [75]:
```
## ^ outcome: first year commits are usually having higher negative sentiment ratio, than later yea
```

Sentiments - TextBlob

In [76]:
```
import textblob #it is awesome library
```

In [77]:
```
textblob.Sentence("i love coffee, its awesome").sentiment
```

Out[77]:
```
Sentiment(polarity=0.75, subjectivity=0.8)
```

In [78]:
```
df["polarity"] = df["message"].apply(lambda x: textblob.Sentence(x).polarity)
df["subjectivity"] = df["message"].apply(lambda x: textblob.Sentence(x).subjectivity)
df.sort_values(["polarity"]).head(3)
```

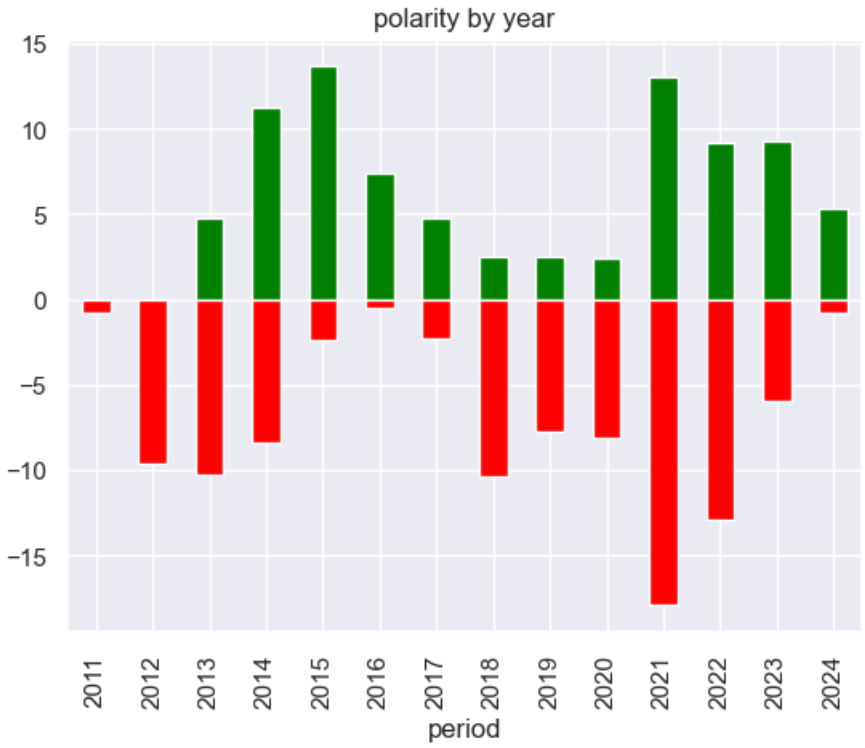
Out[78]:

date	date	email	message	raw	tkns_in_upper_case	nt
2013-04-20	2013-04-20 21:12:21	kfgg@reback.net	DOC Its a bad idea to PEP8 documentation!	DOC: Its a bad idea to PEP8 documentation! \n		
2023-11-26	2023-11-26 04:46:58	72:458@users.noreply.github.com	Adjust tests in base folder for arrow string option	Adjust tests in base folder for arrow string option (#56124)\n\n		
2013-06-01	2013-06-01 19:44:22	zpwbm@gmx.com	BLD test_perf.py, add base pickle target pickle options to test_perf	BLD: test_perf.py, add --base-pickle --target-pickle options to test_perf\n		

In [79]:
```
df = df.set_index("date")
```

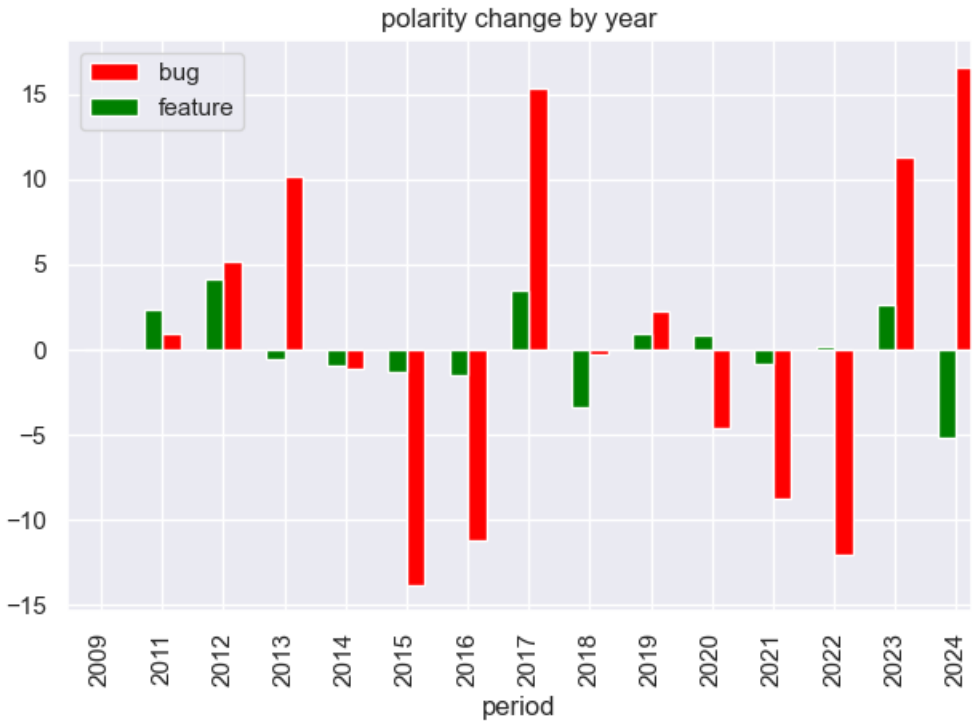
In [80]:
```
df.where(df["polarity"] > 0.5)["polarity"].resample("Y").sum().plot(
    color="green", kind="bar"
)
df.where(df["polarity"] < -0.5).groupby("period")["polarity"].sum().plot(
    color="red", kind="bar", title="polarity by year"
)
plt.show()
```

```
/var/folders/7r/4f58k_j13ks_wpyxs010dk8m0000gn/T/ipykernel_18372/4131801632.py:1: FutureWarning: 'Y
' is deprecated and will be removed in a future version, please use 'YE' instead.
  df.where(df["polarity"] > 0.5)["polarity"].resample("Y").sum().plot(
```



In [ ]:
```
## ^ sinusoid
```

In [81]:
```
import matplotlib.dates as mdates

fig = (
    df[df["is_bug"]]
    .groupby("period")["polarity"]
    .sum()
    .diff()
    .rename("bug")
    .plot(color="red", kind="bar", position=0, width=0.3)
)

fig = (
    df[df["is_feature"]]
    .groupby("period")["polarity"]
    .sum()
    .diff()
    .rename("feature")
    .plot(
        color="green",
        kind="bar",
        width=0.3,
        position=1,
        title="polarity change by year",
    )
)
fig.legend()
plt.tight_layout()
```



In [82]:
```
top_contributors = df[df["email"].isin(top_contributors_emails)]

top_contributors = (
    top_contributors.groupby("email")
    .aggregate(
        commits=("email", "count"),
        polarity=("polarity", "sum"),
        subjectivity=("subjectivity", "sum"),
    )
    .assign(
        polarity_ratio=lambda x: x["polarity"] / x["commits"],
        subjectivity_ratio=lambda x: x["subjectivity"] / x["commits"],
    )
    .sort_values("polarity_ratio", ascending=False)
)
```

In [83]:
```
top_contributors
```

Out[83]:

email	commits	polarity	subjectivity	polarity_ratio	subjectivity_ratio
217581@users.noreply.github.com	602	29.434449	124.961965	0.048894	0.207578
xftndl@gmail.com	3114	110.571125	457.102839	0.035508	0.146790
tjnpok@gmail.com	622	17.292896	60.743862	0.027802	0.097659
dqdmpv@gmail.com	607	14.274501	85.543234	0.023516	0.140928
kfgg@reback.net	4711	102.887320	643.716932	0.021840	0.136641
fnbjmg@gmail.com	723	11.389271	87.858053	0.015753	0.121519
zpwbm@gmx.com	943	14.614559	133.972582	0.015498	0.142071
kpsjtw@gmail.com	1356	11.147729	183.325433	0.008221	0.135196
kcspdl@gmail.com	4812	-0.014831	389.787300	-0.000003	0.081003
72:458@users.noreply.github.com	1120	-27.985060	203.483370	-0.024987	0.181682

In [84]:
```
import numpy as np
```

In [85]:
```
# polarity per year, taking only  abs(0.5)

df.groupby("period").agg(
    stressed_day_polarity_score=("polarity", "sum"),
    n_commits=("email", "count"),
    neg_polarity_n_commits=("polarity", lambda x: sum(x < -0.5)),
    pos_polarity_n_commits=("polarity", lambda x: sum(x > +0.5)),
    emails=("email", lambda x: Counter(x)),
).sort_values("neg_polarity_n_commits", ascending=False).head(5)
```

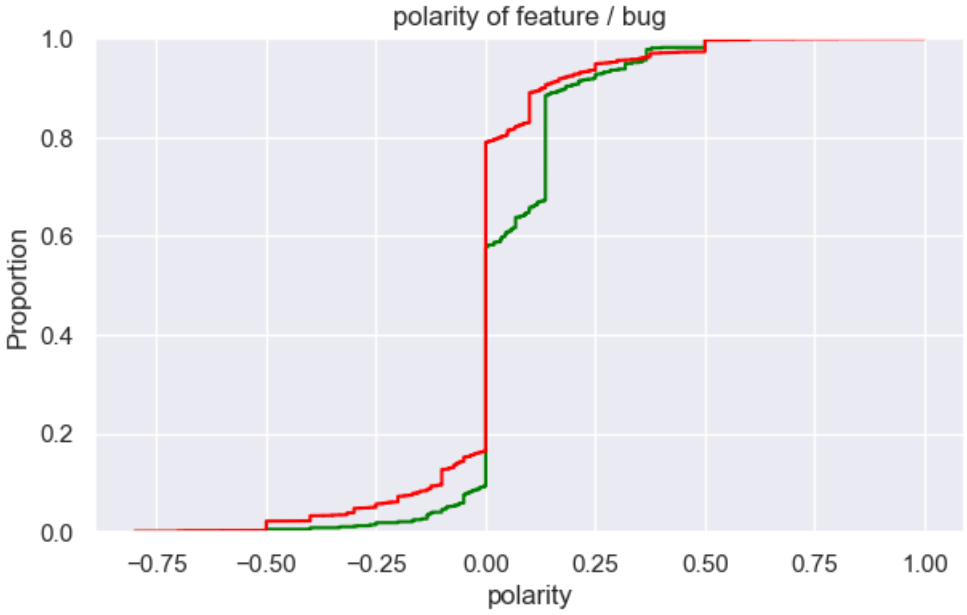
Out[85]:

period	stressed_day_polarity_score	n_commits	neg_polarity_n_commits	pos_polarity_n_commits	
2021	-2.447313	3157	24	15	'kcspdl@ '819351@us 2, 'npjol@us '481229@us '566735@us 103, '72:458@us
2022	6.136355	2434	18	8	{'fr ' '72:458@us 260, 'k '529:93@us '353676@us '539626@us 1, ' 'uxpfsu@us
2013	80.137042	3239	14	21	{'b 'k ' ' 'e 'zpwbm i@gmail.com ' 'diboh@l
2012	77.321215	3238	13	17	{'xf 'ew 'bebnlm@ @gmail.com': 'm 'efcjb 'kt 'kpti@lar
2018	21.575977	2302	13	4	'hg 'k '211871@us 9, 'f 'ktdifo@us 60, 'hbsdj 'upnbvh@us '547:45@user

In [86]:
```
# df[df['date'].between("2020-05-10","2020-05-10")].sort_values('polarity')
```

In [87]:
```
# df['adj']=df['message_tagged_lemmas'].apply(lambda x: set([tk[0] for tk in x if 'RB' in tk[1]]))
fig,ax=plt.subplots(nrows=1, sharex=True, figsize=(7,4))
sns.ecdfplot(df[df['is_feature']], ax=ax,x='polarity', color='green')
sns.ecdfplot(df[df['is_bug']],  ax=ax, x='polarity', color='red')
```

```
plt.title('polarity of feature / bug')
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Deep Learning models for finding emotions

https://huggingface.co/models ## Examples: ## emotions ## joeddav/distilbert-base-uncased-go-emotions-student ## SamLowe/roberta-base-go_emotions

In [88]:
```
# classifier  = pipeline(
# model="lxyuan/distilbert-base-multilingual-cased-sentiments-student",
# truncation=True,
# )
```

In [89]:
```
from transformers import pipeline

# # Load the BERT-Emotions-Classifier
# classifier = pipeline(
#     "text-classification", top_k=3,truncation=True, model="ayoubkirouane/BERT-Emotions-Classifier
# )

classifier = pipeline(
    model="SamLowe/roberta-base-go_emotions",
    truncation=True,
    top_k=3,
    return_all_scores=True,
)
```

```
/opt/homebrew/Caskroom/miniforge/base/envs/env/lib/python3.12/site-packages/huggingface_hub/file_do
wnload.py:1132: FutureWarning: `resume_download` is deprecated and will be removed in version 1.0.0
. Downloads always resume when possible. If you want to force a new download, use `force_download=T
rue`.
  warnings.warn(
```

In [90]:
```
text = ["exciting feature increases the capacity and speed"]
model_outputs = classifier(text, num_workers=8)
```

```
model_outputs
```

Out[90]:
```
[[{'label': 'excitement', 'score': 0.5872608423233032},
  {'label': 'neutral', 'score': 0.25702694058418274},
  {'label': 'approval', 'score': 0.06274145096540451}]]
```

In [91]:
```
def get_emotions(text):
    model_outputs = classifier(text)
    return {
        list(emotion.values())[0]: list(emotion.values())[1]
        for emotion in model_outputs[0]
    }
```

In [92]:
```
get_emotions("i love the most annoying bug, which was crashing the app")
```

Out[92]:
```
{'love': 0.7644850015640259,
 'annoyance': 0.29879993200302124,
 'anger': 0.1696334034204483}
```

In [93]:
```
sample = df.sample(2000)
cols = sample.shape[1]
sample = sample.join(sample["message"].apply(lambda x: pd.Series(get_emotions(x))))
```

In [94]:
```
emotions=sample.iloc[:, cols:].columns
sample.iloc[:, cols:].sum().to_frame()
```

Out[94]:

	0
neutral	1899.011103
approval	36.382189
annoyance	19.963248
disapproval	14.667061
disappointment	2.617199
realization	5.815083
confusion	2.026032
admiration	0.341037
optimism	0.567603
curiosity	0.292628
anger	0.069813
desire	0.034816
caring	0.046107
fear	0.049372
amusement	0.436037
joy	0.128353
gratitude	2.886562
excitement	0.053653
remorse	0.763476
sadness	0.148368

In [95]:
```
sample.nlargest(5, "confusion")
```

Out[95]:

date	email	message	raw	tkns_in_upper_case
2023-07-12 16:30:03	445316@users.noreply.github.com	CLN Replace confusing brackets with backticks * Replace confusing brackets with backticks There are some errors which read like > only list like objects are allowed to be passed to isin(), you > passed a [str] (note how the type `str` is enclosed in square brackets) This is potentially confu...	CLN: Replace confusing brackets with backticks (#54091)\n\n* Replace confusing brackets with backticks\r\n\r\nThere are some errors which read like:\r\n\r\n> only list-like objects are allowed to be passed to isin(), you\r\n> passed a [str]\r\n\r\n(note how the type `str` is enclosed in square b...	TYPE
2023-06-02 17:02:26	58:743@users.noreply.github.com	DOC Adjust build command for building with meson Change default command to show verbose output. Otherwise, it might be confusing for users when the import hangs and their CPU goes to 100% during the rebuild step.	DOC: Adjust build command for building with meson (#53392)\n\nChange default command to show verbose output. \r\n\r\nOtherwise, it might be confusing for users when the import hangs and their CPU goes to 100% during the rebuild step.	CPU
2019-09-12 12:51:15	hfqdfm@gmail.com	Fix a typo in "computation.rst" in document. There's `np.random.np.random` in /doc/source/user_guide/computation.rst, which I believe is a typo. But the weird thing is there's actually `np.random.np` in numpy (1.16.4), but not in numpy (1.17.2). That's maybe why the doc build passed before. Wh...	Fix a typo in "computation.rst" in document. (#28400)\n\nThere's `np.random.np.random` in /doc/source/user_guide/computation.rst, which I believe is a typo. But the weird thing is there's actually `np.random.np` in numpy (1.16.4), but not in numpy (1.17.2). That's maybe why the doc build passed ...	
2012-01-19 20:52:09	efcjbo@onerussian.com	ENH pass figsize into _grouped_plot functions atm figsize is not passed through by boxplot etc, making it (impossible?) to have custom figsize Conflicts pandas/tools/plotting.py	ENH: pass figsize into _grouped_plot functions\n\natm figsize is not passed through by boxplot etc, making it\n(impossible?) to have custom figsize\n\nConflicts:\n\n\tpandas/tools/plotting.py\n	
2022-05-25 21:34:05	7487:3@users.noreply.github.com	Changes as requested in #47058 * Changes as requested in #47058 This is my first open source contribution. Please let me know if there are any mistakes, will rectify them * required changes as per #47058 Please let me	Changes as requested in #47058 (#47119)\n\n* Changes as requested in #47058\r\n\r\nThis is my first open-source contribution. Please let me know if there are any mistakes, will rectify them\r\n\r\n*	

date	email	message	raw	tkns_in_upper_case
		know if have to change anything. * changes to doc as per #47058 If any more ...	required changes as per #47058\r\n\r\nPlease let me know if have to change anything. \r\n\r\n* cha...	

5 rows × 39 columns

In [97]:
```
sample.nlargest(5, "curiosity")
```

Out[97]:

date	email	message	raw	tkns_in_upper_case	ntkns
2022-05-25 21:34:05	7487:3@users.noreply.github.com	Changes as requested in #47058 * Changes as requested in #47058 This is my first open source contribution. Please let me know if there are any mistakes, will rectify them * required changes as per #47058 Please let me know if have to change anything. * changes to doc as per #47058 If any more ...	Changes as requested in #47058 (#47119)\n\n* Changes as requested in #47058\r\n\r\nThis is my first open-source contribution. Please let me know if there are any mistakes, will rectify them\r\n\r\n* required changes as per #47058\r\n\r\nPlease let me know if have to change anything. \r\n\r\n* cha...		
2012-05-30 00:52:19	xftndl@gmail.com	DOC what's new	DOC: what's new\n		
2015-01-12 22:30:32	uipnbt@uiowa.edu	Merge pull request #9230 from jseabold/py3 s3 COMPAT Need to read Bytes on Python	Merge pull request #9230 from jseabold/py3-s3\n\nCOMPAT: Need to read Bytes on Python		
2020-06-23 17:35:56	55:444@users.noreply.github.com	TST Verify whether non writable numpy array is shiftable (21049)	TST: Verify whether non writable numpy array is shiftable (21049) (#34919)\n\n		
2015-05-14 14:52:38	kpsjtw@gmail.com	Merge pull request #10101 from jorisvandenbossche/v0.16.1 docs v0.16.1 docs	Merge pull request #10101 from jorisvandenbossche/v0.16.1-docs\n\nv0.16.1 docs		

5 rows × 39 columns

In [99]:
```
sum_emotions=sample[sample['email'].isin(top_contributors_emails)].groupby('email')[emotions].sum()
```

In [100…
```
sum_emotions.style.format('{:,.2f}').background_gradient(cmap='RdYlGn_r', axis=1)
```

Out[100…

email	neutral	approval	annoyance	disapproval	disappointment	realization	confus
217581@users.noreply.github.com	31.83	0.46	0.36	0.07	0.00	0.01	(
72:458@users.noreply.github.com	68.60	0.99	1.40	1.21	0.24	0.12	(
dqdmpv@gmail.com	34.95	0.59	0.35	0.47	0.00	0.20	(
fnbjmg@gmail.com	45.16	0.60	0.43	0.31	0.03	0.08	(
kcspdl@gmail.com	280.90	3.74	2.78	1.05	0.09	0.26	(
kfgg@reback.net	258.20	4.14	2.26	2.22	0.66	0.77	(
kpsjtw@gmail.com	82.59	1.43	0.78	1.30	0.11	0.20	(
tjnpok@gmail.com	29.90	0.42	0.24	0.03	0.00	0.05	(
xftndl@gmail.com	163.91	3.06	1.61	2.09	0.23	0.44	(
zpwbm@gmx.com	40.37	0.70	0.28	0.02	0.00	0.13	(

In [101…

```
sum_emotions.drop(columns=['neutral','approval']).style.format('{:,.2f}').highlight_max(color='red'
```

Out[101…

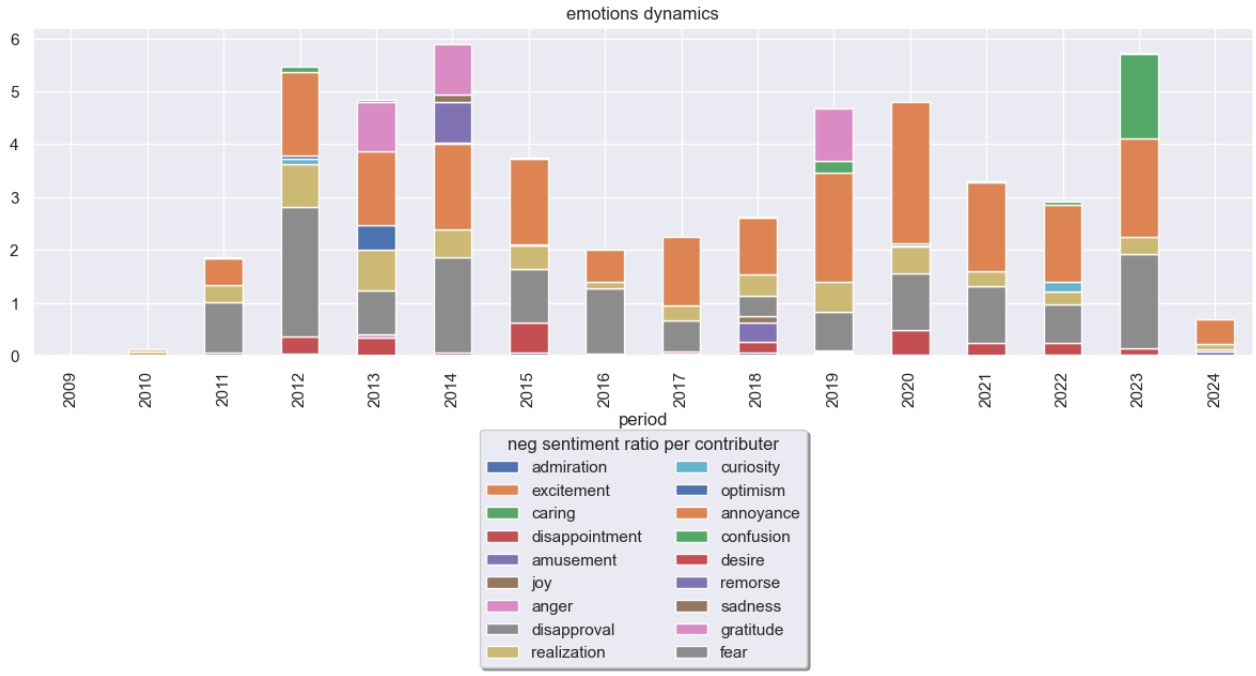
email	annoyance	disapproval	disappointment	realization	confusion	admiration	o
217581@users.noreply.github.com	0.36	0.07	0.00	0.01	0.00	0.00	
72:458@users.noreply.github.com	1.40	1.21	0.24	0.12	0.00	0.01	
dqdmpv@gmail.com	0.35	0.47	0.00	0.20	0.00	0.00	
fnbjmg@gmail.com	0.43	0.31	0.03	0.08	0.00	0.00	
kcspdl@gmail.com	2.78	1.05	0.09	0.26	0.00	0.02	
kfgg@reback.net	2.26	2.22	0.66	0.77	0.00	0.06	
kpsjtw@gmail.com	0.78	1.30	0.11	0.20	0.00	0.00	
tjnpok@gmail.com	0.24	0.03	0.00	0.05	0.00	0.00	
xftndl@gmail.com	1.61	2.09	0.23	0.44	0.03	0.06	
zpwbm@gmx.com	0.28	0.02	0.00	0.13	0.00	0.00	

In [102…

```
emotions_no_neut=set(emotions)
emotions_no_neut.remove('neutral')
emotions_no_neut.remove('approval')
emotions_no_neut=list(emotions_no_neut)
```

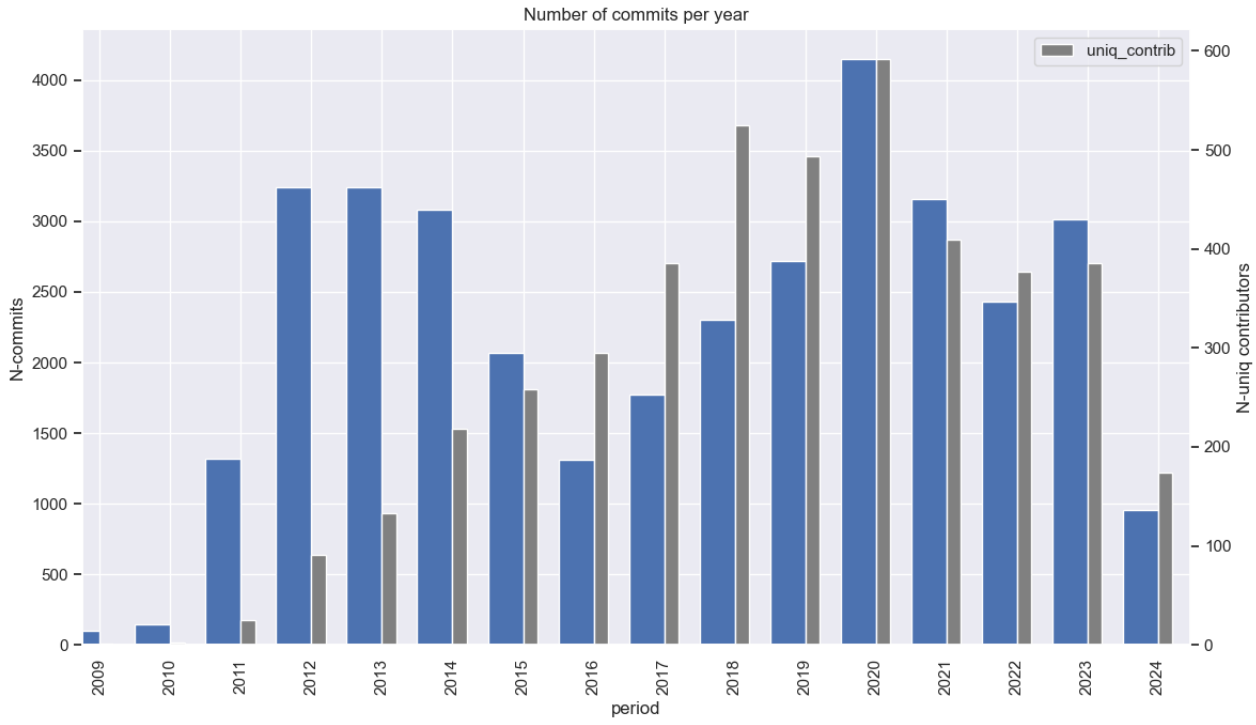
In [103…

```
fig, ax = plt.subplots(1, figsize=(12,7),sharex=True)
sample.groupby('period')[emotions_no_neut].sum().plot(ax=ax, kind='bar',stacked=True, title='emotio


ax.legend(
    loc="upper center",
    bbox_to_anchor=(0.5, -0.2),
    fancybox=True,
    shadow=True,
    ncol=2,
    title="neg sentiment ratio per contributer",
)
plt.tight_layout()
plt.show()
```



In [104…
```
# 2020, high annoyance
```

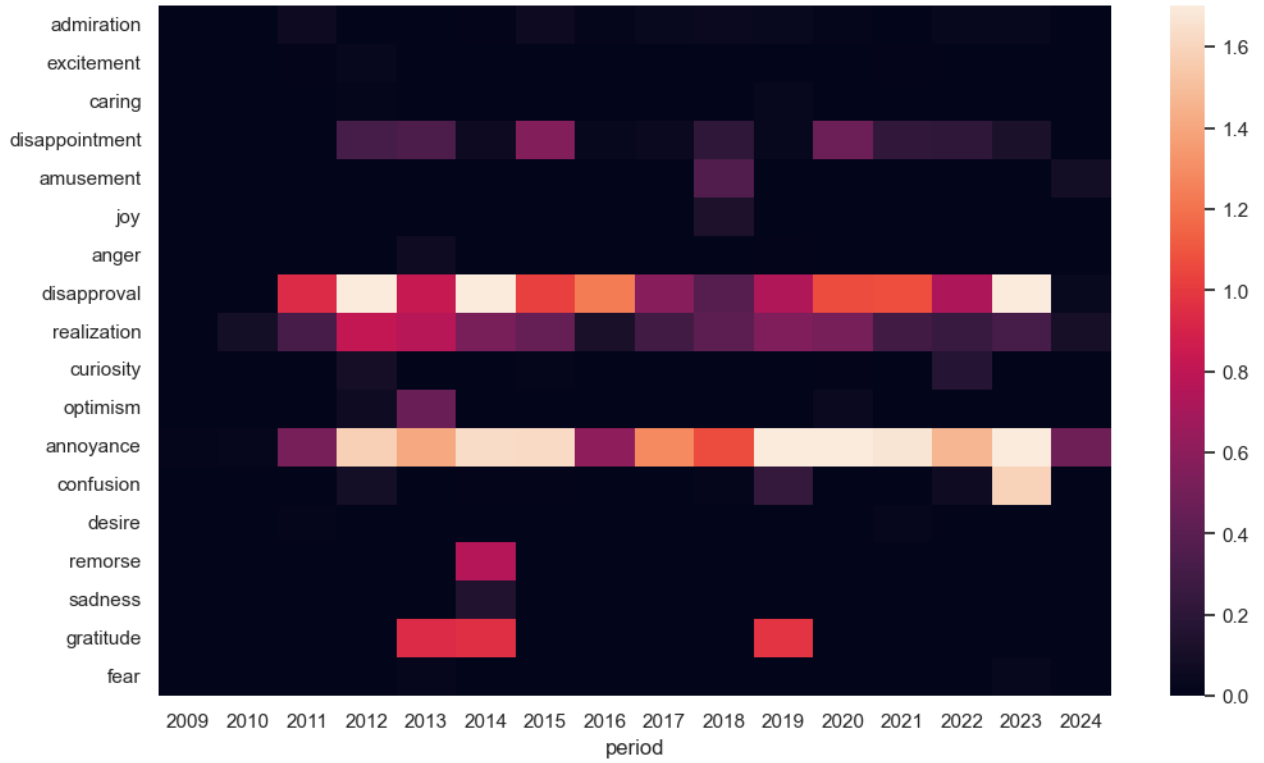
In [105…
```
fig, ax = plt.subplots(1, figsize=(12, 7), sharex=True)
ax2 = ax.twinx()

df.groupby("period").size().plot(
    ax=ax, kind="bar", title="Number of commits per year", position=1
)
df.groupby("period").agg(uniq_contrib=("email", "nunique")).plot(
    ax=ax2, kind="bar", width=0.2, grid=False, position=0, color="grey"
)
ax.set_ylabel("N-commits")
ax2.set_ylabel("N-uniq contributors")
plt.tight_layout()
plt.show()
```



In [106…
```
dfplot=sample.groupby('period')[emotions_no_neut].sum().T
```

In [107…
```
fig, ax = plt.subplots(1, figsize=(12, 7), sharex=True)
sns.heatmap(dfplot, robust=True)
```

Out[107… <Axes: xlabel='period'>

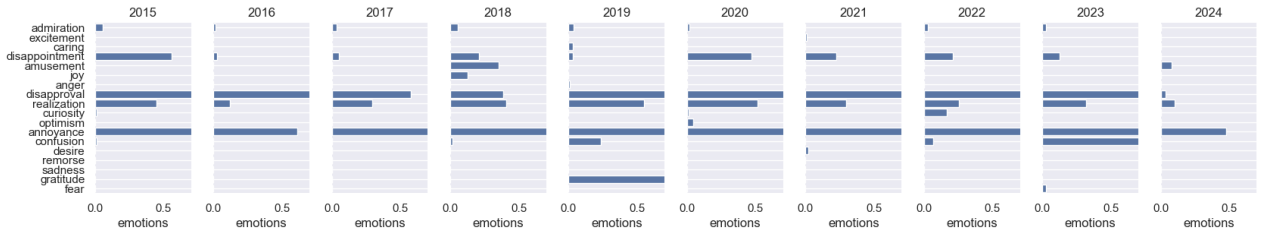


In [108…
```
dfplot=dfplot.reset_index()
```

In [109…
```
dfplot.select_dtypes('number').quantile(0.9).max()
```

Out[109… 1.646579760592431

In [110…
```
g = sns.PairGrid(dfplot,
                 x_vars=dfplot.columns[-10:], y_vars='index',
                 height=3, aspect=.55)

g.map(sns.barplot, orient="h",
      palette="flare_r", linewidth=1, edgecolor="w")
g.set(xlim=(0, 0.7), xlabel="emotions", ylabel="")

titles=dfplot.columns[-10:]
for ax, title in zip(g.axes.flat, titles):
    ax.set(title=title)
    ax.xaxis.grid(False)
    ax.yaxis.grid(True)

sns.despine(left=True, bottom=True)
```



In [ ]:

summarization

https://huggingface.co/facebook/bart-large-cnn

In [111…
```
# ## summary. fb model 1.2 GB
from transformers import pipeline
summarizator = pipeline("summarization", model="facebook/bart-large-cnn")
```

```
/opt/homebrew/Caskroom/miniforge/base/envs/env/lib/python3.12/site-packages/huggingface_hub/file_do
wnload.py:1132: FutureWarning: `resume_download` is deprecated and will be removed in version 1.0.0
. Downloads always resume when possible. If you want to force a new download, use `force_download=T
rue`.
  warnings.warn(
```

In [112…
```
text_sample=  df[(df["email"].isin(top_contributors_emails)) & (df["is_feature"]).fillna(False) & (
text_sample
```

Out[112…
```
"TST  fix and test index division by zero Related  #19336 Author  Brock Mendel <jbrockmendel@gmail
.com> Closes #19347 from jbrockmendel/div_zero2 and squashes the following  s  be1e2e1b8 [Brock Me
ndel] move fixture to conftest 64b0c0853 [Brock Mendel] Merge branch 'master' of https // hub.com/
pandas dev/pandas into div_zero2 aa969f8d2 [Brock Mendel] Merge branch 'master' of https // hub.co
m/pandas dev/pandas into div_zero2 000aefde0 [Brock Mendel] fix long again 9de356ab0 [Brock Mende
l] revert fixture to fix test_range failures b8cf21d3e [Brock Mendel] flake8 remove unused import
afedba98b [Brock Mendel] whatsnew clarification b51c2e14c [Brock Mendel] fixturize 37efd5108 [Broc
k Mendel] make zero a fixture 965f7214e [Brock Mendel] Merge branch 'master' of https // hub.com/p
andas dev/pandas into div_zero2 d648ef698 [Brock Mendel] requested edits 1ef3a6c74 [Brock Mendel]
Merge branch 'master' of https // hub.com/pandas dev/pandas into div_zero2 78de1a4df [Brock Mende
l] Merge branch 'master' of https // hub.com/pandas dev/pandas into div_zero2 0277d9fca [Brock Men
del] add ipython output to whatsnew 5d7e3ea0c [Brock Mendel] Merge branch 'master' of https // hu
b.com/pandas dev/pandas into div_zero2 ea75c3ca0 [Brock Mendel] ipython block 6fc61bd99 [Brock Men
del] elaborate docstring ca3bf4241 [Brock Mendel] Whatsnew section cd543497c [Brock Mendel] move d
ispatch_missing to core.missing 06df02a89 [Brock Mendel] py3 fix 84c74c54a [Brock Mendel] remove o
perator.div for py3 6acc2f78a [Brock Mendel] fix missing import e0e89b978 [Brock Mendel] fix and a
nd tests for divmod 969f342e1 [Brock Mendel] fix and test index division by zero"
```

In [113…
```
summarizator(text_sample,max_length=100)
```

Out[113…
```
[{'summary_text': 'Brock Mendel closes #19347 from jbrockmendel/div_zero2 and squashes the followi
ng  s. #19336 Author  Brock Mendel <jbrock mendel@gmail.com> Closes #19346 from jBrockm endel/ div
_zero1 and squashing the following s.'}]
```

In [114…
```
summarizator(text_sample,max_length=20)
```

```
Your min_length=56 must be inferior than your max_length=20.
/opt/homebrew/Caskroom/miniforge/base/envs/env/lib/python3.12/site-packages/transformers/generatio
n/utils.py:1165: UserWarning: Unfeasible length constraints: `min_length` (56) is larger than the m
aximum possible length (20). Generation will stop at the defined maximum length. You should decreas
e the minimum length and/or increase the maximum length.
  warnings.warn(
```

Out[114…
```
[{'summary_text': 'Brock Mendel closed #19347 from jbrockmendel/div'}]
```

In [ ]:
```
To write a short summary what the contributor was doing, we will run summarize twice
1st layer is over individual commit message, which would produce a short summary
2nd layer on a top of summaries which we've got
```

In [116…
```
sample=df[(df["email"].isin(top_contributors_emails[:1])) & (df['is_feature']) & (df['message_lemma
```

In [117…
```
sample['summary']=sample['message'].apply(lambda x: summarizator(x,min_length=5,  max_length=20, cl
```

In [118…
```
sample[['email','summary','message','is_feature','is_bug']].head(10)
```

Out[118…

date	email	summary	message	is_feature	is_bug
2023-05-03 15:51:59	kcspdl@gmail.com	REF lazify relativedelta imports. API intentionally raise ValueError.	REF lazify relativedelta imports * REF lazify relativedelta imports * API intentionally raise ValueError * whatsnew	True	False
2022-10-20 18:03:54	kcspdl@gmail.com	DEPR enforce Timedelta freq, delta, is_populated dep	DEPR enforce Timedelta freq, delta, is_populated deprecations * DEPR enforce Timedelta freq, delta, is_populated deprecations * update docs, pyi * fix nat test * move whatsnew	True	False
2022-11-07 23:04:22	kcspdl@gmail.com	DEPR enforce a bunch of fixes. disable pylint check. Fix gh	DEPR enforce a bunch * DEPR enforce a bunch * docstring fixup * disable pylint check * pylint fixup * Fix gh ref in whatsnew	True	False
2020-06-01 02:40:01	kcspdl@gmail.com	Mul is a simple way to simplify to_ offset. It's used to simplify	ENH mul(Tick, float); simplify to_offset * ENH mul(Tick, float); simplify to_offset * troubleshoot docbuild * whatsnew, comments Co authored by brock <brock@EnterpriseB.local>	True	False
2023-05-24 20:53:49	kcspdl@gmail.com	DEPR be stricter in assert_almost_equal * 32bit builds	DEPR be stricter in assert_almost_equal * DEPR be stricter in assert_almost_equal * 32bit builds * Fix transform test * ignore warning i cant reproduce localy * pylint fixup * Fix AarrayManager and CoW builds * fix tests * Whatsnew Co authored by Matthew Roeschke < >	True	False
2022-12-21 19:57:24	kcspdl@gmail.com	DEPR enforce inplaceness for df.loc[, foo]=bar	DEPR enforce inplaceness for df.loc[, foo]=bar * DEPR enforce inplaceness for df.loc[, foo]=bar * Fix ArrayManager tests * suggested edits to AM tets * update doctest * CoW test * whatsnew * Use reindex_indexer * suggested test edits	True	False
2024-02-14 19:51:05	kcspdl@gmail.com	TYP make dtype required in _from_sequence_of_strings	TYP make dtype required in _from_sequence_of_strings * TYP make dtype required in _from_sequence_of_strings * GH ref * mypy fixup * Move whatsnew * Update pandas/core/arrays/base.py Co authored by Matthew Roeschke < > Co authored by Matthew Roeschke < >	True	False
2022-11-07 23:08:00	kcspdl@gmail.com	DePR DatetimeIndex indexing with mismatched tzawareness.	DEPR DatetimeIndex indexing with mismatched tzawareness * DEPR DatetimeIndex indexing with mismatched tzawareness * clarify whatsnew	True	False
2020-10-12 01:23:57	kcspdl@gmail.com	CLN dont special case DatetimeArray indexing. Use parent class _valid	CLN dont special case DatetimeArray indexing * CLN dont special case DatetimeArray indexing * use parent class _validate_getitem_key * test, whatsnew * update test	True	False
2022-10-20 20:02:25	kcspdl@gmail.com	DEPR enforce Series/DataFrame awareness mismatch deprecations.	DEPR enforce Series/DataFrame awareness mismatch deprecations * DEPR enforce	True	False

email	summary	message	is_feature	is_bug
date				
	DEPR	deprecation on Series(ts_aware, dtype=naive) * DEPR enforce deprecation on Series(tzaware_seq, dtype=naive) * rename * move whatsnew		

In [119…
```
sample['enriched_summary']=sample.apply(lambda x: x['summary'] + ' It is a feature.' if x['is_featu
```

In [ ]:
```
```

In [120…
```
summary=sample.sort_index(ascending=False).head(10).groupby('email').agg(msg=('enriched_summary',la
print(summary)
#.apply(lambda x: summarizator(x,min_length=50,  max_length=200))
```

```
 TYP  make dtype required in _from_sequence_of_strings  It is a feature.. Mypy fixup * update docte
st * simplify * avoid Series.view * dont It is a feature.. DEPR  resample with PeriodIndex. Update
docstring. Code block in 0 It is a feature.. DEPR  downcasting in NDFrame.where, mask, clip. GH ref
It is a feature.. DEPR  concat ignoring empty objects   * DEPR  Concat with It is a feature.. DEPR
support axis=None in DataFrame reductions. Test, whatsnew, It is a feature.. DEPR  be stricter in a
ssert_almost_equal   * 32bit builds It is a feature.. REF  lazify relativedelta imports. API  inten
tionally raise ValueError. It is a feature.. DEPR  concat ignoring all NA columns    * DEPR  It is
a feature.. The latest version of the Pandas software is now available for download. The latest ver
sion It is a feature.
```

In [121…
```
summarizator(summary,min_length=100,  max_length=200)
```

Out[121…
```
[{'summary_text': 'The latest version of the Pandas software is now available for download. The la
test version It is a feature. Make dtype required in _from_sequence_of_strings. Mypy fixup. Update
doctest. simplify. Avoid Series.view. Update docstring. Code block in 0. Update codestring.where,
mask, clip. GH ref It isA feature.. DEPR  concat ignoring empty objects   * DePR  Concat with It i
sa feature.'}]
```

In [ ]:
```
```

In [ ]:
```
```

using ChatGPT API

In [ ]:
```
from openai import OpenAI
client = OpenAI(api_key='')
response = client.chat.completions.create(
  model="gpt-4o",
  messages=[
    {
      "role": "system",
      "content": "Summarize content, what is bad and what is good"
    },
    {
      "role": "user",
      "content": summary}
  ],
  temperature=0.5,
  max_tokens=200,
  top_p=1
)
```

In [ ]:
```
print(response.to_json())
```

In [ ]:
```
```

In [ ]:
```
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: